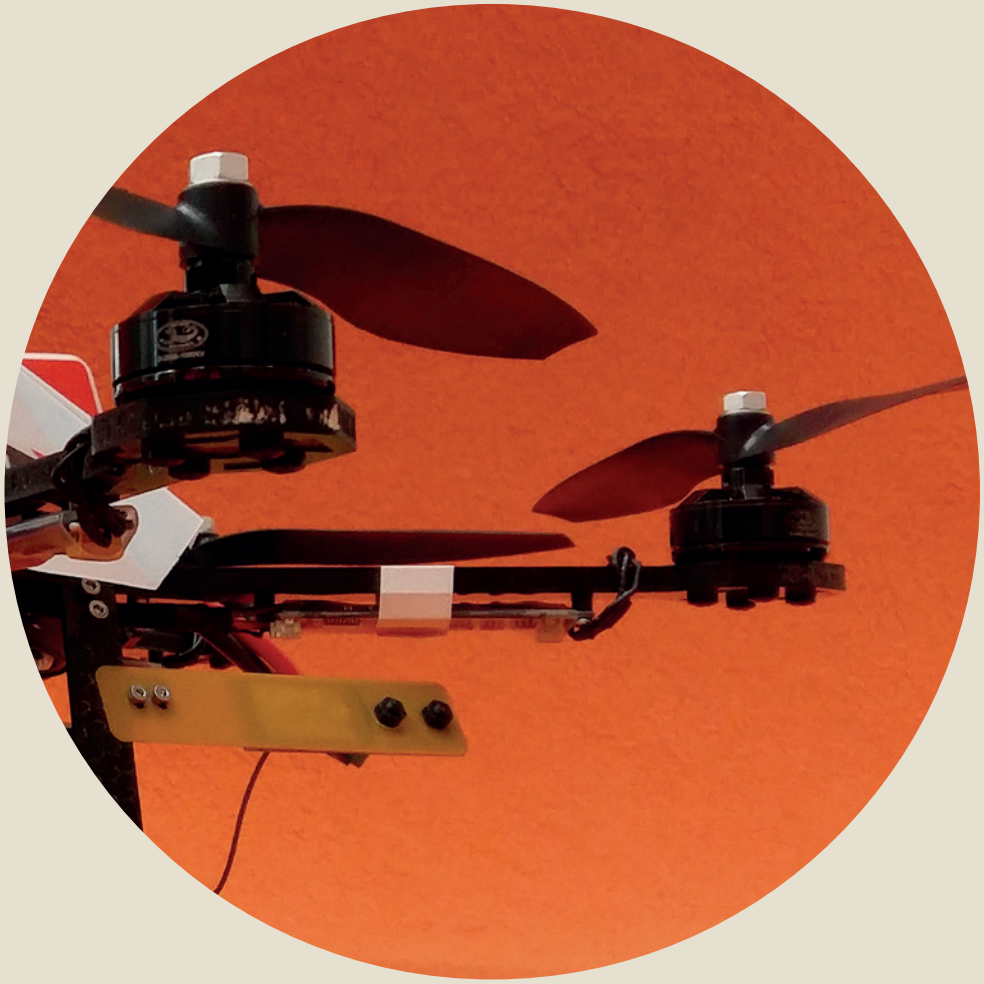
THESES



Aerospace
and Geodesy

Guillermo Paúl Falconí Salazar

Adaptive Fault Tolerant Control for VTOL Aircraft with Actuator Redundancy

TUM.University Press

Lehrstuhl für Flugsystemdynamik
Fakultät für Luftfahrt, Raumfahrt und Geodäsie
Technische Universität München

Adaptive Fault Tolerant Control for VTOL Aircraft with Actuator Redundancy

Guillermo Paúl Falconí Salazar

Vollständiger Abdruck der von der Fakultät für Luftfahrt, Raumfahrt und Geodäsie der Technischen Universität München zur Erlangung des akademischen Grades eines Doktor-Ingenieurs (Dr.-Ing.) genehmigten Dissertation.

Vorsitzender:

Prof. Dr.-Ing. Volker Gümmer

Prüfer der Dissertation:

1. Prof. Dr.-Ing. Florian Holzapfel
2. Prof. Dr.-Ing. Jan Lunze

Die Dissertation wurde am 25.05.2021 bei der Technischen Universität München eingereicht und durch die Fakultät für Luftfahrt, Raumfahrt und Geodäsie am 12.07.2021 angenommen.

The German National Library has registered this publication in the German National Bibliography. Detailed bibliographic data are available on the Internet at https://portal.dnb.de.

Imprint

1. Edition

Layout design and typesetting: Guillermo Paúl Falconí Salazar
Cover design: Caroline Ennemoser
Cover illustration: Guillermo Paúl Falconí Salazar

TUM.University Press
Technical University of Munich
Arcisstrasse 21
80333 Munich

DOI: 10.14459/2021md1595568
ISBN printed edition: 978-3-95884-074-4

www.tum.de

Abstract/Kurzfassung

Abstract

This thesis presents an adaptive fault tolerant controller for a VTOL multicopter system that considers actuator redundancy and limits and implements a strategy for degraded control authority. In a first step, model reference adaptive control (MRAC) is studied for redundant systems and improved control authority conditions based on the developed parameter reduction due to overactuation (PRO) are proposed. Predictor-based adaptive control allocation (P-ACA) is then developed such that adaptive control can incorporate control allocation approaches to exploit actuator redundancy, enforce actuator limits and implement a degraded control strategy. For the multicopter case, this is combined with a prioritizing control allocation (PRIO CA) which permits the prioritization of virtual controls if the desired ones are unattainable. Flight tests including two hexacopter configurations with representative failure cases and high-speed scenarios are presented. A single unified controller handles the nominal and faulty cases.

Kurzfassung

Diese Arbeit präsentiert einen adaptiven fehlertoleranten Regler für VTOL Multicoptern, der Sättigung und Redundanz der Aktuatoren berücksichtigt und eine Strategie für degradierte Steuerbarkeit implementiert. In einem ersten Schritt wird adaptive Modellfolgeregelung redundanter Systeme untersucht und verbesserte Steuerbarkeitsbedingungen werden anhand von der entwickelten Parameterreduktion aufgrund der Aktuatorredundanz (PRO) präsentiert. Die Beobachter-basierte adaptive Steuereingangsallokation (P-ACA) ist danach entwickelt, so dass die Steuereingangsallokation in der adaptiven Regelungsstruktur integriert werden kann. Damit können die Aktuatorredundanz ausgenutzt, die Aktuatorsättigung eingehalten und eine Strategie für degradierte Steuerbarkeit implementiert werden. Für Multicopter-Systeme wird die Methode P-ACA mit einer priorisierenden Steuereingangsallokation (PRIO CA) kombiniert, welche die Priorisierung der virtuellen Regelgrößen im Falle unerreichbaren Kommandos erlaubt. Flugtests mit zwei Hexacopter-Konfigurationen inklusive repräsentativen Fehlerfällen- und Hochgeschwindigkeitsszenarien werden präsentiert. Ein einzelner vereinter Regler wird sowohl für den Nominalfall als für die Fehlerfälle benutzt.

Acknowledgment

First, I would like to express my deepest appreciation to my advisor Prof. Dr.-Ing. Florian Holzapfel who from day one supported my research, put me in a position to take responsibility and achieve my goals, as well as encourage my continuous improvement. I'm profoundly grateful to Prof. Dr.-Ing. Jan Lunze for the interest in my work, his unique perspective, and for the careful review of this thesis. Furthermore, I would like to thank the chairman of the committee Prof. Dr.-Ing. Volker Gümmer. My sincerest thanks go to Dr. Oliver Funke who supported my main research project and to Dr. Girish Chowdhary for providing several collaboration opportunities.

I would like to extend my gratitude to my colleagues at the Institute of Flight System Dynamics for the great working environment. Special thanks go to Maximilian Mühlegg which whom I have tackled several projects and shared most of my time at the institute. I'm very grateful to Thomas Raffler who always had great flight-testing advice and who's contributions were instrumental to a great testing platform. I would also like to thank Christian Heise for his invaluable input during our discussions, our joint projects, and for his proof-reading. Thanks to Valentin Marvakov and David Seiferth for their support during the final flight tests. I would also like to thank all my project partners and former students. Sravan Akkinappalli, Philipp Niermeyer and Jorg Angelov deserve special mention. Finally, thanks to Simon Schatz, Jian Wang, Volker Schneider, Patrick Lauffs, Matthias Rieck, Benedikt Grüter and Kajetan Nürnberger among many others for their support and the extraordinary atmosphere at the institute.

Many thanks to all my friends standing by my side during this whole project. I'm deeply grateful to my parents Katty and Guillermo and to my siblings Rafael, Martina, and Juan Mateo for their unconditional support and belief in me. I want to thank my beloved wife Tatiana for her infinite patience and motivation. Her support, humor, and strikingly logical advice always helped keep me on track. Finally, thank you to my cherished son Mateo who held on just long enough to let me defend the thesis before his birth.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

In the last decades the variety and complexity of missions for which aircraft systems are envisioned are constantly increasing and are pushing the development of flight automation. Especially new designs with vertical take-off and landing (VTOL) capabilities have been developed for applications ranging from small delivery unmanned aerial vehicles (UAV) up to auto-piloted urban air taxis. In the category of consumer products, quadcopters for filming are available, e.g. the *Mavic Air 2* from *DJI* [31] and *EVO* from *Autel Robotics* [9]. Fixed-wing VTOL platforms with variable payload have also made their way to products with applications like agricultural remote sensing, mapping and package delivery. Some examples are the *Trinity* from *Quantum Systems* with a maximum payload of $0.7[kg]$ and 1 hour flight time [127], or the *Wingcopter* UAV with a payload of $2-6[kg]$ and a range of $45-120[km]$ [152]. Another trending application is urban air mobility where flight automation is stepping up to the next level by developing automatic passenger air transportation. There is no available product yet and in most cases not even the necessary regulations, but there exist several enterprises with this goal in mind. Some examples are the *Passenger Air Vehicle* from *Aurora Flight Sciences* [8], *CityAirbus* [2], *Volocopter* [148], *AutoFlightX* [148], or *Cora* from *Wisk* [153].

One of the main challenges and success factors of new aircraft designs is their fault tolerance capability, that is the ability of compensating faults in order to avoid a system failure [79]. In the aerospace industry, safety and reliability have been the main drivers of fault tolerance since both have a strong influence on profit, public image and aircraft certification.Classically, fault tolerance has been addressed through physical redundancy of critical components [12, 40, 99]. Latter, the introduction of digital fly-by-wire systems allowed the implementation of advanced algorithms like for example model-based fault tolerant control [99]. Nowadays, within the UAV development, fault tolerance has been identified as a key step in order to increase system's reliability and therefore to automatically accomplish complete missions [32].Interestingly, an increased level of automation, does not only affect the nominal functionalities but greatly influences fault and failure handling.

The main goal of this thesis is the enhancement of fault tolerance by the means of digital flight control. Specifically, tolerance to faults within the actuation system is addressed. If not taken into account, this kind of faults can lead to performance or control loss. The importance of correctly addressing this kind of faults is based on the

criticality of the resulting consequences. An actuator fault can for example cause a loss of control in-flight (LOC-I) accident. Taking a look at the 5-year report from the International Air Transport Association (IATA) [76], it can bee seen that from 2013-2017 the accidents categorized as LOC-I have not been the most common but have a high fatality risk. They build up about 60% of the fatal aircraft accidents counting for only about 9% of the total accidents [76]. Although these statistics do not directly translate to new VTOL designs, they stress the relevance of avoiding control loss during flight operation. Therefore, in this thesis especial attention is given to overactuated systems, that is systems with redundant actuators. In the aerospace industry, either from reliability or safety reasons, configurations with actuator redundancy commonly result from the requirement that the system should maintain controlled flight in the presence of a single failure. In this context, fault tolerant control enables and maximizes the advantages of the physical actuator redundancy for maintaining controlled flight.

Given that this research on fault tolerant control (FTC) for actuation faults has a clear practical motivation, it is imperative to experimentally validate the developed algorithms. Therefore, the hexacopter as shown in Figure 1.1 is selected as a testing platform. The hexacopter is a VTOL system and a representative of multirotor configurations which have over typical helicopters the following advantages. Multicopters can use fixed-pitch rotors simplifying the mechanical structure [95] and it is possible to directly control the motor speeds simplifying the design of the controller [71]. The hexarotor is a system with actuator redundancy since it has 6 propulsion units, two more than the minimum needed which corresponds to a quadrotor configuration. Given the popularity of multicopters within the research community, a further advantage of selecting the hexacopter is the possibility of comparing the presented solutions with others in the literature.

Figure 1.1: Ascending Technologies Hexacopter: Firefly

1.2 Research Challenges and State of the Art

In order to successfully design a fault tolerant controller with respect to actuation faults and failures, three main challenges have been identified and drive the selection of the methods in this thesis:

1. Uncertainties regarding faults and failures,
2. Actuator redundancy and
3. Actuator limits.

Fault and failures are inherently uncertain, i.e. they are unknown in magnitude, appearance time and duration [138]. Hence, the task at hand is the control of a plant with uncertain dynamics. One possible way to solve it is the reconfiguration approach, which is composed of two main steps: the fault detection and isolation step and the controller reconfiguration step. Another possibility is to design a controller which is robust to a set of possible faults as in robust control. In this thesis, the task is formulated as an adaptive control problem. Adaptive control studies the question of controlling uncertain dynamical systems and it is addressed by online variation/adaptation of the controller parameters. Hence, differently to the reconfiguration approach which mainly selects and switches between controllers, an adaptive controller continuously reacts to the plant dynamics similarly to the behavior of an integrator. The application of adaptive fault tolerant control against actuator faults and failures is one main focus of this thesis. Especially, **adaptive fault tolerant control for overactuated systems** is thoroughly discussed because of its major importance in practice.

The next challenge is actuator redundancy. Although actuator redundancy physically allows the system to tolerate an actuator fault, redundant actuators and their faults, even if known, pose a significant challenge for the flight control algorithms. The challenge arises from the fact that actuator redundancy inevitably results in an ambiguity regarding the distribution of forces and moments to the available actuators. Complexity increases for most VTOL configurations, since the effect of a faulty actuator simultaneously affects different control axes. This differs from traditional fixed-wing concepts where redundancy mainly arises from dividing control surfaces, leading to groups of similar actuators.

The last challenge is the consideration of the actuator physical limitations, for example in the form of amplitude or rate limits. This becomes more relevant during failure scenarios than during normal operation since given the fault of an actuator, the remaining actuators need to compensate for the missing control forces and/or even counteract the disturbance forces corresponding to the faulty actuator. This can prevent the recovery of nominal performance during an actuator fault. In the context of redundant actuators, control allocation is one of the most effective and popular methods of dealing with the additional degrees of freedom and at the same time with actuator limits. Control allocation is the selection of actuator commands such that they satisfy the demanded forces and moments from the control law [81]. In this thesis **control allocation** algorithms are studied within the adaptive control formulation in order to exploit the hardware redundancy while simultaneously cope with the uncertain characteristics of faults and with the physical limits of actuators. In the following the state-of-the-art of the used methods is reviewed and an overview of the existing solutions for multicopter applications is given.

1.2.1 Fault Tolerant Control and Adaptive Control

Fault Tolerant Control (FTC) studies methods to automatically mitigate the effect of faults on the overall system performance and ultimately prevent a system failure. There exist two levels of fault tolerance that are of interest in safety-critical applications like the considered aircraft systems. The first one corresponds to a fail-operational system, i.e. a system that after a given fault is able to maintain the required performance level [12, p.8]. If this is not possible, the next goal is a fail-graceful system, which means that after a given fault the system remains operational but with a degraded performance [12, p.8]. The requirements strongly depend on the application scenario and in this thesis both cases are addressed for the hexacopter platform.

FTC approaches are commonly classified into active and passive approaches [41, 156]. Passive FTC corresponds to controllers that are robust against a certain class of presumed faults [41]. Here, the controller gains and structure are not modified online regardless of the presence of faults [156]. In contrast, active FTC reconfigures control action as a reaction to faults in order to guarantee stability and acceptable performance [156]. Active fault tolerant controllers consist of a fault diagnosis module and a control redesign module [12, p.2]. Classically, after the necessary fault information has been gathered, the controller actively reacts by a discrete change of the parameters and/or the change of the controller structure. For a good overview of FTC, the reader is referred to [121, 156] and to [57, 58, 122] for a newer revision. A review regarding aerospace applications can be found in [54, 126, 137].

In the literature, the classification of adaptive control within FTC is not consistent. This is mainly because the reaction to faults occurs in a continuous way in contrast to switching between controllers. Eterno et. al classify adaptive control under passive FTC since the controller reacts to the faults without any structural changes or discrete switches [41]. Zhang and Jiang classify it under active FTC since the parameter adaptation corresponds to an active reconfiguration [156].

Adaptive control was originally motivated by the idea of designing flight controllers that were able to perform over a large operational envelope as an extension to gain scheduling [132, p.2]. Historically, the theoretical foundations of adaptive control followed the crash of the X-15-3 aircraft in 1967 [38] as a result of a large development effort on closed-loop stability proofs and on improving the understanding on adaptive control. Most of the main text books deal with the control of linear plants [7, 78, 117, 132], sometimes including a nonlinear uncertainty term, and in [89] a nonlinear formulation is presented. The adaptive control approach studied within this thesis is **model reference adaptive control (MRAC)** [78, 92, 117, 132], which is one of the most predominant approaches in the literature and has been extensively studied. This facilitates the analysis of faults, overactuation and input saturation within a well-known framework. The main idea in MRAC is to design a controller that in closed loop with the plant and in the presence of uncertainties emulates a previously defined dynamics model which describes the desired system performance: the reference model.

MRAC approaches are usually divided into direct and indirect architectures or combinations of the two [11, 117, 132]. In direct MRAC, the controller parameters are updated directly without estimating the plant parameters beforehand. The description of how parameters change over time is called update law. In the direct MRAC case, it is driven by the error between the plant response and the desired response which is defined by the reference model. The direct MRAC controller structure is depicted in

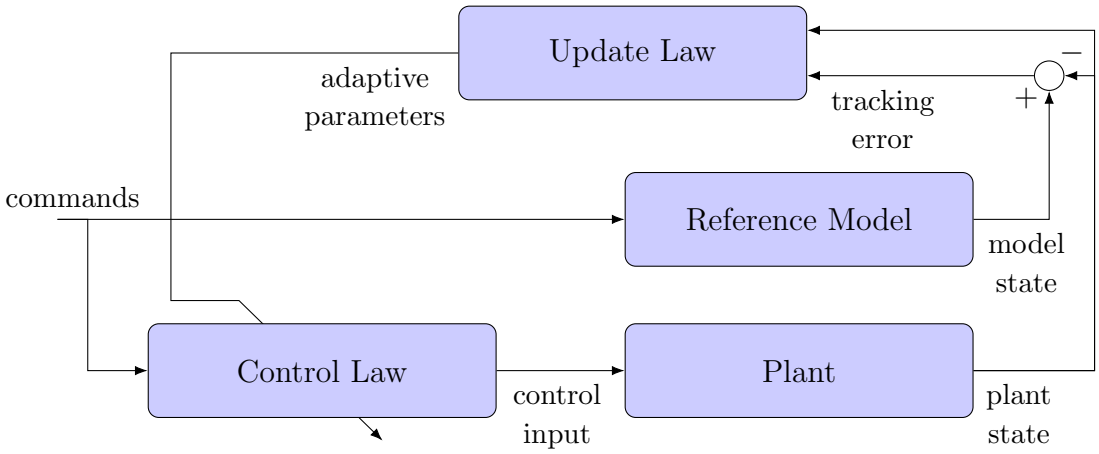


Figure 1.2: Controller Structure - Direct MRAC

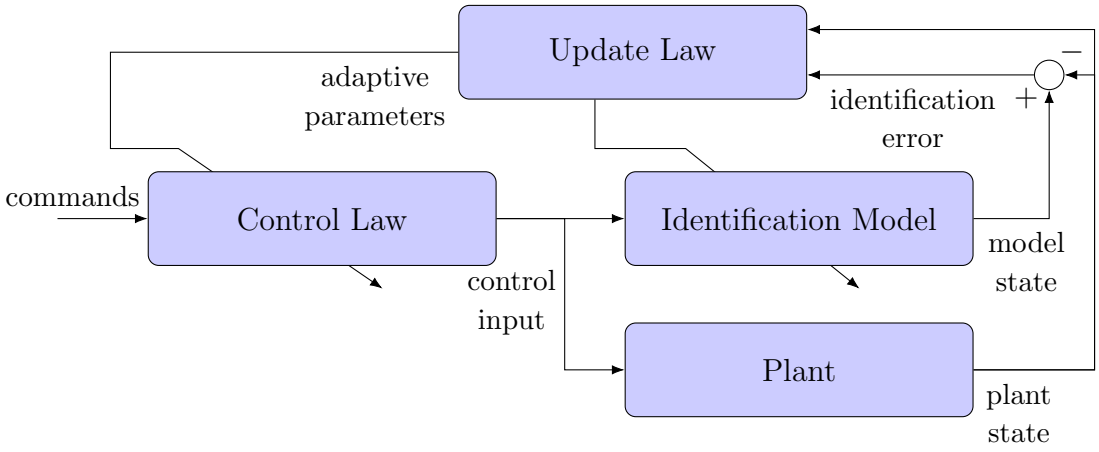


Figure 1.3: Controller Structure - Indirect MRAC

Figure 1.2. In indirect MRAC, the updated parameters correspond to the parameters of an identification model. These parameters are then used within the control law for computing the control input. In this case, the control law is designed such that the identification model with known parameters behaves like the reference model and the reference model is not directly implemented. Furthermore, the update law is driven by the error between the plant response and the response of the identification model. In Figure 1.3 the architecture of indirect MRAC is shown. In this thesis both structures will be discussed in the context of overactuated systems.

The idea of using adaptive control for improving fault tolerance is not something new. Because of its continuous parameter adaptation it has been identified that under ideal conditions, it provides graceful degradation and system recovery [41]. However, the fact that faults reflect on fast varying parameters has been a hurdle for developing successful application cases [12]. This can namely lead to unacceptably high control bandwidths and a decrease of robustness against unmodeled dynamics and noise [41]. The challenge of improving performance and robustness of MRAC in specific has been an active research field that has lead to the development of a large amount of modifications of the update laws, the reference model and the uncertainty parameterization.

The theoretical efforts were followed by various successful applications addressing actuator faults. Direct adaptive control augmented with a neural network was tested on the X-36 tailless aircraft under the effect of simulated failures as part of the RESTORE program [18]. In [21] flight test results on a modified F-15 aircraft with an immobilized control surface were presented. The approach corresponds to a direct adaptive control augmentation of a nonlinear dynamic inversion (NDI) controller. In [32] successful flight tests on a GTMAX helicopter UAV including actuator malfunctions have been presented. A multiple layer approach was proposed. A neural network adaptive controller is the first contingency line against faults. A fault detection and isolation (FDI) module and a reconfiguration mechanism switch between a bank of adaptive controllers for large faults. At a higher level, optimization is used to adjust the flight path to the degradation and finally a mission adaptation component adjusts the mission goals. In [27] successful flight tests of a GT Twinstar with 25% of its wing missing have been presented. Here, direct MRAC in combination with neural networks are compared to derivative free MRAC. Indirect adaptive control, specifically $\mathcal{L}_1$ control, has been flight tested using the NASA's subscale transport aircraft (AirSTAR) [63]. Various failure scenarios were tested including command latency, degraded aircraft stability and control effectiveness degradation.

In the fault tolerant adaptive control literature, it is difficult to find a systematic study and restrictive assumptions are often posed. One example is the assumption that the system has several actuators having a similar characteristics so that the problem can be address independently for each degree of control [151]. This is the case for divided control surfaces in fixed-wing vehicles but it is not applicable for most VTOL configurations. To model actuator uncertainties, the input matrix is usually multiplied by a control effectiveness matrix which is square and its dimensions correspond to the number of inputs. Another common assumption is that this control effectiveness matrix has a full rank, which considers degradation of control but excludes a total actuator failure. This assumption makes sense for and arises from non-redundant systems. For overactuated systems it is too restrictive. One important contribution towards a systematic study is the work of Tao [138] which presents methods and stability proofs of direct adaptive control for systems with actuator failures. Interestingly, to the best of the author's knowledge a comprehensive analysis of the implications of redundant actuators for the uncertainty parameterization is missing. In this work, the **uncertainty parameterization of overactuated systems** without similarity assumptions or full rank restrictions is studied in detail. Initial work has been presented by the author in [46,47].

Apart from actuator redundancy, control input saturation is a main topic in this thesis. In adaptive control, it has a twofold importance. On the one hand it sets limits on disturbance rejection and on the type of trajectories that can be followed. On the other hand if control input saturation is not taken into account, it can produce parameter estimation errors and windup [96, 134]. In the direct MRAC framework, one of the first ideas to counteract control magnitude saturation was the consideration of the control deficit within the reference model like in [85], where a single input system was considered. The positive μ-modification extends these results by guaranteeing that the control input will never saturate [90]. In [134] and [91], extensions for a multi-input plant were presented. Another alternative is pseudo control hedging [73, 82]. It considers a known actuator model, including magnitude and rate saturation or actuator dynamics, within the reference model in order to hide these effects from the parameter update laws. A similar approach is followed in [96]. Here a reference dynamical

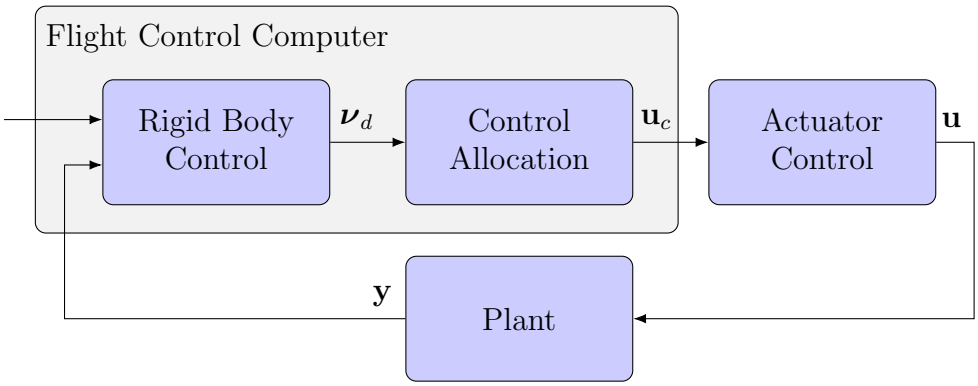


Figure 1.4: Control Allocation Formulation

system is constructed using the information of the control law in order to generate a reference signal that satisfies the control input limitations. Regarding indirect MRAC approaches, known actuator limits are mostly taken into account by integrating the actuator model within the identification model. One example of this is given in [97] which extends the results on $\mathcal{L}_1$ adaptive control for systems with input saturation. This approach is significantly simpler than for direct architectures. Further, different anti windup approaches have been combined with adaptive controllers in order to improve stability and performance results [83, 84, 139]. The presented approaches do not specifically consider the advantages of actuator redundancy. Therefore, in this thesis control allocation is chosen for addressing control input saturation as detailed in the next section.

1.2.2 Control Allocation

Fault tolerance, overactuation and control allocation are strongly related. Usually, fault tolerance requirements lead to an overactuated mechanical design and control allocation methods are a possible way of fully exploiting the gained degrees of freedom. For overactuated mechanical systems, the control algorithms are often separated into different modules as depicted in Figure 1.4 [81]. The rigid body motion control computes virtual controls $\boldsymbol{\nu}_d(t)$ according to the given task. In aerospace applications the virtual controls are usually forces or moments that the flight controller computes in order to fulfill the control task. In multirotor applications, the virtual controls are commonly the total thrust T and the desired control moments L, M and N about each body fixed axis. The next module corresponds to the control allocation. Its primary task is to compute a control input $\mathbf{u}(t)$ such that the desired virtual control $\boldsymbol{\nu}_d(t)$ is achieved under consideration of actuator limits. The last module may consist of a low-level controller for every actuator system, typically as a separated computation unit. For overactuated systems the control allocation solution might not be unique. In this case, it is possible to take into account secondary goals like minimum energy consumption or minimum structural load. Further, it is possible to implement a strategy for degraded control situations.

One of the main design drivers for the control allocation is the trade-off between the computational resources and the control allocation capabilities. The solutions range from fast and simple pseudo-inverse based methods to complex iterative numerical

Computational Resources	Objectives
maximum execution time	**achieving attainable virtual controls**
memory space	**handling unattainable virtual controls**
Actuator Model	**Constraints**
linear or nonlinear	**input magnitude limits**
static or dynamic	input rate limits
time constant or **time varying**	

Table 1.1: Control Allocation Design Drivers

optimization methods where a large variety of constraint types and optimization goals can be specified. In the case of real-time applications, the maximum execution time is the limiting factor. Hence, it is important to select the most simple problem formulation that covers the necessary system characteristics in order to fulfill the control task. The problem formulation is characterized by the actuator model, the control allocation objectives and the constraints as detailed in Table 1.1. This list reflects only the main characteristics and is by no means exhaustive. For fault tolerant systems, an important differentiation is between attainable and unattainable solutions. For the control allocation problem a desired virtual control $\boldsymbol{\nu}_d(t)$ is called attainable if there exists a control input $\mathbf{u}(t)$ that satisfies the constraints and generates $\boldsymbol{\nu}_d(t)$, otherwise it is called unattainable. The set of all attainable virtual controls $\boldsymbol{\nu}_d(t)$ is called the attainable control set (ACS).

The main design aspects corresponding to the VTOL application scenario as presented in Chapter 2 are highlighted in Table 1.1. The actuator model can be linear and static but a time-varying model needs to be accommodated. Moreover, the constraints that need to be taken into account are magnitude limits but not rate limits. The linear formulation is achieved by compensating the effector nonlinearities through a nonlinear inversion mapping. The actuator dynamics and input rate limits can be neglected since the rigid body dynamics are considerably slower. Further, the use of adaptive control for compensating for faults results in a time-varying actuator model. The considered objectives in Table 1.1 basically reflect the main tasks of the control allocation and no secondary goals are included. In the hexacopter case, unattainable solutions will have a central role since their occurrence cannot be avoided during failure scenarios as will be shown latter. There are mainly three approaches in the literature to deal with unattainable virtual controls: The minimization of a norm of the virtual control error [81], the prioritization of specific virtual controls [35] and the preservation of the virtual control vector direction which corresponds to the direct allocation approach in the sense of Durham [36]. Taking the highlighted factors in Table 1.1, a summary of candidate control allocation approaches is given in the following. A general review of control allocation methods can be found in [81, 120].

The simplest actuator model is the well understood problem of a linear, static and time-invariant actuator model. The most common solution is the pseudo-inverse in combination with saturation of the inputs that exceed their limits [120]. The reasons for its popularity are twofold. This approach yields the minimum 2-norm of the control input $\mathbf{u}(t)$ if no inputs saturate and therefore relates to the minimization of the used

control input power. Furthermore, only minimum computational resources are needed since it only involves a matrix multiplication and a saturation function. The main disadvantage is that the simple saturation or clipping of the inputs will alter the resulting virtual controls, even in cases were an attainable solution exists. The use of a weighted pseudo-inverse or tailored generalized inverses can improve the results. Nevertheless, no single generalized inverse is able to map the whole set of attainable virtual controls to the set of available control inputs using the simple saturation [36]. Improvements can be achieved by using iterative methods like the redistributed pseudo-inverse or daisy chaining. Nevertheless, they do not guarantee that attainable solutions are found nor that the allocation error is minimized in some sense [81]. Two drawbacks prevent the direct use of these methods for our application: the need of a continuous computation of a matrix inverse for a time-varying matrix and the inability to implement a suitable strategy for unattainable virtual controls.

Apart from generalized inverse approaches, the constrained linear control allocation problem has been addressed using optimization routines like linear programming [13, 123] and quadratic programming [65]. The most common solution routines are active set and interior point methods [119]. For linear program solving, the simplex method is another established alternative [119]. Although optimization approaches are more flexible regarding optimization goals and constraints, it is hard to give guarantees on a maximum execution time and therefore practically a trade-off with respect to optimality needs to be made [81]. To which extend suboptimal results can be tolerated, is not easy to answer. Therefore, in this thesis an alternative to the mentioned optimization algorithms is proposed. Nonetheless, it is worth to mention that the formal verification of optimization algorithms for safety-critical applications remains an active research field [28, 29, 80, 129, 150].

Control allocation methods in combination with adaptive control can be divided into direct and indirect approaches. Direct approaches are presented in [50,98,143,144]. Within an MRAC formulation, the control allocation matrix is considered as an unknown parameter and it is consequently updated by an update law. The input limits are indirectly enforced by saturating the desired virtual controls and by limiting the control allocation matrix using the projection operator. The bounds on the virtual controls and adaptive parameters are fixed and need to hold simultaneously and are therefore conservative. This holds especially for systems with high coupling of actuator and control axes. Furthermore, a possibility of implementing a strategy for unattainable virtual controls in this framework is not known to the author.

Indirect approaches are more flexible since it is possible to combine the estimation of the control effectiveness matrix with a control allocation approach that is able to handle a time-varying actuator model. In [22], a time-windowed and a recursive parameter estimation algorithms are used to estimate the control effectiveness matrix. This is then combined with a quadratic programming solver to allocate the control inputs. In [42, 43], the control effectiveness matrix is estimated using a concurrent learning method and it is combined with a gradient-based optimization as a control allocation. Here, a prioritization strategy is implemented for unattainable virtual controls. In [140, 141], adaptive control in combination with dynamic control allocation is used. In this case the control allocation problem is not solved every time step, but a control allocation update law, similar to a gradient optimization, solves the optimization problem dynamically.

1.2.3 Multirotor Control

In this section, a review of the state of the art regarding multirotor control is presented. The focus of this review lies on fault tolerant and adaptive control and on experimental results demonstrating the maturity of the developed algorithms. In order to measure the suitability to a real world application, the following factors are taken into account:

- Type of the experimental setting,
- Utilizable flight envelope,
- Presented maneuvers,
- And considered actuator fault types.

The actuator faults are differentiated between **effectiveness reduction** and **total failure**. Further, it is differentiated between a **full** and a **degraded control authority** case. Degraded control authority occurs if after the fault the number of system outputs that can be controlled is reduced, otherwise it is a full authority case. Avoiding degraded control authority is one of the main reasons for actuator redundancy. Nevertheless, this can occur in overactuated systems and in fact the hexacopter is an example of this. This differentiation is made since degraded control authority represents a more complex case for the flight controller. The last important characterization is if the faults are assumed to be known or if the uncertainties are considered in the presented experimental results showing a more realistic scenario.

First experimental autopilots controlled the attitude or position only at low speeds close to hover within a laboratory environment [15,16,23,64,145]. First successful outdoor flights were shown by Hoffmann et al. with velocities up to $3.5\ [m/s]$ using PID control for attitude and position control loops [70]. Higher speeds up to $6\ [m/s]$ were demonstrated by Kendoul et al. using a cascaded nonlinear dynamic inversion approach [86]. Subsequently, the author presented a flight with velocities up to $7.5\ [m/s]$ using a position tracking backstepping controller [49]. Finally, Raffler et al. showed a high speed flight up to $12\ [m/s]$ using a path controller augmented by pseudo control hedging [128]. The controller prioritizes the cross range error over the velocity error along the path. All of these outdoor results used GPS position measurements. Mellinger et al. [107,108] and Lupashin et al. [102] presented other types of high dynamic maneuvers like flying through narrow, vertical gaps and perching on inverted surfaces. They were tested within a laboratory environment and using specialized controllers for each maneuver.

An important design choice that directly affects the utilizable flight envelope is the attitude parameterization. In order to maximize the flight envelope, representations with singularities such as Euler angles are to be avoided. Various authors have proposed quaternion representations [3,20,30,56] while others used the special orthogonal group $SO(3)$ [95]. The $SO(3)$ attitude parameterization has been presented in combination with robust adaptive control [93] and $\mathcal{L}_1$ adaptive control [88]. A parameter variant that exploits the structure of the multirotor dynamics model is the reduced attitude representation in $\mathbb{S}^2$ as presented in [48,55,149]. Details on the different attitude parameterizations are presented in Chapter 2 (Multirotor Dynamics Model).

A general analysis of the influence of faults for multicopter systems can be found in [105,133,146]. It has been found that the standard hexacopter configuration (see Figure

2.1a) is not able to maintain the same number of independently controlled outputs after a failure although it is an overactuated system. With a different selection of the rotational direction of the propellers it is possible to achieve full authority in some specific failure cases but not in all. Therefore, a degraded control strategy that does not control the yaw motion has been proposed [34, 42]. Furthermore, full control authority can be regained by letting one propeller rotate in inverse direction [1] or by tilting the rotation axes of all the rotors [60, 111].

The first flight tests considering actuator effectiveness reduction were all done in a laboratory environment. The first one is presented by Michini et al. considering a 50% degradation which is not known to the controller [112]. The controller consists of a quaternion-based attitude inner loop and a velocity outer loop that includes an $\mathcal{L}_1$ adaptive control augmentation. Dydek et al. presented the first direct MRAC position controller derived using a linear hover model of the quadrotor and applied to the unknown control effectiveness reduction case [37]. These results were improved in [39] using a combined MRAC approach. Specifically, smoother parameter estimates allowed for higher adaptation gains. Furthermore, the degradation was not simulated but the blades of one propeller were cut to reduce its original length by 25%. Again for the unknown control effectiveness reduction case, Chamseddine et al. compared different MRAC variations and an LQR controller [24]. For the unknown control effectiveness reduction case, Amoozgar et al. proposed a fuzzy gain scheduled PID controller and compared its performance with a conventional PID controller [6]. The first outdoor experiments using adaptive control were presented by Mallikarjunan et al. for a quadrotor and a hexarotor [103]. Different attitude controllers are compared using backstepping and $\mathcal{L}_1$ PWC adaptive control as a basis. In the paper, it is mentioned that the hexacopter with actuator failure was successfully flight tested, but there is no further information or results presented.

Achtelik et al. presented the first experimental results for the case of a known total actuator failure [1]. The platform used is a hexarotor which is able to rotate its rotors in both directions in order to avoid degraded control authority. The strategy is to use four rotors to control thrust, roll and pitch moments and to use the remaining rotor only for yaw control. This last rotor is the one that changes its rotational direction. Müller et al. presented the first degraded control strategy for a known total actuator failure validated experimentally [114]. The flight tests were done using an optical positioning system and included the cases of one and two propeller failures for a quadrotor system. The strategy is to control a selected body-fixed yaw-like axis as a reduced attitude instead of controlling the full three-dimensional attitude. Therefore, the rotation about the axis remains uncontrolled but by controlling its direction position control could still be achieved. The rotation axis depends on the failure case and therefore a dedicated controller was developed for each failure case. The absolute value of the rotational rate during position hold gives an impression of the dynamics of the system. It was about 822 $[deg/s]$ for the case of one actuator failure and 1553 $[deg/s]$ for the case of two actuator failures. Again for the known actuator failure case a degraded control strategy was presented by Du et al. for a hexacopter attitude control [34]. The strategy consists on disregarding yaw moment commands through an update of the control allocation matrix. This leads to the possibility of controlling position of the hexarotor while the yaw motion is uncontrolled. During the experiment, the yaw rate reached approximately 300 $[deg/s]$.

The first results regarding an unknown total actuator failure are found in [147] and

	Platform			Validation			Fault Type							
	Quadrotor	Hexarotor	Other	Simulation	Lab. experiment	Experiment	None	Effectiveness Reduction	Failure (full authority)	Failure (degraded authority)	Unknown Fault	Velocity [m/s]	Mass [kg]
[23]	✓				✓		✓						0.5[1]
[145]	✓				✓		✓					0.25	0.5
[70]	✓					✓	✓					3.5	1.6[2]
[16]	✓				✓		✓						0.52[1]
[64]	✓				✓		✓						0.57
[86]	✓					✓	✓					6	0.6[2]
[112]	✓				✓			✓			✓	1	0.5
[37]	✓				✓			✓			✓	< 1	0.5[1]
[107]	✓				✓		✓					3.6	0.5[1]
[24]	✓				✓			✓			✓	< 1	1.4[1]
[131]	✓				✓			✓					1.4[1]
[1]		✓			✓	✓			✓			2	1.2[1]
[6]	✓				✓			✓			✓	0	1.4[1]
[104]			✓	✓					✓				1.2
[103]	✓	✓	✓			✓				✓*	✓		
[128]	✓					✓	✓					12	0.5[1]
[5]			✓	✓				✓	✓		✓		1.3
[33]		✓		✓				✓			✓	1	1.54
[39]	✓				✓			✓			✓		0.5[1]
[114]	✓				✓					✓			0.5
[130]	✓			✓				✓					1
[34]		✓				✓				✓		1	1.54
[147]		✓			✓				✓		✓	1	1.03[1]
[154]		✓				✓			✓			3	
[53]	✓				✓		✓					5.7	0.5
[109]	✓					✓		✓				1	0.42
[157]	✓					✓	✓						2
[60]		✓				✓			✓		✓		1.4
[118]		✓		✓	✓			✓			✓		1.54
[135]		✓		✓				✓	✓		✓		2
[106]		✓				✓			✓				

1: Values taken from a related source.

2: Mass varies with payload and this is one configuration.

✓*: Information is missing.

Table 1.2: Classification of Multirotor Control

in [42]. Vey and Lunze [147] presented an active FTC approach for position control of a hexacopter which combines an FDI module consisting of a bank of observers with a reconfiguration module based on the virtual actuator concept. In this case the reconfiguration is equivalent to an update of the control allocation. The solution works for a full control authority case corresponding to 4 out of 6 possible actuator failures in the selected configuration. On the other hand, Falconi et al. [42] presented a controller that is able to handle the degraded control authority case. The FTC approach is composed of an adaptive estimation of the control effectiveness of the propellers and an optimization-based control allocation. The control allocation prioritizes thrust, pitch and roll moments over the yaw moments similarly to the results of [114] and [34]. In the experiment, the hover yaw rate is between 80-120 $[deg/s]$. The first position controller based on GPS measurements used for compensating a total failure in the full control authority case was presented by Yang et al. [154]. Mazeh et al. presented afterwards results handling the known failure of two actuators [106]. In both cases, an active FTC structure was used and the failure was assumed to be known. The reconfiguration consisted in an exchange of the control allocation matrix depending on the failure case. The first outdoor experiments using position control compensating an unknown total failure in the degraded control authority case were presented by Falconi et al. [43] extending the results in [42].

Table 1.2 gives a general overview of multirotor control regarding platform, type of validation, considered fault types, maximal velocity and aircraft mass. The classification is not focused on the type of controllers that have been proposed, but rather on categories that allow to compare the results towards their applicability on a real world scenario. The considered validation forms are simulation, laboratory experiment or experiment. An experiment is classified as laboratory experiment if either the position and/or attitude measurements come from an infrastructure that is external to the vehicle, e.g. from a camera motion system, or if external computational resources are needed and hence the algorithms do not run onboard of the vehicle. The considered fault types correspond to the description given at the beginning of this section. Another important factor for the usefulness of a multirotor controller is the maximal horizontal velocity that can be reached. This allows for a larger flight envelope and shows robustness of the controller against the increased aerodynamic disturbances at higher velocities. Finally, the last category is the weight of the used platform. The author's publications are summarized in Table 1.3 in the same manner as in Table 1.2. The overview shows that although there are several results addressing the identified challenges in Section 1.2, very few are able to tackle them simultaneously and even less show representative experimental results. One clear milestone in multirotor fault tolerant control is dealing with an unknown total actuator failure. For the hexarotor system, the degraded authority failure case has increased complexity due to the drastically reduced control authority and therefore the need for careful handling of actuator redundancy and limits.

	Platform			Validation			Fault Type						
	Quadrotor	Hexarotor	Other	Simulation	Lab. experiment	Experiment	None	Effectiveness Reduction	Failure (full authority)	Failure (degraded authority)	Unknown Fault	Velocity [m/s]	Mass [kg]
First Author Publications													
[44]	✓			✓			✓					1	0.5[1]
[48]		✓		✓			✓					2	1.2[1]
[49]		✓				✓	✓					7.5	1.2[1]
[45]		✓		✓				✓				2	1.2[1]
[42]		✓				✓				✓	✓	0	1.2[1]
[47]		✓		✓					✓				1.2[1]
[50]		✓		✓					✓			5	1.2[1]
[51]		✓				✓		✓			✓		1.2[1]
[52]		✓				✓		✓			✓		1.2[1]
[46]		✓		✓					✓				1.2[1]
[43]		✓				✓		✓		✓	✓	2	1.2[1]
Other Publications													
[3]		✓		✓		✓	✓						1.2[1]
[66]		✓		✓		✓		✓			✓		1.2[1]
[115]		✓		✓		✓		✓			✓		1.2
[4]		✓		✓		✓		✓			✓		1.2[1]
[62]		✓		✓				✓					1.2[1]
[61]		✓		✓				✓					1.2[1]

1: values taken from a related source.

Table 1.3: Classification of Multirotor Control - Author's publications

1.3 Problem Statement and Objectives

This thesis addresses the challenge of developing an adaptive fault tolerant controller for a VTOL multicopter system that considers actuator redundancy and actuator limits and implements a strategy for degraded control authority. Considering the presented state of the art, the main objectives are separated in two categories: control theoretical and application related objectives.

The control theoretical objectives are the following:

1. **Systematic analysis of adaptive control methods for systems with redundant actuators**: Analysis of the impact of the structure of overactuated systems regarding uncertainty parameterization.

2. **Study of adaptive control methods regarding actuator fault tolerance in the case of overactuated systems**: Given that actuator redundancy provides additional degrees of freedom for controlling a plant, it is expected that fault tolerance is possible. Therefore, the questions arise whether adaptive control approaches can verify this hypothesis given the fact that faults are unknown and which conditions are necessary in this case.

3. **Derivation of an approach that exploits actuator redundancy for fault tolerance within the adaptive control framework**.

The following goals are application related and intend to increase the usability of the proposed theoretical solutions to real world problems. The goals are:

4. **Testing the theory on real hardware**.

5. **Running the algorithms online onboard of the aircraft**.

6. **Lean, compact and unified structure of the controller**.

7. **Developing of a controller that deals with unknown actuator faults and failures**: Although it is a natural requirement for fault tolerant systems, there are several partial results in the literature only dealing with control reconfiguration under a given known fault.

8. **Inclusion of a strategy for the degraded control authority case**: This is a necessary step in order to improve fault tolerance of the hexacopter system.

9. **The controller should not be a limitation of the physical system**: This goal is rather ambitious and the best possible result. It means that the designed controller should be able to stabilize the plant and maintain certain performance level as long as this is physically possible.

1.4 Contribution

The contributions of this thesis compared to the state of the art are stated in the following.

Parameter Reduction due to Overactuation (PRO)

This approach is a new uncertainty parameterization within an adaptive control framework developed for systems with redundant actuators. It reduces the uncertain parameters to the minimum set without reducing the amount of information gathered. This improves the usage of computational resources without affecting the controller performance. Initial work has been presented in [46, 47].

Control Authority Conditions for Overactuated Systems

Based on the new uncertainty parameterization (PRO), control authority conditions are formulated for systems with redundant actuators such that actuator failure cases can be considered within the stability proofs of model reference adaptive control (MRAC). The conditions are derived for direct and indirect structures. Initial work has been presented in [46, 47].

Predictor-Based Adaptive Control Allocation (P-ACA)

Based on Predictor-Based MRAC and the Parameter Reduction due to Overactuation (PRO), a framework for integrating control allocation algorithms is presented. Depending on the control allocation approach this permits the consideration of actuator limits and a better use of the redundancy degrees of freedom.

Prioritizing Control Allocation (PRIO CA)

In order to take actuator limits into account and to implement a strategy for degraded control authority for VTOL aircraft, the Prioritizing Control Allocation (PRIO CA) is developed. In the case of unattainable or suboptimal solutions, the algorithm prioritizes virtual controls in a given order such that control degradation is reduced to a minimum. This approach has the advantage of having a fixed maximum number of iterations to return a solution.

Extension of the SVD Udpate Algorithm to Non-Square Matrices

The Prioritizing Control Allocation (PRIO CA) needs the knowledge of the Singular Value Decomposition (SVD) of the input matrix. This is done using the SVD update algorithm presented in [69]. In this work it is extended to non-square matrices and a numerical drift correction term is added.

Attitude Parameterization and Yaw Motion Decoupling

In order to maximize the utilizable flight envelope, the reduced attitude vector in $\mathbb{S}^2$ is used. The advantage is that non-uniqueness and singularities are avoided. Furthermore, the stability properties are shown decoupled from the yaw motion. This enhances the advantages of the Prioritizing Control Allocation (PRIO CA) during degraded control situations. This is the first time that this type of parameterization is analyzed using an adaptive controller. Non-adaptive controllers have been presented

in [48, 51, 55, 149] and previous adaptive geometric results use a full attitude parameterization in $SO(3)$ [88, 93].

Experimental Validation

The presented controller is validated using the hexacopter testbed. During the experiments, the faults are unknown to the controller, all computations are done onboard, and no external sensors were used. Furthermore, a single unified controller handles the nominal and all the faulty cases. The test cases include two hexacopter configurations with representative failure cases for demonstrating repeatable results. The experiments include post-failure performance analysis and include software induced and hardware failure scenarios.

Experimental Validation - High Speed

The test cases include hover flight as well as high speed cruise flight. These are the first results showing the failure scenario occurring during high speed cruise flight at $10.9[m/s]$. This shows the robustness of the controller against unmodeled aerodynamic effects which increase overproportionally with respect to the airspeed.

1.5 Scope of the Thesis

This thesis deals with continuous and deterministic adaptive control approaches. Discrete and stochastic adaptive control approaches are out of the scope. Furthermore, the case of state feedback is analyzed and the output feedback case is not considered. The control allocation is restricted to linear actuator models which may result from a transformation of a nonlinear model. High level control and mission management are not part of the focus of the thesis. The studied hexacopter controllers have velocity, attitude or angular rates as a tracking output.

1.6 Structure of the Thesis

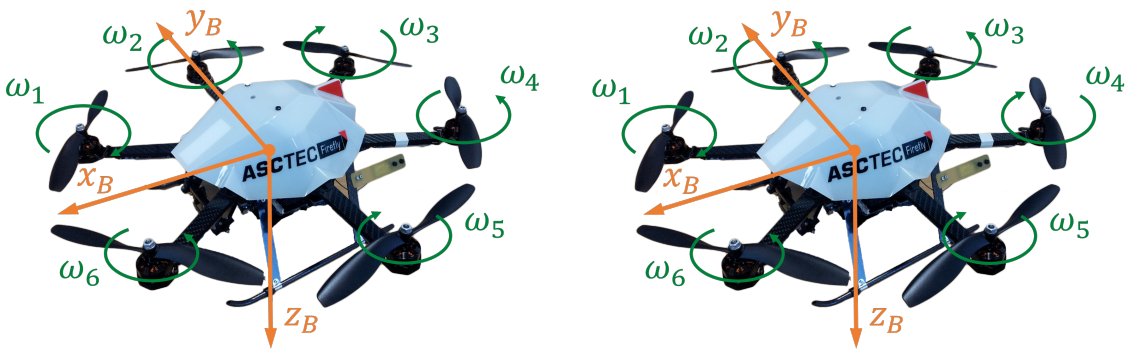
The thesis is organized as follows. In **Chapter 2 (Multirotor Dynamics Model)**, the hexarotor testbed is introduced regarding the important aspects for developing the flight controller: inertia characteristics, sensors, actuators and flight control computer. In addition, the dynamics model of a multirotor suitable for control law design is derived and the main assumptions are stated. In **Chapter 3 (Adaptive Fault Tolerant Control)**, the Parameter Reduction due to Overactuation (PRO) method is presented as a tool for analyzing and designing adaptive fault tolerant controllers for overactuated systems. The implications of actuator redundancy are examined in direct and indirect model reference adaptive control (MRAC). In **Chapter 4 (Adaptive Control Allocation)**, control allocation is integrated within the Predictor-Based MRAC framework and a reduced order predictor is derived. Subsequently, the Prioritizing Control Allocation (PRIO CA) strategy is described along with the Singular Value Decomposition (SVD) update algorithm.

In **Chapter 5 (Multirotor Controllers)**, the adaptive fault tolerant controller is derived for multicopters based on the presented approaches. The attitude controller uses a geometric approach in order to maximize the flight envelope and takes the degraded control authority case into account. In **Chapter 6 (Experimental Results)**, flight test results containing total actuator failure are presented. The degraded and full authority cases are included as well as high velocity flights. In **Chapter 7 (Conclusion)**, the results are summarized. Based upon them, further research topics are postulated.

Chapter 2

Multirotor Dynamics Model

In this chapter, the dynamics model of a multirotor system is derived. The main testbed in this work is the AscTec hexacopter *Firefly* [1] which is depicted in Figure 2.1. It is a VTOL aircraft with redundant actuators. It has 6 propulsion units consisting of a fixed-pitch propeller, an electric motor and a motor speed controller. Fixed-pitch propellers simplify the mechanical structure as no swashplate is required [71, 95]. Furthermore because of the propeller size, no flap or lag hinges are needed. The control inputs of the system are the 6 rotational rates of the propellers. The propellers are able to produce collective thrust and moments about the three body-fixed axes. In this thesis, two actuator configurations are used which only differ in the rotation direction of the third and fourth propellers as shown in Figures 2.1a and 2.1b. This has implications regarding the fault tolerance of the system as explained in Section 2.3.



(a) Actuator Configuration 1

(b) Actuator Configuration 2

Figure 2.1: AscTec Hexacopter Firefly [1] - Gumstix Configuration

Depending on the planed mission, different types of controllers are needed for multirotor platforms. The dynamics of these platforms are characterized by relatively fast poles because of the small aircraft inertia and the high actuator bandwidth. Therefore, multicopters are always equipped with at least an attitude controller. Rate-command-attitude-hold (RCAH) controllers are also available but require higher pilot skills. Another usual controller is a velocity-command-position-hold (VCPH) for piloted flights and position-command-position-hold (PCPH) for pre-programmed flights. The latter controllers have the highest automation degree and require low pilot skills for a stable flight.

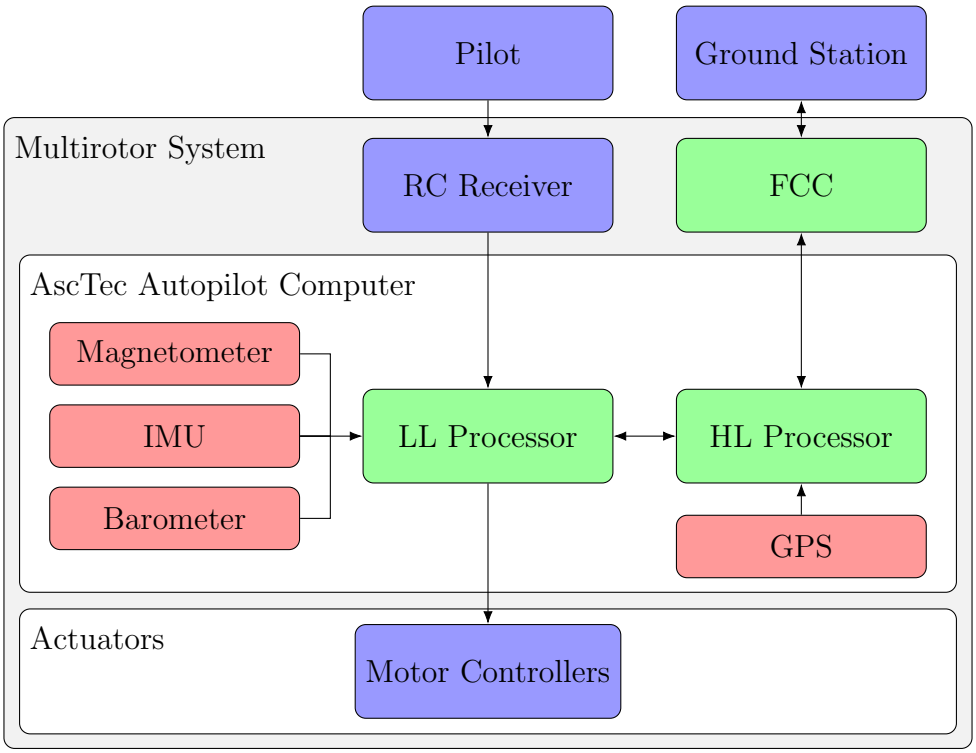


Figure 2.2: Multirotor System Architecture

2.1 System Architecture

The system architecture can be seen in Figure 2.2. The available sensors are a 3-axis magnetometer, a barometer, a global positioning system (GPS) module and an inertial measurement unit (IMU). The IMU comprises a 3-axis accelerometer and a 3-axis gyro. The main sensor characteristics can be seen in Table 2.1. The hexarotor is equipped with two computing units: the AscTec autopilot computer and the flight control computer (FCC). The implemented algorithms run on the FCC which is a Gumstix Overo FireSTORM Computer-On-Module (CoM). On the AscTec autopilot board there are two processors. The Low-Level (LL) processor provides the AscTec autopilot functions and can be used in flight tests as a back-up controller. The LL processor also receives the pilot inputs and sends the commands to the motor controllers of the propulsion units. The High-Level (HL) processor is connected to the GPS module and is used as a communication link between FCC and the LL processor. The FCC directly communicates with the ground station via a wireless serial communication link.

2.2 Rigid Body Motion

In order to describe the rigid body motion of a multirotor, the coordinate frames listed in Table 2.2 are used. The Earth centered inertial frame (ECI) system has its origin at the Earth's center and does not rotate. For aircraft applications, the translation of its origin can be neglected and this system is considered the inertial reference frame which is used for formulating Newton's second law. Closely related, the Earth-centered Earth-Fixed frame (ECEF) system has its origin at the Earth's center but it does rotate about

Gyroscopes			Accelerometers		
Description	Value	Unit	Description	Value	Unit
Bias	[0.5,-0.3,0.2]	deg/s	Bias	[0.03;-0.04;0.08]	m/s^2
Noise variance	[2.25,2.25,2.25]	deg^2/s^2	Noise variance	[0.04,0.04,0.04]	m^2/s^4
Scale Factor	[0.01,0.03,0.04]	-	Scale Factor	[0.1,0.05,0.08]	-
Alignment	$\mathbf{I}_3$	-	Alignment	$\mathbf{I}_3$	-
Resolution	0.015	deg/s	Resolution	0.001	m/s^2
Sample time	0.001	s	Sample time	0.003	s
Latency	0.001	s	Latency	0.003	s
Lower Limit	-350	deg/s	Lower Limit	-25	m/s^s
Upper Limit	350	deg/s	Upper Limit	25	m/s^2

Table 2.1: Sensor Characteristics

its z-axis following the Earth's rotation rate $\vec{\boldsymbol{\omega}}^{IE}$. In order to describe the attitude of an aircraft, the north-east-down frame (NED) system is used as a reference. Its origin is an aircraft fixed reference point (R). The z-axis points downwards perpendicular to the tangent plane on the Earth's surface. The x and y axes point to the north and east respectively. The body fixed system has its origin at the reference point of the aircraft and its orientation is coupled to the aircraft orientation. The body-fixed frame corresponding to the hexarotor can be seen in Figure 2.1 respectively. Finally, the local navigation system has its origin at a fixed point on the Earth's surface. Its axes are fixed and coincide with the NED axes given that the reference point R is the origin of the N frame. It is an Euclidean space that can be used for aircraft of limited range and per definition its rotation respect to the ECEF frame is $\vec{\boldsymbol{\omega}}^{EN} = 0$. A detailed description of the different coordinate frames can be found in Appendix A.

In order to specify the coordinate system information, the following nomenclature is used in this thesis. Let us consider the angular rate $\left(\vec{\boldsymbol{\omega}}^{IB}\right)_B(t) \in \mathbb{R}^3$. The superscript IB denotes that it corresponds to the rotation of the body-fixed frame B with respect to the ECI frame I. The subscript outside the brackets B denotes that the vector is expressed in the body-fixed frame. Bold variables refer to a non-scalar variable and the arrow $\overrightarrow{(\cdot)}$ corresponds to a vector in the Euclidean space $\mathbb{R}^3$. An example of a linear acceleration is given by $\left(\dot{\vec{\mathbf{v}}}^R\right)_B^{IB}(t) \in \mathbb{R}^3$. Here, the superscript R defines the point whose motion is described. The subscript outside the brackets B refers to the coordinate system the variable is expressed in and the superscript IB defines in which frame the derivatives have been taken. In this case we have two derivatives that can be expanded as follows

$$\left(\dot{\vec{\mathbf{v}}}^R\right)_B^{IB}(t) = \frac{d}{dt}\left(\vec{\mathbf{v}}^R\right)_B^I(t),$$
$$\left(\vec{\mathbf{v}}^R\right)_B^I(t) = \mathbf{M}_{BI} \cdot \frac{d}{dt}\left(\vec{\mathbf{r}}^R\right)_I^I(t).$$

Hence, the first derivative was taken in the I-frame and the second one in the B-frame. In between a coordinate transformation was needed using the rotation matrix $\mathbf{M}_{BI}(t) \in \mathbb{R}^3$, which rotates a vector from the I-frame into the B-frame. Finally, the velocities correspond to kinematic velocities if it is not otherwise stated (in contrast to aerodynamic velocities).

Coordinate System	Abbreviation	Index
Earth-Centered Inertial System	ECI	I
Earth-Centered Earth-Fixed System	ECEF	E
North-East-Down System	NED	O
Body Fixed System	B	B
i-th Propeller Fixed System	-	P_i
Local Navigation System	N	N

Table 2.2: Coordinate Systems

The control dynamics model should contain the main effects that dictate the rigid body motion but be as simple as possible in order to allow a stability assessment and a real time implementation of the control algorithms. Note that this requirements differ from the ones of a simulation model. Therefore, the following assumptions are made.

Assumption 2.1 (Flat and non-rotating Earth) *Because of the usual range and achievable velocities of a multirotor system, the Earth can be modeled as flat and non-rotating. This means that the transport rate $\vec{\boldsymbol{\omega}}^{EO}$ and the Earth's rotation $\vec{\boldsymbol{\omega}}^{IE}$ are negligible.*

Assumption 2.2 (Rigid Body) *Any two points of the aircraft have a constant distance between each other. Furthermore, the aircraft has a constant mass and a constant moment of inertia (mass distribution).*

Assumption 2.3 (Center of Gravity) *The systems considered here have an axis-symmetric mass distribution and therefore it is assumed that the center of gravity coincides with the origin of the body-fixed axes.*

Assumption 2.4 (Forces in the Rotor Plane) *The considered multicopter systems have an even number of equal propulsion units and all rotors are aligned in the same plane. Because there is an equal number of rotors spinning clockwise and counter-clockwise, it is assumed that the forces in the rotor plane are small and cancel each other so that they can be neglected.*

Assumption 2.5 (Motor Dynamics) *The motor dynamics are much faster than the rigid body dynamics and can therefore be neglected.*

Assumption 2.6 (Aerodynamic Forces and Moments) *Aerodynamic forces and moments regarding the body of the aircraft are neglected since they are much smaller than the propulsion forces and can be neglected, i.e.* $\vec{\mathbf{F}}_A^R = \mathbf{0}, \overrightarrow{\mathbf{M}}_A^R = \mathbf{0}$.

The motion of a rigid body can be completely described by its states: position, attitude, linear and angular velocities. Therefore in the following sections, the differential equations governing these states are derived under the given assumptions.

2.2.1 Dynamics

In this thesis, the focus lies on controlling the rigid body motion of the multirotor systems. The dynamics model is based on the conservation of linear and angular momentum as dictated by Newton's second law. It follows that the changes of linear and

angular momentum with respect to an inertial frame correspond to the total forces and moments acting on the system respectively. Under the Assumptions 2.1 (Flat and non-rotating Earth), 2.2 (Rigid Body) and 2.3 (Center of Gravity), the dynamic equations describing the motion with respect to the center of gravity G can be expressed in the body fixed frame as follows [19]

$$\begin{aligned} m \cdot \left(\dot{\vec{\mathbf{v}}}^{R}\right)_{B}^{IB}(t) &= -m\left(\left(\vec{\boldsymbol{\omega}}^{IB}\right)_{B}(t) \times \left(\vec{\mathbf{v}}^{R}\right)_{B}^{I}(t)\right) + \left(\vec{\mathbf{F}}_{T}^{R}\right)_{B}(t), \\ \left(\mathbf{I}^{R}\right)_{BB}\left(\dot{\vec{\boldsymbol{\omega}}}^{IB}\right)_{B}^{B}(t) &= -\left(\vec{\boldsymbol{\omega}}^{IB}\right)_{B}(t) \times \left(\left(\mathbf{I}^{R}\right)_{BB}\left(\vec{\boldsymbol{\omega}}^{IB}\right)_{B}(t)\right) + \left(\overrightarrow{\mathbf{M}}_{T}^{R}\right)_{B}(t). \end{aligned} \tag{2.1}$$

Here, the selection of the reference point R as the center of gravity allows for decoupled translational and rotational dynamics. The states are the angular rates of the rigid body with respect to the ECI frame $\vec{\boldsymbol{\omega}}^{IB}(t) \in \mathbb{R}^3$ and the velocity of the reference point with respect to the ECI frame $\left(\vec{\mathbf{v}}^{R}\right)^{I}(t) \in \mathbb{R}^3$. The mass of the system is $m > 0 \in \mathbb{R}$ and the moment of inertia with respect to the reference point in the B frame is $\left(\mathbf{I}^{R}\right)_{BB} \in \mathbb{R}^{3\times 3}$. $\vec{\mathbf{F}}_{T}^{R}(t) \in \mathbb{R}^3$ is the total force vector and $\overrightarrow{\mathbf{M}}_{T}^{R}(t) \in \mathbb{R}^3$ is the total moment vector with respect to the reference point.

The translational dynamics can be further simplified by describing the acceleration with respect to the local navigation frame. The acceleration relative to the ECI system is

$$\left(\dot{\vec{\mathbf{v}}}^{R}\right)^{II} = \left(\dot{\vec{\mathbf{v}}}^{R}\right)^{IB} + \vec{\boldsymbol{\omega}}^{IB} \times \left(\vec{\mathbf{v}}^{R}\right)^{I} = \left(\dot{\vec{\mathbf{v}}}^{R}\right)^{IB} + \vec{\boldsymbol{\omega}}^{IB} \times \left(\vec{\mathbf{v}}^{R}\right)^{I}.$$

From Assumption 2.1 (Flat and non-rotating Earth) and the definition of the N-frame, it holds that $\vec{\boldsymbol{\omega}}^{IN} = \vec{\boldsymbol{\omega}}^{IE} + \vec{\boldsymbol{\omega}}^{EN} = \mathbf{0}$. Hence, $\left(\dot{\vec{\mathbf{v}}}^{R}\right)^{NN} = \left(\dot{\vec{\mathbf{v}}}^{R}\right)^{II}$. Finally, inserting the results in (2.1) leads to the simplified dynamic equations are

$$m \cdot \left(\dot{\vec{\mathbf{v}}}^{R}\right)_{N}^{NN}(t) = \left(\vec{\mathbf{F}}_{T}^{R}\right)_{N}(t), \tag{2.2a}$$

$$\left(\mathbf{I}^{R}\right)_{BB}\left(\dot{\vec{\boldsymbol{\omega}}}^{NB}\right)_{B}^{B}(t) = -\left(\vec{\boldsymbol{\omega}}^{NB}\right)_{B}(t) \times \left(\left(\mathbf{I}^{R}\right)_{BB}\left(\vec{\boldsymbol{\omega}}^{NB}\right)_{B}(t)\right) + \left(\overrightarrow{\mathbf{M}}_{T}^{R}\right)_{B}(t). \tag{2.2b}$$

The total force is the sum of propulsion (P), aerodynamic (A) and gravitational (G) forces. The total moment is computed analogously and hence

$$\begin{aligned} \left(\vec{\mathbf{F}}_{T}^{R}\right)_{B}(t) &= \left(\vec{\mathbf{F}}_{P}^{R}\right)_{B}(t) + \left(\vec{\mathbf{F}}_{A}^{R}\right)_{B}(t) + \left(\vec{\mathbf{F}}_{G}^{R}\right)_{B}(t), \\ \left(\overrightarrow{\mathbf{M}}_{T}^{R}\right)_{B}(t) &= \left(\overrightarrow{\mathbf{M}}_{P}^{R}\right)_{B}(t) + \left(\overrightarrow{\mathbf{M}}_{A}^{R}\right)_{B}(t) + \left(\overrightarrow{\mathbf{M}}_{G}^{R}\right)_{B}(t). \end{aligned}$$

In the next section the different forces and moments are described in detail.

2.2.2 Forces and Moments

The gravitational forces are modeled as a constant force field

$$\left(\vec{\mathbf{F}}_{G}^{R}\right)_{N} = \left(\vec{\mathbf{F}}_{G}^{R}\right)_{O} = m\left(\vec{\mathbf{g}}^{G}\right)_{O} = m\begin{bmatrix}0\\0\\g\end{bmatrix}. \tag{2.3}$$

Here, $g \in \mathbb{R}$ is the constant gravitational acceleration and $\vec{\mathbf{g}} \in \mathbb{R}^3$ is the gravitational vector. As a result of Assumption 2.3 (Center of Gravity), the gravitational moments

are $\overrightarrow{\mathbf{M}}_G^R = \overrightarrow{\mathbf{M}}_G^G = \mathbf{0}$. From Assumption 2.6 (Aerodynamic Forces and Moments), the aerodynamic forces and moments are neglected, i.e. $\vec{\mathbf{F}}_A^R = \overrightarrow{\mathbf{M}}_A^R = \mathbf{0}$. With increasing velocity, these forces and moments increase in magnitude and robustness of the controller against these unmodeled dynamics needs to be checked. The results are presented in Chapter 6 (Experimental Results).

For multirotor systems, the propulsion forces and moments have the largest influence on the system's dynamics. A common feature of the considered configurations is that the rotation axis of all propellers is aligned with the body-fixed z-axis $\vec{\mathbf{z}}_B$. The most common propeller model for multirotors assumes a hover state. In this case, the force and the moment along the rotation axis of the propeller are proportional to the squared rotational speed. Hence, for the i-th propeller it holds that

$$\begin{aligned} Z_{P_i}(t) &= -k_T \cdot \omega_{P_i}^2(t), \\ N_{P_i}(t) &= -sgn\left(\omega_{P_i}(t)\right) \cdot k_M \cdot \omega_{P_i}^2(t), \end{aligned} \tag{2.4}$$

where $Z_{P_i}(t) \in \mathbb{R}$ and $N_{P_i}(t) \in \mathbb{R}$ are the third components of the i-th propeller force and moment vectors given in the B-frame respectively. $\omega_{P_i}(t) \in \mathbb{R}$ is the propellers' rotational rate about $\vec{\mathbf{z}}_B$ with respect to the B-frame given in the B-frame. The propeller specific parameters are reduced to the thrust moment coefficients $k_T > 0 \in \mathbb{R}$ and $k_M > 0 \in \mathbb{R}$. Further, $sgn(\cdot)$ represents the sign function. In this model it has been assumed that the propeller rotates in its design direction and therefore in the B-frame the force $Z_{P_i}(t)$ does not depend on the specific rotation direction but the yaw moment $N_{P_i}(t)$ does. Since the forces and moments in the propeller plane are negligible as stated in Assumption 2.4 (Forces in the Rotor Plane), the total propulsion forces are given by

$$\left(\vec{\mathbf{F}}_P^R\right)_B(t) = \sum_{i=1}^{6} \begin{pmatrix} 0 \\ 0 \\ Z_{P_i}(t) \end{pmatrix} = -T(t) \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \tag{2.5}$$

where $T(t) = \sum_{i=1}^{6} k_T \cdot \omega_{P_i}^2(t) \geq 0$ is the total thrust.

The propulsion moments have two components. The roll and pitch moments arise due to the lever arm of the thrust forces and the yaw moment corresponds to the support moment. The total moments can be computed as

$$\begin{aligned} \left(\overrightarrow{\mathbf{M}}_P^R\right)_B(t) &= \sum_{i=1}^{6} \left(\left(\vec{\mathbf{r}}^{RP_i}\right)_B \times \left(\vec{\mathbf{F}}_{P_i}^R\right)_B(t) + \left(\overrightarrow{\mathbf{M}}_{P_i}^R\right)_B(t) \right), \\ &= \sum_{i=1}^{6} \left(\begin{pmatrix} l\cos(\alpha_i) \\ l\sin(\alpha_i) \\ 0 \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \\ Z_{P_i}(t) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ N_{P_i}(t) \end{pmatrix} \right), \end{aligned} \tag{2.6}$$

where $\vec{\mathbf{r}}^{RP_i} \in \mathbb{R}^3$ is the position vector of the i-th propeller with respect to the reference point, $l > 0 \in \mathbb{R}$ is the length of the propeller arms and $\alpha_i \in \mathbb{R}$ is the angle between the body-fixed x-axis $\vec{\mathbf{x}}_B$ and the rotor arm. For the hexarotor configurations in Figure 2.1 it holds that $\alpha_i = \{30°, 90°, 150°, 210°, 270°, 330°\}$.

Given Assumption 2.5 (Motor Dynamics), the angular velocities are assumed to be directly controlled. By defining the control input of the system as the squared angular velocities

$$\mathbf{u}(t) = \left(\omega_1^2(t), \quad \omega_2^2(t), \quad \ldots, \quad \omega_6^2(t)\right)^T \in \mathbb{R}^6, \tag{2.7}$$

it is possible to write a linear relation between control input $\mathbf{u}(t)$ and the total thrust $T(t)$ and the propulsion moments $\overrightarrow{\mathbf{M}}_P^R(t)$ as follows

$$\boldsymbol{\nu}(t) := \begin{pmatrix} T(t) \\ \left(\overrightarrow{\mathbf{M}}_P^R\right)_B(t) \end{pmatrix} = \begin{bmatrix} \mathbf{B}_{Tu} \\ \mathbf{B}_{Mu} \end{bmatrix} \mathbf{u}(t) = \mathbf{B}_a \mathbf{u}(t). \tag{2.8}$$

Here, $\boldsymbol{\nu}(t) \in \mathbb{R}^4$ is defined as the virtual control vector and the matrix matrix $\mathbf{B}_a \in \mathbb{R}^{4\times 6}$ is partitioned into $\mathbf{B}_{Tu} \in \mathbb{R}^{1\times 6}$ and $\mathbf{B}_{Mu} \in \mathbb{R}^{3\times 6}$. For the hexarotor configurations in Figures 2.1a and 2.1b the matrix $\mathbf{B}_a \in \mathbb{R}^{4\times 6}$ is given by [45]

$$\text{Hexarotor 1 Figure 2.1a} \quad : \quad \mathbf{B}_a = \begin{bmatrix} k_T & k_T & k_T & k_T & k_T & k_T \\ -\frac{1}{2}lk_T & -lk_T & -\frac{1}{2}lk_T & \frac{1}{2}lk_T & lk_T & \frac{1}{2}lk_T \\ \frac{\sqrt{3}}{2}lk_T & 0 & -\frac{\sqrt{3}}{2}lk_T & -\frac{\sqrt{3}}{2}lk_T & 0 & \frac{\sqrt{3}}{2}lk_T \\ -k_M & k_M & -k_M & k_M & -k_M & k_M \end{bmatrix}, \tag{2.9}$$

$$\text{Hexarotor 2 Figure 2.1b} \quad : \quad \mathbf{B}_a = \begin{bmatrix} k_T & k_T & k_T & k_T & k_T & k_T \\ -\frac{1}{2}lk_T & -lk_T & -\frac{1}{2}lk_T & \frac{1}{2}lk_T & lk_T & \frac{1}{2}lk_T \\ \frac{\sqrt{3}}{2}lk_T & 0 & -\frac{\sqrt{3}}{2}lk_T & -\frac{\sqrt{3}}{2}lk_T & 0 & \frac{\sqrt{3}}{2}lk_T \\ -k_M & k_M & k_M & -k_M & -k_M & k_M \end{bmatrix}. \tag{2.10}$$

For $k_T, k_M, l > 0$ the matrix $\mathbf{B}_a$ has a full row rank for the presented configurations. Hence, for a given desired $\boldsymbol{\nu}_d(t) \in \mathbb{R}^4$ it is possible to find a function which maps it into the input space $\mathbf{u}(t) = \mathbf{u}(\boldsymbol{\nu}_d(t))$ so that $\boldsymbol{\nu}(t) = \mathbf{B}_a\mathbf{u}(\boldsymbol{\nu}_d(t)) = \boldsymbol{\nu}_d(t)$. The hexacopter is an overactuated system since the row rank of $\mathbf{B}_a$ is smaller than the number of actuators, i.e. $4 < 6$. In the quadrotor case, the matrix $\mathbf{B}_a \in \mathbb{R}^{4\times 4}$ is square and a unique solution exists [110]. In the hexarotor case, the solution is not unique [110] and therefore, it makes sense to separate the rigid body control and the control allocation as shown in Figure 1.4. The rigid body control calculates the desired virtual control $\boldsymbol{\nu}_d(t)$ and the control allocation maps it into the input space to compute the command $\mathbf{u}(t)$.

2.2.3 Kinematics

The kinematics of a rigid body can be described by the position and attitude differential equations. For control purposes, the position is described in the local navigation frame and its differential equation is given by

$$\left(\dot{\vec{\mathbf{r}}}^R\right)_N^N(t) = \left(\vec{\mathbf{v}}^R\right)_N^N(t). \tag{2.11}$$

The attitude of a rigid body is fully and uniquely described by the rotation matrix [136], i.e. an orthogonal matrix in $\mathbb{R}^{3\times 3}$ whose determinant equals one. This type of matrices define the special orthogonal group $SO(3)$. In order to describe the attitude of the aircraft the rotation matrix from the body fixed frame to the local navigation frame $\mathbf{M}_{NB} \in SO(3)$ is used. A useful representation of this matrix is given by the unit vectors pointing along the direction of the body-fixed axes expressed in the N-frame as follows:

$$\mathbf{M}_{NB}(t) = \left[\left(\vec{\mathbf{x}}_B\right)_N(t), \quad \left(\vec{\mathbf{y}}_B\right)_N(t), \quad \left(\vec{\mathbf{z}}_B\right)_N(t)\right]. \tag{2.12}$$

Its derivative is related to the angular rates by the strapdown equation as

$$\dot{\mathbf{M}}_{NB}(t) = \mathbf{M}_{NB}(t) \cdot \mathbf{\Omega}\left(\left(\vec{\boldsymbol{\omega}}^{NB}\right)_B(t)\right). \tag{2.13}$$

Here, the function $\mathbf{\Omega}(\cdot) : \mathbb{R}^3 \rightarrow \mathbb{R}^{3\times3}$ describes the mapping from a 3-dimensional vector to its corresponding skew symmetric matrix

$$\mathbf{\Omega}\left(\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}\right) := \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix}. \tag{2.14}$$

By definition it holds that $\mathbf{u} \times \mathbf{v} = \mathbf{\Omega}(\mathbf{u})\,\mathbf{v}$ for two vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^3$. Lemma D.1 (Cross Product) summarizes the needed properties of (2.14).

2.2.4 State Space Model

The state-space model is separated into rigid body motion and actuator model. The motion of the rigid boy is described by the dynamic equations (2.2) and kinematic equations (2.11) and (2.13). By inserting the forces and moments (2.3), (2.4), (2.5) and (2.6), the rigid body equations of motion are

$$\begin{aligned}
\left(\dot{\vec{\mathbf{r}}}^{R}\right)_N^N(t) &= \left(\vec{\mathbf{v}}^{R}\right)_N^N(t), \\
m\left(\dot{\vec{\mathbf{v}}}^{R}\right)_N^{NN}(t) &= m\left(\vec{\mathbf{g}}^{G}\right)_N - T(t)\left(\vec{\mathbf{z}}_B\right)_N(t), \\
\dot{\mathbf{M}}_{NB}(t) &= \mathbf{M}_{NB}(t)\,\mathbf{\Omega}\left(\left(\vec{\boldsymbol{\omega}}^{NB}\right)_B(t)\right), \\
\left(\mathbf{I}^{R}\right)_{BB}\left(\dot{\vec{\boldsymbol{\omega}}}^{IB}\right)_B^B(t) &= -\left(\vec{\boldsymbol{\omega}}^{IB}\right)_B(t) \times \left(\left(\mathbf{I}^{R}\right)_{BB}\left(\vec{\boldsymbol{\omega}}^{IB}\right)_B(t)\right) + \left(\overrightarrow{\mathbf{M}}_P^R\right)_B(t).
\end{aligned} \tag{2.15}$$

Here, the total thrust $T(t)$ and propulsion moments $\left(\overrightarrow{\mathbf{M}}_P^R\right)_B(t)$ are the inputs of the system. The vector $\left(\vec{\mathbf{z}}_B\right)_N(t) \in \mathbb{R}^3$ is the body-fixed z-axis given in the N-frame and is computed from $\left(\vec{\mathbf{z}}_B\right)_N(t) = \mathbf{M}_{NB} \cdot [0,0,1]^T$. The actuator model relates the virtual controls $\boldsymbol{\nu}(t)$ with the inputs $\mathbf{u}(t)$ of the system as in (2.8). In order to model faults, multiplicative $\mathbf{\Lambda} \in \mathbb{R}^{6\times6}$ and additive $\mathbf{d}_{\boldsymbol{\nu}} \in \mathbb{R}^4$ parameters can be used [79]:

$$\boldsymbol{\nu}(t) = \mathbf{B}_a \mathbf{\Lambda} \mathbf{u}(t) + \mathbf{d}_{\boldsymbol{\nu}}. \tag{2.16}$$

This model covers degradation of control effectiveness, actuator failures, stuck actuators and control direction changes as it is detailed in Chapter 3 (Adaptive Fault Tolerant Control). Furthermore, through the mapping (2.7) the actuator model (2.16) has become linear which drastically simplifies the control allocation task. The advantage of the separation between rigid body and actuator models is that the tasks of stabilizing the aircraft motion and allocating the necessary forces and moments to the actuators can be separated as well. In this case, if a different multirotor configuration is used, it will primarily affect the control allocation and not the controller itself. In Table 2.3 the main inertia, geometry and actuator parameters of the system are summarized. The moment of inertia is a diagonal matrix with the entries $\left(\mathbf{I}^{R}\right)_{BB} = diag([\left(I_{xx}^{R}\right)_{BB}, \left(I_{yy}^{R}\right)_{BB}, \left(I_{zz}^{R}\right)_{BB}])$. Furthermore, the actuator parameters hold for all 6 propulsion units.

Description	Value	Unit
Mass m	1.164	kg
Moment of Inertia $\left(I^R_{xx}\right)_{BB}$	$10.0 \cdot 10^{-3}$	$kg \cdot m^2$
Moment of Inertia $\left(I^R_{yy}\right)_{BB}$	$10.2 \cdot 10^{-3}$	$kg \cdot m^2$
Moment of Inertia $\left(I^R_{zz}\right)_{BB}$	$18.1 \cdot 10^{-3}$	$kg \cdot m^2$
Arm length l	0.215	m
Thrust Coefficient k_T	$3.53 \cdot 10^{-6}$	$N \cdot \frac{s^2}{rad^2}$
Moment Coefficient k_M	$1.29 \cdot 10^{-7}$	$Nm \cdot \frac{s^2}{rad^2}$
Min. Angular Velocity $\omega_{P,min}$	1260	rpm
Max. Angular Velocity $\omega_{P,max}$	10300	rpm

Table 2.3: Plant Parameters

2.2.5 Full and Reduced Attitude Control

There are mainly two types of control goals regarding rigid body attitude control: full and reduced attitude control [26]. In full attitude control, the three rotational degrees of freedom are to be controlled. In reduced attitude control or pointing applications, only the alignment of a single body-fixed vector is required and the rotation about this vector is not relevant [26]. In this case, two attitude degrees of freedom are controlled and a reduced attitude vector can be used instead of the full rotation matrix. This vector has a length of one and defines the unit sphere $\mathbb{S}^2$ in $\mathbb{R}^3$.

For the presented type of multirotors the translational and yaw motions are inherently decoupled and therefore reduce attitude control can be used for maintaining this decoupling. This can be seen by analyzing the influence that the attitude has on the translational differential equations. The position of the system is controlled by the total propulsion force (2.5), which can be written as $\vec{\mathbf{F}}^R_P(t) = -T(t) \cdot \vec{\mathbf{z}}_B(t)$. The vector $\vec{\mathbf{z}}_B(t)$ is a unit vector pointing along the body-fixed z-axis. Hence, the position of the hexacopter is controlled by changing the total thrust $T(t)$ and by tilting the vector $\vec{\mathbf{z}}_B(t)$, which can be seen as a reduced attitude parameter.

Given that $\left(\vec{\mathbf{z}}_B\right)_B = [0,0,1]^T$ is constant in the body-fixed frame and using the strapdown equation (2.13), the kinematics the reduced attitude parameter $\left(\vec{\mathbf{z}}_B\right)_N = \mathbf{M}_{NB}\left(\vec{\mathbf{z}}_B\right)_B$ are given by

$$\left(\dot{\vec{\mathbf{z}}}_B\right)^N_N(t) = \mathbf{M}_{NB}(t) \cdot \mathbf{\Omega}\left(\left(\vec{\boldsymbol{\omega}}^{NB}\right)_B(t)\right) \cdot \left(\vec{\mathbf{z}}_B\right)_B(t) = \mathbf{M}_{NB}(t) \begin{pmatrix} q(t) \\ -p(t) \\ 0 \end{pmatrix}. \tag{2.17}$$

Here, the vector elements of the angular rate correspond to $\left(\vec{\boldsymbol{\omega}}^{NB}\right)_B(t) = (p,q,r)^T(t)$. From (2.17) it can be seen that the yaw rate $r(t)$, i.e. the rotation about the vector $\vec{\mathbf{z}}_B(t)$, neither influences the tilting of the total force vector $\vec{\mathbf{F}}^R_P(t)$ nor the translational motion.

In this work, the ability to independently control position and yaw motions motivates the use of reduced attitude control with the vector $\vec{\mathbf{z}}_B$ as the reduced attitude vector. This decoupling facilitates the controller tuning and enables better performance by avoiding coupling of motions with different time constants and noise characteristics. On the one hand yaw rate measurements have usually a higher noise level compared to roll and pitch rate measurements due to the motors mounting direction. On the other hand, roll and pitch motions are much faster compared to the yaw motion of

Attitude Representation	Singularity Free	Unique	Number of Parameters	Number of Constraints	Degrees of Freedom
Rotation Matrix	✓	✓	9	6	3
Euler Angles	x	x	3	0	3
Quaternions	✓	x	4	1	3
Reduced-Attitude Vector	✓	✓	3	1	2

Table 2.4: Attitude Parameters' Comparison

the system. The reason is that roll and pitch moments are directly generated from the propellers' force and the yaw moment arises from the supporting torque of the motors. Furthermore, roll and pitch moments of inertia are also considerably smaller than the yaw moment of inertia. The last reason for the choice of reduced attitude control is that if facilitates the strategy for degraded control authority as discussed in Chapter 4 (Adaptive Control Allocation). In this case, the performance of yaw motion control might degrade and the decoupling avoids propagation to position control.

2.2.6 Attitude Parameterization

Attitude parameterization is a well studied field and is relevant for defining the flight envelope of the system. In this section, relevant facts are shortly summarized in order to underline the selection of the reduced attitude vector in this work. For a more in deep treatment, the reader is referred to [26, 136].

For control purposes the direct use of the rotation matrix is commonly not preferred because of the interpretation effort and the large number of parameters. 9 parameters and 6 constraints are necessary for describing the 3 degrees of freedom of the attitude motion. Nevertheless, there is no other parameterization that is able to represent the attitude globally and uniquely [26, 136] and therefore any other representation needs a careful interpretation of the results in $SO(3)$. Differently from the rotation matrices, two of the most common parameterizations are the Euler angles and the attitude quaternions. The Euler angles are composed of only 3 parameters but are not unique for some attitudes and show singularities on these attitudes. This representation is mainly suited for systems with a limited motion. The quaternions correspond to a 4 parameter representation with 1 constraint. It is a global but not unique parameterization since every attitude can be represented by two quaternions. If not taken into account, this can lead to the unwinding phenomenon, which in simplified terms means that the path towards the desired attitude does not correspond to the shortest path [10, 26]. This is a strongly undesired behavior. The reduced attitude vector is a global, unique and singularity free parameterization with two degrees of freedom. Therefore, it is equivalent to the rotation matrix but for the reduced attitude case. The advantage in this lower dimensional case is the ease of interpretation and the reduction of the number of parameters to 3. An overview of the listed parameterizations is depicted in Table 2.4.

This work mainly deals with reduced attitude control and uses the reduced attitude vector representation. This guarantees that no singularities artificially limit the achievable flight envelope and that the results can be directly interpreted in the corre-

sponding physical space. Nevertheless, the Euler angles $(\Phi, \Theta, \Psi)^T \in \mathbb{R}^3$ are used for displaying results because of their wide use and for facilitating the interpretation. Φ is the bank angle, Θ the pitch angle and Ψ is the azimuth angle. The rotation sequence is about the z, y and finally x axes. The relation between the rotation matrix and the Euler angles is given by [136]

$$\mathbf{M}_{BN} = \begin{bmatrix} \cos\Theta\cos\Psi & \cos\Theta\sin\Psi & -\sin\Theta \\ \sin\Phi\sin\Theta\cos\Psi - \cos\Phi\sin\Psi & \sin\Phi\sin\Theta\sin\Psi + \cos\Phi\cos\Psi & \sin\Phi\cos\Theta \\ \cos\Phi\sin\Theta\cos\Psi + \sin\Phi\sin\Psi & \cos\Phi\sin\Theta\sin\Psi - \sin\Phi\cos\Psi & \cos\Phi\cos\Theta \end{bmatrix}$$

and consequently from (2.12) the relation between the reduced attitude vector $(\vec{\mathbf{z}}_B)_O$ and the Euler angles is given by

$$(\vec{\mathbf{z}}_B)_N = \begin{pmatrix} \cos\Phi\sin\Theta\cos\Psi + \sin\Phi\sin\Psi \\ \cos\Phi\sin\Theta\sin\Psi - \sin\Phi\cos\Psi \\ \cos\Phi\cos\Theta \end{pmatrix}. \tag{2.18}$$

2.3 Attainable Control Set

Given the state-space model for a multirotor system, the question arises how to quantify the physical fault tolerance capabilities of the different configurations. One possible measure is the attainable control set (ACS) which describes the set of virtual controls that are achievable given the actuator limits. Formally, the input set $\mathcal{U} \subset \mathbb{R}^6$ is defined as

$$\mathcal{U} := \{\mathbf{u} \in \mathbb{R}^6 \mid \omega_{P,min}^2 \leq u_i \leq \omega_{P,max}^2, \forall i = 1...6\}, \tag{2.19}$$

where $\omega_{P,min}$ and $\omega_{P,max}$ are the minimum and maximum rotational rates of the propulsion units from Table 2.3. Therefore, the set $\mathcal{U}$ is a 6-dimensional hypercube. The ACS is defined as the output image $\mathcal{V}$ of the linear mapping (2.16) on $\mathcal{U}$

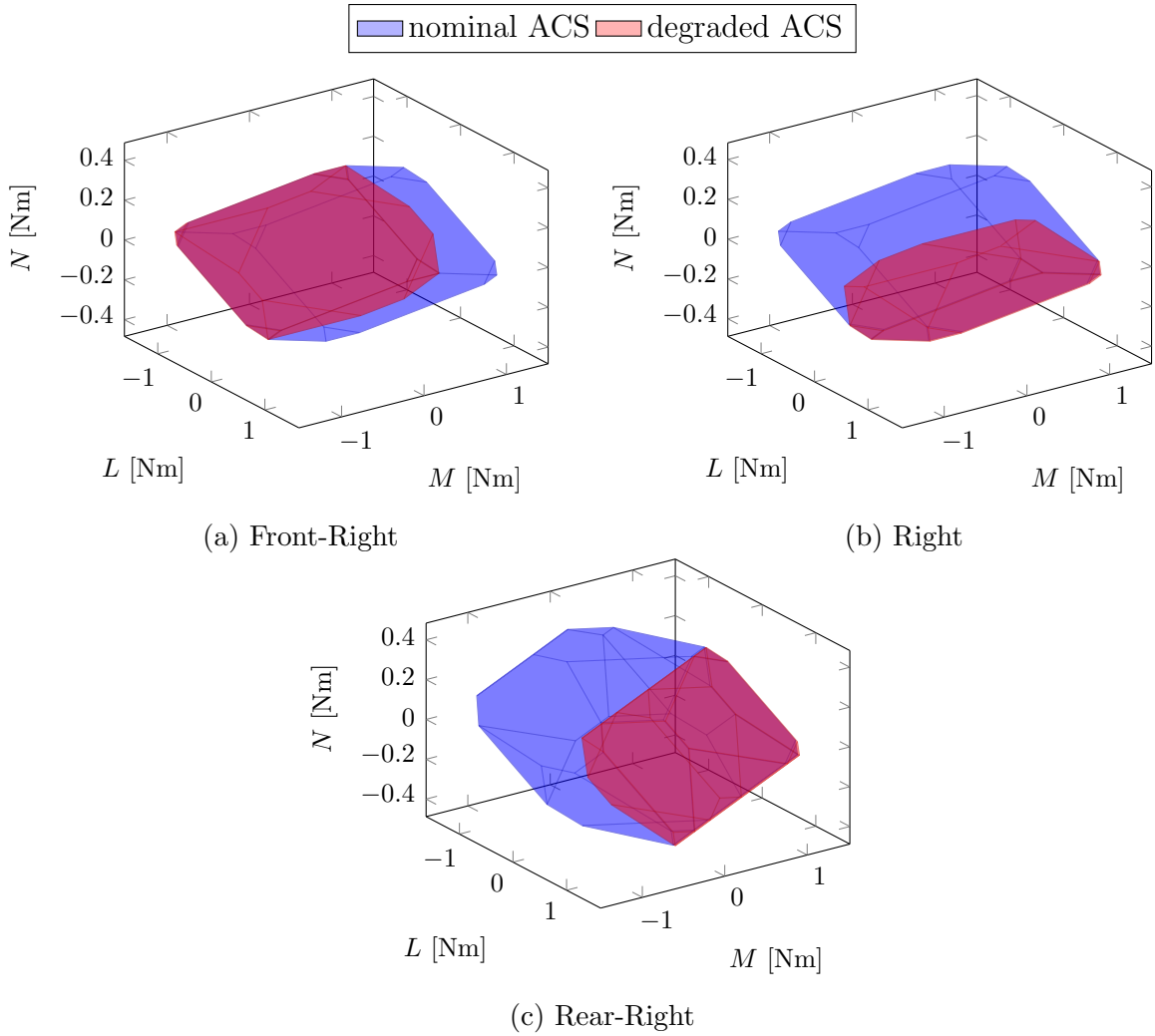
$$\mathcal{V} := \{\boldsymbol{\nu} \in \mathbb{R}^4 \mid \boldsymbol{\nu} = \mathbf{B}_a\boldsymbol{\Lambda}\mathbf{u} + \mathbf{d}_\nu, \mathbf{u} \in \mathcal{U}\}. \tag{2.20}$$

Because the input set $\mathcal{U}$ is a convex polyhedra and it is linearly mapped to the set $\mathcal{V}$, the ACS $\mathcal{V}$ is also a convex polyhedra.

In the following, the ACS for different total failure conditions is described. All the attainable control sets in this work have been computed using the Multi-Parametric Toolbox MPT 3.0 [68]). In the nominal case, $\mathbf{d}_\nu = \mathbf{0}$ and $\boldsymbol{\Lambda}$ is a diagonal unit matrix. In a failure case, the corresponding diagonal elements of the control effectiveness matrix $\boldsymbol{\Lambda}$ are set to zero. First, the hexacopter with actuator configuration 1 as in Figure 2.1a is analyzed. As the ACS for multirotor systems is 4-dimensional, only cuts of this polyhedron can be visualized. In Figure 2.3 the thrust has been fixed to hover thrust $T = 11.87[N]$ and the available moment space is shown. Each of the three plots shows the ACS in nominal conditions and the effect of a failure of each of the right actuators. Because of the symmetry of the configuration, the degraded ACS of the left actuators has been omitted. As can be seen, any failure significantly reduces the ACS and the volume of the 3-dimensional polyhedrons reduces for any single failure case to 47% of the nominal one.

The possibility of a steady-state hover flight is given if the point $\boldsymbol{\nu} = (m \cdot g, 0, 0, 0)^T$ lies within the ACS. However, this is not the only possibility of hovering. Because

of the decoupling between yaw and position motion, the value of the yaw moment can be $N \neq 0$. Therefore, an interesting ACS cut is setting roll and pitch moments to zero $L = M = 0$ as shown in Figure 2.4. Each of the three plots shows the ACS in nominal conditions and the effect of a failure of each of the right actuators. In this case, the cut of the degraded ACS only depends on the rotation direction of the propeller. Interestingly, regardless of the weight of the system, this configuration is only able to produce yaw moment in one direction in the case of a failure during hover flight.



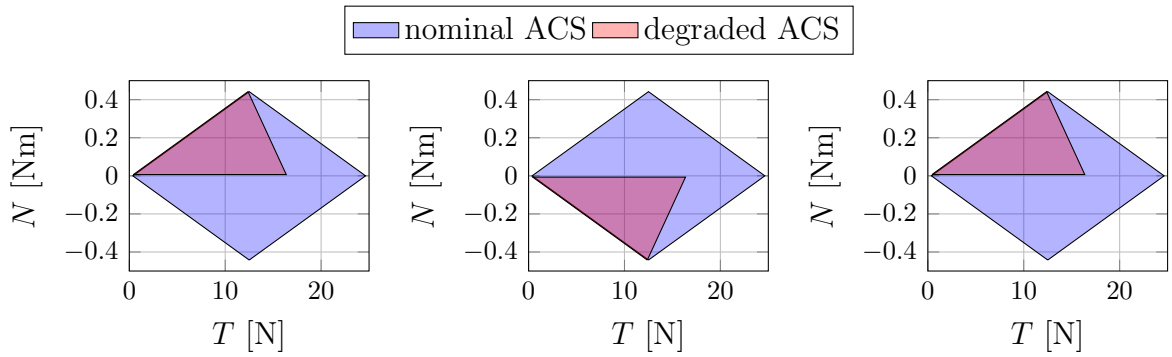
(a) Front-Right

(b) Right

(c) Rear-Right

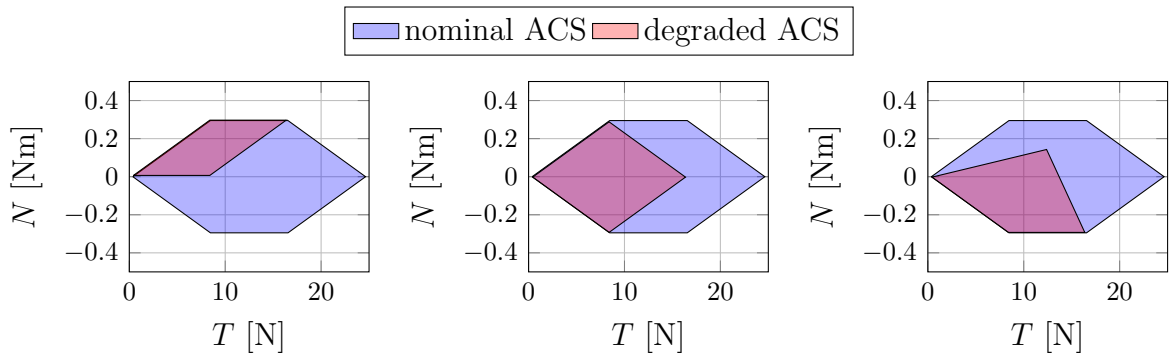
Figure 2.3: Hexacopter Actuator Configuration 1 - Comparison of the Attainable Control Set (ACS) for $T = mg$ between nominal Conditions and Failure of the front-right, right and rear-right Actuators

Because of the last fact, several alternative configurations have been proposed in the literature. Full control authority can be regained by letting one propeller rotate in inverse direction [1], by tilting the rotation axis of all the rotors [60, 111] or by using more than 6 propulsion units. Nevertheless, the simplest adaption is the change of the rotational directions of the rotors as in the hexarotor configuration 2 (Figure 2.1b), which is analyzed in the following. In Figure 2.6 the ACS moment space with a cut at hover thrust $T = 11.87[N]$ is depicted. Each of the three plots shows the ACS in nominal conditions and the effect of a failure of each of the right actuators. Differently



(a) Front-Right (b) Right (c) Rear-Right

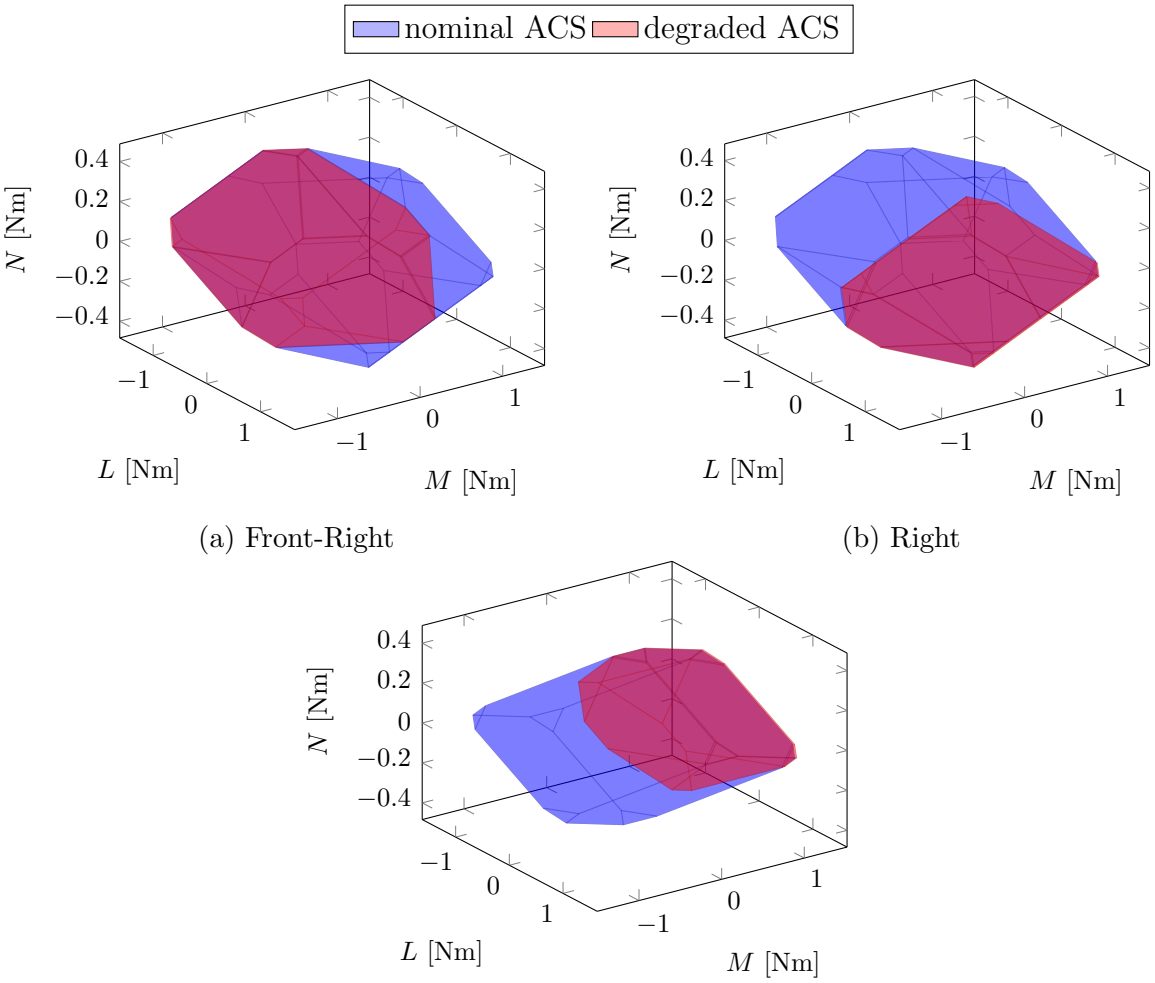
Figure 2.4: Hexacopter Actuator Configuration 1 - Comparison of the Attainable Control Set (ACS) for $L = M = 0$ between nominal Conditions and Failure of the front-right, right and rear-right Actuators



(a) Front-Right (b) Right (c) Rear-Right

Figure 2.5: Hexacopter Actuator Configuration 2 - Comparison of the Attainable Control Set (ACS) for $L = M = 0$ between nominal Conditions and Failure of the front-right, right and rear-right Actuators

from the actuator configuration 1, each actuator failure has a qualitatively different influence on the ACS. In this configuration, a failure of the actuators 2,3,4 or 5 leads to a 3-dimensional polyhedron that has the same volume as in configuration 1. In the failure case of any of the front actuators, the volume further reduces by 50% and have therefore a higher criticality. This fact is clearer from Figure 2.5. Here, the ACS cut from setting roll and pitch moments to zero $L = M = 0$ is depicted for comparing the failures of each of the three right actuators. The ACS cut of the left actuators is the reflection of the corresponding right actuator's ACS cut about the $N = 0$ line. It can be seen that in case of a failure of one of the front actuators, the yaw moment can only be produce in one direction. Compared to configuration 1, the situation has not improved. If one of the other 4 actuators fail, hovering with a stable yaw motion is possible. Therefore, in configuration 2 the failure of 4 out of 6 actuators can be fully compensated [133], i.e. the point $\boldsymbol{\nu} = (m \cdot g, 0, 0, 0)^T$ lies within the ACS. The disadvantage is that a failure of one of the 2 front actuators leads to a further reduction of the ACS volume.



(a) Front-Right

(b) Right

(c) Rear-Right

Figure 2.6: Hexacopter Actuator Configuration 2 - Comparison of the Attainable Control Set (ACS) for $T = mg$ between nominal Conditions and Failure of the front-right, right and rear-right Actuators

Chapter 3

Adaptive Fault Tolerant Control

In this chapter, the theoretical framework for solving fault tolerant control problems for systems with redundant actuators using adaptive control is presented. First, suitable models for the dynamical system, actuators and possible faults are derived. Subsequently, state-of-the-art adaptive control approaches are analyzed regarding fault tolerance and actuator redundancy. Based on the gained insights, the parameter reduction due to overactuation (PRO) approach is introduced as one of the main contributions of this thesis. This approach explicitly takes the special characteristics of systems with redundant actuators into account in order to achieve the control task in the presence of actuation faults. The applicability of the PRO approach mainly depends on the parameterization of the uncertainty rather than on a special controller structure. Therefore, the conditions under which the PRO approach can be used are studied. For facilitating the analysis, the well known MRAC framework is used in this chapter including the extension of direct and indirect adaptive control approaches using PRO.

The remainder of the chapter is organized as follows. In Section 3.1, the control task is defined and suitable models for the dynamical system, actuators and possible faults are derived. In Section 3.2, direct MRAC is introduced and in Section 3.3 it is extended via the PRO approach. In Section 3.4, predictor-based MRAC is introduced and in Section 3.5 it is extended via the PRO approach. The PRO approach is summarized in Section 3.6. Finally, a bibliographical remarks are given in Section 3.7. The mathematical tools used within the stability assessment are collected in Appendix B.

3.1 System Description and Control Task

In order to describe the MRAC control laws, the following system representation in which the unknown parameters are clearly defined and appear linearly in the plant dynamics is considered:

$$\dot{\mathbf{x}}_p(t) = \mathbf{A}_p\mathbf{x}_p(t) + \mathbf{B}_p\mathbf{\Lambda}\mathbf{u}(t) + \mathbf{K}_\phi\boldsymbol{\phi}(\mathbf{x}_p(t)) + \mathbf{d}. \tag{3.1}$$

Here, the state of the plant is $\mathbf{x}_p(t) \in \mathbb{R}^n$, the control input is $\mathbf{u}(t) \in \mathbb{R}^m$, the unknown dynamic matrix is $\mathbf{A}_p \in \mathbb{R}^{n\times n}$, the known input matrix is $\mathbf{B}_p \in \mathbb{R}^{n\times m}$, the unknown control effectiveness matrix is $\mathbf{\Lambda} \in \mathbb{R}^{m\times m}$ and the constant unknown disturbance is $\mathbf{d} \in \mathbb{R}^n$. The term $\mathbf{h}(\mathbf{x}_p(t)) = \mathbf{K}_\phi\boldsymbol{\phi}(\mathbf{x}_p(t))$ represents a linearly parameterizable nonlinear function $\mathbf{h}(\mathbf{x}_p)$, where $\mathbf{K}_\phi \in \mathbb{R}^{n\times k}$ is an unknown matrix and $\boldsymbol{\phi}(\mathbf{x}_p) : \mathbb{R}^n \to \mathbb{R}^k$

is a known regressor vector composed of k basis functions $\phi_i(\mathbf{x}_p)$, which are locally Lipschitz-continuous in $\mathbf{x}_p$ [92, pp. 281]. The system (3.1) is called overactuated if $\operatorname{rank}(\mathbf{B}_p) = r < m$.

In MRAC, the control task is that the state trajectories of the plant asymptotically approach the state trajectories of a reference model, which represents the desired dynamics of the system. In the following, let the corresponding reference model be

$$\dot{\mathbf{x}}_m(t) = \mathbf{A}_m \mathbf{x}_m(t) + \mathbf{B}_m \mathbf{r}(t), \tag{3.2}$$

with the state $\mathbf{x}_m(t) \in \mathbb{R}^n$, the bounded and piecewise-continuous reference signal $\mathbf{r}(t) : \mathbb{R}^+ \to \mathbb{R}^r$, the Hurwitz dynamic matrix $\mathbf{A}_m \in \mathbb{R}^{n\times n}$ and the input matrix $\mathbf{B}_m \in \mathbb{R}^{n\times r}$. The reference model has several important characteristics. Since $\mathbf{A}_m$ is Hurwitz, it is a stable low pass filter and a bounded-input bounded-state linear system with respect to $\mathbf{r}(t)$. Hence, $\mathbf{x}_m(t)$ is a bounded signal. Furthermore, there exists a symmetric positive definite matrix $\mathbf{P} \in \mathbb{R}^{n\times n}$ for every symmetric positive definite matrix $\mathbf{Q} \in \mathbb{R}^{n\times n}$ such that the Lyapunov equation holds [87, Th. 4.6]:

$$\mathbf{P}\mathbf{A}_m + \mathbf{A}_m^T \mathbf{P} = -\mathbf{Q}. \tag{3.3}$$

The reference model (3.2) must be achievable by the plant (3.1) and therefore the matrices $\mathbf{A}_m$ and $\mathbf{B}_m$ cannot be arbitrarily chosen. This restriction is reflected in the so-called matching conditions, which depend on the control approach and are therefore introduced later on. The control task is formalized as in [117] with the restriction to state trajectory tracking.

Control Task 3.1 (Model Reference Adaptive Control) *Given the plant* (3.1) *and the reference model* (3.2)*, find a control law* $\mathbf{u}(t, \mathbf{x})$ *such that the tracking error asymptotically vanishes*

$$\lim_{t\to\infty} ||\mathbf{e}_c(t)|| = \lim_{t\to\infty} ||\mathbf{x}_p(t) - \mathbf{x}_m(t)|| = 0.$$

Before getting into the specific control approaches that solve the Control Task 3.1, the actuator and fault models of the system (3.1) are analyzed. Hence, consider the alternative representation in which the static actuator model is separated:

$$\dot{\mathbf{x}}_p(t) = \mathbf{A}_p \mathbf{x}_p(t) + \mathbf{B}_\nu \boldsymbol{\nu}(t) + \mathbf{K}_\phi \boldsymbol{\phi}(\mathbf{x}_p(t)) + \mathbf{d}_x, \tag{3.4}$$

$$\boldsymbol{\nu}(t) = \mathbf{B}_a \boldsymbol{\Lambda} \mathbf{u}(t) + \mathbf{d}_{\boldsymbol{\nu}}. \tag{3.5}$$

For aircraft and most mechanical systems, (3.4) describes the rigid-body dynamics and (3.5) is the actuator model. Here $\boldsymbol{\nu}(t) \in \mathbb{R}^r$ is the virtual control, $\mathbf{B}_\nu \in \mathbb{R}^{n\times r}$ is the known rigid body input matrix and $\mathbf{B}_a \in \mathbb{R}^{r\times m}$ is the known actuator mapping matrix such that $\mathbf{B}_p = \mathbf{B}_\nu \cdot \mathbf{B}_a$. The constant rigid body disturbance is $\mathbf{d}_x \in \mathbb{R}^n$ and the constant actuator disturbance is $\mathbf{d}_{\boldsymbol{\nu}} \in \mathbb{R}^r$ and both satisfy the following condition $\mathbf{d} = \mathbf{B}_\nu \mathbf{d}_{\boldsymbol{\nu}} + \mathbf{d}_x$. Hence, note that by inserting (3.5) into (3.4), the system dynamics (3.1) are recovered.

The key to achieve the representation (3.4)-(3.5) is the right choice of the virtual controls $\boldsymbol{\nu}(t)$. They can be understood as a minimal representation of the control input's influence on the system. Therefore, they are restricted to the degrees of freedom that can be controlled independently, i.e. to r. Given that $\operatorname{rank}(\mathbf{B}_p) = r$, the full rank factorization $\mathbf{B}_p = \mathbf{B}_\nu \cdot \mathbf{B}_a$ with $\operatorname{rank}(\mathbf{B}_\nu) = \operatorname{rank}\mathbf{B}_a = r$ always exists, see Lemma B.4 (Rank) in Appendix B. For mechanical systems, forces and moments are usually a

natural choice of virtual controls $\boldsymbol{\nu}(t)$. The system is called overactuated if it has more actuators than virtual controls, i.e. $r < m$ [81].

The considered actuator model (3.5) is linear and static. This is not necessary for applying the proposed PRO approach, but it helps to illustrate its main characteristics in a simple framework. In (3.5), faults are modeled as multiplicative faults $\boldsymbol{\Lambda}$ and additive faults $\mathbf{d}_{\boldsymbol{\nu}}$ [79]. The fault model is common within the framework of adaptive control and both $\boldsymbol{\Lambda}$ and $\mathbf{d}_{\boldsymbol{\nu}}$ are unknown constant parameters. The type of faults that can be taken into account depend on the restrictions made on the control effectiveness matrix $\boldsymbol{\Lambda}$. The most common restrictions and the corresponding faults can be seen in Table 3.1.

The restrictions on the control effectiveness matrix $\boldsymbol{\Lambda}$ arise due to the fact that it is an unknown matrix. Hence, the restrictions allow finding a solution of the adaptive control problem and are related to the controllability of the plant. This will be clear from the subsequent sections. Nevertheless, in order to understand Table 3.1, let us first restrict the control effectiveness matrix $\boldsymbol{\Lambda}$ to a diagonal matrix and set $\mathbf{d}_{\boldsymbol{\nu}} = \mathbf{0}$. In this case the virtual control can be computed from

$$\boldsymbol{\nu}(t) = \sum_{i=1}^{m} \left(\mathbf{b}_{A,ci} \cdot \lambda_{ii} \cdot u_i(t) \right),$$

where $\mathbf{b}_{A,ci} \in \mathbb{R}^r$ is the i-th column of $\mathbf{B}_a$, $\lambda_{ii} \in \mathbb{R}$ is the i-th diagonal element of $\boldsymbol{\Lambda}$ and $u_i \in \mathbb{R}$ is the i-th element of $\mathbf{u}$. Note that the knowledge about the plant is given by the vectors $\mathbf{b}_{A,ci}$ which physically represent the influence directions of each control input u_i. It can be seen that in order to map an actuator failure, $\lambda_{ii} = 0$ needs to be allowed. Hence, if we restrict all λ_{ii}'s to $\lambda_{ii} > 0$, i.e. a positive definite $\boldsymbol{\Lambda}$, it is only possible to model a degradation of the control effectiveness. If the restriction is $\lambda_{ii} \neq 0$, i.e. a non-singular $\boldsymbol{\Lambda}$, it is also possible to map a control direction inversion for $\lambda_{ii} < 0$. For a positive semi-definite $\boldsymbol{\Lambda}$, i.e. $\lambda_{ii} \geq 0$, an actuator failure can be modeled but not a control direction inversion. Finally, in order take a stuck actuator into account, the corresponding λ_{ii} needs to be zero and the disturbance offset $\mathbf{d}_{\boldsymbol{\nu}}$ would correspond to the resulting virtual controls of the stuck actuator.

If the diagonal restriction is dropped, the virtual control can be computed from

$$\boldsymbol{\nu}(t) = \sum_{j=1}^{m} \left(\left(\sum_{i=1}^{m} \mathbf{b}_{A,ci} \cdot \lambda_{ij} \right) \cdot u_j(t) \right) + \mathbf{d}_{\boldsymbol{\nu}},$$

where $\lambda_{ij} \in \mathbb{R}$ is the element of $\boldsymbol{\Lambda}$ corresponding to the i-th row and j-th column. Hence, the direction in which the j-th actuator influences the virtual controls is given by $\sum_{i=1}^{m} (\mathbf{b}_{A,ci} \cdot \lambda_{ij})$, which corresponds to a linear combination of the columns of $\mathbf{B}_a$. Physically, it means that the off-diagonal elements of $\boldsymbol{\Lambda}$ allow to model a direction change of the control inputs. In this case, a total actuator failure of the j-th actuator corresponds to a zero column of $\boldsymbol{\Lambda}$, i.e. $\lambda_{ij} = 0, \forall i$. Geometrically, positive definiteness of $\boldsymbol{\Lambda}$ means that the faulty input vector $\mathbf{u}_f(t) = \boldsymbol{\Lambda}\mathbf{u}(t)$ is not rotated more than $\pm 90[^\circ]$ with respect to $\mathbf{u}$. This can be seen from

$$\mathbf{u}^T(t)\,\mathbf{u}_f(t) = \mathbf{u}^T(t)\,\boldsymbol{\Lambda}\mathbf{u}(t) > 0.$$

The reason of the presented analysis is that in adaptive control the most common assumption on the control effectiveness matrix $\boldsymbol{\Lambda}$ is positive definiteness. From Table

Assumption on $\mathbf{\Lambda}$	**Corresponding Faults**
positive definite	degradation of control effectiveness
invertible (non-singular)	degradation of control effectiveness, control direction inversion
positive semi-definite	degradation of control effectiveness, actuator failure, stuck actuator
no restriction	degradation of control effectiveness, actuator failure, stuck actuator, control direction inversion

Table 3.1: Control effectiveness matrix and the corresponding faults

3.1, it can be seen that this prevents the application of the adaptive methods to total failure cases. In the stability assessment, this is information is used e.g. to conclude that the uncertain plant (3.1) is controllable. For systems that are not overactuated, a non-singular matrix $\mathbf{\Lambda}$ is a necessary condition if all the virtual controls $\boldsymbol{\nu}(t)$ are needed to achieve the control task. Since in this case the input matrix $\mathbf{B}_a$ is a square matrix, it follows that the product $\mathbf{B}_a\mathbf{\Lambda}$ is singular for a singular control effectiveness matrix $\mathbf{\Lambda}$. In this case, equation (3.5) cannot be solved for $\mathbf{u}(t)$. This is different for an overactuated system because $\mathbf{B}_a$ is a full row rank matrix with more columns than rows. Then, equation (3.5) might still have a solution for $\mathbf{u}(t)$ for a singular control effectiveness matrix $\mathbf{\Lambda}$. This difference is the key to the application of adaptive methods to the problem of fault tolerant control for systems with redundant actuators and the basis for the PRO approach.

In order to illustrate the theory, the following example of a simple overactuated linear system is introduced.

Example 3.1 (Simple Overactuated Linear System) *Consider the following second order linear system corresponding to the approximated roll dynamics of an airplane [19, pp. 328] with the addition of a redundant input:*

$$\underbrace{\begin{pmatrix}\dot{p}(t)\\ \dot{\Phi}(t)\end{pmatrix}}_{\dot{\mathbf{x}}_p} = \underbrace{\begin{bmatrix}L_p & 0\\ 1 & 0\end{bmatrix}}_{\mathbf{A}_p}\underbrace{\begin{pmatrix}p(t)\\ \Phi(t)\end{pmatrix}}_{\mathbf{x}_p} + \underbrace{\begin{bmatrix}L_\xi & L_\xi\\ 0 & 0\end{bmatrix}}_{\mathbf{B}_p}\underbrace{\begin{bmatrix}\lambda_1 & 0\\ 0 & \lambda_2\end{bmatrix}}_{\mathbf{\Lambda}}\underbrace{\begin{pmatrix}\xi_1(t)\\ \xi_2(t)\end{pmatrix}}_{\mathbf{u}}. \tag{3.6}$$

The states of the system are the bank angle $\Phi(t) \in \mathbb{R}$ and the roll angular rate $p(t) \in \mathbb{R}$. The inputs are the deflections of two aileron pairs $\xi_1(t), \xi_2(t) \in \mathbb{R}$. The plant parameters are $L_p \in \mathbb{R}$ and $L_\xi < 0 \in \mathbb{R}$. Faults are modeled in terms of effectiveness of the two control inputs $\lambda_1, \lambda_2 \in \mathbb{R}$.

The goal of model reference adaptive control is that the plant (3.6) behaves like the following reference model:

$$\underbrace{\begin{pmatrix}\dot{p}_m(t)\\ \dot{\Phi}_m(t)\end{pmatrix}}_{\mathbf{x}_m} = \underbrace{\begin{bmatrix}-k_d & -k_p\\ 1 & 0\end{bmatrix}}_{\mathbf{A}_m}\underbrace{\begin{pmatrix}p_m(t)\\ \Phi_m(t)\end{pmatrix}}_{\mathbf{x}_m} + \underbrace{\begin{bmatrix}k_p\\ 0\end{bmatrix}}_{\mathbf{B}_m}\underbrace{\Phi_r(t)}_{\mathbf{r}}. \tag{3.7}$$

Here, $\Phi_m(t), p_m(t) \in \mathbb{R}$ are the states of the reference model, $\Phi_r(t) \in \mathbb{R}$ is the reference bank angle and $k_p, k_d > 0 \in \mathbb{R}$ are positive constants. Hence, the reference model is a stable PT2 element [101, Sec. 5.6].

Note the plant (3.6) *has two control inputs which have exactly the same effect on the plant and therefore the system is overactuated. Formally, this is reflected in the fact that* $r = \operatorname{rank}(\mathbf{B}_p) = 1 < n = 2$*, which means that there is only one virtual control* $\nu(t)$ *available.*

In the next sections, the Control Task 3.1 is solved using different state-of-the-art MRAC approaches [78, 87, 92, 117]: direct and predictor based MRAC. They are analyzed regarding fault tolerance and actuator redundancy and subsequently the PRO approach is presented as a solution to the current limitations.

3.2 Direct MRAC

In direct MRAC, the Control Task 3.1 is solved by formulating a control law with time varying gains in order to compensate for the uncertainties within the plant modeling. The controller gains are adapted *directly* without estimating the unknown plant parameters beforehand. The differential equation describing the parameter change over time is called update law. The controller structure is depicted in Figure 3.1. The standard control law is composed of online adjusted controller gains $\hat{\boldsymbol{\Theta}}(t)$ and a known regressor vector $\boldsymbol{\omega}(t, \mathbf{x}_p(t))$. For the plant (3.1), the control law is given by [92]

$$\mathbf{u}(t) = \hat{\boldsymbol{\Theta}}_x(t)\,\mathbf{x}_p(t) + \hat{\boldsymbol{\Theta}}_r(t)\,\mathbf{r}(t) + \hat{\boldsymbol{\Theta}}_\phi(t)\,\boldsymbol{\phi}(\mathbf{x}_p(t)) + \hat{\boldsymbol{\Theta}}_d(t) = \hat{\boldsymbol{\Theta}}(t)\,\boldsymbol{\omega}(t, \mathbf{x}_p), \tag{3.8}$$

with $\hat{\boldsymbol{\Theta}}_x \in \mathbb{R}^{m\times n}$, $\hat{\boldsymbol{\Theta}}_r \in \mathbb{R}^{m\times r}$, $\hat{\boldsymbol{\Theta}}_\phi \in \mathbb{R}^{m\times k}$, $\hat{\boldsymbol{\Theta}}_d \in \mathbb{R}^m$, $q = n + r + k + 1$ and

$$\begin{aligned}\boldsymbol{\omega}(t) &= [\mathbf{x}_p^T(t), \mathbf{r}^T(t), \boldsymbol{\phi}^T(\mathbf{x}_p(t)), 1]^T \in \mathbb{R}^q,\\ \hat{\boldsymbol{\Theta}}(t) &= [\hat{\boldsymbol{\Theta}}_x(t), \hat{\boldsymbol{\Theta}}_r(t), \hat{\boldsymbol{\Theta}}_\phi(t), \hat{\boldsymbol{\Theta}}_d(t)] \in \mathbb{R}^{m\times q}.\end{aligned}$$

Then, by inserting the control law (3.8) in the plant dynamics (3.1), the closed-loop dynamics are

$$\begin{aligned}\dot{\mathbf{x}}_p(t) = &\left(\mathbf{A}_p + \mathbf{B}_p\boldsymbol{\Lambda}\hat{\boldsymbol{\Theta}}_x(t)\right)\mathbf{x}_p(t) + \mathbf{B}_p\boldsymbol{\Lambda}\hat{\boldsymbol{\Theta}}_r(t)\,\mathbf{r}(t)\\ &+ \left(\mathbf{K}_\phi + \mathbf{B}_p\boldsymbol{\Lambda}\hat{\boldsymbol{\Theta}}_\phi(t)\right)\boldsymbol{\phi}(\mathbf{x}_p(t)) + \left(\mathbf{d} + \mathbf{B}_p\boldsymbol{\Lambda}\hat{\boldsymbol{\Theta}}_d(t)\right).\end{aligned} \tag{3.9}$$

By comparing the reference model (3.2) and the close-loop dynamics (3.9), a necessary condition for model following can be formulated in terms of the ideal parameters

$$\boldsymbol{\Theta} = [\boldsymbol{\Theta}_x, \boldsymbol{\Theta}_r, \boldsymbol{\Theta}_\phi, \boldsymbol{\Theta}_d] \in \mathbb{R}^{m\times q} \tag{3.10}$$

as follows [92, pp. 282]:

Assumption 3.1 (Matching Condition - direct MRAC) *The ideal parameters* $\boldsymbol{\Theta}$ *comply with the following condition*

$$\mathbf{Y} := \begin{bmatrix}\mathbf{A}_m - \mathbf{A}_p, & \mathbf{B}_m, & -\mathbf{K}_\phi, & -\mathbf{d}\end{bmatrix} = \mathbf{B}_p\boldsymbol{\Lambda}\boldsymbol{\Theta}. \tag{3.11}$$

Note that this condition restricts the uncertainties to the range of $\mathbf{B}_p$, the so-called matched uncertainties, and can therefore be directly compensated by the control input.

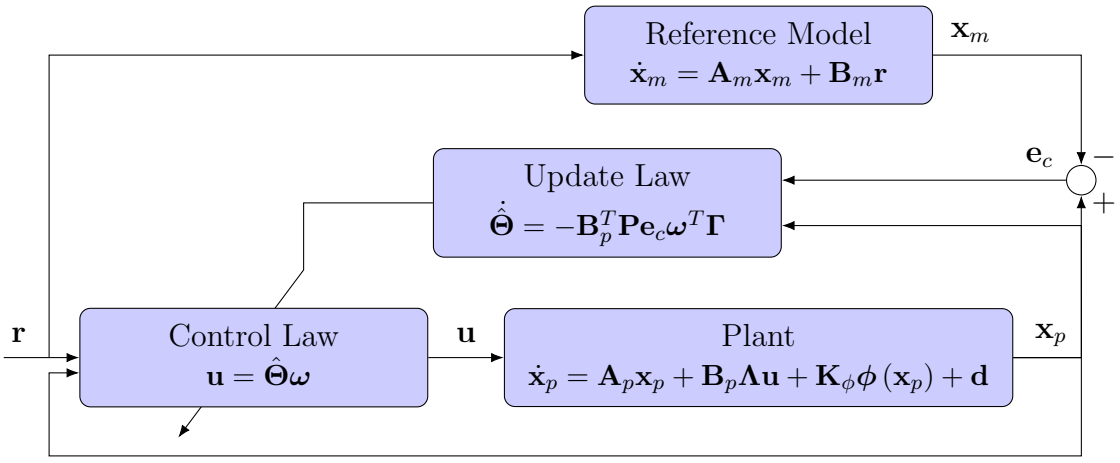


Figure 3.1: Controller Structure - Direct MRAC

The parameter error is then defined as $\widetilde{\mathbf{\Theta}}(t) := \hat{\mathbf{\Theta}}(t) - \mathbf{\Theta}$. Using (3.11), the closed-loop plant dynamics (3.9) can be rewritten as the known reference model dynamics plus a parameter error term

$$\dot{\mathbf{x}}_p(t) = \mathbf{A}_m\mathbf{x}_p(t) + \mathbf{B}_m\mathbf{r}(t) + \mathbf{B}_p\mathbf{\Lambda}\widetilde{\mathbf{\Theta}}(t)\boldsymbol{\omega}(t).$$

Given the reference model (3.2), the dynamics of the tracking error $\mathbf{e}_c(t) = \mathbf{x}_p(t) - \mathbf{x}_m(t)$ are

$$\dot{\mathbf{e}}_c(t) = \mathbf{A}_m\mathbf{e}_c(t) + \mathbf{B}_p\mathbf{\Lambda}\widetilde{\mathbf{\Theta}}(t)\boldsymbol{\omega}(t). \tag{3.12}$$

Given the error dynamics (3.12) corresponding to the control law (3.8), the update law for $\hat{\mathbf{\Theta}}(t)$ is derived from the Lyapunov stability analysis such that the Control Task 3.1 is achieved. The Lyapunov candidate function is given by [92, pp. 283]

$$V\left(\mathbf{e}_c, \widetilde{\mathbf{\Theta}}\right) = \frac{1}{2}\mathbf{e}_c^T(t)\mathbf{P}\mathbf{e}_c(t) + \frac{1}{2}tr\left(\widetilde{\mathbf{\Theta}}(t)\mathbf{\Gamma}^{-1}\widetilde{\mathbf{\Theta}}^T(t)\mathbf{\Lambda}\right),$$

using the symmetric positive definite matrix $\mathbf{P}$ from (3.3). From Lemma B.8 (Positive Definite Trace), it follows that the Lyapunov function is positive definite with respect to the errors $\mathbf{e}_c$ and $\widetilde{\mathbf{\Theta}}$ in the case of a positive definite matrix $\mathbf{\Lambda}$ and a symmetric positive definite matrix $\mathbf{\Gamma} \in \mathbb{R}^{q\times q}$. Hence, in the standard MRAC proof, the following assumption regarding the control authority has been made [92]:

Assumption 3.2 (Control Authority - direct MRAC) *The matrix $\mathbf{\Lambda}$ is positive definite.*

Assumption 3.2 can be interpreted as requiring a minimum control authority plus the knowledge of the effectiveness direction of the actuators given by the known matrix $\mathbf{B}_p$. Using Lemma B.2 (Trace and Scalar Product) and the Lyapunov equation (3.3), the derivative of the Lyapunov function along the trajectories of (3.12) is given by

$$\begin{aligned}
\dot{V}(t) &= -\frac{1}{2}\mathbf{e}_c^T(t)\mathbf{Q}\mathbf{e}_c(t) + \mathbf{e}_c^T(t)\mathbf{P}\mathbf{B}_p\mathbf{\Lambda}\widetilde{\mathbf{\Theta}}(t)\boldsymbol{\omega}(t) + tr\left(\widetilde{\mathbf{\Theta}}(t)\mathbf{\Gamma}^{-1}\dot{\widetilde{\mathbf{\Theta}}}^T(t)\mathbf{\Lambda}\right), \\
&= -\frac{1}{2}\mathbf{e}_c^T(t)\mathbf{Q}\mathbf{e}_c(t) + tr\left(\widetilde{\mathbf{\Theta}}(t)\boldsymbol{\omega}(t)\cdot\mathbf{e}_c^T(t)\mathbf{P}\mathbf{B}_p\mathbf{\Lambda}\right) + tr\left(\widetilde{\mathbf{\Theta}}(t)\mathbf{\Gamma}^{-1}\dot{\widetilde{\mathbf{\Theta}}}^T(t)\mathbf{\Lambda}\right), \\
&= -\frac{1}{2}\mathbf{e}_c^T(t)\mathbf{Q}\mathbf{e}_c(t) + tr\left(\widetilde{\mathbf{\Theta}}(t)\left(\boldsymbol{\omega}(t)\mathbf{e}_c^T(t)\mathbf{P}\mathbf{B}_p + \mathbf{\Gamma}^{-1}\dot{\widetilde{\mathbf{\Theta}}}^T(t)\right)\mathbf{\Lambda}\right).
\end{aligned}$$

By choosing the standard direct MRAC update law [92, pp. 285]

$$\dot{\widetilde{\mathbf{\Theta}}}(t) = -\left(\mathbf{\Gamma}\boldsymbol{\omega}(t)\,\mathbf{e}_c^T(t)\,\mathbf{P}\mathbf{B}_p\right)^T = -\mathbf{B}_p^T\mathbf{P}\mathbf{e}_c(t)\,\boldsymbol{\omega}^T(t)\,\mathbf{\Gamma}, \tag{3.13}$$

the derivative of the Lyapunov function $\dot{V}(\mathbf{e}_c, \widetilde{\mathbf{\Theta}}) = -\frac{1}{2}\mathbf{e}_c^T(t)\,\mathbf{Q}\mathbf{e}_c(t)$ is negative semi-definite. From Theorem 4.8 (Uniform Stability) [87], it follows that the equilibrium $(\mathbf{e}_c, \widetilde{\mathbf{\Theta}}) = (\mathbf{0}, \mathbf{0})$ is globally uniformly stable and $\mathbf{e}_c, \widetilde{\mathbf{\Theta}} \in \mathcal{L}_\infty$.

In the following, the goal is to show asymptotic convergence of the tracking error $\mathbf{e}_c(t)$ to zero based on Barbalat's Lemma (Lemma B.9). In order to show that the function $\dot{V}(\mathbf{e}_c(t)) = -\frac{1}{2}\mathbf{e}_c^T(t)\,\mathbf{Q}\mathbf{e}_c(t) \to 0$ for $t \to \infty$, it needs to be shown that $\dot{V}(\mathbf{e}_c(t))$ is a uniform continuous function of t and that the limit

$$\lim_{t\to\infty} \int_{\tau=0}^{t} \dot{V}(\mathbf{e}_c(\tau))\,d\tau \tag{3.14}$$

exists and is finite.

Since $V(\mathbf{e}_c(t), \widetilde{\mathbf{\Theta}}(t))$ is monotonically non-increasing and bounded from below by zero, i.e. $0 \leq V(\mathbf{e}_c(t), \widetilde{\mathbf{\Theta}}(t)) \leq V(\mathbf{e}_c(0), \widetilde{\mathbf{\Theta}}(0))$, it converges as $t \to \infty$. Then,

$$\left|\int_{\tau=0}^{t} \dot{V}(\mathbf{e}_c(\tau))\,d\tau\right| = \left|V(\mathbf{e}_c(t), \widetilde{\mathbf{\Theta}}(t)) - V(\mathbf{e}_c(0), \widetilde{\mathbf{\Theta}}(0))\right| < \infty,$$

which implies that (3.14) exists and is finite.

The next step is to show that $\dot{V}(\mathbf{e}_c(t))$ is uniformly continuous. Because $\mathbf{x}_m(t)$ and $\mathbf{e}_c(t)$ are bounded, $\mathbf{x}_p(t)$ is bounded too. This implies boundedness of $\boldsymbol{\phi}(\mathbf{x}_p(t))$ and $\boldsymbol{\omega}(\mathbf{x}_p(t), \mathbf{r}(t), \boldsymbol{\phi}(\mathbf{x}_p(t)))$. From the error dynamics (3.12) it follows that $\dot{\mathbf{e}}_c(t)$ is uniformly bounded in t for all $t \geq 0$. Therefore, $\ddot{V}(t) = -\mathbf{e}_c^T(t)\,\mathbf{Q}\dot{\mathbf{e}}_c(t)$ is also bounded uniformly in t for all $t \geq 0$. Then, $\dot{V}(\mathbf{e}_c(t))$ is uniformly continuous for $t \geq 0$. Finally, using Lemma B.9 (Barbalat) it follows that $\dot{V}(\mathbf{e}_c(t)) \to 0$ and hence $\mathbf{e}_c(t) \to \mathbf{0}$ for $t \to \infty$. The presented result regarding MRAC corresponds to Theorem 9.2 in [92] and is summarized in the next theorem.

Theorem 3.3 (Direct MRAC) *Consider the plant dynamics* (3.1)*, the reference model* (3.2)*, the control law* (3.8) *and the update law* (3.13)*. Furthermore let Assumption 3.1 (Matching Condition - direct MRAC) and Assumption 3.2 (Control Authority - direct MRAC) hold. Then, the equilibrium* $(\mathbf{e}_c, \widetilde{\mathbf{\Theta}}) = (\mathbf{0}, \mathbf{0})$ *is globally uniformly stable and* $\mathbf{e}_c(t) \to \mathbf{0}$ *for* $t \to \infty$*.* ■

In the context of redundant actuators and fault tolerance, this result is not sufficient because Assumption 3.2 (Control Authority - direct MRAC) excludes the total failure of an actuator as seen in Table 3.1. Within the proof, a positive definite $\mathbf{\Lambda}$ is needed in order to build a positive definite Lyapunov candidate.

Example 3.2 (Application of direct MRAC) *In order to understand the physical meaning of Assumption 3.2 (Control Authority - direct MRAC) and see that in the case of overactuated systems it is not a necessary condition, direct MRAC is applied to the system described in Example 3.1. In this example, there is neither a nonlinear term nor a constant disturbance in the plant dynamics* (3.6) *(stuck actuators are not considered). Hence, the regressor vector is*

$\boldsymbol{\omega}(t) = [\mathbf{x}_p^T(t), \mathbf{r}^T(t)]^T \in \mathbb{R}^3$, the controller gains are $\hat{\boldsymbol{\Theta}}(t) \in \mathbb{R}^{2\times 3}$ and the control law is $\mathbf{u}(t) = \hat{\boldsymbol{\Theta}}(t)\boldsymbol{\omega}(t)$ from (3.8). The gains corresponding to each actuator represent one row of $\hat{\boldsymbol{\Theta}}(t)$ and are defined by

$$\hat{\boldsymbol{\Theta}}(t) = \begin{bmatrix} \hat{\boldsymbol{\Theta}}_{\xi_1}^T(t) \\ \hat{\boldsymbol{\Theta}}_{\xi_2}^T(t) \end{bmatrix},$$

with $\hat{\boldsymbol{\Theta}}_{\xi_1}, \hat{\boldsymbol{\Theta}}_{\xi_2} \in \mathbb{R}^3$. Assumption 3.1 (Matching Condition - direct MRAC) reads for this example as

$$\mathbf{Y} := \begin{bmatrix} -kd - Lp & -kp & kp \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} L_\xi & L_\xi \\ 0 & 0 \end{bmatrix} \boldsymbol{\Lambda}\boldsymbol{\Theta}.$$

It has a solution as long as the matrix $[L_\xi, L_\xi] \cdot \boldsymbol{\Lambda}$ has a pseudo-inverse. Interestingly, by using the update law (3.13) it can be seen that the derivatives of every row

$$\dot{\hat{\boldsymbol{\Theta}}}(t) = \begin{bmatrix} \dot{\hat{\boldsymbol{\Theta}}}_{\xi_1}^T(t) \\ \dot{\hat{\boldsymbol{\Theta}}}_{\xi_2}^T(t) \end{bmatrix} = -\begin{bmatrix} L_\xi & 0 \\ L_\xi & 0 \end{bmatrix} \mathbf{P}\mathbf{e}_c(t)\boldsymbol{\omega}^T(t)\boldsymbol{\Gamma} \tag{3.15}$$

are exactly the same. This makes sense since through the definition of $\mathbf{B}_p$ it is assumed that both actuators have the same effect on the plant. From Theorem 3.3 (Direct MRAC), it is known that as long as $\boldsymbol{\Lambda}$ is positive definite the control objective is achieved.

The question now is what happens if one of the two actuators fails. Without loss of generality, let the second actuator have a failure $\lambda_2 = 0$. In this case, Theorem 3.3 (Direct MRAC) fails to prove stability since $\boldsymbol{\Lambda}$ is positive semi-definite and Assumption 3.2 (Control Authority - direct MRAC) does not hold. To show stability, the plant (3.6) can be rewritten as

$$\begin{pmatrix} \dot{p}(t) \\ \dot{\Phi}(t) \end{pmatrix} = \begin{bmatrix} L_p & 0 \\ 1 & 0 \end{bmatrix} \begin{pmatrix} p(t) \\ \Phi(t) \end{pmatrix} + \begin{bmatrix} L_\xi \\ 0 \end{bmatrix} \lambda_1 \xi_1(t). \tag{3.16}$$

The matching condition reduces to

$$\mathbf{Y} = L_\xi \cdot \lambda_1 \cdot \boldsymbol{\Theta}_{\xi_1}^T$$

and has a solution for $\lambda_1 \neq 0$. The update law for the first actuator is then

$$\dot{\hat{\boldsymbol{\Theta}}}_{\xi_1}^T(t) = -\begin{bmatrix} L_\xi & 0 \end{bmatrix} \mathbf{P}\mathbf{e}_c(t)\boldsymbol{\omega}^T(t)\boldsymbol{\Gamma},$$

which is exactly the same as before in (3.15). Using Theorem 3.3 (Direct MRAC), now it is possible to prove that the equilibrium $(\mathbf{e}_c, \widetilde{\boldsymbol{\Theta}}_{\xi_1}) = (\mathbf{0}, \mathbf{0})$ is globally uniformly stable and $\mathbf{e}_c(t) \to \mathbf{0}$ for $t \to \infty$ as long as $\lambda_1 > 0$. Because of the update laws' equality $\dot{\hat{\boldsymbol{\Theta}}}_{\xi_1}(t) = \dot{\hat{\boldsymbol{\Theta}}}_{\xi_2}(t)$, it also follows that $\hat{\boldsymbol{\Theta}}(t)$ is bounded.

For the case that $\lambda_2 > 0$ and $\lambda_1 = 0$, a similar result holds. Hence, the control objective is achieved using the update law (3.15) as long as $\lambda_1, \lambda_2 \geq 0$ and either $\lambda_1 > 0$ or $\lambda_2 > 0$ holds. In the case that both actuators fail, i.e. $\lambda_1 = \lambda_2 = 0$, it is clear that the control objective cannot be achieved since the plant dynamics are

$$\begin{pmatrix} \dot{\Phi}(t) \\ \dot{p}(t) \end{pmatrix} = \begin{bmatrix} L_p & 0 \\ 1 & 0 \end{bmatrix} \begin{pmatrix} \Phi(t) \\ p(t) \end{pmatrix}.$$

3.3 Direct MRAC PRO

For the controller, fault tolerance means that the Control Task 3.1 should be achieved even in the presence of faults. In the direct MRAC case, corresponding to Theorem 3.3 (Direct MRAC), the goal is to prove stability of the tracking error $\mathbf{e}_c(t)$ and parameter errors $\widetilde{\mathbf{\Theta}}(t)$ and asymptotic convergence of the tracking error $\mathbf{e}_c(t)$ to zero for the largest possible set of unknown faults, i.e. of $\mathbf{\Lambda}$. From the last section, it is clear that the standard direct MRAC stability proof is not able to take actuator failures into account. Assumption 3.2 (Control Authority - direct MRAC) was identified as the limiting factor and by Example 3.1, it was shown that this assumption is too restrictive for systems with actuator redundancy. In this section, the Parameter Reduction due to Overactuation (PRO) approach is introduced as a solution to this problem. Its application to direct MRAC was first introduced in [47].

In Example 3.1, the actuator failure case has been shown to be stable by reducing the plant and taking the parameters corresponding to the failed actuator out of the analysis first as they do not influence the system dynamics. The standard direct MRAC control and update laws were still used. This changed Assumption 3.2 to a reduced set which corresponded to the original $\mathbf{\Lambda} \in \mathbb{R}^{2\times 2}$ being singular, e.g. $\lambda_1 > 0$ and $\lambda_2 = 0$. In this section, this idea is used to formulate a general approach for selecting the reduced set of parameters that are going to be taken into account within the stability analysis. In this way Assumption 3.2 can be relaxed in order to accommodate actuator failures.

In Example 3.1, the columns of $\mathbf{B}_p$ are the same and therefore the update law (3.15) for the two actuators is identical. This means that if the row $\hat{\mathbf{\Theta}}_{\xi_1}^T(t)$ is known, there is no extra information in $\hat{\mathbf{\Theta}}_{\xi_2}^T(t)$ and the problem is overparameterized due to the actuator redundancy. In this section, it will be shown that this overparameterization always exists for overactuated systems and that a reduction of the parameters to a minimum set is the key to handling failures within the adaptive control framework.

In order to clarify if there is a set of parameters that contains no extra information, notice that using the direct MRAC update law (3.13), the parameters are updated only in the range of $\mathbf{B}_p^T$. In an overactuated system, the range of $\mathbf{B}_p^T$ does not have full dimension because $\mathbf{B}_p$ has a null space due to $\operatorname{rank}(\mathbf{B}_p) = r < m$. Therefore, a subset of the parameters $\hat{\mathbf{\Theta}}(t)$ contains no extra information, i.e. the subset is not updated by the update law. In order to identify this subset, the singular value decomposition (SVD) of $\mathbf{B}_p$ can be used. With $\operatorname{rank}(\mathbf{B}_p) = r$ and $r \leq \min(n, m)$ the SVD is given by

$$\mathbf{B}_p = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \tag{3.17}$$

where $\mathbf{U} \in \mathbb{R}^{n\times n}, \mathbf{\Sigma} \in \mathbb{R}^{n\times m}, \mathbf{V} \in \mathbb{R}^{m\times m}$ [74, Th. 2.6.3]. $\mathbf{U}, \mathbf{V}$ are orthogonal matrices, i.e. $\mathbf{U}\mathbf{U}^T = \mathbf{I}$ and $\mathbf{V}\mathbf{V}^T = \mathbf{I}$. $\mathbf{\Sigma}$ is a rectangular diagonal matrix with non-negative entries on the diagonal which can be written as

$$\mathbf{\Sigma} = \begin{bmatrix} \mathbf{D} & \mathbf{0}_{r\times m-r} \\ \mathbf{0}_{n-r\times r} & \mathbf{0}_{n-r\times m-r} \end{bmatrix},$$

where $\mathbf{D} \in \mathbb{R}^{r\times r}$ is a diagonal positive definite matrix. The diagonal elements are the singular values of $\mathbf{B}_p$ and have a decreasing order. Taking this structure into account, it is clear that if we partition $\mathbf{U} = [\mathbf{U}_r, \mathbf{U}_n]$ and $\mathbf{V} = [\mathbf{V}_r, \mathbf{V}_n]$, such that $\mathbf{U}_r \in \mathbb{R}^{n\times r}$,

$\mathbf{U}_n \in \mathbb{R}^{n\times(n-r)}$, $\mathbf{V}_r \in \mathbb{R}^{m\times r}$ and $\mathbf{V}_n \in \mathbb{R}^{m\times(m-r)}$, the matrix $\mathbf{B}_p$ can be alternatively written as

$$\mathbf{B}_p = \mathbf{U}_r \mathbf{D} \mathbf{V}_r^T. \tag{3.18}$$

Remark 3.4 *The SVD from $\mathbf{B}_p$ has a physical interpretation. The columns of $\mathbf{U}_r$ are a basis of the range of $\mathbf{B}_p$ meaning that they are the directions in the state space that can be influenced by the control input. On the other hand, the columns of $\mathbf{V}_r$ represent the directions in the input space that have an effect on the state space. Similarly, the columns of $\mathbf{V}_n$ are a basis of the null space of $\mathbf{B}_p$, i.e. the actuator combinations that have no influence in the plant dynamics. In an overactuated system, the dimension of the null space is always greater than zero because* $\operatorname{rank}(\mathbf{B}_p) = r < m$.

Example 3.3 (Illustration of the SVD of $\mathbf{B}_p$) *Remark 3.4 can be illustrated using Example 3.1. Taking a look at the SVD of $\mathbf{B}_p$*

$$\underbrace{\begin{bmatrix} L_\xi & L_\xi \\ 0 & 0 \end{bmatrix}}_{\mathbf{B}_p} = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{\mathbf{U}} \underbrace{\begin{bmatrix} L_\xi\sqrt{2} & 0 \\ 0 & 0 \end{bmatrix}}_{\mathbf{\Sigma}} \underbrace{\begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix}}_{\mathbf{V}^T} = \underbrace{\begin{bmatrix} 1 \\ 0 \end{bmatrix}}_{\mathbf{U}_r} \underbrace{\begin{bmatrix} L_\xi\sqrt{2} \end{bmatrix}}_{\mathbf{D}} \underbrace{\begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix}}_{\mathbf{V}_r^T}, \tag{3.19}$$

it can be seen that the system is overactuated from $\operatorname{rank}(\mathbf{B}_p) = 1 < 2$. *Furthermore, the inputs influence only in the $\mathbf{U}_r$ direction, i.e. only the roll angular acceleration $\dot{p}(t)$. The null space of $\mathbf{B}_p$ corresponds to $\mathbf{V}_n^T = [-\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}]$. In this case the actuators cancel each other because $\xi_1 = -\xi_2$. A graphical interpretation of the SVD can be seen in Figure 3.2.*

In the following consider the ideal parameters as defined in (3.10). In order to separate the directions in the parameter space that contain information from the ones that don't, the new parameters $\mathbf{\Theta}_b \in \mathbb{R}^{m\times q}$ are defined by the invertible transformation

$$\mathbf{\Theta}_b = \mathbf{V}^T \mathbf{\Theta}. \tag{3.20}$$

It allows the partition of the parameters in two main groups

$$\mathbf{\Theta}_b = \begin{bmatrix} \mathbf{\Theta}_r \\ \mathbf{\Theta}_n \end{bmatrix} = \begin{bmatrix} \mathbf{V}_r^T \mathbf{\Theta} \\ \mathbf{V}_n^T \mathbf{\Theta} \end{bmatrix}, \tag{3.21}$$

where $\mathbf{\Theta}_r \in \mathbb{R}^{r\times q}$ are the parameters in the range of $\mathbf{B}_p^T$ and $\mathbf{\Theta}_n \in \mathbb{R}^{(m-r)\times q}$ lies in the null space of $\mathbf{B}_p$. The original parameters can be computed by the inverse transformation of (3.20) as

$$\mathbf{\Theta} = \mathbf{V}\mathbf{\Theta}_b = \mathbf{V}_r\mathbf{\Theta}_r + \mathbf{V}_n\mathbf{\Theta}_n. \tag{3.22}$$

Let the parameter estimates $\hat{\mathbf{\Theta}}(t)$ be analogously transformed as in (3.20) and partitioned as in (3.21) such that $\hat{\mathbf{\Theta}}_b(t) \in \mathbb{R}^{m\times q}$, $\hat{\mathbf{\Theta}}_r(t) \in \mathbb{R}^{r\times q}$ and $\hat{\mathbf{\Theta}}_n(t) \in \mathbb{R}^{(m-r)\times q}$. Their update laws are computed using the update law from the last section (3.13), the SVD of $\mathbf{B}_p$ (3.18) and the orthogonality of $\mathbf{V}$ (Lemma B.10) as follows

$$\dot{\hat{\mathbf{\Theta}}}_r(t) = \mathbf{V}_r^T \dot{\hat{\mathbf{\Theta}}}(t) = -\mathbf{V}_r^T \underbrace{\mathbf{V}_r \mathbf{D} \mathbf{U}_r^T}_{=\mathbf{B}_p^T} \mathbf{P}\mathbf{e}_c(t)\, \boldsymbol{\omega}^T(t)\, \mathbf{\Gamma} = -\mathbf{D}\mathbf{U}_r^T \mathbf{P}\mathbf{e}_c(t)\, \boldsymbol{\omega}^T(t)\, \mathbf{\Gamma}, \tag{3.23a}$$

$$\dot{\hat{\mathbf{\Theta}}}_n(t) = \mathbf{V}_n^T \dot{\hat{\mathbf{\Theta}}}(t) = -\mathbf{V}_n^T \underbrace{\mathbf{V}_r \mathbf{D} \mathbf{U}_r^T}_{=\mathbf{B}_p^T} \mathbf{P}\mathbf{e}_c(t)\, \boldsymbol{\omega}^T(t)\, \mathbf{\Gamma} = \mathbf{0}. \tag{3.23b}$$

It is clear that the parameters $\hat{\boldsymbol{\Theta}}(t)$ are only updated in the subspace corresponding to $\hat{\boldsymbol{\Theta}}_r(t)$. Hence, we can write the parameter $\hat{\boldsymbol{\Theta}}(t)$ as follows

$$\begin{aligned}\hat{\boldsymbol{\Theta}}(t) &= \hat{\boldsymbol{\Theta}}(0) + \int_{\tau=0}^{t} \dot{\hat{\boldsymbol{\Theta}}}(\tau)\, d\tau, \\ &= \mathbf{V}_r \hat{\boldsymbol{\Theta}}_r(0) + \mathbf{V}_n \hat{\boldsymbol{\Theta}}_n(0) + \int_{\tau=0}^{t} \mathbf{V}_r \dot{\hat{\boldsymbol{\Theta}}}_r(\tau) + \mathbf{V}_n \dot{\hat{\boldsymbol{\Theta}}}_n(\tau)\, d\tau, \\ &= \mathbf{V}_r \hat{\boldsymbol{\Theta}}_r(t) + \mathbf{V}_n \hat{\boldsymbol{\Theta}}_n(0).\end{aligned}$$

This means that $\hat{\boldsymbol{\Theta}}(t)$ is fully defined by $\hat{\boldsymbol{\Theta}}_r(t)$ and the initial condition $\hat{\boldsymbol{\Theta}}_n(0)$. Therefore, only the parameters $\hat{\boldsymbol{\Theta}}_r(t)$ need to be updated. This leads to a parameter reduction for overactuated systems from $m \cdot q$ to $r \cdot q$, which reduces the computational effort considerably for a large number of regressor functions (large q). This special way of decreasing the number adaptive parameters is called Parameter Reduction due to Overactuation (PRO). It is the key that enables us to include actuator failures within adaptive control stability proofs of overactuated systems in the subsequent sections. The control law (3.8) can be equivalently implemented using (3.22) as

$$\mathbf{u}(t) = \left(\mathbf{V}_r \hat{\boldsymbol{\Theta}}_r(t) + \mathbf{V}_n \hat{\boldsymbol{\Theta}}_n(0)\right) \cdot \boldsymbol{\omega}(t, \mathbf{x}_p). \tag{3.24}$$

Knowing that the parameters $\hat{\boldsymbol{\Theta}}(t)$ are only updated in the range of $\mathbf{B}_p^T$, the question arises if Assumption 3.2 (Control Authority - direct MRAC) can be relaxed. There-

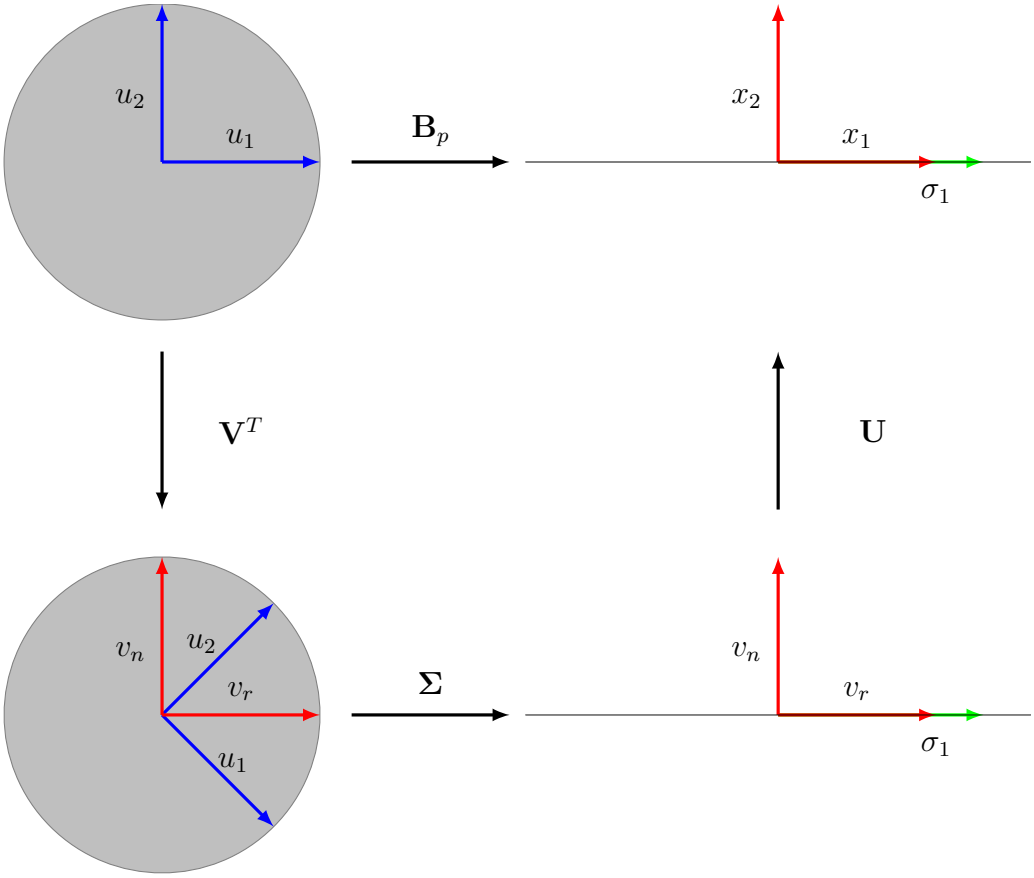


Figure 3.2: Singular Value Decomposition - Example 3.1

fore, it first needs to be checked if the matching condition (3.11) can be satisfied. If $\mathbf{Y} = \mathbf{B}_p\mathbf{\Lambda}\mathbf{\Theta}$ has a solution, the solution for $\mathbf{\Theta}$ is a connected convex set given by

$$\mathbf{\Theta} = (\mathbf{B}_p\mathbf{\Lambda})^+ \mathbf{Y} + \left(\mathbf{I} - (\mathbf{B}_p\mathbf{\Lambda})^+ \mathbf{B}_p\mathbf{\Lambda}\right) \mathbf{W}, \tag{3.25}$$

where $\mathbf{W} \in \mathbb{R}^{m\times q}$ is an arbitrary matrix representing the offset of the solution within the null space of $\mathbf{B}_p\mathbf{\Lambda}$ [59, Sec. 3.5.1.4]. In order to achieve a solution out of the set (3.25) with the update law (3.13) and accordingly with (3.23), at least one solution must be such that $\mathbf{V}_n^T\mathbf{\Theta} = \hat{\mathbf{\Theta}}_n(0)$. Inserting the parameter transformation (3.20) and the SVD of $\mathbf{B}_p$ (3.18) in the matching condition (3.11) leads to

$$\begin{aligned}\mathbf{Y} = \mathbf{B}_p\mathbf{\Lambda}\mathbf{\Theta} &= \mathbf{U}_r\mathbf{D}\mathbf{V}_r^T\mathbf{\Lambda}\mathbf{V}\mathbf{\Theta}_b,\\ &= \mathbf{U}_r\mathbf{D}\mathbf{V}_r^T\mathbf{\Lambda}\left(\mathbf{V}_r\mathbf{\Theta}_r + \mathbf{V}_n\mathbf{\Theta}_n\right),\\ &= \mathbf{U}_r\mathbf{D}\left(\mathbf{V}_r^T\mathbf{\Lambda}\mathbf{V}_r\mathbf{\Theta}_r + \mathbf{V}_r^T\mathbf{\Lambda}\mathbf{V}_n\mathbf{\Theta}_n\right).\end{aligned}$$

In order to solve this equation for every initial value $\hat{\mathbf{\Theta}}_n(0)$, the following relaxed main assumption is formulated instead of Assumption 3.2 (Control Authority - direct MRAC).

Assumption 3.5 (Control Authority - direct MRAC PRO) *Given the SVD of* $\mathbf{B}_p$ (3.18)*, the matrix* $\mathbf{\Lambda}_{rr} := \mathbf{V}_r^T\mathbf{\Lambda}\mathbf{V}_r$ *is positive definite.*

Proposition 3.6 (Ideal Parameters - MRAC PRO) *Let Assumptions 3.1 (Matching Condition - direct MRAC) and 3.5 (Control Authority - direct MRAC PRO) hold. Then, the solution set of ideal parameters* (3.25) *always contains the unique ideal parameter* $\mathbf{\Theta}$ *compatible with the update laws* (3.23) *such that* $\mathbf{V}_n^T\mathbf{\Theta} = \mathbf{\Theta}_n = \hat{\mathbf{\Theta}}_n(0)$*.*

Proof: First note that regardless of the value of $\mathbf{\Lambda}$, for a solution of the matching condition (3.11) to exist, $\mathbf{Y}$ must lie in the range of $\mathbf{B}_p$, i.e. $\mathbf{U}_n^T\mathbf{Y} = \mathbf{0}$. Therefore using Lemma B.10 (Orthogonal Matrix) it holds that

$$\mathbf{Y} = \left(\mathbf{I} - \mathbf{U}_n\mathbf{U}_n^T\right)\mathbf{Y} = \mathbf{U}_r\mathbf{U}_r^T\mathbf{Y}.$$

Then, as $\mathbf{\Lambda}_{rr}$ is assumed to be positive definite, it is invertible and the following ideal parameters exist

$$\mathbf{\Theta}_r = \mathbf{\Lambda}_{rr}^{-1}\left(\mathbf{D}^{-1}\mathbf{U}_r^T\mathbf{Y} - \mathbf{V}_r^T\mathbf{\Lambda}\mathbf{V}_n\hat{\mathbf{\Theta}}_n(0)\right), \quad \mathbf{\Theta}_n = \hat{\mathbf{\Theta}}_n(0),$$

such that $\mathbf{Y} = \mathbf{B}_p\mathbf{\Lambda}\left(\mathbf{V}_r\mathbf{\Theta}_r + \mathbf{V}_n\mathbf{\Theta}_n\right)$. ■

It is important to note that Assumption 3.5 (Control Authority - direct MRAC PRO) includes all the cases where $\mathbf{\Lambda}$ is positive definite and hence it is more general than Assumption 3.2 (Control Authority - direct MRAC). This is formulated in the following proposition.

Proposition 3.7 *Let the matrix* $\mathbf{\Lambda}$ *be positive definite, then* $\mathbf{\Lambda}_{rr} = \mathbf{V}_r^T\mathbf{\Lambda}\mathbf{V}_r$ *is also positive definite.*

Proof: Because $\mathbf{V}$ is an orthogonal matrix, it holds that $\mathbf{V}^T\mathbf{\Lambda}\mathbf{V}$ is positive definite if and only if $\mathbf{\Lambda}$ is positive definite, see Lemma B.6 (Orthogonal Transformation Invariance). If $\mathbf{V}^T\mathbf{\Lambda}\mathbf{V}$ is positive definite, then all of its principal submatrices are also positive definite [74, Obs. 7.1.2]. Noting that the $\mathbb{R}^{r\times r}$ leading submatrix is $\mathbf{V}_r^T\mathbf{\Lambda}\mathbf{V}_r = \mathbf{\Lambda}_{rr}$ completes the proof. ■

In the following, the main result of this section is presented as the extension of Theorem 3.3 (Direct MRAC) for systems with redundant inputs using the PRO approach.

Theorem 3.8 (direct MRAC PRO) *Compared to Theorem 3.3 (Direct MRAC), Assumption 3.2 (Control Authority - direct MRAC) is relaxed to Assumption 3.5 (Control Authority - direct MRAC PRO) to take actuator failures into account in the case of systems with redundant control inputs. Furthermore, the reduced set of parameters is used.*

Consider the plant dynamics (3.1)*, the reference model* (3.2)*, the control law* (3.24) *(or equivalently* (3.8)*) and the update laws* (3.23) *(or equivalently* (3.13)*). Let Assumptions 3.1 (Matching Condition - direct MRAC) and 3.5 (Control Authority - direct MRAC PRO) hold. In this case, the unique ideal parameters* $\mathbf{\Theta}$ *exist as derived in Proposition 3.6 (Ideal Parameters - MRAC PRO) and the equilibrium* $(\mathbf{e}_c, \widetilde{\mathbf{\Theta}}) = (\mathbf{0}, \mathbf{0})$ *is globally uniformly stable and* $\mathbf{e}_c(t) \to \mathbf{0}$ *for* $t \to \infty$*.*

Proof: Consider the following Lyapunov candidate function

$$V(\mathbf{e}_c, \widetilde{\mathbf{\Theta}}_r, \widetilde{\mathbf{\Theta}}_n) = \frac{1}{2}\mathbf{e}_c^T(t)\,\mathbf{P}\mathbf{e}_c(t) + \frac{1}{2}tr\left(\widetilde{\mathbf{\Theta}}_r(t)\,\mathbf{\Gamma}^{-1}\widetilde{\mathbf{\Theta}}_r^T(t)\,\mathbf{\Lambda}_{rr}\right) + \frac{1}{2}tr\left(\widetilde{\mathbf{\Theta}}_n(t)\,\widetilde{\mathbf{\Theta}}_n^T(t)\right),$$

where $\widetilde{\mathbf{\Theta}}_r(t) = \hat{\mathbf{\Theta}}_r(t) - \mathbf{\Theta}_r$ and $\widetilde{\mathbf{\Theta}}_n(t) = \hat{\mathbf{\Theta}}_n(t) - \mathbf{\Theta}_n$ are the parameter errors. From Lemma B.8 (Positive Definite Trace) it is known that it is positive definite for a symmetric positive definite $\mathbf{\Gamma}$ and a positive definite $\mathbf{\Lambda}_{rr}$. Using Proposition 3.6, the error $\widetilde{\mathbf{\Theta}}(t)$ can be written as $\widetilde{\mathbf{\Theta}}(t) = \mathbf{V}_r\widetilde{\mathbf{\Theta}}_r(t) + \mathbf{V}_n\widetilde{\mathbf{\Theta}}_n(t) = \mathbf{V}_r\widetilde{\mathbf{\Theta}}_r(t)$ and the error dynamics (3.12) only include the parameter error $\widetilde{\mathbf{\Theta}}_r(t)$ as

$$\dot{\mathbf{e}}_c(t) = \mathbf{A}_m\mathbf{e}_c(t) + \mathbf{B}_p\mathbf{\Lambda}\mathbf{V}_r\widetilde{\mathbf{\Theta}}_r(t)\,\boldsymbol{\omega}(t). \tag{3.26}$$

In the following, the time dependency is not written out for readability. Using the update law (3.23b), the error dynamics (3.26) and the SVD of $\mathbf{B}_p$ (3.18), the derivative of the Lyapunov function is

$$\begin{aligned}\dot{V} &= -\frac{1}{2}\mathbf{e}_c^T\mathbf{Q}\mathbf{e}_c + \mathbf{e}_c^T\mathbf{P}\mathbf{B}_p\mathbf{\Lambda}\widetilde{\mathbf{\Theta}}\boldsymbol{\omega} + tr\left(\widetilde{\mathbf{\Theta}}_r\mathbf{\Gamma}^{-1}\dot{\widetilde{\mathbf{\Theta}}}_r^T\mathbf{\Lambda}_{rr}\right) + tr\left(\widetilde{\mathbf{\Theta}}_n\dot{\widetilde{\mathbf{\Theta}}}_n^T\right),\\ &= -\frac{1}{2}\mathbf{e}_c^T\mathbf{Q}\mathbf{e}_c + \mathbf{e}_c^T\mathbf{P}\left(\mathbf{U}_r\mathbf{D}\mathbf{V}_r^T\right)\mathbf{\Lambda}\mathbf{V}_r\widetilde{\mathbf{\Theta}}_r\boldsymbol{\omega} + tr\left(\widetilde{\mathbf{\Theta}}_r\mathbf{\Gamma}^{-1}\dot{\widetilde{\mathbf{\Theta}}}_r^T\mathbf{\Lambda}_{rr}\right).\end{aligned}$$

Using Lemma B.2 (Trace and Scalar Product), the definition $\mathbf{\Lambda}_{rr} = \mathbf{V}_r^T\mathbf{\Lambda}\mathbf{V}_r$ and the update law (3.23a), the derivative of the Lyapunov function is as follows

$$\begin{aligned}\dot{V} &= -\frac{1}{2}\mathbf{e}_c^T\mathbf{Q}\mathbf{e}_c + tr\left(\widetilde{\mathbf{\Theta}}_r\boldsymbol{\omega}\cdot\mathbf{e}_c^T\mathbf{P}\mathbf{U}_r\mathbf{D}\mathbf{\Lambda}_{rr}\right) + tr\left(\widetilde{\mathbf{\Theta}}_r\mathbf{\Gamma}^{-1}\dot{\widetilde{\mathbf{\Theta}}}_r^T\mathbf{\Lambda}_{rr}\right),\\ &= -\frac{1}{2}\mathbf{e}_c^T\mathbf{Q}\mathbf{e}_c + tr\left(\widetilde{\mathbf{\Theta}}_r\left(\boldsymbol{\omega}\mathbf{e}_c^T\mathbf{P}\mathbf{U}_r\mathbf{D} + \mathbf{\Gamma}^{-1}\dot{\widetilde{\mathbf{\Theta}}}_r^T\right)\mathbf{\Lambda}_{rr}\right),\\ &= -\frac{1}{2}\mathbf{e}_c^T\mathbf{Q}\mathbf{e}_c.\end{aligned}$$

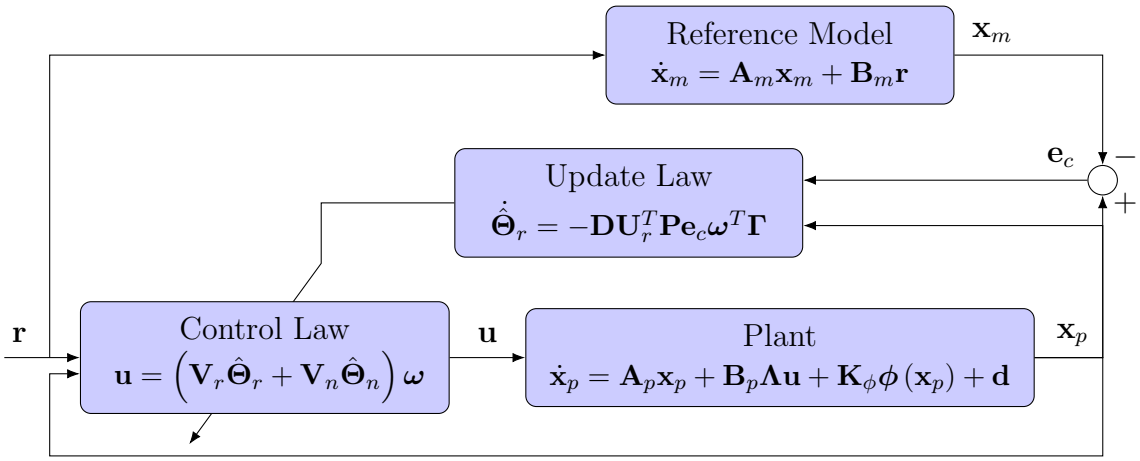


Figure 3.3: Controller Structure - Direct MRAC PRO

Subsequently, using the same argumentation as in standard direct MRAC it can be concluded that the equilibrium $(\mathbf{e}_c, \widetilde{\mathbf{\Theta}}_r, \widetilde{\mathbf{\Theta}}_n) = (\mathbf{0}, \mathbf{0}, \mathbf{0})$ is uniformly stable and $\mathbf{e}_c(t) \to \mathbf{0}$ for $t \to \infty$. Because the parameter transformation (3.20) is invertible, this also means that the equilibrium $(\mathbf{e}_c, \widetilde{\mathbf{\Theta}}) = (\mathbf{0}, \mathbf{0})$ is uniformly stable. ■

Remark 3.9 *If the system* (3.1) *is not overactuated, i.e.* $r = m$*, the matrix* $\mathbf{V}_r = \mathbf{V}$ *and the standard direct MRAC argumentation is recovered. In this case Assumption 3.5 (Control Authority - direct MRAC PRO) is equivalent to Assumption 3.2 (Control Authority - direct MRAC).*

Remark 3.10 *The most effective version of the algorithm in terms of needed memory and operations can be implemented by selecting the initial condition* $\hat{\mathbf{\Theta}}_n(0) = \mathbf{0}$*. The control law* (3.24) *can then be implemented as*

$$\mathbf{u}(t) = \mathbf{V}_r\hat{\mathbf{\Theta}}_r(t) \cdot \boldsymbol{\omega}(t, \mathbf{x}_p).$$

Note that in Theorem 3.8 (direct MRAC PRO) the control and update laws of direct MRAC were not changed, only a parameter transformation was used for the stability proof. In this way it was possible to show that actuator failures can be taken into account in an overactuated system as long as Assumption 3.5 (Control Authority - direct MRAC PRO) holds. Furthermore, the controller in Section 3.2 can be implemented using the reduced set of parameters presented in this section. This implementation corresponds to the reference model (3.2), the control law (3.8), the parameter transformation (3.22) and the update law (3.23a) and is given in Figure 3.3. In addition, a comparison between direct MRAC and direct MRAC PRO can be seen in Table 3.2. Subsequently, the presented results are illustrated using Example 3.1.

Example 3.4 (Application of direct MRAC PRO) *In order to illustrate the presented results, let us first analyze Assumption 3.5 (Control Authority - direct MRAC PRO) which considers faults and replaces the more restrictive Assumption 3.2 (Control Authority - direct MRAC). Using the SVD of the example's* $\mathbf{B}_p$ (3.19)*, it states that the reduced control effectiveness matrix*

$$\Lambda_{rr} = \mathbf{V}_r^T\mathbf{\Lambda}\mathbf{V}_r = \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{bmatrix} = \frac{1}{2}(\lambda_1 + \lambda_2)$$

must be positive definite. This includes a single actuator failure as already proved in the last section. Interestingly, it also includes the direction inversion of an actuator as long as the other one has a larger effectiveness. As expected, a simultaneous failure of both actuators is excluded.

Using the SVD of the example's $\mathbf{B}_p$ (3.19)*, the implementation with the reduced parameters* $\hat{\boldsymbol{\Theta}}_r(t)$ *uses the following update law and parameter transformation*

$$\dot{\hat{\boldsymbol{\Theta}}}_r(t) = -L_\xi \sqrt{2} \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{P}\mathbf{e}_c(t)\, \boldsymbol{\omega}^T(t)\, \boldsymbol{\Gamma},$$
$$\hat{\boldsymbol{\Theta}}(t) = \begin{bmatrix} \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{bmatrix} \hat{\boldsymbol{\Theta}}_r(t) + \begin{bmatrix} -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{bmatrix} \hat{\boldsymbol{\Theta}}_n(0).$$

Only the parameters $\hat{\boldsymbol{\Theta}}_r \in \mathbb{R}^{1\times 3}$ *are updated instead of* $\hat{\boldsymbol{\Theta}} \in \mathbb{R}^{2\times 3}$ *reducing the number of parameters by a half.*

The methodology presented in this section allows the inclusion of failures within the actuation system and is not limited to redundancy of actuators with the same effect on the plant. This is demonstrated using a hexacopter system in Chapter 5. So far the parameter reduction approach has been presented within the direct MRAC framework. In the next sections, it is shown that this approach is also applicable to indirect adaptive control architectures.

Direct MRAC		
Plant	$\dot{\mathbf{x}}_p(t) = \mathbf{A}_p\mathbf{x}_p(t) + \mathbf{B}_p\boldsymbol{\Lambda}\mathbf{u}(t) + \mathbf{K}_\phi\boldsymbol{\phi}(\mathbf{x}_p(t)) + \mathbf{d}$	(3.1)
Reference Model	$\dot{\mathbf{x}}_m(t) = \mathbf{A}_m\mathbf{x}_m(t) + \mathbf{B}_m\mathbf{r}(t)$	(3.2)
Control Law	$\mathbf{u}(t) = \hat{\boldsymbol{\Theta}}(t)\,\boldsymbol{\omega}(t, \mathbf{x}_p)$	(3.8)
Matching Cond.	$[\mathbf{A}_m - \mathbf{A}_p, \quad \mathbf{B}_m, \quad -\mathbf{K}_\phi, \quad -\mathbf{d}] = \mathbf{B}_p\boldsymbol{\Lambda}\boldsymbol{\Theta}$	(3.11)
Error Dynamics	$\dot{\mathbf{e}}_c(t) = \mathbf{A}_m\mathbf{e}_c(t) + \mathbf{B}_p\boldsymbol{\Lambda}\widetilde{\boldsymbol{\Theta}}(t)\,\boldsymbol{\omega}(t)$	(3.12)
Update Law	$\dot{\hat{\boldsymbol{\Theta}}}(t) = -\mathbf{B}_p^T\mathbf{P}\mathbf{e}_c(t)\,\boldsymbol{\omega}^T(t)\,\boldsymbol{\Gamma}$	(3.13)
Control Authority	$\boldsymbol{\Lambda}$ positive definite	Asm. 3.2
Direct MRAC PRO		
Plant	$\dot{\mathbf{x}}_p(t) = \mathbf{A}_p\mathbf{x}_p(t) + \mathbf{B}_p\boldsymbol{\Lambda}\mathbf{u}(t) + \mathbf{K}_\phi\boldsymbol{\phi}(\mathbf{x}_p(t)) + \mathbf{d}$	(3.1)
Reference Model	$\dot{\mathbf{x}}_m(t) = \mathbf{A}_m\mathbf{x}_m(t) + \mathbf{B}_m\mathbf{r}(t)$	(3.2)
Control Law	$\mathbf{u}(t) = \left(\mathbf{V}_r\hat{\boldsymbol{\Theta}}_r(t) + \mathbf{V}_n\hat{\boldsymbol{\Theta}}_n(0)\right)\boldsymbol{\omega}(t, \mathbf{x}_p)$	(3.24)
Matching Cond.	$[\mathbf{A}_m - \mathbf{A}_p, \quad \mathbf{B}_m, \quad -\mathbf{K}_\phi, \quad -\mathbf{d}] = \mathbf{B}_p\boldsymbol{\Lambda}\boldsymbol{\Theta}$	(3.11)
Error Dynamics	$\dot{\mathbf{e}}_c(t) = \mathbf{A}_m\mathbf{e}_c(t) + \mathbf{B}_p\boldsymbol{\Lambda}\mathbf{V}_r\widetilde{\boldsymbol{\Theta}}_r(t)\,\boldsymbol{\omega}(t)$	(3.26)
Update Laws	$\dot{\hat{\boldsymbol{\Theta}}}_r(t) = -\mathbf{D}\mathbf{U}_r^T\mathbf{P}\mathbf{e}_c(t)\,\boldsymbol{\omega}^T(t)\,\boldsymbol{\Gamma}$	(3.23a)
	$\dot{\hat{\boldsymbol{\Theta}}}_n(t) = \mathbf{0}$	(3.23b)
Control Authority	$\boldsymbol{\Lambda}_{rr} = \mathbf{V}_r^T\boldsymbol{\Lambda}\mathbf{V}_r$ positive definite	Asm. 3.5

□ : Reduced parameters version.

□ : Inclusion of full actuator failure.

Table 3.2: Comparison between direct MRAC and direct MRAC PRO

3.4 Predictor-Based MRAC

In this section, indirect adaptive control is addressed. The selected approach is predictor based MRAC (PMRAC) because of its relevance in model reference adaptive control. It is for example the starting point for $\mathcal{L}1$ adaptive control [75]. In the following, the standard predictor-based MRAC is shortly reviewed and in the next section, it is extended to systems with redundant inputs. In the literature, the term predictor-based MRAC is used for a variety of approaches. Here, it refers to the indirect MRAC approach that uses a state predictor to estimate the plant state $\mathbf{x}_p(t)$ as well as the unknown parameters. The control law is then defined such that the predictor replicates the dynamics of the reference model (3.2) and indirectly it achieves that the plant behaves like reference model (3.2). Its structure is depicted in Figure 3.4.

For identification purposes, the plant (3.1) is rewritten using the following parameterization

$$\dot{\mathbf{x}}_p(t) = \mathbf{A}_m \mathbf{x}_p(t) + \mathbf{B}_p \mathbf{\Lambda} \mathbf{u}(t) + \mathbf{B}_p \mathbf{\Theta} \boldsymbol{\omega}(t), \tag{3.27}$$

where $\boldsymbol{\omega}(t) \in \mathbb{R}^q$ is the known regressor vector and $\mathbf{\Theta} \in \mathbb{R}^{m \times q}$ is the unknown parameter matrix. The specific parameterization intends to facilitate the derivation of a control law in the case of known parameters and considers only matched uncertainties since the focus lies on failure analysis. Matched uncertainties lie in the range of $\mathbf{B}_p$ and can therefore be directly compensated by the control input. The known regressor vector $\boldsymbol{\omega}(t)$ and the unknown ideal parameter matrix $\mathbf{\Theta}$ are defined as

$$\begin{aligned} \boldsymbol{\omega}(t) &:= [\mathbf{x}_p^T(t), \boldsymbol{\phi}^T(\mathbf{x}_p(t)), 1]^T \in \mathbb{R}^q, \\ \mathbf{\Theta} &:= [\mathbf{\Theta}_{p,x}, \mathbf{\Theta}_{p,\phi}, \mathbf{\Theta}_{p,d}] \in \mathbb{R}^{m \times q}, \end{aligned}$$

with $q = n + k + 1$, $\mathbf{\Theta}_{p,x} \in \mathbb{R}^{m \times n}$, $\mathbf{\Theta}_{p,\phi} \in \mathbb{R}^{m \times k}$ and $\mathbf{\Theta}_{p,d} \in \mathbb{R}^m$. In order to emphasize the analogy of the parameter transformation and reduction within the different control approaches, the same notation as in direct MRAC is used although the parameters $\mathbf{\Theta}$ and the regressor vector $\boldsymbol{\omega}(t)$ are differently defined. The comparison between the parameterized plant (3.27) and its original representation (3.1) leads to the subsequent matching condition.

Assumption 3.11 (Matching Condition - PMRAC) *The ideal parameters $\mathbf{\Theta}$ satisfy*

$$\mathbf{Y} := \begin{bmatrix} \mathbf{A}_p - \mathbf{A}_m, & \mathbf{K}_\phi, & \mathbf{d} \end{bmatrix} = \mathbf{B}_p \mathbf{\Theta}. \tag{3.28}$$

The state predictor of the parameterized plant (3.27) is defined as

$$\dot{\hat{\mathbf{x}}}_p(t) = \mathbf{A}_m \hat{\mathbf{x}}_p(t) + \mathbf{B}_p \hat{\mathbf{\Lambda}}(t) \mathbf{u}(t) + \mathbf{B}_p \hat{\mathbf{\Theta}}(t) \boldsymbol{\omega}(t), \tag{3.29}$$

where $\hat{\mathbf{x}}_p(t) \in \mathbb{R}^n$, $\hat{\mathbf{\Theta}}(t) \in \mathbb{R}^{m \times q}$ and $\hat{\mathbf{\Lambda}}(t) \in \mathbb{R}^{m \times m}$ are the estimates of $\mathbf{x}_p(t)$, $\mathbf{\Theta}$ and $\mathbf{\Lambda}$ respectively. Given the plant (3.27) and its state predictor (3.29), model reference tracking is achieved in two steps, which are typical for indirect approaches. The first one consists in deriving a control law $\mathbf{u}(t)$ for the completely known predictor model (3.29) such that it behaves like the reference model (3.2). The second step is the identification of the unknown plant parameters $\mathbf{\Lambda}$ and $\mathbf{\Theta}$ so that the control law $\mathbf{u}(t)$ is also valid for the plant dynamics (3.1).

The standard control law is given by

$$\mathbf{u}(t) = \hat{\mathbf{\Lambda}}^{-1}(t) \left(\mathbf{K}_r \mathbf{r}(t) - \hat{\mathbf{\Theta}}(t) \boldsymbol{\omega}(t) \right), \tag{3.30}$$

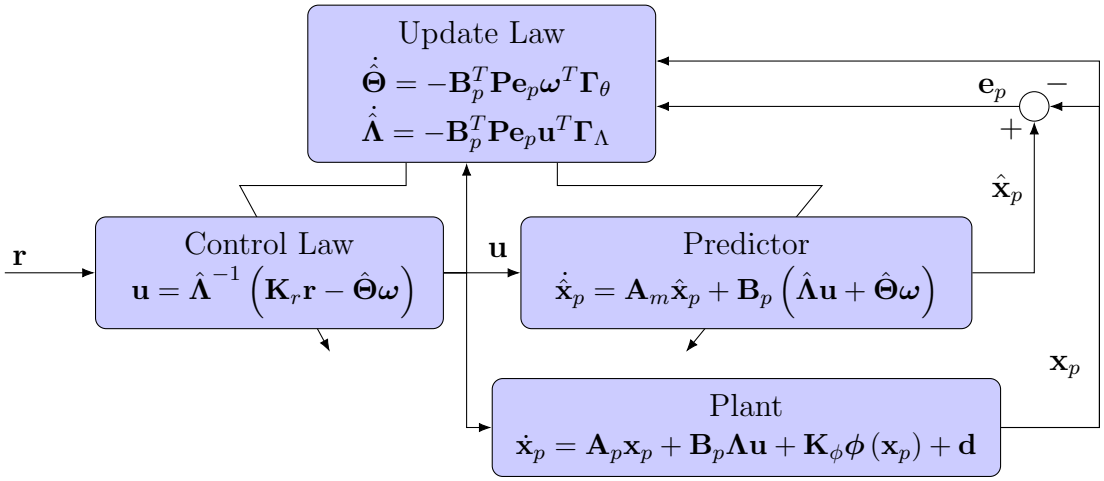


Figure 3.4: Controller Structure - Predictor MRAC

where $\mathbf{K}_r \in \mathbb{R}^{m \times r}$ is chosen such that $\mathbf{B}_m = \mathbf{B}_p \mathbf{K}_r$. Inserting the control law (3.30) into the predictor dynamics (3.29) leads to closed-loop predictor dynamics which are equal to the reference model dynamics

$$\dot{\hat{\mathbf{x}}}_p (t) = \mathbf{A}_m \hat{\mathbf{x}}_p (t) + \mathbf{B}_p \mathbf{K}_r \mathbf{r} (t) = \mathbf{A}_m \hat{\mathbf{x}}_p (t) + \mathbf{B}_m \mathbf{r} (t) . \tag{3.31}$$

Note that the control law (3.30) can only be implemented if the following assumption holds.

Assumption 3.12 (Control Authority - PMRAC) *The matrix $\hat{\boldsymbol{\Lambda}} (t)$ is invertible and as a consequence, the matrix $\boldsymbol{\Lambda}$ must be invertible too.*

Hence, the effectiveness of the different actuators is allowed to change as long as $\boldsymbol{\Lambda}$ is nonsingular, i.e. as long as there is no total failure as depicted in Table 3.1. The next step is to choose the update laws of $\hat{\boldsymbol{\Lambda}} (t)$ and $\hat{\boldsymbol{\Theta}} (t)$ such that the dynamics of the predictor estimation error $\mathbf{e}_p (t) = \hat{\mathbf{x}}_p (t) - \mathbf{x}_p (t)$ tend to zero and the parameter errors $\widetilde{\boldsymbol{\Lambda}} (t) = \hat{\boldsymbol{\Lambda}} (t) - \boldsymbol{\Lambda}$ and $\widetilde{\boldsymbol{\Theta}} (t) = \hat{\boldsymbol{\Theta}} (t) - \boldsymbol{\Theta}$ have a stable zero equilibrium. The predictor error dynamics are computed from (3.27) and (3.29) as follows

$$\dot{\mathbf{e}}_p (t) = \mathbf{A}_m \mathbf{e}_p (t) + \mathbf{B}_p \widetilde{\boldsymbol{\Lambda}} (t) \mathbf{u} (t) + \mathbf{B}_p \widetilde{\boldsymbol{\Theta}} (t) \boldsymbol{\omega} (t) . \tag{3.32}$$

The Lyapunov candidate function is given by

$$V \left(\mathbf{e}_p, \widetilde{\boldsymbol{\Theta}}, \widetilde{\boldsymbol{\Lambda}} \right) = \frac{1}{2} \mathbf{e}_p^T (t) \mathbf{P} \mathbf{e}_p (t) + \frac{1}{2} tr \left(\widetilde{\boldsymbol{\Theta}} (t) \boldsymbol{\Gamma}_\theta^{-1} \widetilde{\boldsymbol{\Theta}}^T (t) \right) + \frac{1}{2} tr \left(\widetilde{\boldsymbol{\Lambda}} (t) \boldsymbol{\Gamma}_\Lambda^{-1} \widetilde{\boldsymbol{\Lambda}}^T (t) \right) , \tag{3.33}$$

with the symmetric positive definite matrix $\mathbf{P}$ solving the Lyapunov equation (3.3). From Lemma B.8 (Positive Definite Trace), it follows that the Lyapunov function is positive definite in the case of symmetric positive definite matrices $\boldsymbol{\Gamma}_\theta \in \mathbb{R}^{q \times q}$ and $\boldsymbol{\Gamma}_\Lambda \in \mathbb{R}^{m \times m}$, which correspond to the adaptation rates. In the following, the time dependency is not written out for readability. The derivative of the Lyapunov function is calculated using the predictor error dynamics (3.32), the Lyapunov equation (3.3)

and Lemma B.2 (Trace and Scalar Product)

$$\begin{aligned}\dot{V} &= -\frac{1}{2}\mathbf{e}_p^T\mathbf{Q}\mathbf{e}_p + \mathbf{e}_p^T\mathbf{P}\mathbf{B}_p\left(\widetilde{\boldsymbol{\Lambda}}\mathbf{u} + \widetilde{\boldsymbol{\Theta}}\boldsymbol{\omega}\right) + tr\left(\widetilde{\boldsymbol{\Theta}}\boldsymbol{\Gamma}_\theta^{-1}\dot{\widetilde{\boldsymbol{\Theta}}}^T\right) + tr\left(\widetilde{\boldsymbol{\Lambda}}\boldsymbol{\Gamma}_\Lambda^{-1}\dot{\widetilde{\boldsymbol{\Lambda}}}^T\right),\\ &= -\frac{1}{2}\mathbf{e}_p^T\mathbf{Q}\mathbf{e}_p + tr\left(\widetilde{\boldsymbol{\Theta}}\left(\boldsymbol{\omega}\mathbf{e}_p^T\mathbf{P}\mathbf{B}_p + \boldsymbol{\Gamma}_\theta^{-1}\dot{\widetilde{\boldsymbol{\Theta}}}^T\right)\right) + tr\left(\widetilde{\boldsymbol{\Lambda}}\left(\mathbf{u}\mathbf{e}_p^T\mathbf{P}\mathbf{B}_p + \boldsymbol{\Gamma}_\Lambda^{-1}\dot{\widetilde{\boldsymbol{\Lambda}}}^T\right)\right).\end{aligned}$$

By choosing the standard predictor-based MRAC update laws

$$\begin{aligned}\dot{\hat{\boldsymbol{\Theta}}}(t) &= -\left(\boldsymbol{\Gamma}_\theta\boldsymbol{\omega}(t)\,\mathbf{e}_p^T(t)\,\mathbf{P}\mathbf{B}_p\right)^T = -\mathbf{B}_p^T\mathbf{P}\mathbf{e}_p(t)\,\boldsymbol{\omega}^T(t)\,\boldsymbol{\Gamma}_\theta,\\ \dot{\hat{\boldsymbol{\Lambda}}}(t) &= -\left(\boldsymbol{\Gamma}_\Lambda\mathbf{u}(t)\,\mathbf{e}_p^T(t)\,\mathbf{P}\mathbf{B}_p\right)^T = -\mathbf{B}_p^T\mathbf{P}\mathbf{e}_p(t)\,\mathbf{u}^T(t)\,\boldsymbol{\Gamma}_\Lambda,\end{aligned} \tag{3.34}$$

the derivative of the Lyapunov function $\dot{V}(t) = -\frac{1}{2}\mathbf{e}_p^T(t)\,\mathbf{Q}\mathbf{e}_p(t)$ is negative semi-definite. From Theorem 4.8 (Uniform Stability) in [87], the equilibrium $(\mathbf{e}_p, \widetilde{\boldsymbol{\Theta}}, \widetilde{\boldsymbol{\Lambda}}) = (\mathbf{0}, \mathbf{0}, \mathbf{0})$ is uniformly stable and $\mathbf{e}_p(t)$, $\widetilde{\boldsymbol{\Theta}}(t)$, $\widetilde{\boldsymbol{\Lambda}}(t) \in \mathcal{L}_\infty$. Therefore, $\hat{\boldsymbol{\Theta}}(t)$ and $\hat{\boldsymbol{\Lambda}}(t)$ are bounded too.

Subsequently, the goal is to show asymptotic convergence of the estimation error $\mathbf{e}_p(t)$ to zero based on Barbalat's Lemma (Lemma B.9). Following the same arguments as in Section 3.2, it is known that $\lim_{t\to\infty}\int_{\tau=0}^{t}\dot{V}(\mathbf{e}_p(\tau))\,d\tau$ exists and is finite. Therefore, it remains to show that $\dot{V}(\mathbf{e}_p(t))$ is a uniform continuous function of t. Given that the input $\mathbf{r}(t)$ is bounded and from the predictor dynamics (3.31) it holds that $\hat{\mathbf{x}}_p(t) \in \mathcal{L}_\infty$ and hence $\mathbf{x}_p(t) \in \mathcal{L}_\infty$. This implies boundedness of $\boldsymbol{\phi}(\mathbf{x}_p(t))$ and $\boldsymbol{\omega}(\mathbf{x}_p(t), \boldsymbol{\phi}(\mathbf{x}_p(t)))$. From Assumption 3.12 (Control Authority - PMRAC), it follows that the control law $\mathbf{u}(t)$ (3.30) is bounded and hence $\dot{\mathbf{e}}_p(t)$ (3.32) is bounded uniformly in t for all $t \geq 0$. Therefore, $\ddot{V}(t) = -\mathbf{e}_p^T(t)\,\mathbf{Q}\dot{\mathbf{e}}_p(t)$ is also bounded uniformly in t for all $t \geq 0$. Then, $\dot{V}(\mathbf{e}_p(t))$ is uniformly continuous for $t \geq 0$. Using Lemma B.9 (Barbalat) it follows that $\dot{V}(\mathbf{e}_p(t)) \to 0$ and hence $\mathbf{e}_p(t) \to \mathbf{0}$ for $t \to \infty$. Finally, because of the closed-loop predictor dynamics (3.31), the tracking error $\mathbf{e}_c(t) = \mathbf{x}_p(t) - \mathbf{x}_m(t) = \mathbf{x}_p(t) - \hat{\mathbf{x}}_p(t) = -\mathbf{e}_p(t) \to \mathbf{0}$ for $t \to \infty$. The presented result is summarized in the next theorem.

Theorem 3.13 (PMRAC) *Consider the plant dynamics* (3.1)*, the reference model* (3.2)*, the state predictor* (3.29)*, the control law* (3.30) *and the update laws* (3.34)*. Furthermore let Assumptions 3.11 (Matching Condition - PMRAC) and 3.12 (Control Authority - PMRAC) hold. Then, the equilibrium* $(\mathbf{e}_p, \widetilde{\boldsymbol{\Theta}}, \widetilde{\boldsymbol{\Lambda}}) = (\mathbf{0}, \mathbf{0}, \mathbf{0})$ *is globally uniformly stable and* $\mathbf{e}_p(t), \mathbf{e}_c(t) \to \mathbf{0}$ *for* $t \to \infty$*.* ■

Example 3.5 (Application of PMRAC) *In order to better understand the physical meaning of Assumption 3.12 (Control Authority - PMRAC) and see that in the case of overactuated systems it is not a necessary condition, predictor-based MRAC is applied to Example 3.1 (Simple Overactuated Linear System). First, the plant dynamics* (3.6) *are reparameterized to the form* (3.27)*. In this example, there is no constant disturbance and no nonlinearity in the plant dynamics* (3.6)*. Hence, the regressor vector is* $\boldsymbol{\omega}(t) = \mathbf{x}_p(t) \in \mathbb{R}^2$ *and the unknown parameters are* $\boldsymbol{\Theta} \in \mathbb{R}^{2\times 2}$ *such that*

$$\begin{pmatrix}\dot{p}(t)\\ \dot{\Phi}(t)\end{pmatrix} = \mathbf{A}_m\begin{pmatrix}p(t)\\ \Phi(t)\end{pmatrix} + \mathbf{B}_p\boldsymbol{\Lambda}\begin{pmatrix}\xi_1(t)\\ \xi_2(t)\end{pmatrix} + \mathbf{B}_p\boldsymbol{\Theta}\begin{pmatrix}p(t)\\ \Phi(t)\end{pmatrix}. \tag{3.35}$$

The matching condition (3.28) *translates to*

$$\underbrace{\begin{bmatrix} L_p + kd & kp \\ 0 & 0 \end{bmatrix}}_{\mathbf{A}_p - \mathbf{A}_m} = \underbrace{\begin{bmatrix} L_\xi & L_\xi \\ 0 & 0 \end{bmatrix}}_{\mathbf{B}_p} \mathbf{\Theta}$$

and has the following set of solutions [59, Sec. 3.5.1.4]

$$\mathbf{\Theta} = \underbrace{\begin{bmatrix} \frac{1}{2L_\xi} & 0 \\ \frac{1}{2L_\xi} & 0 \end{bmatrix}}_{\mathbf{B}_p^+} \underbrace{\begin{bmatrix} L_p + kd & kp \\ 0 & 0 \end{bmatrix}}_{\mathbf{A}_p - \mathbf{A}_m} + \underbrace{\begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{bmatrix}}_{\mathbf{I} - \mathbf{B}_p^+ \mathbf{B}_p} \mathbf{W}. \tag{3.36}$$

The matrix $\mathbf{W} \in \mathbb{R}^{2\times 2}$ *is an arbitrary matrix representing the offset of the solution within the null space of* $\mathbf{B}_p$*. Note that due to overactuation, i.e.* $r < m$*, the matrix* $\mathbf{B}_p$ *has a non-empty null space leading to the non-uniqueness of the ideal parameter* $\mathbf{\Theta}$*. The null space is spanned by the columns of* $\mathbf{I} - \mathbf{B}_p^+ \mathbf{B}_p$*.*

The state predictor is then given as

$$\begin{pmatrix} \dot{\hat{p}}(t) \\ \dot{\hat{\Phi}}(t) \end{pmatrix} = \mathbf{A}_m \begin{pmatrix} \hat{p}(t) \\ \hat{\Phi}(t) \end{pmatrix} + \mathbf{B}_p \hat{\mathbf{\Lambda}} \begin{pmatrix} \xi_1(t) \\ \xi_2(t) \end{pmatrix} + \mathbf{B}_p \hat{\mathbf{\Theta}} \begin{pmatrix} p(t) \\ \Phi(t) \end{pmatrix},$$

with $\hat{\mathbf{\Lambda}}, \hat{\mathbf{\Theta}} \in \mathbb{R}^{2\times 2}$ *and leads to the control law*

$$\mathbf{u}(t) = \hat{\mathbf{\Lambda}}^{-1}(t) \left(\mathbf{K}_r r(t) - \hat{\mathbf{\Theta}}(t) \boldsymbol{\omega}(t) \right). \tag{3.37}$$

Here, the feed-forward gain $\mathbf{K}_r \in \mathbb{R}^{2\times 1}$ *satisfies*

$$\underbrace{\begin{bmatrix} kp \\ 0 \end{bmatrix}}_{\mathbf{B}_m} = \underbrace{\begin{bmatrix} L_\xi & L_\xi \\ 0 & 0 \end{bmatrix}}_{\mathbf{B}_p} \underbrace{\begin{bmatrix} 1 \\ 1 \end{bmatrix} \frac{kp}{2L_\xi}}_{\mathbf{K}_r}.$$

Analogously to the ideal parameter $\mathbf{\Theta}$ (3.36)*, the choice of* $\mathbf{K}_r$ *is not unique. The corresponding update laws* (3.34) *are given by*

$$\dot{\hat{\mathbf{\Theta}}}(t) = -\begin{bmatrix} L_\xi & 0 \\ L_\xi & 0 \end{bmatrix} \mathbf{P}\mathbf{e}_p(t) \mathbf{x}_p^T(t) \mathbf{\Gamma}_\theta,$$
$$\dot{\hat{\mathbf{\Lambda}}}(t) = -\begin{bmatrix} L_\xi & 0 \\ L_\xi & 0 \end{bmatrix} \mathbf{P}\mathbf{e}_p(t) \mathbf{u}^T(t) \mathbf{\Gamma}_\Lambda.$$

Similarly to the direct MRAC case, the rows of $\dot{\hat{\mathbf{\Theta}}}(t)$ *and* $\dot{\hat{\mathbf{\Lambda}}}(t)$ *are exactly the same and this points to an overparameterization due to overactuation.*

For an invertible $\hat{\mathbf{\Lambda}}(t)$*, Theorem 3.13 (PMRAC) can be applied and the model following goal is achieved for Example 3.1. The question now is what happens if one of the two actuators fails because in that case* $\mathbf{\Lambda}$ *becomes singular. In order to answer this question, let the second actuator have a failure such that the plant* (3.35) *can be rewritten as*

$$\begin{pmatrix} \dot{p}(t) \\ \dot{\Phi}(t) \end{pmatrix} = \mathbf{A}_m \begin{pmatrix} p(t) \\ \Phi(t) \end{pmatrix} + \begin{bmatrix} L_\xi \\ 0 \end{bmatrix} \lambda_1 \xi_1(t) + \begin{bmatrix} L_\xi \\ 0 \end{bmatrix} \mathbf{\Theta}_1 \begin{pmatrix} p(t) \\ \Phi(t) \end{pmatrix}.$$

with the unknown parameters $\mathbf{\Theta}_1 \in \mathbb{R}^{1\times 2}$*. The matching condition* (3.28) *translates to*

$$\underbrace{\begin{bmatrix} L_p + kd & kp \\ 0 & 0 \end{bmatrix}}_{\mathbf{A}_p - \mathbf{A}_m} = \underbrace{\begin{bmatrix} L_\xi \\ 0 \end{bmatrix}}_{\mathbf{B}_{p1}} \mathbf{\Theta}_1,$$

which has the following unique solution

$$\mathbf{\Theta}_1 = \underbrace{\begin{bmatrix} \frac{1}{L_\xi} & 0 \end{bmatrix}}_{\mathbf{B}_{p1}^+} \underbrace{\begin{bmatrix} L_p + kd & kp \\ 0 & 0 \end{bmatrix}}_{\mathbf{A}_p - \mathbf{A}_m} + \underbrace{0}_{\mathbf{I} - \mathbf{B}_{p1}^+ \mathbf{B}_{p1}} \cdot \mathbf{W}.$$

The state predictor is then given as

$$\begin{pmatrix} \dot{\hat{p}}(t) \\ \dot{\hat{\Phi}}(t) \end{pmatrix} = \mathbf{A}_m \begin{pmatrix} \hat{p}(t) \\ \hat{\Phi}(t) \end{pmatrix} + \begin{bmatrix} L_\xi \\ 0 \end{bmatrix} \hat{\lambda}_1 \xi_1(t) + \begin{bmatrix} L_\xi \\ 0 \end{bmatrix} \hat{\mathbf{\Theta}}_1 \begin{pmatrix} p(t) \\ \Phi(t) \end{pmatrix},$$

with $\hat{\lambda}_1(t) \in \mathbb{R}$*,* $\hat{\mathbf{\Theta}}_1(t) \in \mathbb{R}^{1\times 2}$ *and leads to the control law*

$$u_1(t) = \xi_1(t) = \hat{\lambda}_1^{-1}(t) \left(k_{r1} \cdot r(t) - \hat{\mathbf{\Theta}}_1(t) \boldsymbol{\omega}(t) \right). \tag{3.38}$$

The feed-forward gain is $k_{r1} = \frac{kp}{L_\xi}$ *and is unique for this case. The corresponding update laws* (3.34) *are given by*

$$\dot{\hat{\mathbf{\Theta}}}_1(t) = -\begin{bmatrix} L_\xi & 0 \end{bmatrix} \mathbf{P} \mathbf{e}_p(t) \mathbf{x}_p^T(t) \mathbf{\Gamma}_\theta,$$
$$\dot{\hat{\lambda}}_1(t) = -\begin{bmatrix} L_\xi & 0 \end{bmatrix} \mathbf{P} \mathbf{e}_p(t) \xi_1(t) \mathbf{\Gamma}_\Lambda.$$

For an invertible $\hat{\lambda}_1(t)$*, Theorem 3.13 can be applied and the model following goal is achieved. A similar result holds for the case that* λ_2 *is invertible and* $\lambda_1 = 0$ *and leads to the control law*

$$u_2(t) = \xi_2(t) = \hat{\lambda}_2^{-1}(t) \left(k_{r2} \cdot r(t) - \hat{\mathbf{\Theta}}_2(t) \boldsymbol{\omega}(t) \right) \tag{3.39}$$

with the feed-forward gain $k_{r2} = \frac{kp}{L_\xi}$*. Hence, in this example the control objective can be achieved using predictor-based MRAC in cases where only one actuator fails.*

Differently from the direct MRAC result, in order to fulfill the control objective, different control laws resulted for each fault case: nominal conditions (3.37)*, failure of actuator 2* (3.38) *and failure of actuator 1* (3.39)*. Due to the computation of* $\hat{\mathbf{\Lambda}}^{-1}(t)$*,* (3.37) *cannot be implemented in any total failure case. Due to* $\hat{\lambda}_1^{-1}(t)$*,* (3.38) *cannot be implemented for the failure of actuator 1. And due to* $\hat{\lambda}_2^{-1}(t)$*,* (3.39) *cannot be implemented for the failure of actuator 2.*

In the next section, the parameter reduction due to overactuation (PRO) approach is applied in order to formulate a control law that is able to handle the nominal as well as the failure cases. The result is based on changing Assumption 3.12 (Control Authority - PMRAC).

3.5 Predictor-Based MRAC PRO

In this section, the PRO approach is used to reformulate PMRAC such that actuator failures can be included. The main result of this section was previously presented in [46] and here a refinement of the control law is shown. First, note that differently from direct MRAC, the Assumption 3.12 (Control Authority - PMRAC) is not needed to achieve stability of the predictor error dynamics with respect to $(\mathbf{e}_p, \widetilde{\mathbf{\Theta}}, \widetilde{\mathbf{\Lambda}}) = (\mathbf{0}, \mathbf{0}, \mathbf{0})$. Assumption 3.12 (Control Authority - PMRAC) is however necessary for the implementation of the control law (3.30). Hence, in this case the control law needs to be changed to take actuator failures into account. One solution is to compute the pseudo-inverse of $\mathbf{B}_p\hat{\mathbf{\Lambda}}(t)$ instead of $\hat{\mathbf{\Lambda}}^{-1}(t)$. Using the parameter transformation presented in Section 3.6, a novel control law allows to handle failure cases without the computation of a pseudo-inverse.

First, the control input $\mathbf{u}$ is separated analogously to the parameter transformation (3.21) using the right-singular vectors $\mathbf{V} = [\mathbf{V}_r, \mathbf{V}_n]$ of $\mathbf{B}_p$ (3.17)

$$\mathbf{u}(t) = \mathbf{V}_r\mathbf{u}_r(t) + \mathbf{V}_n\mathbf{u}_n(t),$$

with $\mathbf{u}_r \in \mathbb{R}^r, \mathbf{u}_n \in \mathbb{R}^{m-r}$. Because $\mathbf{V}$ is orthogonal, this is only a transformation and no degree of freedom has been lost. Subsequently, using the compact SVD of $\mathbf{B}_p = \mathbf{U}_r\mathbf{D}\mathbf{V}_r^T$ (3.18) and the parameter definitions

$$\hat{\mathbf{\Theta}}_r = \mathbf{V}_r^T\hat{\mathbf{\Theta}} \in \mathbb{R}^{r\times q}, \quad \hat{\mathbf{\Lambda}}_{rr} = \mathbf{V}_r^T\hat{\mathbf{\Lambda}}\mathbf{V}_r \in \mathbb{R}^{r\times r}, \quad \hat{\mathbf{\Lambda}}_{rn} = \mathbf{V}_r^T\hat{\mathbf{\Lambda}}\mathbf{V}_n \in \mathbb{R}^{r\times(m-r)}, \tag{3.40}$$

the predictor (3.29) is reformulated

$$\begin{aligned}
\dot{\hat{\mathbf{x}}}_p(t) &= \mathbf{A}_m\hat{\mathbf{x}}_p(t) + \mathbf{U}_r\mathbf{D}\mathbf{V}_r^T\left(\hat{\mathbf{\Lambda}}(t)\,\mathbf{u}(t) + \hat{\mathbf{\Theta}}(t)\,\boldsymbol{\omega}(t)\right),\\
&= \mathbf{A}_m\hat{\mathbf{x}}_p(t) + \mathbf{U}_r\mathbf{D}\mathbf{V}_r^T\left(\hat{\mathbf{\Lambda}}(t)\,\mathbf{V}_r\mathbf{u}_r(t) + \hat{\mathbf{\Lambda}}(t)\,\mathbf{V}_n\mathbf{u}_n(t) + \hat{\mathbf{\Theta}}(t)\,\boldsymbol{\omega}(t)\right), \qquad (3.41)\\
&= \mathbf{A}_m\hat{\mathbf{x}}_p(t) + \mathbf{U}_r\mathbf{D}\left(\hat{\mathbf{\Lambda}}_{rr}(t)\,\mathbf{u}_r(t) + \hat{\mathbf{\Lambda}}_{rn}(t)\,\mathbf{u}_n(t) + \hat{\mathbf{\Theta}}_r(t)\,\boldsymbol{\omega}(t)\right).
\end{aligned}$$

Note that here $\hat{\mathbf{\Theta}}_r(t)$ and $\hat{\mathbf{\Lambda}}_{rr}(t)$ are reduced parameters analogously transformed as in direct MRAC PRO. The motivation for this transformation will be clear from the subsequent derivation of the control law. The following main assumption is formulated instead of Assumption 3.12 (Control Authority - PMRAC).

Assumption 3.14 (Control Authority - PMRAC PRO) *The matrix $\hat{\mathbf{\Lambda}}_{rr} = \mathbf{V}_r^T\hat{\mathbf{\Lambda}}\mathbf{V}_r$ is invertible and as a consequence the matrix $\mathbf{\Lambda}_{rr} = \mathbf{V}_r^T\mathbf{\Lambda}\mathbf{V}_r$ must be invertible too.*

Using Assumption 3.14, a novel control law instead of (3.30) is proposed. By comparing (3.41) with the reference model (3.2), it can be seen that the following equation

$$\mathbf{B}_m\mathbf{r}(t) = \mathbf{U}_r\mathbf{D}\left(\hat{\mathbf{\Lambda}}_{rr}(t)\,\mathbf{u}_r(t) + \hat{\mathbf{\Lambda}}_{rn}(t)\,\mathbf{u}_n(t) + \hat{\mathbf{\Theta}}_r(t)\,\boldsymbol{\omega}(t)\right)$$

must hold for the predictor dynamics to be equal to the reference model dynamics. Using the fact that $\mathbf{B}_m = \mathbf{B}_p\mathbf{K}_r = \mathbf{U}_r\mathbf{D}\mathbf{V}_r^T\mathbf{K}_r$, the following control law

$$\begin{aligned}
\mathbf{u}_r(t) &= \hat{\mathbf{\Lambda}}_{rr}^{-1}(t)\left(\mathbf{V}_r^T\mathbf{K}_r\mathbf{r}(t) - \hat{\mathbf{\Lambda}}_{rn}(t)\,\mathbf{u}_n(t) - \hat{\mathbf{\Theta}}_r(t)\,\boldsymbol{\omega}(t)\right),\\
\mathbf{u}(t) &= \mathbf{V}_r\mathbf{u}_r(t) + \mathbf{V}_n\mathbf{u}_n(t),
\end{aligned} \tag{3.42}$$

is proposed. Note that given (3.42) the choice of $\mathbf{u}_n(t)$ does not affect the dynamics (3.41) and hence can be used to meet other requirements like e.g. the minimization of the needed input energy. This degree of freedom arises from the overactuation of the system. Nevertheless, the control law $\mathbf{u}_n(t)$ needs to be bounded and piecewise continuous in t. The parameters $\hat{\mathbf{\Lambda}}(t)$ and $\hat{\mathbf{\Theta}}(t)$ can still be updated using (3.34) and then transformed to $\hat{\mathbf{\Theta}}_r(t), \hat{\mathbf{\Lambda}}_{rr}(t), \hat{\mathbf{\Lambda}}_{rn}(t)$ using (3.40) in order to implement the control law (3.42). In this case, the goal of taking failures into account is already achieved. However, because of the structure of the update law (3.34), it is known from Section 3.3 that the parameters $\hat{\mathbf{\Theta}}(t)$ and $\hat{\mathbf{\Lambda}}(t)$ can be reduced in the case of overactuation. Using the same parameter transformation $\hat{\mathbf{\Theta}}_r(t) = \mathbf{V}_r^T \hat{\mathbf{\Theta}}(t)$ and $\hat{\mathbf{\Lambda}}_r(t) = \mathbf{V}_r^T \hat{\mathbf{\Lambda}}(t)$ leads to the following reduced update laws

$$\begin{aligned}\dot{\hat{\mathbf{\Theta}}}_r(t) &= -\left(\mathbf{\Gamma}_\theta \boldsymbol{\omega}(t)\, \mathbf{e}_p^T(t)\, \mathbf{P}\mathbf{U}_r\mathbf{D}\right)^T = -\mathbf{D}\mathbf{U}_r^T\mathbf{P}\mathbf{e}_p(t)\, \boldsymbol{\omega}^T(t)\, \mathbf{\Gamma}_\theta,\\ \dot{\hat{\mathbf{\Lambda}}}_r(t) &= -\left(\mathbf{\Gamma}_\Lambda \mathbf{u}(t)\, \mathbf{e}_p^T(t)\, \mathbf{P}\mathbf{U}_r\mathbf{D}\right)^T = -\mathbf{D}\mathbf{U}_r^T\mathbf{P}\mathbf{e}_p(t)\, \mathbf{u}^T(t)\, \mathbf{\Gamma}_\Lambda.\end{aligned} \tag{3.43}$$

As in the direct MRAC PRO case, the parameters in the null space of $\mathbf{B}_p$ are constant, i.e. $\hat{\mathbf{\Theta}}_n = \mathbf{V}_n^T\hat{\mathbf{\Theta}}(0)$ and $\hat{\mathbf{\Lambda}}_n = \mathbf{V}_n^T\hat{\mathbf{\Lambda}}(0)$. The final controller structure can be seen in Figure 3.5 and the next theorem summarizes the results.

Theorem 3.15 (PMRAC PRO) *Compared to Theorem 3.13 (PMRAC), the Assumption 3.12 (Control Authority - PMRAC) is relaxed to Assumption 3.14 (Control Authority - PMRAC PRO) and the control law* (3.30) *is changed to* (3.42)*. In this way actuator failures are taken into account for systems with redundant control inputs. Furthermore, the reduced set of parameters is used.*

Consider the plant dynamics (3.1)*, the reference model* (3.2)*, the control law* (3.42) *and the update laws* (3.43)*. Furthermore let Assumptions 3.11 (Matching Condition - PMRAC) and 3.14 (Control Authority - PMRAC PRO) hold. Then, the equilibrium* $(\mathbf{e}_p, \widetilde{\mathbf{\Theta}}_r, \widetilde{\mathbf{\Lambda}}_r) = (\mathbf{0}, \mathbf{0}, \mathbf{0})$ *is globally uniformly stable and* $\mathbf{e}_p(t), \mathbf{e}_c(t) \to \mathbf{0}$ *for* $t \to \infty$*.*

Proof: Inserting the control law (3.42) in the predictor dynamics (3.41) leads to

$$\dot{\hat{\mathbf{x}}}_p(t) = \mathbf{A}_m\hat{\mathbf{x}}_p(t) + \mathbf{U}_r\mathbf{D}\mathbf{V}_r^T\mathbf{K}_r\mathbf{r}(t) = \mathbf{A}_m\hat{\mathbf{x}}_p(t) + \mathbf{B}_m\mathbf{r}(t). \tag{3.44}$$

Hence, the predictor dynamics equal the reference model dynamics as long as Assumption 3.14 (Control Authority - PMRAC PRO) holds.

Using the matching condition (3.28), the plant (3.1) can be represented as in (3.27). Then, using the reduced parameters $\mathbf{\Theta}_r = \mathbf{V}_r^T\mathbf{\Theta}$ and $\mathbf{\Lambda}_r = \mathbf{V}_r^T\mathbf{\Lambda}$ and their estimates $\hat{\mathbf{\Theta}}_r(t) = \mathbf{V}_r^T\hat{\mathbf{\Theta}}(t)$ and $\hat{\mathbf{\Lambda}}_r(t) = \mathbf{V}_r^T\hat{\mathbf{\Lambda}}(t)$, the plant and predictor dynamics can be rewritten as

$$\begin{aligned}\dot{\mathbf{x}}_p(t) &= \mathbf{A}_m\mathbf{x}_p(t) + \mathbf{U}_r\mathbf{D}\left(\mathbf{\Lambda}_r\mathbf{u}(t) + \mathbf{\Theta}_r\boldsymbol{\omega}(t)\right),\\ \dot{\hat{\mathbf{x}}}_p(t) &= \mathbf{A}_m\hat{\mathbf{x}}_p(t) + \mathbf{U}_r\mathbf{D}\left(\hat{\mathbf{\Lambda}}_r(t)\,\mathbf{u}(t) + \hat{\mathbf{\Theta}}_r(t)\,\boldsymbol{\omega}(t)\right).\end{aligned}$$

Therefore, the state estimation error dynamics are

$$\dot{\mathbf{e}}_p(t) = \mathbf{A}_m\mathbf{e}_p(t) + \mathbf{U}_r\mathbf{D}\left(\widetilde{\mathbf{\Lambda}}_r(t)\,\mathbf{u}(t) + \widetilde{\mathbf{\Theta}}_r(t)\,\boldsymbol{\omega}(t)\right), \tag{3.45}$$

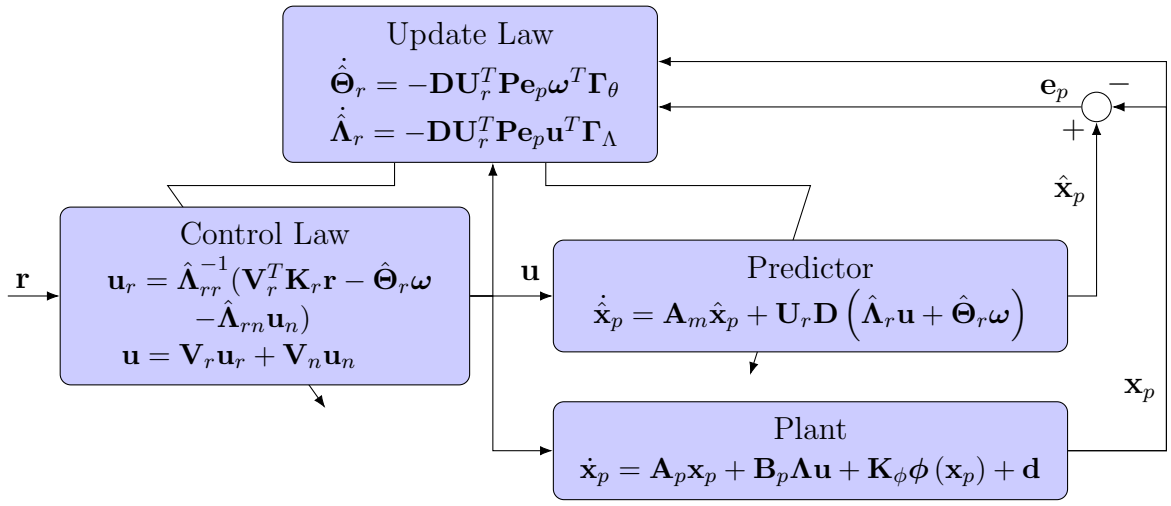


Figure 3.5: Controller Structure - Predictor MRAC PRO

with the reduced parameter errors $\widetilde{\boldsymbol{\Theta}}_r(t) = \hat{\boldsymbol{\Theta}}_r(t) - \boldsymbol{\Theta}_r$ and $\widetilde{\boldsymbol{\Lambda}}_r(t) = \hat{\boldsymbol{\Lambda}}_r(t) - \boldsymbol{\Lambda}_r$. The Lyapunov candidate function is formulated as

$$V\left(\mathbf{e}_p, \widetilde{\boldsymbol{\Theta}}_r, \widetilde{\boldsymbol{\Lambda}}_r\right) = \frac{1}{2}\mathbf{e}_p^T(t)\,\mathbf{P}\mathbf{e}_p(t) + \frac{1}{2}tr\left(\widetilde{\boldsymbol{\Theta}}_r(t)\,\boldsymbol{\Gamma}_\theta^{-1}\widetilde{\boldsymbol{\Theta}}_r^T(t)\right) + \frac{1}{2}tr\left(\widetilde{\boldsymbol{\Lambda}}_r(t)\,\boldsymbol{\Gamma}_\Lambda^{-1}\widetilde{\boldsymbol{\Lambda}}_r^T(t)\right),$$

with the symmetric positive definite matrix $\mathbf{P}$ solving the Lyapunov equation (3.3). The parameters in the null space of $\mathbf{B}_p$, $\hat{\boldsymbol{\Theta}}_n(t) = \mathbf{V}_n^T\hat{\boldsymbol{\Theta}}(t)$ and $\hat{\boldsymbol{\Lambda}}_n(t) = \mathbf{V}_n^T\hat{\boldsymbol{\Lambda}}(t)$, are not considered within the Lyapunov candidate function since they are constant. From Lemma B.8 (Positive Definite Trace) it follows that the Lyapunov function is positive definite in the case of symmetric positive definite matrices $\boldsymbol{\Gamma}_\theta \in \mathbb{R}^{q\times q}$ and $\boldsymbol{\Gamma}_\Lambda \in \mathbb{R}^{m\times m}$, which correspond to the adaptation rates. In the following the time dependency is not written out for readability. The derivative of the Lyapunov function is calculated using the predictor error dynamics (3.45), the Lyapunov equation (3.3) and Lemma B.2 (Trace and Scalar Product)

$$\begin{aligned}\dot{V} &= -\frac{1}{2}\mathbf{e}_p^T\mathbf{Q}\mathbf{e}_p + \mathbf{e}_p^T\mathbf{P}\mathbf{U}_r\mathbf{D}\left(\widetilde{\boldsymbol{\Lambda}}_r\mathbf{u} + \widetilde{\boldsymbol{\Theta}}_r\boldsymbol{\omega}\right) + tr\left(\widetilde{\boldsymbol{\Theta}}_r\boldsymbol{\Gamma}_\theta^{-1}\dot{\widetilde{\boldsymbol{\Theta}}}_r^T\right) + tr\left(\widetilde{\boldsymbol{\Lambda}}_r\boldsymbol{\Gamma}_\Lambda^{-1}\dot{\widetilde{\boldsymbol{\Lambda}}}_r^T\right),\\ &= -\frac{1}{2}\mathbf{e}_p^T\mathbf{Q}\mathbf{e}_p + tr\left(\widetilde{\boldsymbol{\Theta}}_r\left(\boldsymbol{\omega}\mathbf{e}_p^T\mathbf{P}\mathbf{U}_r\mathbf{D} + \boldsymbol{\Gamma}_\theta^{-1}\dot{\widetilde{\boldsymbol{\Theta}}}_r^T\right)\right) + tr\left(\widetilde{\boldsymbol{\Lambda}}_r\left(\mathbf{u}\mathbf{e}_p^T\mathbf{P}\mathbf{U}_r\mathbf{D} + \boldsymbol{\Gamma}_\Lambda^{-1}\dot{\widetilde{\boldsymbol{\Lambda}}}_r^T\right)\right).\end{aligned}$$

Finally, using the reduced update laws (3.43), the derivative of the Lyapunov function $\dot{V}(t) = -\frac{1}{2}\mathbf{e}_p^T(t)\,\mathbf{Q}\mathbf{e}_p(t)$ becomes negative semi-definite. For implementing the control law (3.42), the parameter definitions $\hat{\boldsymbol{\Lambda}}_{rr}(t) = \hat{\boldsymbol{\Lambda}}_r(t)\,\mathbf{V}_r$ and $\hat{\boldsymbol{\Lambda}}_{rn}(t) = \hat{\boldsymbol{\Lambda}}_r(t)\,\mathbf{V}_n$ are used. Now the same argumentation as in the standard predictor-based MRAC case can be used in order to show error stability and reference model tracking. ■

Remark 3.16 *Interestingly, if the adaptation gain $\boldsymbol{\Gamma}_\Lambda$ is chosen as $\mathbf{I}\cdot\gamma$ with $\gamma > 0 \in \mathbb{R}$, the control effectiveness update laws can be separated as*

$$\begin{aligned}\dot{\hat{\boldsymbol{\Lambda}}}_{rr} &= \dot{\hat{\boldsymbol{\Lambda}}}_r\mathbf{V}_r = -\mathbf{D}\mathbf{U}_r^T\mathbf{P}\mathbf{e}_p\left(\mathbf{V}_r\mathbf{u}_r + \mathbf{V}_n\mathbf{u}_n\right)^T\mathbf{V}_r\gamma = -\mathbf{D}\mathbf{U}_r^T\mathbf{P}\mathbf{e}_p\mathbf{u}_r^T\gamma,\\ \dot{\hat{\boldsymbol{\Lambda}}}_{rn} &= \dot{\hat{\boldsymbol{\Lambda}}}_r\mathbf{V}_n = -\mathbf{D}\mathbf{U}_r^T\mathbf{P}\mathbf{e}_p\left(\mathbf{V}_r\mathbf{u}_r + \mathbf{V}_n\mathbf{u}_n\right)^T\mathbf{V}_n\gamma = -\mathbf{D}\mathbf{U}_r^T\mathbf{P}\mathbf{e}_p\mathbf{u}_n^T\gamma.\end{aligned}$$

Furthermore, if the control law is chosen such that $\mathbf{u}_n = \mathbf{0}$, only $\hat{\mathbf{\Lambda}}_{rr}$ needs to be updated. This corresponds to the result presented in [46], which has been generalized in this section.

Remark 3.17 *If the system is not overactuated, i.e. $r = m$, the matrix $\mathbf{V}_r = \mathbf{V}$ and the standard predictor-based MRAC formulation is recovered. In this case the parameters $\hat{\mathbf{\Theta}}_r(t)$ and $\hat{\mathbf{\Lambda}}_r(t)$ are orthonormal transformations of the original ones (combinations of rotations and reflections). Since in this case $\mathbf{V}_r$ is an orthogonal matrix, Assumptions 3.12 (Control Authority - PMRAC) and 3.14 (Control Authority - PMRAC PRO) are equivalent. Furthermore, since there is no null space of $\mathbf{B}_p$ and hence no $\mathbf{V}_n$, the control law (3.42) is composed only of $\mathbf{u}_r$ and there is no freedom left in order to choose $\mathbf{u}_n(t)$.*

Remark 3.18 *In order to guarantee that Assumption 3.14 (Control Authority - PMRAC PRO) holds, the SVD-Update proposed in [69] can be used. Given the derivative $\dot{\hat{\mathbf{\Lambda}}}_{rr}(t)$, this algorithm limits the minimum singular value from below to be positive.*

Example 3.6 (Application of PMRAC PRO) *In order to illustrate the presented results, let us apply PMRAC PRO to the plant in Example 3.1 (Simple Overactuated Linear System). The parameterized plant (3.35) with the regressor vector $\boldsymbol{\omega}(t) = \mathbf{x}_p(t) \in \mathbb{R}^2$ and the unknown parameters $\mathbf{\Theta} \in \mathbb{R}^{2\times 2}$ is already known from Example 3.5 (Application of PMRAC). Given the SVD of the example's input matrix $\mathbf{B}_p$ (3.19), the predictor dynamics using the reduced parameters $\mathbf{\Theta}_r = \mathbf{V}_r^T\mathbf{\Theta}$ and $\mathbf{\Lambda}_r = \mathbf{V}_r^T\mathbf{\Lambda}$ are then given as*

$$\begin{pmatrix}\dot{p}(t)\\ \dot{\Phi}(t)\end{pmatrix} = \mathbf{A}_m\begin{pmatrix}p(t)\\ \Phi(t)\end{pmatrix} + \underbrace{\begin{bmatrix}1\\0\end{bmatrix}L_\xi\sqrt{2}}_{=\mathbf{U}_r\cdot\mathbf{D}}\left(\mathbf{\Lambda}_r\begin{pmatrix}\xi_1(t)\\ \xi_2(t)\end{pmatrix} + \mathbf{\Theta}_r\begin{pmatrix}p(t)\\ \Phi(t)\end{pmatrix}\right). \tag{3.46}$$

The corresponding state predictor is then given as

$$\begin{pmatrix}\dot{\hat{p}}(t)\\ \dot{\hat{\Phi}}(t)\end{pmatrix} = \mathbf{A}_m\begin{pmatrix}\hat{p}(t)\\ \hat{\Phi}(t)\end{pmatrix} + \begin{bmatrix}1\\0\end{bmatrix}L_\xi\sqrt{2}\left(\hat{\mathbf{\Lambda}}_r\begin{pmatrix}\xi_1(t)\\ \xi_2(t)\end{pmatrix} + \hat{\mathbf{\Theta}}_r\begin{pmatrix}p(t)\\ \Phi(t)\end{pmatrix}\right).$$

In order to achieve the parameterization as in (3.41), the following transformations are required

$$\begin{aligned} u_r(t) &= \mathbf{V}_r^T\mathbf{u} = \tfrac{\sqrt{2}}{2}(\xi_1(t) + \xi_2(t)), & u_n(t) &= \mathbf{V}_n^T\mathbf{u} = \tfrac{\sqrt{2}}{2}(-\xi_1(t) + \xi_2(t)),\\ \Lambda_{rn} &= \mathbf{V}_r^T\mathbf{\Lambda}\mathbf{V}_n = \tfrac{1}{2}(-\lambda_1 + \lambda_2), & \Lambda_{rr} &= \mathbf{V}_r^T\mathbf{\Lambda}\mathbf{V}_r = \tfrac{1}{2}(\lambda_1 + \lambda_2), \end{aligned}$$

where the matrices $\mathbf{V}_r^T = \begin{bmatrix}\frac{\sqrt{2}}{2}, & \frac{\sqrt{2}}{2}\end{bmatrix}$ and $\mathbf{V}_n^T = \begin{bmatrix}-\frac{\sqrt{2}}{2}, & \frac{\sqrt{2}}{2}\end{bmatrix}$ are given by the SVD of $\mathbf{B}_p$ (3.19). This leads to the dynamics

$$\begin{pmatrix}\dot{\hat{p}}(t)\\ \dot{\hat{\Phi}}(t)\end{pmatrix} = \mathbf{A}_m\begin{pmatrix}\hat{p}(t)\\ \hat{\Phi}(t)\end{pmatrix} + \begin{bmatrix}1\\0\end{bmatrix}L_\xi\sqrt{2}\left(\hat{\Lambda}_{rr}(t)\,u_r(t) + \hat{\Lambda}_{rn}(t)(t)\,u_n(t) + \hat{\mathbf{\Theta}}_r\begin{pmatrix}p(t)\\ \Phi(t)\end{pmatrix}\right),$$

with the corresponding control law (3.42)

$$\begin{aligned} u_r(t) &= \hat{\Lambda}_{rr}^{-1}(t)\left(\mathbf{V}_r^T\mathbf{K}_r r(t) - \hat{\Lambda}_{rn}(t)\,u_n(t) - \hat{\mathbf{\Theta}}_r(t)\,\mathbf{x}_p(t)\right),\\ \mathbf{u}(t) &= \mathbf{V}_r u_r(t) + \mathbf{V}_n u_n(t). \end{aligned}$$

In this example both $u_r(t), u_n(t) \in \mathbb{R}$ *are scalar signals and are defined such that* u_n *can be chosen independently from the stability assessment because it represents the control allocation degrees of freedom due to overactuation. Furthermore, note that the reduced feed-forward gain* $\mathbf{V}_r^T\mathbf{K}_r$ *is uniquely defined such that* $\mathbf{B}_m = \mathbf{B}_p\mathbf{K}_r = \mathbf{U}_r\mathbf{D}\mathbf{V}_r^T\mathbf{K}_r$ *as follows*

$$\underbrace{\begin{bmatrix} kp \\ 0 \end{bmatrix}}_{\mathbf{B}_m} = \underbrace{\begin{bmatrix} 1 \\ 0 \end{bmatrix}}_{\mathbf{U}_r} \underbrace{\begin{bmatrix} L_\xi\sqrt{2} \end{bmatrix}}_{\mathbf{D}} \underbrace{\frac{\sqrt{2}}{2}\frac{k_p}{L_\xi}}_{\mathbf{V}_r^T\mathbf{K}_r}.$$

In order to implement the proposed control law (3.42)*, Assumption 3.14 (Control Authority - PMRAC PRO) replaced Assumption 3.12 (Control Authority - PMRAC). It states that the reduced control effectiveness matrix* $\Lambda_{rr} = \frac{1}{2}(\lambda_1 + \lambda_2)$ *must be invertible. This includes a failure of any single aileron pair, but as expected the simultaneous failure of the two actuators is excluded. Interestingly, the case in which one of the actuators has an unknown direction inversion and both actuators have exactly the same effectiveness is also excluded.*

The corresponding reduced update laws (3.34) *are given by*

$$\dot{\hat{\mathbf{\Theta}}}_r(t) = -L_\xi\sqrt{2}\begin{bmatrix} 1 & 0 \end{bmatrix}\mathbf{P}\mathbf{e}_p(t)\,\mathbf{x}_p^T(t)\,\mathbf{\Gamma}_\theta,$$
$$\dot{\hat{\mathbf{\Lambda}}}_r(t) = -L_\xi\sqrt{2}\begin{bmatrix} 1 & 0 \end{bmatrix}\mathbf{P}\mathbf{e}_p(t)\,\boldsymbol{\xi}(t)\,\mathbf{\Gamma}_\Lambda.$$

Since only the parameters $\hat{\mathbf{\Theta}}_r(t)$, $\hat{\mathbf{\Lambda}}_r(t) \in \mathbb{R}^{1\times 2}$ *need to be updated instead of* $\hat{\mathbf{\Theta}}(t)$, $\hat{\mathbf{\Lambda}}(t) \in \mathbb{R}^{2\times 2}$*, this reduces the number of parameters by a half.*

In this example, PMRAC PRO has been applied to the system described in Example 3.1. The resulting adaptive control law is able to fulfill the control objective for different failure cases including a total failure of any of the actuators. Table 3.3 summarizes the comparison between PMRAC with PMRAC PRO.

Predictor-Based MRAC		
Plant	$\dot{\mathbf{x}}_p(t) = \mathbf{A}_p\mathbf{x}_p(t) + \mathbf{B}_p\boldsymbol{\Lambda}\mathbf{u}(t) + \mathbf{K}_\phi\boldsymbol{\phi}(\mathbf{x}_p(t)) + \mathbf{d}$	(3.1)
Reference Model	$\dot{\mathbf{x}}_m(t) = \mathbf{A}_m\mathbf{x}_m(t) + \mathbf{B}_m\mathbf{r}(t)$	(3.2)
Predictor Model	$\dot{\hat{\mathbf{x}}}_p(t) = \mathbf{A}_m\hat{\mathbf{x}}_p(t) + \mathbf{B}_p\left(\hat{\boldsymbol{\Lambda}}(t)\mathbf{u}(t) + \hat{\boldsymbol{\Theta}}(t)\boldsymbol{\omega}(t)\right)$	(3.29)
Control Law	$\mathbf{u}(t) = \hat{\boldsymbol{\Lambda}}^{-1}(t)\left(\mathbf{K}_r\mathbf{r}(t) - \hat{\boldsymbol{\Theta}}(t)\boldsymbol{\omega}(t)\right)$	(3.30)
Matching Cond.	$[\mathbf{A}_p - \mathbf{A}_m, \quad \mathbf{K}_\phi, \quad \mathbf{d}] = \mathbf{B}_p\boldsymbol{\Theta}$	(3.28)
Error Dynamics	$\dot{\mathbf{e}}_p(t) = \mathbf{A}_m\mathbf{e}_p(t) + \mathbf{B}_p\tilde{\boldsymbol{\Lambda}}(t)\mathbf{u}(t) + \mathbf{B}_p\tilde{\boldsymbol{\Theta}}(t)\boldsymbol{\omega}(t)$	(3.32)
Update Laws	$\dot{\hat{\boldsymbol{\Theta}}}(t) = -\mathbf{B}_p^T\mathbf{P}\mathbf{e}_p(t)\boldsymbol{\omega}^T(t)\boldsymbol{\Gamma}_\theta,$ $\dot{\hat{\boldsymbol{\Lambda}}}(t) = -\mathbf{B}_p^T\mathbf{P}\mathbf{e}_p(t)\mathbf{u}^T(t)\boldsymbol{\Gamma}_\Lambda$	(3.34)
Control Authority	$\boldsymbol{\Lambda}$ and its estimate $\hat{\boldsymbol{\Lambda}}(t)$ are invertible	Asm. 3.12
Predictor-Based MRAC PRO		
Plant	$\dot{\mathbf{x}}_p(t) = \mathbf{A}_p\mathbf{x}_p(t) + \mathbf{B}_p\boldsymbol{\Lambda}\mathbf{u}(t) + \mathbf{K}_\phi\boldsymbol{\phi}(\mathbf{x}_p(t)) + \mathbf{d}$	(3.1)
Reference Model	$\dot{\mathbf{x}}_m(t) = \mathbf{A}_m\mathbf{x}_m(t) + \mathbf{B}_m\mathbf{r}(t)$	(3.2)
Predictor Model	$\dot{\hat{\mathbf{x}}}_p(t) = \mathbf{A}_m\hat{\mathbf{x}}_p(t) + \mathbf{U}_r\mathbf{D}(\hat{\boldsymbol{\Lambda}}_r(t)\mathbf{u}(t) + \hat{\boldsymbol{\Theta}}_r(t)\boldsymbol{\omega}(t))$	(3.41)
Control Law	$\mathbf{u}_r(t) = \hat{\boldsymbol{\Lambda}}_{rr}^{-1}(t)(\mathbf{V}_r^T\mathbf{K}_r\mathbf{r}(t) - \hat{\boldsymbol{\Lambda}}_{rn}(t)\mathbf{u}_n(t)$ $-\hat{\boldsymbol{\Theta}}_r(t)\boldsymbol{\omega}(t))$ $\mathbf{u}(t) = \mathbf{V}_r\mathbf{u}_r(t) + \mathbf{V}_n\mathbf{u}_n(t)$	(3.42)
Matching Cond.	$[\mathbf{A}_p - \mathbf{A}_m, \quad \mathbf{K}_\phi, \quad \mathbf{d}] = \mathbf{B}_p\boldsymbol{\Theta}$	(3.28)
Error Dynamics	$\dot{\mathbf{e}}_p(t) = \mathbf{A}_m\mathbf{e}_p(t) + \mathbf{U}_r\mathbf{D}\left(\tilde{\boldsymbol{\Lambda}}_r(t)\mathbf{u}(t) + \tilde{\boldsymbol{\Theta}}_r(t)\boldsymbol{\omega}(t)\right)$	(3.45)
Update Laws	$\dot{\hat{\boldsymbol{\Theta}}}_r(t) = -\mathbf{D}\mathbf{U}_r^T\mathbf{P}\mathbf{e}_p(t)\boldsymbol{\omega}^T(t)\boldsymbol{\Gamma}_\theta$ $\dot{\hat{\boldsymbol{\Lambda}}}_r(t) = -\mathbf{D}\mathbf{U}_r^T\mathbf{P}\mathbf{e}_p(t)\mathbf{u}^T(t)\boldsymbol{\Gamma}_\Lambda$	(3.43)
Control Authority	$\boldsymbol{\Lambda}_{rr} = \mathbf{V}_r^T\boldsymbol{\Lambda}\mathbf{V}_r$ and $\hat{\boldsymbol{\Lambda}}_{rr}(t)$ are invertible	Asm. 3.14

□ : Reduced parameters version.

□ : Inclusion of full actuator failure.

Table 3.3: Comparison between PMRAC and PMRAC PRO

3.6 Parameter Reduction due to Overactuation

In this section, the PRO approach presented in the preceding sections is summarized and generalized. The main goal of the approach is to formulate control authority conditions that can be applied to overactuated systems such that actuator failure cases can be considered within the stability proofs of model reference adaptive control. These control authority conditions directly affect the control effectiveness matrix $\boldsymbol{\Lambda}$ and usually impose non-singularity of this matrix as seen in Sections 3.2 and 3.4. As proposed in Sections 3.3 and 3.5, the PRO approach solves this issue via a parameter transformation and reduction. This led to the introduction of the reduced control effectiveness matrix

$\mathbf{\Lambda}_{rr} \in \mathbb{R}^{r\times r}$ whose dimensions reflect the degrees of freedom available for control of the plant. The control authority conditions are then applied to the reduced control effectiveness matrix $\mathbf{\Lambda}_{rr}$ instead of the control effectiveness matrix $\mathbf{\Lambda}$ in order to proof stability of the closed-loop and thereby taking actuator failure cases into account. In the following, these ideas are elaborated and summarized.

The starting point of the PRO approach is avoiding overparameterization due to actuator redundancy without losing any information. Therefore, consider an overactuated system of the form (3.1). The input matrix $\mathbf{B}_p \in \mathbb{R}^{n\times m}$ has the $\operatorname{rank}(\mathbf{B}_p) = r \leq \min(n, m)$ and the following singular value decomposition

$$\mathbf{B}_p = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \tag{3.47}$$

where $\mathbf{U} \in \mathbb{R}^{n\times n}, \mathbf{\Sigma} \in \mathbb{R}^{n\times m}, \mathbf{V} \in \mathbb{R}^{m\times m}$ [74, Th. 2.6.3]. $\mathbf{U}, \mathbf{V}$ are orthogonal matrices, i.e. $\mathbf{U}\mathbf{U}^T = \mathbf{I}$ and $\mathbf{V}\mathbf{V}^T = \mathbf{I}$. $\mathbf{\Sigma}$ is a rectangular diagonal matrix with non-negative entries on the diagonal which can be written as

$$\mathbf{\Sigma} = \begin{bmatrix} \mathbf{D} & \mathbf{0}_{r\times m-r} \\ \mathbf{0}_{n-r\times r} & \mathbf{0}_{n-r\times m-r} \end{bmatrix},$$

where $\mathbf{D} \in \mathbb{R}^{r\times r}$ is a diagonal positive definite matrix. The diagonal elements are the singular values of $\mathbf{B}_p$ and have a decreasing order. Taking this structure into account, it is clear that if we partition $\mathbf{U} = [\mathbf{U}_r, \mathbf{U}_n]$ and $\mathbf{V} = [\mathbf{V}_r, \mathbf{V}_n]$, such that $\mathbf{U}_r \in \mathbb{R}^{n\times r}$, $\mathbf{U}_n \in \mathbb{R}^{n\times(n-r)}$, $\mathbf{V}_r \in \mathbb{R}^{m\times r}$ and $\mathbf{V}_n \in \mathbb{R}^{m\times(m-r)}$, the matrix $\mathbf{B}_p$ can be alternatively written as

$$\mathbf{B}_p = \mathbf{U}_r\mathbf{D}\mathbf{V}_r^T. \tag{3.48}$$

In the following theorem, this SVD is used to achieve parameter reduction.

Theorem 3.19 (Parameter Reduction due to Overactuation) *Let $\hat{\mathbf{\Theta}}(t) \in \mathbb{R}^{m\times q}$ be the parameter estimate of the ideal parameter $\mathbf{\Theta} \in \mathbb{R}^{m\times q}$. It is possible to reduce the set of parameters without any information loss, if the system is overactuated and the parameter update law has the form*

$$\dot{\hat{\mathbf{\Theta}}}(t) = -\mathbf{B}_p^T\mathbf{C}(t). \tag{3.49}$$

Here, $\mathbf{C}(t) \in \mathbb{R}^{n\times q}$ is a continuous function which depends on the specific control approach. Using the SVD of $\mathbf{B}_p$ (3.48), the reduced parameter set is defined as $\hat{\mathbf{\Theta}}_r(t) = \mathbf{V}_r^T\hat{\mathbf{\Theta}}(t) \in \mathbb{R}^{r\times q}$ and its update law is given by

$$\dot{\hat{\mathbf{\Theta}}}_r(t) = -\mathbf{D}\mathbf{U}_r^T\mathbf{C}(t).$$

The original parameters are computed from

$$\hat{\mathbf{\Theta}}(t) = \mathbf{V}_r\hat{\mathbf{\Theta}}_r(t) + \mathbf{V}_n\hat{\mathbf{\Theta}}_n(0),$$

where $\hat{\mathbf{\Theta}}_n(t) = \mathbf{V}_n^T\hat{\mathbf{\Theta}}(t) \in \mathbb{R}^{(m-r)\times q}$. Because only the parameters $\hat{\mathbf{\Theta}}_r(t)$ need to be updated, the parameters are reduced from $(m \cdot q)$ to $(r \cdot q)$. Note that the choice $\hat{\mathbf{\Theta}}_n(t=0) = 0$ further simplifies the computation of the original parameters.

Proof: First note that the following parameter transformation

$$\mathbf{\Theta}_b(t) = \begin{bmatrix} \mathbf{\Theta}_r(t) \\ \mathbf{\Theta}_n(t) \end{bmatrix} = \begin{bmatrix} \mathbf{V}_r^T\mathbf{\Theta}(t) \\ \mathbf{V}_n^T\mathbf{\Theta}(t) \end{bmatrix} = \mathbf{V}^T\mathbf{\Theta}(t) \tag{3.50}$$

is invertible and that the original parameters can be computed from its inverse as

$$\boldsymbol{\Theta}(t) = \mathbf{V}\boldsymbol{\Theta}_b(t) = \mathbf{V}_r\boldsymbol{\Theta}_r(t) + \mathbf{V}_n\boldsymbol{\Theta}_n(t). \tag{3.51}$$

The parameter $\boldsymbol{\Theta}_r \in \mathbb{R}^{r\times q}$ lies in the range of $\mathbf{B}_p^T$ and $\boldsymbol{\Theta}_n \in \mathbb{R}^{(m-r)\times q}$ corresponds to the null space of $\mathbf{B}_p$. Let the parameter estimates $\hat{\boldsymbol{\Theta}}(t)$ be analogously transformed as in (3.50) such that $\hat{\boldsymbol{\Theta}}_b(t) \in \mathbb{R}^{m\times q}$, $\hat{\boldsymbol{\Theta}}_r(t) \in \mathbb{R}^{r\times q}$ and $\hat{\boldsymbol{\Theta}}_n(t) \in \mathbb{R}^{(m-r)\times q}$. The update laws of the transformed parameters are computed using the SVD of $\mathbf{B}_p$ (3.48), the original update law (3.49) and the fact that $\mathbf{V}$ is an orthogonal matrix (Lemma B.10) as follows

$$\dot{\hat{\boldsymbol{\Theta}}}_r(t) = \mathbf{V}_r^T\dot{\hat{\boldsymbol{\Theta}}}(t) = -\mathbf{V}_r^T\mathbf{V}_r\mathbf{D}\mathbf{U}_r^T\mathbf{C}(t) = -\mathbf{D}\mathbf{U}_r^T\mathbf{C}(t), \tag{3.52}$$

$$\dot{\hat{\boldsymbol{\Theta}}}_n(t) = \mathbf{V}_n^T\dot{\hat{\boldsymbol{\Theta}}}(t) = -\mathbf{V}_n^T\mathbf{V}_r\mathbf{D}\mathbf{U}_r^T\mathbf{C}(t) = \mathbf{0}. \tag{3.53}$$

It is possible to see that the parameters $\hat{\boldsymbol{\Theta}}(t)$ are only updated in the subspace corresponding to $\hat{\boldsymbol{\Theta}}_r(t)$. Then, the parameter $\hat{\boldsymbol{\Theta}}(t)$ can be written as follows

$$\begin{aligned}
\hat{\boldsymbol{\Theta}}(t) &= \hat{\boldsymbol{\Theta}}(0) + \int_{\tau=0}^{t} \dot{\hat{\boldsymbol{\Theta}}}(\tau)\, d\tau, \\
&= \mathbf{V}_r\hat{\boldsymbol{\Theta}}_r(0) + \mathbf{V}_n\hat{\boldsymbol{\Theta}}_n(0) + \int_{\tau=0}^{t} \mathbf{V}_r\dot{\hat{\boldsymbol{\Theta}}}_r(\tau) + \mathbf{V}_n\dot{\hat{\boldsymbol{\Theta}}}_n(\tau)\, d\tau, \\
&= \mathbf{V}_r\hat{\boldsymbol{\Theta}}_r(t) + \mathbf{V}_n\hat{\boldsymbol{\Theta}}_n(0).
\end{aligned}$$

Hence, $\hat{\boldsymbol{\Theta}}(t)$ is fully defined by $\hat{\boldsymbol{\Theta}}_r(t)$ and the initial condition $\hat{\boldsymbol{\Theta}}_n(0)$. The parameter reduction follows from the fact that only $\hat{\boldsymbol{\Theta}}_r \in \mathbb{R}^{r\times q}$ is updated and it has a smaller dimension than $\hat{\boldsymbol{\Theta}} \in \mathbb{R}^{m\times q}$ if the system is overactuated, i.e. $r < m$. ■

Theorem 3.19 is restricted to parameters that have an update law of the form (3.49). This arises from the fact that a reduction of overparameterization due to overactuation is only possible for parameters that are directly related to the number of inputs. These parameters usually correspond to terms of one of the following two forms: $\mathbf{B}_p\boldsymbol{\Theta}\boldsymbol{\omega}(t)$ or $\mathbf{B}_p\boldsymbol{\Lambda}\boldsymbol{\Theta}\boldsymbol{\omega}(t)$. The first one usually corresponds to a reparameterization of plant parameters that lie in the range of $\mathbf{B}_p$ as in the case of predictor-based MRAC. The second one usually corresponds to controller gains that are adapted directly like in the direct MRAC case. The presented reduced parameter $\boldsymbol{\Theta}_r$ does not only have implementation advantages but is the key to developing fault tolerant adaptive control for overactuated systems. This was demonstrated in Section 3.3 for direct MRAC and in Section 3.5 for PMRAC.

The division of the parameters into $\boldsymbol{\Theta}_r$ and $\boldsymbol{\Theta}_n$ has an intuitive interpretation if these parameters correspond to a term of the form $\mathbf{B}_p\boldsymbol{\Theta}\boldsymbol{\omega}(t)$. From the SVD of $\mathbf{B}_p$ (3.48) and the parameter transformation (3.50), it follows that

$$\mathbf{B}_p\boldsymbol{\Theta}\boldsymbol{\omega}(t) = \mathbf{U}_r\mathbf{D}\mathbf{V}_r^T\left(\mathbf{V}_r\boldsymbol{\Theta}_r + \mathbf{V}_n\boldsymbol{\Theta}_n\right)\boldsymbol{\omega}(t) = \mathbf{U}_r\mathbf{D}\boldsymbol{\Theta}_r\boldsymbol{\omega}(t).$$

Hence, reduced parameters $\boldsymbol{\Theta}_r$ are the only ones that influence the system dynamics and correspond to a minimal parameterization. The parameters $\boldsymbol{\Theta}_n$ do not affect the dynamics and this is the reason why there is no update in this direction, i.e. $\dot{\hat{\boldsymbol{\Theta}}}_n(t) = \mathbf{0}$.

In the case that the parameters appear in a term of the form $\mathbf{B}_p\boldsymbol{\Lambda}\boldsymbol{\Theta}\boldsymbol{\omega}(t)$ the interpretation is only correct for $\boldsymbol{\Lambda} = \mathbf{I}$. If $\boldsymbol{\Lambda}$ changes, there exists a coupling in the sense that

$\boldsymbol{\Theta}_r$ is partially in the null space of $\mathbf{B}_p\boldsymbol{\Lambda}$ and $\boldsymbol{\Theta}_n$ is not totally in the null space of $\mathbf{B}_p\boldsymbol{\Lambda}$. Nevertheless, if $\operatorname{rank}(\mathbf{B}_p) = \operatorname{rank}(\mathbf{B}_p\boldsymbol{\Lambda}) = r$, the virtual controls are still r-dimensional. The reduced parameters $\boldsymbol{\Theta}_r \in \mathbb{R}^{r\times q}$ still represent the number of parameters that are fixed by the matching conditions and $\boldsymbol{\Theta}_n \in \mathbb{R}^{(m-r)\times q}$ has the number of parameters that can be chosen freely.

The first step of the PRO approach is the parameter reduction as described by Theorem 3.19 (Parameter Reduction due to Overactuation). The second step is the reformulation of the control authority assumptions using a reduced control efficiency matrix. Before analyzing the conditions needed within adaptive control, control authority is studied as a characteristic of the plant under the failure conditions modeled by the control efficiency matrix $\boldsymbol{\Lambda}$. Control authority is directly related to the term $\mathbf{B}_p\boldsymbol{\Lambda}\mathbf{u}(t)$ and how the control input is chosen. In order to maintain the full control authority in the faulty case, the following condition is needed.

Assumption 3.20 (Control Authority - Fault Case) *For a given vector $\boldsymbol{\chi}(t) \in \mathbb{R}^n$, there exists a control input $\mathbf{u}(t) \in \mathbb{R}^m$ such that*

$$\boldsymbol{\chi}(t) = \mathbf{B}_p\boldsymbol{\Lambda}\mathbf{u}(t) \tag{3.54}$$

given that the solution of

$$\boldsymbol{\chi}(t) = \mathbf{B}_p\mathbf{u}(t) \tag{3.55}$$

exists (possibly with a different input $\mathbf{u}(t)$). This condition translates to the rank condition

$$\operatorname{rank}(\mathbf{B}_p) = \operatorname{rank}(\mathbf{B}_p\boldsymbol{\Lambda}) = r. \tag{3.56}$$

This assumption mainly states that the influence of the input $\mathbf{u}(t)$ on the state derivatives maintains its same r directions during nominal operation $\boldsymbol{\Lambda} = \mathbf{I}$ as well as during a faulty case $\boldsymbol{\Lambda} \neq \mathbf{I}$. This is a condition on $\boldsymbol{\Lambda}$ that is required to operate with full authority during failure conditions. Because $\operatorname{rank}(\mathbf{B}_p\boldsymbol{\Lambda}) \leq \min(\operatorname{rank}(\mathbf{B}_p), \operatorname{rank}(\boldsymbol{\Lambda}))$ holds for the rank of any matrix product and $\operatorname{rank}(\mathbf{B}_p) = r$, it is necessary that $r \leq \operatorname{rank}(\boldsymbol{\Lambda})$. Physically it means that at most $m - r$ actuators can fail and it corresponds to the level of overactuation in the system. Assumption 3.20 can be translated to a condition on the control effectiveness matrix $\boldsymbol{\Lambda}$ as stated by the following proposition.

Proposition 3.21 *Assumption 3.20 (Control Authority - Fault Case) holds if the matrix*

$$\mathbf{V}_r^T\boldsymbol{\Lambda}\boldsymbol{\Lambda}^T\mathbf{V}_r \tag{3.57}$$

is invertible. This matrix is not to be confused with $\boldsymbol{\Lambda}_{rr} = \mathbf{V}_r^T\boldsymbol{\Lambda}\mathbf{V}_r$.

Proof: From (3.55), it is known that $\boldsymbol{\chi}(t)$ lies within the range of $\mathbf{B}_p$, i.e. $\mathbf{U}_n^T\boldsymbol{\chi}(t) = \mathbf{0}$. Therefore using Lemma B.10 (Orthogonal Matrix) it holds that

$$\boldsymbol{\chi}(t) = \mathbf{U}_r\mathbf{U}_r^T \cdot \boldsymbol{\chi}(t) = \left(\mathbf{I} - \mathbf{U}_n\mathbf{U}_n^T\right) \cdot \boldsymbol{\chi}(t).$$

Using the SVD of $\mathbf{B}_p = \mathbf{U}_r\mathbf{D}\mathbf{V}_r^T$ (3.48), and by multiplying condition (3.54) with $\mathbf{D}^{-1}\mathbf{U}_r^T$ from the left it follows that

$$\mathbf{D}^{-1}\mathbf{U}_r^T \cdot \boldsymbol{\chi}(t) = \mathbf{V}_r^T\boldsymbol{\Lambda} \cdot \mathbf{u}(t).$$

Since $\mathbf{D}^{-1}\mathbf{U}_r^T \cdot \boldsymbol{\chi}(t) \in \mathbb{R}^r$ and $\mathbf{V}_r^T\boldsymbol{\Lambda} \in \mathbb{R}^{r\times m}$, this equation has a solution for any vector $\boldsymbol{\chi}(t)$ if $\mathbf{V}_r^T\boldsymbol{\Lambda}$ has full row rank. From Lemma B.4 (Rank) in Appendix B, it follows that the symmetric matrix $\mathbf{V}_r^T\boldsymbol{\Lambda}\boldsymbol{\Lambda}^T\mathbf{V}_r$ has a full rank too. Therefore, it is invertible and the proposition is proven. ■

Assumption 3.20 and Proposition 3.21 are conditions on the control effectiveness matrix $\boldsymbol{\Lambda}$ that guarantee the same control authority in the nominal and faulty cases. They represent the limit of the physical capabilities of the plant. In adaptive control approaches, this condition is usually more restrictive because of the fact that $\boldsymbol{\Lambda}$ is unknown. In direct MRAC Assumption 3.2 (Control Authority - direct MRAC) states that positive definiteness of $\boldsymbol{\Lambda}$ is necessary. In predictor-based MRAC Assumtion 3.12 (Control Authority - PMRAC) states that $\boldsymbol{\Lambda}$ needs to be invertible. In both cases a total actuator failure is ruled out as seen in Table 3.1. In Sections 3.3 and 3.5, fault-tolerant versions of direct and predictor-based MRAC have been presented. The next theorem summarizes the strategy applied.

Theorem 3.22 (Fault Tolerant Control Authority Conditions) *In order to take faults and failures within adaptive control approaches in the case of overactuated systems into account, the following strategy is proposed. First, identify the control authority conditions which assume that the control effectiveness matrix $\boldsymbol{\Lambda}$ is positive definite or invertible. Subsequently, these assumptions are exchanged by the same type of condition applied to the reduced control effectiveness matrix defined by*

$$\boldsymbol{\Lambda}_{rr} = \mathbf{V}_r^T\boldsymbol{\Lambda}\mathbf{V}_r.$$

Here, $\mathbf{V}_r \in \mathbb{R}^{m\times r}$ is a partition of $\mathbf{V} = [\mathbf{V}_r, \mathbf{V}_n]$ defined by the SVD of $\mathbf{B}_p$ (3.47). An overview is given by the following table.

Original Assumption	*New Assumption*
$\boldsymbol{\Lambda}$ *is positive definite.*	$\boldsymbol{\Lambda}_{rr}$ *is positive definite.*
$\boldsymbol{\Lambda}$ *is invertible.*	$\boldsymbol{\Lambda}_{rr}$ *is invertible.*

For non-overactuated systems with $r = m$, original and new assumptions are equivalent and actuator failures are not allowed. In the case of overactuated systems $r < m$, a maximum of $m - r$ actuator failures can be taken into account.

Proof: First consider the following invertible transformation

$$\boldsymbol{\Lambda}_{bb} = \mathbf{V}^T\boldsymbol{\Lambda}\mathbf{V} = \begin{bmatrix}\mathbf{V}_r^T\\ \mathbf{V}_n^T\end{bmatrix}\boldsymbol{\Lambda}\begin{bmatrix}\mathbf{V}_r & \mathbf{V}_n\end{bmatrix} = \begin{bmatrix}\mathbf{V}_r^T\boldsymbol{\Lambda}\mathbf{V}_r & \mathbf{V}_r^T\boldsymbol{\Lambda}\mathbf{V}_n\\ \mathbf{V}_n^T\boldsymbol{\Lambda}\mathbf{V}_r & \mathbf{V}_n^T\boldsymbol{\Lambda}\mathbf{V}_n\end{bmatrix}.$$

This transformation preserves the eigenvalues of the control effectiveness matrix $\boldsymbol{\Lambda}$, see Lemma B.6 (Orthogonal Transformation Invariance). In the non-overactuated case with $r = m$, $\mathbf{V}_r = \mathbf{V}$ is an orthogonal matrix and therefore the original and new assumptions are equivalent and actuator failures cannot be taken into account.

In the overactuated case $r < m$, it can be seen that $\boldsymbol{\Lambda}_{rr}$ is the $r \times r$ submatrix of the transformed matrix $\boldsymbol{\Lambda}_{bb}$. Because of the new control authority conditions, the restrictions to a positive definite or to an invertible matrix only apply to this reduced control effectiveness matrix $\boldsymbol{\Lambda}_{rr}$ instead of $\boldsymbol{\Lambda}$. Therefore, failures up to $m - r$ actuators can be taken into account. ■

Corollary 3.23 *If $\mathbf{\Lambda}$ is symmetric positive semi-definite, Assumption 3.20 is equivalent to $\mathbf{\Lambda}_{rr}$ being positive definite and to $\mathbf{\Lambda}_{rr}$ being invertible. Therefore, in this case, a controller using the conditions in Theorem 3.22 exploits the maximum physical capability of the plant. As shown in Table 3.1, a symmetric positive semi-definite $\mathbf{\Lambda}$ includes the cases of degradation of control effectiveness, actuator failure and stuck actuator.*

Proof: If $\mathbf{\Lambda}$ is symmetric positive semi-definite with rank $k \leq m$, its full rank factorization is given by [59, Eq. 3.157]

$$\mathbf{\Lambda} = \mathbf{L} \cdot \mathbf{L}^T, \tag{3.58}$$

where $\mathbf{L} \in \mathbb{R}^{m \times k}$ is full column rank. In the following, the rank condition (3.56) in Assumption 3.20 (Control Authority - Fault Case) is checked. From the SVD of $\mathbf{B}_p$ (3.48) and Lemma B.4 (Rank) and noting that $\mathbf{U}_r$ and $\mathbf{D}$ are full column rank matrices it follows that

$$\operatorname{rank}\left(\mathbf{B}_p\mathbf{\Lambda}\right) = \operatorname{rank}\left(\mathbf{U}_r\mathbf{D}\mathbf{V}_r^T\mathbf{\Lambda}\right) = \operatorname{rank}\left(\mathbf{V}_r^T\mathbf{\Lambda}\right).$$

Since $\mathbf{L}$ is full column rank, inserting (3.58) and using Lemma B.4 (Rank) leads to

$$\begin{aligned}
\operatorname{rank}\left(\mathbf{B}_p\mathbf{\Lambda}\right) &= \operatorname{rank}\left(\mathbf{V}_r^T\mathbf{L} \cdot \mathbf{L}^T\right), \\
&= \operatorname{rank}\left(\mathbf{V}_r^T\mathbf{L}\right), \\
&= \operatorname{rank}\left(\left(\mathbf{V}_r^T\mathbf{L}\right) \cdot \left(\mathbf{L}^T\mathbf{V}_r\right)\right), \\
&= \operatorname{rank}\left(\left(\mathbf{V}_r^T\mathbf{\Lambda}\mathbf{V}_r\right)\right), \\
&= \operatorname{rank}\left(\mathbf{\Lambda}_{rr}\right).
\end{aligned}$$

Hence, Assumption 3.20 is equivalent to $\operatorname{rank}\left(\mathbf{B}_p\right) = \operatorname{rank}\left(\mathbf{\Lambda}_{rr}\right) = r$. Since $\mathbf{\Lambda}_{rr} \in \mathbb{R}^{r \times r}$, this is equivalent to $\mathbf{\Lambda}_{rr}$ being invertible [74, Sec. 0.5].

From Lemma B.6 (Orthogonal Transformation Invariance) it follows that $\mathbf{\Lambda}_{bb} = \mathbf{V}^T\mathbf{\Lambda}\mathbf{V}$ is positive semi-definite. Therefore, all of its principal submatrices including $\mathbf{V}_r^T\mathbf{\Lambda}\mathbf{V}_r = \mathbf{\Lambda}_{rr}$ are also positive semi-definite [74, Obs. 7.1.2]. Hence, $\mathbf{\Lambda}_{rr}$ has non-negative eigenvalues. From the condition $\operatorname{rank}\left(\mathbf{\Lambda}_{rr}\right) = r$ it follows that 0 is not an eigenvalue of $\mathbf{\Lambda}_{rr}$ and the equivalence with positive definiteness of $\mathbf{\Lambda}_{rr}$ is proved. ■

The application of Theorem 3.22 can be seen in the comparison Tables 3.2 (direct MRAC) and 3.3 (PMRAC). The selection of the reduced control effectiveness matrix $\mathbf{\Lambda}_{rr}$ is underlined by the equivalence of the conditions in Theorem 3.22 to Assumption 3.20 in the case of a symmetric positive semi-definite control effectiveness matrix $\mathbf{\Lambda}$. In this case the adaptive control assumptions do not restrict the physical capabilities of the system and the best possible result is achieved as required from the thesis objective 9 (The controller should not be a limitation of the physical system). The restriction to positive semi-definiteness can be seen as the correct knowledge of the control input directions. Specifically, the condition $\mathbf{u}^T(t) \cdot \left(\mathbf{\Lambda}\mathbf{u}(t)\right) \geq 0$ means that the vector $\mathbf{u}(t)$ cannot be rotated more than $\pm 90[^\circ]$ by the matrix $\mathbf{\Lambda}$.

A limitation of using the reduced control effectiveness matrix $\mathbf{\Lambda}_{rr}$ is that if $\mathbf{\Lambda}$ is invertible, it is not true that $\mathbf{\Lambda}_{rr}$ is invertible too. For most applications, this is not much of a limitation since the uncertainty about the control input directions is not higher than $\pm 90[^\circ]$ and therefore Corollary 3.23 can be applied.

Example 3.7 (Controllability Conditions) *In order to illustrate the controllability conditions, consider the system in Example 3.1 which has the following control efectiveness matrix*

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}.$$

As derived in Example 3.4, the reduced control effectiveness matrix is given by

$$\Lambda_{rr} = \frac{1}{2}\left(\lambda_1 + \lambda_2\right).$$

The matrix $\mathbf{\Lambda}$ is positive definite if both $\lambda_1, \lambda_2 > 0$. This implies that $\Lambda_{rr} > 0$, i.e. Λ_{rr} is positive definite too.

The situation with an invertible $\mathbf{\Lambda}$ is different. $\mathbf{\Lambda}$ is invertible as long as $\lambda_1, \lambda_2 \neq 0$. This excludes actuator failures. In the case of the reduced Λ_{rr}, actuator failures can be taken into account as long as $\lambda_1 + \lambda_2 \neq 0$. Here, the failure of a single actuator can be taken into account. But if $\lambda_1 = -\lambda_2$ and $\lambda_1, \lambda_2 \neq 0$, only the original control effectiveness matrix $\mathbf{\Lambda}$ is invertible. This case corresponds to a unknown direction inversion of one of the actuators in such a way that the two actuators cancel each other and $\Lambda_{rr} = 0$.

In this section, the thesis objectives 1 (Systematic analysis of adaptive control methods for systems with redundant actuators) and 2 (Study of adaptive control methods regarding actuator fault tolerance in the case of overactuated systems) have been addressed and the PRO approach has been presented. Its first step is a parameter reduction due to overactuation as in Theorem 3.19 (Parameter Reduction due to Overactuation). The second step uses the reduced control effectiveness matrix $\mathbf{\Lambda}_{rr}$ in order to formulate the control authority conditions within a given control approach as stated in Theorem 3.22 (Fault Tolerant Control Authority Conditions). Using the PRO approach, fault tolerant versions of adaptive control approaches for overactuated systems can be obtained. In this chapter, this approach has been applied to direct MRAC in Section 3.3 and to PMRAC in Section 3.5.

3.7 Bibliographical Remarks

Although the adaptive control literature addresses degradation and failure of actuators, the specific stability assessment of overactuated systems has not received much attention. The main restrictions for the applicability to actuator failures correspond to assumptions regarding the control effectiveness matrix. The basis for this chapter was presented in [47] for direct MRAC PRO and in [46] for PMRAC PRO. To the best knowledge of the author, this was the first time that actuator failures were integrated into the assumptions regarding the control effectiveness matrix. Previously, Tao presented an equivalent condition for the direct MRAC case in his book [138]. The used failure model corresponds to (3.5) with a diagonal positive semi-definite control effectiveness matrix $\mathbf{\Lambda}$, which can only have the values $\{0, 1\}$. This model in combination with Proposition 2.5.2 in [138] is equivalent to Assumptions 3.1 (Matching Condition - direct MRAC) and 3.5 (Control Authority - direct MRAC PRO) in Section 3.3. Since $\mathbf{\Lambda}$ does not need to be positive semi-definite for Assumption 3.5 (Control Authority - direct MRAC PRO) to hold, it corresponds to a slightly generalized condition. Furthermore, Tao focuses on groups of actuators that present equal characteristics. In this work, the used formulation along the lines of [92] facilitates the application towards multivariable systems with high coupling of actuator and control axes like VTOL aircraft. Another differentiation is the application to indirect adaptive control.

Chapter 4

Adaptive Control Allocation

In the last chapter, fault tolerance adaptive control for overactuated systems was analyzed and the PRO approach was introduced. The next major step towards real life applications is the handling of control input saturations by taking advantage of the actuator redundancy. In this chapter, the integration of the control allocation approach within the predictor based MRAC framework is proposed. An indirect method is selected since the integration is simpler than in the direct MRAC case.

In the following, the Control Task 3.1 (Model Reference Adaptive Control) and the reference model (3.2) remain the same as in the previous chapter. Equivalent to (3.1) and (3.4)-(3.5), the following plant representation with separated rigid body and actuator models is used

$$\dot{\mathbf{x}}_p(t) = \mathbf{A}_p \mathbf{x}_p(t) + \mathbf{B}_\nu \boldsymbol{\nu}(t) + \mathbf{K}_\phi \boldsymbol{\phi}(\mathbf{x}_p(t)) + \mathbf{d}, \tag{4.1a}$$

$$\boldsymbol{\nu}(t) = \mathbf{B}_a \boldsymbol{\Lambda} \mathbf{u}(t). \tag{4.1b}$$

Here, the full rank factorization $\mathbf{B}_p = \mathbf{B}_\nu \cdot \mathbf{B}_a$ with $\operatorname{rank}(\mathbf{B}_\nu) = \operatorname{rank}\mathbf{B}_a = r$ is used. Furthermore, the constant actuator disturbance $\mathbf{d}_\nu$ from (3.5) is considered in $\mathbf{d} = \mathbf{B}_\nu \mathbf{d}_\nu + \mathbf{d}_x$ without loss of generality. The control allocation task can then be formalized as:

Control Task 4.1 (Control Allocation) *Given the desired virtual control input $\boldsymbol{\nu}_d(t, \mathbf{x}_p) \in \mathbb{R}^r$, compute the control input $\mathbf{u}(t, \boldsymbol{\nu}_d) \in \mathbb{R}^m$ such that*

$$\boldsymbol{\nu}_d(t) = \mathbf{B}_a \boldsymbol{\Lambda} \mathbf{u}(t) \tag{4.2}$$

and such that the input lies within the input set $\mathcal{U} \subset \mathbb{R}^m$ defined as

$$\mathcal{U} := \{\mathbf{u} \in \mathbb{R}^m \mid u_{i,min} \leq u_i \leq u_{i,max}, \forall i = 1...m\},$$

where u_i is the i-th element of $\mathbf{u}(t)$ and $u_{i,min}, u_{i,max} \in \mathbb{R}$ are the corresponding minimum and maximum values. The control input set $\mathcal{U}$ is an m-dimensional hyperrectangle.

Because of the saturations, the Control Task 4.1 cannot be solved for every desired virtual control $\boldsymbol{\nu}_d(t)$. As in Chapter 2 (Multirotor Dynamics Model), the attainable control space ACS is defined as the output image $\mathcal{V}$ of the linear mapping $(4.1b)$ on $\mathcal{U}$

$$\mathcal{V} := \{\boldsymbol{\nu} \in \mathbb{R}^r \mid \boldsymbol{\nu} = \mathbf{B}_a \boldsymbol{\Lambda} \mathbf{u}, \mathbf{u} \in \mathcal{U}\}.$$

Because the input set $\mathcal{U}$ is a convex polyhedra and it is linearly mapped to the set $\mathcal{V}$, it follows that the ACS $\mathcal{V}$ is also a convex polyhedra.

There is a great variety of control allocation methods in the literature as shown in Chapter 1 (Introduction). The main burden to directly apply most of the available methods to the problem at hand is the assumption of a known and constant matrix $\mathbf{B}_a\mathbf{\Lambda}$. In this chapter, this issue is tackled using adaptive control. The remainder of the chapter is organized as follows. In Section 4.1, the PMRAC approach is formulated such that the rigid body control and the control allocation are separated tasks. In Section 4.2, the structure of the dynamical system is exploited to formulate a reduced-order version of the presented approach. In Section 4.3, the Prioritizing Control Allocation approach is introduced followed by the required SVD update algorithm in Section 4.4. Finally, bibliographical remarks are given in Section 4.5.

4.1 Predictor-Based Adaptive Control Allocation

In this section, a special form of predictor-based MRAC is derived such that the integration of control allocation algorithms is facilitated. In a first step, the plant (4.1) is rewritten using the following parameterization

$$\dot{\mathbf{x}}_p(t) = \mathbf{A}_m\mathbf{x}_p(t) + \mathbf{B}_\nu\boldsymbol{\nu}(t) + \mathbf{B}_\nu\mathbf{\Theta}_\nu\boldsymbol{\omega}(t), \tag{4.3a}$$

$$\boldsymbol{\nu}(t) = \mathbf{U}_{a,r}\mathbf{D}_a\mathbf{\Lambda}_{a,r}\mathbf{u}(t). \tag{4.3b}$$

where $\boldsymbol{\omega}(t) \in \mathbb{R}^q$ is the known regressor vector and $\mathbf{\Theta}_\nu \in \mathbb{R}^{r\times q}$ is the unknown parameter matrix. Here, the PRO parameter transformation presented in Chapter 3 has been applied to the actuator model $(4.1\mathrm{b})$. The SVD decomposition of the matrix $\mathbf{B}_a$ is analogously defined as (3.48) leading to

$$\mathbf{B}_a = \mathbf{U}_{a,r}\mathbf{D}_a\mathbf{V}_{a,r}^T \tag{4.4}$$

with $\mathbf{U}_{a,r} \in \mathbb{R}^{r\times r}$, $\mathbf{D}_a \in \mathbb{R}^{r\times r}$ and $\mathbf{V}_{a,r} \in \mathbb{R}^{m\times r}$. The matrix $\mathbf{\Lambda}_{a,r} \in \mathbb{R}^{r\times m}$ is defined as

$$\mathbf{\Lambda}_{a,r} = \mathbf{V}_{a,r}^T\mathbf{\Lambda}. \tag{4.5}$$

The specific parameterization of (4.3) intends to facilitate the derivation of a control law in the case of known parameters and the separation of the control allocation task. The known regressor vector $\boldsymbol{\omega}(t)$ and the unknown ideal parameter matrix $\mathbf{\Theta}_\nu$ are defined as

$$\boldsymbol{\omega}(t) := \left[\mathbf{x}_p^T(t), \boldsymbol{\phi}^T(\mathbf{x}_p(t)), 1\right]^T \in \mathbb{R}^q,$$
$$\mathbf{\Theta}_\nu := \left[\mathbf{\Theta}_{p,x}, \mathbf{\Theta}_{p,\phi}, \mathbf{\Theta}_{p,d}\right] \in \mathbb{R}^{r\times q},$$

with $q = n+k+1$, $\mathbf{\Theta}_{p,x} \in \mathbb{R}^{r\times n}$, $\mathbf{\Theta}_{p,\phi} \in \mathbb{R}^{r\times k}$ and $\mathbf{\Theta}_{p,d} \in \mathbb{R}^r$. The comparison between the parameterized plant (4.3) and its original representation (4.1) leads to the subsequent matching condition.

Assumption 4.1 (Matching Condition - P-ACA) *The ideal parameters $\mathbf{\Theta}_\nu$ satisfy*

$$\mathbf{Y} := \left[\mathbf{A}_p - \mathbf{A}_m, \quad \mathbf{K}_\phi, \quad \mathbf{d}\right] = \mathbf{B}_\nu\mathbf{\Theta}_\nu. \tag{4.6}$$

Remark 4.2 (Uniqueness of the ideal parameter $\boldsymbol{\Theta}_\nu$) *Note that if a solution to the matching condition* (4.6) *exists, the solution* $\boldsymbol{\Theta}_\nu$ *is unique. This follows from the fact that* $\mathbf{B}_\nu$ *has a full column rank by definition [110, p. 260]. Hence, the parameter matrix* $\boldsymbol{\Theta}_\nu$ *is not overparameterized due to overactuation.*

The state predictor of the parameterized plant (4.3) is defined as

$$\dot{\hat{\mathbf{x}}}_p(t) = \mathbf{A}_m \hat{\mathbf{x}}_p(t) + \mathbf{B}_\nu \hat{\boldsymbol{\nu}}(t) + \mathbf{B}_\nu \hat{\boldsymbol{\Theta}}_\nu(t)\,\boldsymbol{\omega}(t), \tag{4.7a}$$

$$\hat{\boldsymbol{\nu}}(t) = \mathbf{U}_{a,r}\mathbf{D}_a \hat{\boldsymbol{\Lambda}}_{a,r}(t)\,\mathbf{u}(t), \tag{4.7b}$$

where $\hat{\mathbf{x}}_p(t) \in \mathbb{R}^n$, $\hat{\boldsymbol{\nu}}(t) \in \mathbb{R}^r$, $\hat{\boldsymbol{\Theta}}_\nu(t) \in \mathbb{R}^{r\times q}$ and $\hat{\boldsymbol{\Lambda}}_{a,r}(t) \in \mathbb{R}^{r\times m}$ are the estimates of $\mathbf{x}_p$, $\boldsymbol{\nu}(t)$, $\boldsymbol{\Theta}_\nu$ and $\boldsymbol{\Lambda}_{a,r}$ respectively. Given the reparameterized plant (4.3) and its state predictor (4.7), model reference tracking is achieved in two steps which are typical for indirect approaches. The first one consists in deriving a control law $\mathbf{u}(t)$ for the completely known predictor model (4.7) such that it behaves like the reference model (3.2). The second step is the identification of the unknown plant parameters $\boldsymbol{\Lambda}_{a,r}$ and $\boldsymbol{\Theta}_\nu$ so that the control law $\mathbf{u}(t)$ is also valid for the plant dynamics (4.3).

Because of the separation between rigid body $(4.3a)$ and actuator dynamics $(4.3b)$, the control law consists of two steps. The first step, is the selection of the desired virtual control $\boldsymbol{\nu}_d(t)$ such that for $\boldsymbol{\nu}(t) = \boldsymbol{\nu}_d(t)$ the dynamics $(4.7a)$ replicate the reference model (3.2). The virtual control law is then given by

$$\boldsymbol{\nu}_d(t) = \mathbf{K}_r \mathbf{r}(t) - \hat{\boldsymbol{\Theta}}_\nu(t)\,\boldsymbol{\omega}(t), \tag{4.8}$$

where $\mathbf{K}_r \in \mathbb{R}^{r\times r}$ is chosen such that $\mathbf{B}_m = \mathbf{B}_\nu \mathbf{K}_r$. Note that analogously to the unknown parameters $\boldsymbol{\Theta}_\nu$, the feed-forward gain $\mathbf{K}_r$ is uniquely defined using this parameterization. The second step is the control allocation task, which is formulated in the following assumption.

Assumption 4.3 (Control Allocation) *Given the desired virtual control input* $\boldsymbol{\nu}_d(t)$ *and the known and time varying control effectiveness matrix* $\hat{\boldsymbol{\Lambda}}_{a,r}(t)$ *compute the control inputs* $\mathbf{u}\left(\hat{\boldsymbol{\Lambda}}_{a,r}, \boldsymbol{\nu}_d\right)$ *such that*

$$\boldsymbol{\nu}_d(t) = \mathbf{U}_{a,r}\mathbf{D}_a \hat{\boldsymbol{\Lambda}}_{a,r}(t)\,\mathbf{u}(t) \tag{4.9}$$

and such that the input lies within the input set $\mathcal{U} \subset \mathbb{R}^m$ *defined as*

$$\mathcal{U} := \{\mathbf{u} \in \mathbb{R}^m \mid u_{i,min} \le u_i \le u_{i,max}, \forall i = 1...m\},$$

where u_i *is the* i*-th element of* $\mathbf{u}(t)$ *and* $u_{i,min}, u_{i,max} \in \mathbb{R}$ *are the corresponding minimum and maximum values. The control input set* $\mathcal{U}$ *is an* m*-dimensional hyperrectangle.*

Provided that Assumption 4.3 holds, inserting the control law (4.8) into the predictor dynamics $(4.7a)$ leads to the closed-loop predictor dynamics which are equal to the reference model dynamics (3.2)

$$\dot{\hat{\mathbf{x}}}_p(t) = \mathbf{A}_m \hat{\mathbf{x}}_p(t) + \mathbf{B}_\nu \mathbf{K}_r \mathbf{r}(t) = \mathbf{A}_m \hat{\mathbf{x}}_p(t) + \mathbf{B}_m \mathbf{r}(t). \tag{4.10}$$

The next step is to choose the update laws of $\hat{\boldsymbol{\Lambda}}_{a,r}(t)$ and $\hat{\boldsymbol{\Theta}}_\nu(t)$ such that the dynamics of the predictor estimation error $\mathbf{e}_p(t) = \hat{\mathbf{x}}_p(t) - \mathbf{x}_p(t)$ tend to zero and the

parameter errors $\widetilde{\mathbf{\Lambda}}_{a,r}(t) = \hat{\mathbf{\Lambda}}_{a,r}(t) - \mathbf{\Lambda}_{a,r}$ and $\widetilde{\mathbf{\Theta}}_{\nu}(t) = \hat{\mathbf{\Theta}}_{\nu}(t) - \mathbf{\Theta}_{\nu}$ have a stable zero equilibrium. The predictor error dynamics are computed from the plant dynamics (4.3) and the predictor dynamics (4.7) as follows

$$\dot{\mathbf{e}}_p(t) = \mathbf{A}_m \mathbf{e}_p(t) + \mathbf{B}_{\nu}\widetilde{\boldsymbol{\nu}}(t) + \mathbf{B}_{\nu}\widetilde{\mathbf{\Theta}}_{\nu}(t)\,\boldsymbol{\omega}(t), \tag{4.11a}$$

$$\widetilde{\boldsymbol{\nu}}(t) = \mathbf{U}_{a,r}\mathbf{D}_a\widetilde{\mathbf{\Lambda}}_{a,r}(t)\,\mathbf{u}(t), \tag{4.11b}$$

In the following the time dependency is not written out for readability. The Lyapunov candidate function is given by

$$V\left(\mathbf{e}_p, \widetilde{\mathbf{\Theta}}_{\nu}, \widetilde{\mathbf{\Lambda}}_{a,r}\right) = \frac{1}{2}\mathbf{e}_p^T\mathbf{P}\mathbf{e}_p + \frac{1}{2}tr\left(\widetilde{\mathbf{\Theta}}_{\nu}\mathbf{\Gamma}_{\theta}^{-1}\widetilde{\mathbf{\Theta}}_{\nu}^T\right) + \frac{1}{2}tr\left(\widetilde{\mathbf{\Lambda}}_{a,r}\mathbf{\Gamma}_{\Lambda}^{-1}\widetilde{\mathbf{\Lambda}}_{a,r}^T\right), \tag{4.12}$$

with the symmetric positive definite matrix $\mathbf{P}$ corresponding to the Lyapunov equation (3.3). From Lemma B.8 (Positive Definite Trace) it follows that the Lyapunov function is positive definite in the case of symmetric positive definite matrices $\mathbf{\Gamma}_{\theta} \in \mathbb{R}^{q\times q}$ and $\mathbf{\Gamma}_{\Lambda} \in \mathbb{R}^{m\times m}$, which correspond to the adaptation rates. The derivative of the Lyapunov function is calculated using the predictor error dynamics (4.11), the Lyapunov equation (3.3) and Lemma B.2 (Trace and Scalar Product)

$$\begin{aligned}
\dot{V} &= -\frac{1}{2}\mathbf{e}_p^T\mathbf{Q}\mathbf{e}_p + \mathbf{e}_p^T\mathbf{P}\mathbf{B}_{\nu}\left(\mathbf{U}_{a,r}\mathbf{D}_a\widetilde{\mathbf{\Lambda}}_{a,r}\mathbf{u} + \widetilde{\mathbf{\Theta}}_{\nu}\boldsymbol{\omega}\right) \\
&\quad + tr\left(\widetilde{\mathbf{\Theta}}_{\nu}\mathbf{\Gamma}_{\theta}^{-1}\dot{\widetilde{\mathbf{\Theta}}}_{\nu}^T\right) + tr\left(\widetilde{\mathbf{\Lambda}}_{a,r}\mathbf{\Gamma}_{\Lambda}^{-1}\dot{\widetilde{\mathbf{\Lambda}}}_{a,r}^T\right), \\
&= -\frac{1}{2}\mathbf{e}_p^T\mathbf{Q}\mathbf{e}_p + tr\left(\widetilde{\mathbf{\Theta}}_{\nu}\left(\boldsymbol{\omega}\mathbf{e}_p^T\mathbf{P}\mathbf{B}_{\nu} + \mathbf{\Gamma}_{\theta}^{-1}\dot{\widetilde{\mathbf{\Theta}}}_{\nu}^T\right)\right) \\
&\quad + tr\left(\widetilde{\mathbf{\Lambda}}_{a,r}\left(\mathbf{u}\mathbf{e}_p^T\mathbf{P}\mathbf{B}_{\nu}\mathbf{U}_{a,r}\mathbf{D}_a + \mathbf{\Gamma}_{\Lambda}^{-1}\dot{\widetilde{\mathbf{\Lambda}}}_{a,r}^T\right)\right).
\end{aligned}$$

The PMRAC update laws are then selected as

$$\begin{aligned}
\dot{\hat{\mathbf{\Theta}}}_{\nu}(t) &= -\left(\mathbf{\Gamma}_{\theta}\boldsymbol{\omega}(t)\,\mathbf{e}_p^T(t)\,\mathbf{P}\mathbf{B}_{\nu}\right)^T &&= -\mathbf{B}_{\nu}^T\mathbf{P}\mathbf{e}_p(t)\,\boldsymbol{\omega}^T(t)\,\mathbf{\Gamma}_{\theta}, \\
\dot{\hat{\mathbf{\Lambda}}}_{a,r}(t) &= -\left(\mathbf{\Gamma}_{\Lambda}\mathbf{u}(t)\,\mathbf{e}_p^T(t)\,\mathbf{P}\mathbf{B}_{\nu}\mathbf{U}_{a,r}\mathbf{D}_a\right)^T &&= -\mathbf{D}_a\mathbf{U}_{a,r}^T\mathbf{B}_{\nu}^T\mathbf{P}\mathbf{e}_p(t)\,\mathbf{u}^T(t)\,\mathbf{\Gamma}_{\Lambda},
\end{aligned} \tag{4.13}$$

such that the derivative of the Lyapunov function $\dot{V}(t) = -\frac{1}{2}\mathbf{e}_p^T(t)\,\mathbf{Q}\mathbf{e}_p(t)$ is negative semi-definite. From Theorem 4.8 (Uniform Stability) in [87], it follows that the equilibrium $(\mathbf{e}_p, \widetilde{\mathbf{\Theta}}_{\nu}, \widetilde{\mathbf{\Lambda}}_{a,r}) = (\mathbf{0}, \mathbf{0}, \mathbf{0})$ is uniformly stable and $\mathbf{e}_p(t)$, $\widetilde{\mathbf{\Theta}}_{\nu}(t)$, $\widetilde{\mathbf{\Lambda}}_{a,r}(t) \in \mathcal{L}_{\infty}$. Therefore $\hat{\mathbf{\Theta}}_{\nu}(t)$ and $\hat{\mathbf{\Lambda}}_{a,r}(t)$ are bounded.

Subsequently, the goal is to show asymptotic convergence of the estimation error $\mathbf{e}_p(t)$ to zero based on Barbalat's Lemma (Lemma B.9). Following the same arguments as in Section 3.2, it is known that $\lim_{t\to\infty}\int_{\tau=0}^{t}\dot{V}(\mathbf{e}_p(\tau))\,d\tau$ exists and is finite. Therefore, it remains to show that $\dot{V}(\mathbf{e}_p(t))$ is a uniform continuous function of t. Given that the input $\mathbf{r}(t)$ is bounded and from the predictor dynamics (4.10) it holds that $\hat{\mathbf{x}}_p(t) \in \mathcal{L}_{\infty}$ and hence $\mathbf{x}_p(t) \in \mathcal{L}_{\infty}$. This implies boundedness of $\boldsymbol{\phi}(\mathbf{x}_p(t))$ and $\boldsymbol{\omega}(\mathbf{x}_p(t), \boldsymbol{\phi}(\mathbf{x}_p(t)))$. It follows that the desired virtual control $\boldsymbol{\nu}_d(t)$ (4.8) and hence the control law $\mathbf{u}(t)$ (4.9) are bounded. The estimation error derivative $\dot{\mathbf{e}}_p(t)$ (4.11) is then bounded uniformly in t for all $t \geq 0$. Therefore, $\ddot{V}(t) = -\mathbf{e}_p^T(t)\,\mathbf{Q}\dot{\mathbf{e}}_p(t)$ is also bounded uniformly in t for all

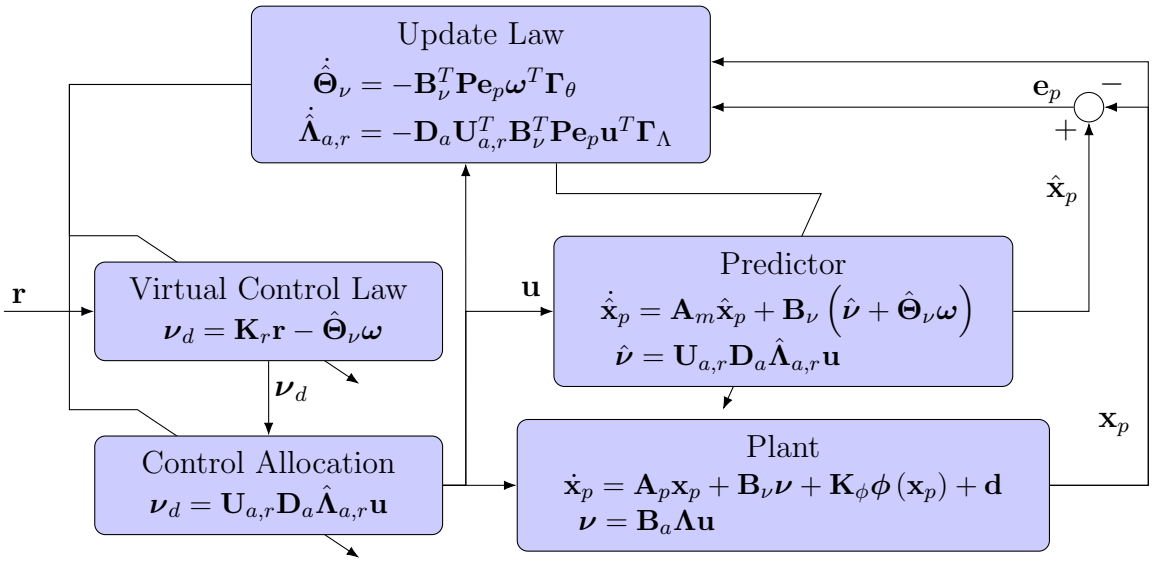


Figure 4.1: Controller Structure - Predictor-Based Adaptive Control Allocation

$t \geq 0$. Then, $\dot{V}(\mathbf{e}_p(t))$ is uniformly continuous for $t \geq 0$. Using Lemma B.9 (Barbalat) it follows that $\dot{V}(\mathbf{e}_p(t)) \to 0$ and hence $\mathbf{e}_p(t) \to \mathbf{0}$ for $t \to \infty$. Finally, because of the closed-loop predictor dynamics (4.10), the tracking error $\mathbf{e}_c(t) = \mathbf{x}_p(t) - \mathbf{x}_m(t) = \mathbf{x}_p(t) - \hat{\mathbf{x}}_p(t) = -\mathbf{e}_p(t) \to \mathbf{0}$ for $t \to \infty$. The presented result is summarized in the next theorem.

Theorem 4.4 (P-ACA) *Consider the plant dynamics* (4.1)*, the reference model* (3.2)*, the state predictor* (4.7)*, the control law* (4.8) *and the update laws* (4.13)*. Furthermore let Assumptions 4.1 (Matching Condition - P-ACA) and 4.3 (Control Allocation) hold. Then, the equilibrium* $(\mathbf{e}_p, \widetilde{\boldsymbol{\Theta}}_\nu, \widetilde{\boldsymbol{\Lambda}}_{a,r}) = (\mathbf{0}, \mathbf{0}, \mathbf{0})$ *is uniformly stable and* $\mathbf{e}_p(t), \mathbf{e}_c(t) \to \mathbf{0}$ *for* $t \to \infty$*.* ■

Theorem 4.4 formulates a possible solution of the Control Task 3.1 (Model Reference Adaptive Control) and separates the design of the virtual control law and the control allocation. The virtual control law is given by (4.8). The control allocation task has been reformulated as Assumption 4.3 (Control Allocation) and replaces the fault tolerant control authority conditions discussed in Theorem 3.22. This assumption is slightly different from Control Task 4.1 (Control Allocation) as formulated in the beginning of this chapter. since the matrix $\mathbf{U}_{a,r} \mathbf{D}_a \hat{\boldsymbol{\Lambda}}_{a,r}(t) \in \mathbb{R}^{r \times m}$ is known and time varying. This is the main step into combining PMRAC with control allocation approaches such that input saturations can be taken into account. The controller structure is depicted in Figure 4.1. In the following, an example illustrates the application of the presented approach.

Example 4.1 (Application of Predictor-Based Adaptive Control Allocation) *For illustrating the results in this section, consider the plant* (3.6) *given in Example 3.1. Subsequently, the input matrix* $\mathbf{B}_p$ *is separated into* $\mathbf{B}_p = \mathbf{B}_\nu \mathbf{B}_a$ *such that* $\mathbf{B}_\nu \in \mathbb{R}^{2 \times r}$ *and* $\mathbf{B}_a \in \mathbb{R}^{r \times 2}$ *with* $r = \operatorname{rank}(\mathbf{B}_p) = 1$*. In this way the virtual control* $\nu(t) \in \mathbb{R}$ *has the minimal dimension and*

the plant (3.6) *can be written in the form* (4.1) *as*

$$\underbrace{\begin{pmatrix}\dot{p}(t)\\ \dot{\Phi}(t)\end{pmatrix}}_{\dot{\mathbf{x}}_p} = \underbrace{\begin{bmatrix}L_p & 0\\ 1 & 0\end{bmatrix}}_{\mathbf{A}_p}\underbrace{\begin{pmatrix}p(t)\\ \Phi(t)\end{pmatrix}}_{\mathbf{x}_p} + \underbrace{\begin{bmatrix}L_\xi\\ 0\end{bmatrix}}_{\mathbf{B}_\nu}\nu\left(t\right),$$
$$\nu\left(t\right) = \underbrace{\begin{bmatrix}1 & 1\end{bmatrix}}_{\mathbf{B}_a}\underbrace{\begin{bmatrix}\lambda_1 & 0\\ 0 & \lambda_2\end{bmatrix}}_{\mathbf{\Lambda}}\underbrace{\begin{pmatrix}\xi_1(t)\\ \xi_2(t)\end{pmatrix}}_{\mathbf{u}}.$$

Subsequently, with the regressor vector $\boldsymbol{\omega}\left(t\right) = \mathbf{x}_p\left(t\right) \in \mathbb{R}^2$ *and the unknown parameter* $\mathbf{\Theta}_\nu \in \mathbb{R}^{1\times 2}$*, the plant is parameterized as in* (4.3) *as follows*

$$\begin{pmatrix}\dot{p}(t)\\ \dot{\Phi}(t)\end{pmatrix} = \mathbf{A}_m\begin{pmatrix}p(t)\\ \Phi(t)\end{pmatrix} + \begin{bmatrix}L_\xi\\ 0\end{bmatrix}\left(\nu\left(t\right) + \mathbf{\Theta}_\nu\mathbf{x}_p\left(t\right)\right),$$
$$\nu\left(t\right) = \sqrt{2}\mathbf{\Lambda}_{a,r}\mathbf{u}\left(t\right).$$

Here, the control efficiency matrix has been reduced to $\mathbf{\Lambda}_{a,r} = \mathbf{V}_{a,r}^T\mathbf{\Lambda}$*. The matrix* $\mathbf{V}_{a,r}$ *is defined by the SVD of the matrix* $\mathbf{B}_a$ *given as*

$$\underbrace{\begin{bmatrix}1 & 1\end{bmatrix}}_{\mathbf{B}_a} = \underbrace{1}_{\mathbf{U}_{a,r}}\underbrace{\sqrt{2}}_{\mathbf{D}_a}\underbrace{\begin{bmatrix}\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2}\end{bmatrix}}_{\mathbf{V}_{a,r}^T}.$$

The corresponding state predictor is then given as

$$\begin{pmatrix}\dot{\hat{p}}(t)\\ \dot{\hat{\Phi}}(t)\end{pmatrix} = \mathbf{A}_m\begin{pmatrix}\hat{p}(t)\\ \hat{\Phi}(t)\end{pmatrix} + \begin{bmatrix}L_\xi\\ 0\end{bmatrix}\left(\hat{\nu}\left(t\right) + \hat{\mathbf{\Theta}}_\nu\left(t\right)\mathbf{x}_p\left(t\right)\right),$$
$$\hat{\nu}\left(t\right) = \sqrt{2}\hat{\mathbf{\Lambda}}_{a,r}\left(t\right)\mathbf{u}\left(t\right).$$

with the corresponding virtual control law (4.8)

$$\nu_d\left(t\right) = \underbrace{\frac{kp}{L_\xi}}_{\mathbf{K}_r}r\left(t\right) - \hat{\mathbf{\Theta}}_\nu\left(t\right)\mathbf{x}_p\left(t\right).$$

The corresponding reduced update laws (4.13) *are given by*

$$\dot{\hat{\mathbf{\Theta}}}_\nu\left(t\right) = -\begin{bmatrix}L_\xi & 0\end{bmatrix}\mathbf{P}\mathbf{e}_p\left(t\right)\mathbf{x}_p^T\left(t\right)\mathbf{\Gamma}_\theta,$$
$$\dot{\hat{\mathbf{\Lambda}}}_{a,r}\left(t\right) = -\sqrt{2}\begin{bmatrix}L_\xi & 0\end{bmatrix}\mathbf{P}\mathbf{e}_p\left(t\right)\mathbf{u}^T\left(t\right)\mathbf{\Gamma}_\Lambda.$$

Finally assuming that Assumption 4.3 (Control Allocation) holds, Theorem 4.4 (P-ACA) can be used to proof stability of the parameter and tracking errors and asymptotic convergence of the tracking error to zero. Interesting to see is that the parameter $\mathbf{\Theta}_\nu \in \mathbb{R}^{1\times 2}$ *has the same dimensions as the reduced parameter* $\mathbf{\Theta}_r \in \mathbb{R}^{1\times 2}$ *in Example 3.6 (Application of PMRAC PRO). The step that is missing here is the control allocation design which selects the inputs* $\mathbf{u}\left(t\right)$ *such that* $\nu_d\left(t\right) = \sqrt{2}\hat{\mathbf{\Lambda}}_{a,r}\mathbf{u}\left(t\right)$*, i.e. such that Assumption 4.3 (Control Allocation) holds.*

In order to highlight the main difference with respect to the previous presented controllers and as a road map for the following sections, the design steps of P-ACA are summarized in the following remark.

Remark 4.5 (P-ACA Design Steps)

1. *Definition of the state predictor,*
2. *Design of a control law* $\mathbf{u}(t)$ *for the known predictor dynamics such that the closed-loop meets the design goal. In P-ACA this step has been separated into two.*
   (a) *Design a virtual control law* $\boldsymbol{\nu}_d(t)$ *for the known rigid-body predictor dynamics such that the closed-loop meets the design goal.*
   (b) *Find a control allocation that maps the desired virtual control* $\boldsymbol{\nu}_d(t)$ *to control input* $\mathbf{u}(t)$.
3. *Compute the update laws for the adaptive parameters such that the control input* $\mathbf{u}(t)$ *in closed loop with the unknown plant dynamics meets the design goal.*

In P-ACA the separation into rigid body and actuator models leads to the uniqueness of the feed-forward gain $\mathbf{K}_r$ in (4.8) and to the uniqueness of the ideal parameter $\boldsymbol{\Theta}_\nu$ as shown in Remark 4.2. Therefore, the redundancy of the system is fully addressed within the control allocation. Before describing the control allocation, in the next section the implemented predictor model is further simplified as motivated from the thesis objective 6.

4.2 Reduced Order P-ACA

In most mechanical systems, the uncertainties are related to forces and moments. In these cases only the equations describing the dynamics of the system contain unknown variables while the kinematic equations are completely known. This special structure can be exploited to reduce the order of the state predictor within the PMRAC framework leading to an implementation advantage.

Specifically, in this section the considered plant structure is a special case of (4.1) and is given by

$$\begin{pmatrix}\dot{\mathbf{x}}_{p1}(t)\\ \dot{\mathbf{x}}_{p2}(t)\end{pmatrix} = \begin{bmatrix}\mathbf{A}_{p11} & \mathbf{A}_{p12}\\ \mathbf{A}_{p21} & \mathbf{A}_{p22}\end{bmatrix}\begin{pmatrix}\mathbf{x}_{p1}(t)\\ \mathbf{x}_{p2}(t)\end{pmatrix} + \begin{bmatrix}\mathbf{0}\\ \mathbf{B}_{\nu 2}\end{bmatrix}\boldsymbol{\nu}(t) + \begin{bmatrix}\mathbf{0}\\ \mathbf{K}_{\phi 2}\end{bmatrix}\boldsymbol{\phi}(\mathbf{x}_p(t)) + \begin{bmatrix}\mathbf{0}\\ \mathbf{d}_2\end{bmatrix}, \quad (4.14a)$$

$$\boldsymbol{\nu}(t) = \mathbf{B}_a\boldsymbol{\Lambda}\mathbf{u}(t). \quad (4.14b)$$

The plant state is partitioned into the kinematic state variables $\mathbf{x}_{p1}(t) \in \mathbb{R}^{n_1}$ and the dynamic state variables $\mathbf{x}_{p2}(t) \in \mathbb{R}^{n_2}$ with $n = n_1 + n_2$. The dynamic matrix is partitioned into its known submatrices $\mathbf{A}_{p11} \in \mathbb{R}^{n_1 \times n_1}$ and $\mathbf{A}_{p12} \in \mathbb{R}^{n_1 \times n_2}$ and its unknown submatrices $\mathbf{A}_{p21} \in \mathbb{R}^{n_2 \times n_1}$ and $\mathbf{A}_{p22} \in \mathbb{R}^{n_2 \times n_2}$. The virtual controls $\boldsymbol{\nu}(t)$ only affect the derivative $\dot{\mathbf{x}}_{p2}(t)$ directly via the known input submatrix $\mathbf{B}_{\nu 2} \in \mathbb{R}^{n_2 \times m}$. The constant unknown disturbance $\mathbf{d}_2 \in \mathbb{R}^{n_2}$ is also restricted to the dynamic state variables. The term $\mathbf{h}_2(\mathbf{x}_p) = \mathbf{K}_{\phi 2}\boldsymbol{\phi}(\mathbf{x}_p)$ represents a linearly parameterizable nonlinear function $\mathbf{h}_2(\mathbf{x}_p)$, where $\mathbf{K}_{\phi 2} \in \mathbb{R}^{n_2 \times k}$ is an unknown matrix and $\boldsymbol{\phi}(\mathbf{x}_p) : \mathbb{R}^n \to \mathbb{R}^k$ is a known regressor vector composed of k basis functions $\phi_i(\mathbf{x}_p)$ which are locally Lipschitz-continuous in $\mathbf{x}_p(t)$ [92]. The important characteristic of system (4.14) is that the kinematic differential equation $\dot{\mathbf{x}}_{p1}(t)$ is completely known and that the uncertainties and control inputs affect only the differential equation of the dynamic state variables $\dot{\mathbf{x}}_{p2}(t)$.

Example 4.2 (State Partition) *Example 4.1 has the structure of the system* (4.14)*, where* $x_{p1}(t) = \Phi(t)$ *is the kinematic state and* $x_{p2}(t) = p(t)$ *is the dynamic state. The dynamic matrix partitions are then* $A_{p11} = 0$*,* $A_{p12} = 1$*,* $A_{p21} = 0$ *and* $A_{p22} = L_p$*. The input submatrix is* $\mathbf{B}_{\nu 2} = L_\xi$*.*

In the same way as in (4.3) in Section 4.1, the plant (4.14) is rewritten using the following parameterization

$$\begin{pmatrix} \dot{\mathbf{x}}_{p1}(t) \\ \dot{\mathbf{x}}_{p2}(t) \end{pmatrix} = \begin{bmatrix} \mathbf{A}_{m11} & \mathbf{A}_{m12} \\ \mathbf{A}_{m21} & \mathbf{A}_{m22} \end{bmatrix} \begin{pmatrix} \mathbf{x}_{p1}(t) \\ \mathbf{x}_{p2}(t) \end{pmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{B}_{\nu 2} \end{bmatrix} \boldsymbol{\nu}(t) + \begin{bmatrix} \mathbf{0} \\ \mathbf{B}_{\nu 2} \end{bmatrix} \boldsymbol{\Theta}_\nu \boldsymbol{\omega}(t), \tag{4.15a}$$

$$\boldsymbol{\nu}(t) = \mathbf{U}_{a,r} \mathbf{D}_a \boldsymbol{\Lambda}_{a,r} \mathbf{u}(t). \tag{4.15b}$$

where $\boldsymbol{\omega}(t) \in \mathbb{R}^q$ is the known regressor vector and $\boldsymbol{\Theta}_\nu \in \mathbb{R}^{r \times q}$ is the unknown parameter matrix. The specific parameterization intends to facilitate the derivation of a control law in the case of known parameters. The known regressor vector $\boldsymbol{\omega}(t)$ and the unknown ideal parameter matrix $\boldsymbol{\Theta}_\nu$ are defined as

$$\begin{aligned} \boldsymbol{\omega}(t) &:= \left[\mathbf{x}_p^T(t), \boldsymbol{\phi}^T(\mathbf{x}_p(t)), 1\right]^T \in \mathbb{R}^q, \\ \boldsymbol{\Theta}_\nu &:= \left[\boldsymbol{\Theta}_{p,x}, \boldsymbol{\Theta}_{p,\phi}, \boldsymbol{\Theta}_{p,d}\right] \in \mathbb{R}^{r \times q}, \end{aligned}$$

with $q = n + k + 1$, $\boldsymbol{\Theta}_{p,x} \in \mathbb{R}^{r \times n}$, $\boldsymbol{\Theta}_{p,\phi} \in \mathbb{R}^{r \times k}$ and $\boldsymbol{\Theta}_{p,d} \in \mathbb{R}^r$. This definition is the same as in Section 4.1. The comparison between the parameterized plant (4.15) and its original representation (4.14) leads to the subsequent matching condition.

Assumption 4.6 (Matching Condition - Reduced Order P-ACA) *The ideal parameters* $\boldsymbol{\Theta}_\nu$ *satisfy the following condition*

$$\mathbf{Y} := \left[\mathbf{A}_p - \mathbf{A}_m, \quad \begin{bmatrix} \mathbf{0} \\ \mathbf{K}_{\phi 2} \end{bmatrix}, \quad \begin{bmatrix} \mathbf{0} \\ \mathbf{d}_2 \end{bmatrix}\right] = \begin{bmatrix} \mathbf{0} \\ \mathbf{B}_{\nu 2} \end{bmatrix} \boldsymbol{\Theta}_\nu = \mathbf{B}_\nu \boldsymbol{\Theta}_\nu. \tag{4.16}$$

This matching condition is only a special case of Assumption 4.1 (Matching Condition - P-ACA) where the range of $\mathbf{B}_\nu$ does not span the whole state space. Therefore, Remark 4.2 (Uniqueness of the ideal parameter $\boldsymbol{\Theta}_\nu$) holds is true in this case as well. Directly from (4.16) it follows that the submatrices of the reference model must be chosen such that $\mathbf{A}_{m11} = \mathbf{A}_{p11}$ and $\mathbf{A}_{m12} = \mathbf{A}_{p12}$, i.e. it is necessary that the reference model mimics the known plant kinematics. The reason is that these correspond to a kinematic relation that cannot be changed by the control input.

Given the special structure of the parameterized plant (4.15), it is clear that the kinematic state variables do not need to be considered in the state predictor since no unknown parameters influence their derivatives. Hence, a reduced order state predictor can be formulated as

$$\dot{\hat{\mathbf{x}}}_{p2}(t) = \begin{bmatrix} \mathbf{A}_{m21} & \mathbf{A}_{m22} \end{bmatrix} \begin{pmatrix} \mathbf{x}_{p1}(t) \\ \hat{\mathbf{x}}_{p2}(t) \end{pmatrix} + \mathbf{B}_{\nu 2} \hat{\boldsymbol{\nu}}(t) + \mathbf{B}_{\nu 2} \hat{\boldsymbol{\Theta}}_\nu(t) \boldsymbol{\omega}(t) - \mathbf{L} \mathbf{e}_{p2}(t), \tag{4.17a}$$

$$\hat{\boldsymbol{\nu}}(t) = \mathbf{U}_{a,r} \mathbf{D}_a \hat{\boldsymbol{\Lambda}}_{a,r}(t) \mathbf{u}(t), \tag{4.17b}$$

where $\hat{\mathbf{x}}_{p2}(t) \in \mathbb{R}^{n_2}$, $\hat{\boldsymbol{\nu}}(t) \in \mathbb{R}^r$, $\hat{\boldsymbol{\Theta}}_\nu(t) \in \mathbb{R}^{r \times q}$ and $\hat{\boldsymbol{\Lambda}}_{a,r}(t) \in \mathbb{R}^{r \times m}$ are the estimates of $\mathbf{x}_{p2}(t)$, $\boldsymbol{\nu}(t)$, $\boldsymbol{\Theta}_\nu$ and $\boldsymbol{\Lambda}_{a,r}$ respectively. The identification error is defined as $\mathbf{e}_{p2}(t) = \hat{\mathbf{x}}_{p2}(t) - \mathbf{x}_{p2}(t)$. Differently from the full state predictors (3.29), (3.41) and (4.7), an

extra identification feedback term $-\mathbf{L}\mathbf{e}_{p2}$ with the positive definite Luenberger gain $\mathbf{L} \in \mathbb{R}^{n_2 \times n_2}$ has been added. This term is used for stabilizing the identification error dynamics.

In Predictor-Based Adaptive Control Allocation (P-ACA) as presented Remark 4.5, model reference tracking is achieved in 4 steps . The first step is the definition of the state predictor (4.17). The second step is the selection of the desired virtual control $\boldsymbol{\nu}_d(t)$ such that for $\boldsymbol{\nu}(t) = \boldsymbol{\nu}_d(t)$ the known predictor model (4.17) replicates the reference model (3.2). The third step is the control allocation task, which is formulated in Assumption 4.3. The fourth step is the identification of the unknown plant parameters $\boldsymbol{\Lambda}_{a,r}$ and $\boldsymbol{\Theta}_\nu$ so that the control law $\mathbf{u}(t)$ is also valid for the plant dynamics (4.14). For the reduced order P-ACA, the second step needs to be slightly adapted since the predictor model (4.17) can not fully represent the reference model. Therefore, the estimated state $\hat{\mathbf{x}}_{p2}(t)$ is extended by the plant state $\mathbf{x}_{p1}(t)$ to build the full predictor state

$$\hat{\boldsymbol{\chi}}(t) := \begin{pmatrix} \mathbf{x}_{p1}(t) \\ \hat{\mathbf{x}}_{p2}(t) \end{pmatrix} \tag{4.18}$$

with the following derivative

$$\begin{aligned} \begin{pmatrix} \dot{\mathbf{x}}_{p1}(t) \\ \dot{\hat{\mathbf{x}}}_{p2}(t) \end{pmatrix} &= \begin{bmatrix} \mathbf{A}_{m11} & \mathbf{A}_{m12} \\ \mathbf{A}_{m21} & \mathbf{A}_{m22} \end{bmatrix} \begin{pmatrix} \mathbf{x}_{p1}(t) \\ \hat{\mathbf{x}}_{p2}(t) \end{pmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{B}_{\nu 2} \end{bmatrix} \left(\hat{\boldsymbol{\nu}}(t) + \hat{\boldsymbol{\Theta}}_\nu(t)\, \boldsymbol{\omega}(t) \right) \\ &\quad - \begin{bmatrix} \mathbf{A}_{m12} \\ \mathbf{L} \end{bmatrix} \mathbf{e}_{p2}(t), \end{aligned} \tag{4.19a}$$

$$\hat{\boldsymbol{\nu}}(t) = \mathbf{U}_{a,r} \mathbf{D}_a \hat{\boldsymbol{\Lambda}}_{a,r}(t)\, \mathbf{u}(t), \tag{4.19b}$$

The term $-\mathbf{A}_{m12} \cdot \mathbf{e}_p(t)$ arises from the fact that the full state predictor is a mixture between plant and predictor states. In the given formulation the last term can be seen as a disturbance term coming from the estimation error $\mathbf{e}_{p2}(t)$. Interestingly, the control input cannot compensate for this term since it does not fully lie in the range of $\mathbf{B}_\nu$. Therefore, the best that can be expected from the control law $\mathbf{u}(t)$ are the following closed-loop predictor dynamics

$$\dot{\hat{\boldsymbol{\chi}}}(t) = \mathbf{A}_m \hat{\boldsymbol{\chi}}(t) + \mathbf{B}_m \mathbf{r}(t) + \mathbf{H}\mathbf{e}_{p2}(t), \tag{4.20}$$

with $\mathbf{H} \in \mathbb{R}^{n \times n_2}$. This corresponds to the reference model dynamics (3.2) plus an identification disturbance term. Using the P-ACA virtual control law (4.8), the closed-loop predictor dynamics are given by (4.20) with

$$\mathbf{H} = -\begin{bmatrix} \mathbf{A}_{m12} \\ \mathbf{L} \end{bmatrix}.$$

Since $\mathbf{A}_m$ is a Hurwitz matrix, it is clear that the closed-loop predictor dynamics (4.20) are Input-to-State Stable (ISS) with respect to the inputs $\mathbf{r}(t)$ and $\mathbf{e}_{p2}(t)$ [87, Definition 4.7]. One implication is that the state $\hat{\boldsymbol{\chi}}(t)$ is bounded if the inputs $\mathbf{r}(t)$ and $\mathbf{e}_{p2}(t)$ are bounded. Differently from the previous results, this cannot be directly stated since boundedness of $\mathbf{e}_{p2}(t)$ needs to be shown first.

The next step is to choose the update laws of $\hat{\boldsymbol{\Lambda}}_{a,r}(t)$ and $\hat{\boldsymbol{\Theta}}_\nu(t)$ such that the dynamics of the predictor estimation error $\mathbf{e}_{p2}(t) = \hat{\mathbf{x}}_{p2}(t) - \mathbf{x}_{p2}(t)$ tend to zero and the

parameter errors $\widetilde{\mathbf{\Lambda}}_{a,r}(t) = \hat{\mathbf{\Lambda}}_{a,r}(t) - \mathbf{\Lambda}_{a,r}$ and $\widetilde{\mathbf{\Theta}}_{\nu}(t) = \hat{\mathbf{\Theta}}_{\nu}(t) - \mathbf{\Theta}_{\nu}$ have a stable equilibrium. The predictor error dynamics are computed from the plant dynamics (4.15) and the predictor dynamics (4.17) as follows

$$\dot{\mathbf{e}}_{p2}(t) = (\mathbf{A}_{m22} - \mathbf{L})\,\mathbf{e}_{p2}(t) + \mathbf{B}_{\nu 2}\widetilde{\boldsymbol{\nu}}(t) + \mathbf{B}_{\nu 2}\widetilde{\mathbf{\Theta}}_{\nu}(t)\,\boldsymbol{\omega}(t), \tag{4.21a}$$

$$\widetilde{\boldsymbol{\nu}}(t) = \mathbf{U}_{a,r}\mathbf{D}_a\widetilde{\mathbf{\Lambda}}_{a,r}(t)\,\mathbf{u}(t), \tag{4.21b}$$

The Luenberger gain $\mathbf{L}$ is chosen such that the dynamic matrix $\mathbf{A}_{m22} - \mathbf{L}$ is Hurwitz. Therefore, there exists a symmetric positive definite matrix $\mathbf{P}_L \in \mathbb{R}^{n_2 \times n_2}$ for a every symmetric positive definite matrix $\mathbf{Q}_L \in \mathbb{R}^{n_2 \times n_2}$ such that the Lyapunov equation holds [87, Th. 4.6]

$$\mathbf{P}_L(\mathbf{A}_{m22} - \mathbf{L}) + (\mathbf{A}_{m22} - \mathbf{L})^T\mathbf{P}_L = -\mathbf{Q}_L. \tag{4.22}$$

In the following the time dependency is not written out for readability. Using this symmetric positive definite matrix $\mathbf{P}_L$, the Lyapunov candidate function is given by

$$V\left(\mathbf{e}_{p2}, \widetilde{\mathbf{\Theta}}_{\nu}, \widetilde{\mathbf{\Lambda}}_{a,r}\right) = \frac{1}{2}\mathbf{e}_{p2}^T\mathbf{P}_L\mathbf{e}_{p2} + \frac{1}{2}tr\left(\widetilde{\mathbf{\Theta}}_{\nu}\mathbf{\Gamma}_{\theta}^{-1}\widetilde{\mathbf{\Theta}}_{\nu}^T\right) + \frac{1}{2}tr\left(\widetilde{\mathbf{\Lambda}}_{a,r}\mathbf{\Gamma}_{\Lambda}^{-1}\widetilde{\mathbf{\Lambda}}_{a,r}^T\right),$$

From Lemma B.8 (Positive Definite Trace) it follows that the Lyapunov function is positive definite in the case of symmetric positive definite matrices $\mathbf{\Gamma}_{\theta} \in \mathbb{R}^{q \times q}$ and $\mathbf{\Gamma}_{\Lambda} \in \mathbb{R}^{m \times m}$, which correspond to the adaptation rates. The derivative of the Lyapunov function is calculated using the predictor error dynamics (4.21), the Lyapunov equation (4.22) and Lemma B.2 (Trace and Scalar Product) as

$$\begin{aligned}
\dot{V} &= -\frac{1}{2}\mathbf{e}_{p2}^T\mathbf{Q}_L\mathbf{e}_{p2} + \mathbf{e}_{p2}^T\mathbf{P}_L\mathbf{B}_{\nu 2}\left(\mathbf{U}_{a,r}\mathbf{D}_a\widetilde{\mathbf{\Lambda}}_{a,r}\mathbf{u} + \widetilde{\mathbf{\Theta}}_{\nu}\boldsymbol{\omega}\right) \\
&\quad + tr\left(\widetilde{\mathbf{\Theta}}_{\nu}\mathbf{\Gamma}_{\theta}^{-1}\dot{\widetilde{\mathbf{\Theta}}}_{\nu}^T\right) + tr\left(\widetilde{\mathbf{\Lambda}}_{a,r}\mathbf{\Gamma}_{\Lambda}^{-1}\dot{\widetilde{\mathbf{\Lambda}}}_{a,r}^T\right), \\
&= -\frac{1}{2}\mathbf{e}_{p2}^T\mathbf{Q}\mathbf{e}_{p2} + tr\left(\widetilde{\mathbf{\Theta}}_{\nu}\left(\boldsymbol{\omega}\mathbf{e}_{p2}^T\mathbf{P}_L\mathbf{B}_{\nu 2} + \mathbf{\Gamma}_{\theta}^{-1}\dot{\widetilde{\mathbf{\Theta}}}_{\nu}^T\right)\right) \\
&\quad + tr\left(\widetilde{\mathbf{\Lambda}}_{a,r}\left(\mathbf{u}\mathbf{e}_{p2}^T\mathbf{P}_L\mathbf{B}_{\nu 2}\mathbf{U}_{a,r}\mathbf{D}_a + \mathbf{\Gamma}_{\Lambda}^{-1}\dot{\widetilde{\mathbf{\Lambda}}}_{a,r}^T\right)\right).
\end{aligned}$$

The reduced-order P-ACA update laws are then chosen as

$$\begin{aligned}
\dot{\hat{\mathbf{\Theta}}}_{\nu}(t) &= -\left(\mathbf{\Gamma}_{\theta}\boldsymbol{\omega}(t)\,\mathbf{e}_{p2}^T(t)\,\mathbf{P}_L\mathbf{B}_{\nu 2}\right)^T &&= -\mathbf{B}_{\nu 2}^T\mathbf{P}_L\mathbf{e}_{p2}(t)\,\boldsymbol{\omega}^T(t)\,\mathbf{\Gamma}_{\theta}, \\
\dot{\hat{\mathbf{\Lambda}}}_{a,r}(t) &= -\left(\mathbf{\Gamma}_{\Lambda}\mathbf{u}(t)\,\mathbf{e}_{p2}^T(t)\,\mathbf{P}_L\mathbf{B}_{\nu 2}\mathbf{U}_{a,r}\mathbf{D}_a\right)^T &&= -\mathbf{D}_a\mathbf{U}_{a,r}^T\mathbf{B}_{\nu 2}^T\mathbf{P}_L\mathbf{e}_{p2}(t)\,\mathbf{u}^T(t)\,\mathbf{\Gamma}_{\Lambda},
\end{aligned} \tag{4.23}$$

such that the derivative of the Lyapunov function $\dot{V}(t) = -\frac{1}{2}\mathbf{e}_{p2}^T(t)\,\mathbf{Q}_L\mathbf{e}_{p2}(t)$ is negative semi-definite. From Theorem 4.8 (Uniform Stability) in [87], it follows that the equilibrium $(\mathbf{e}_{p2}, \widetilde{\mathbf{\Theta}}_{\nu}, \widetilde{\mathbf{\Lambda}}_{a,r}) = (\mathbf{0}, \mathbf{0}, \mathbf{0})$ is uniformly stable and $\mathbf{e}_{p2}(t)$, $\widetilde{\mathbf{\Theta}}_{\nu}(t)$, $\widetilde{\mathbf{\Lambda}}_{a,r}(t) \in \mathcal{L}_{\infty}$. Therefore $\hat{\mathbf{\Theta}}_{\nu}(t)$ and $\hat{\mathbf{\Lambda}}_{a,r}(t)$ are bounded.

Subsequently, the goal is to show asymptotic convergence of the estimation error $\mathbf{e}_{p2}(t)$ to zero based on Barbalat's Lemma (Lemma B.9). Following the same arguments as in Section 3.2, it is known that $\lim_{t\to\infty}\int_{\tau=0}^{t}\dot{V}(\mathbf{e}_{p2}(\tau))\,d\tau$ exists and is finite. Therefore, it remains to show that $\dot{V}(\mathbf{e}_{p2}(t))$ is a uniform continuous function of t. From the closed-loop predictor dynamics (4.20), it follows that $\hat{\boldsymbol{\chi}}(t)$ is bounded given that

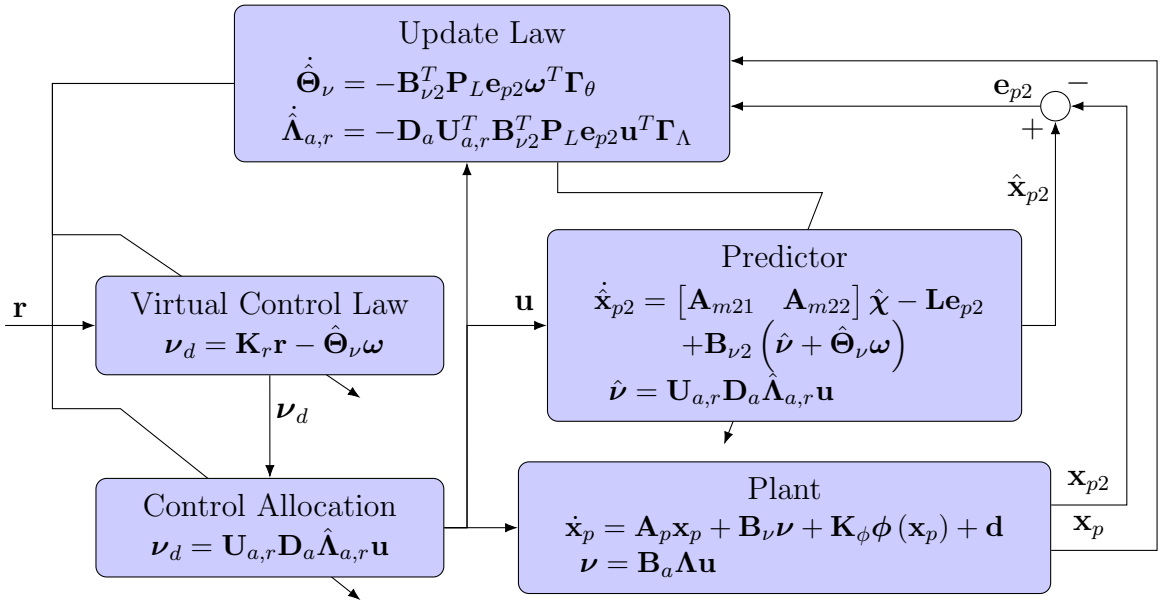


Figure 4.2: Controller Structure - Reduced Order P-ACA

the two inputs $\mathbf{r}(t)$ and $\mathbf{e}_{p2}(t)$ are bounded. Hence, $\mathbf{x}_p(t)$ is bounded, which further implies that $\boldsymbol{\phi}(\mathbf{x}_p(t))$ and the regressor vector $\boldsymbol{\omega}(\mathbf{x}_p(t), \boldsymbol{\phi}(\mathbf{x}_p(t)))$ are bounded. It follows that the desired virtual control $\boldsymbol{\nu}_d(t)$ (4.8) and hence the control law $\mathbf{u}(t)$ (4.9) are bounded. The estimation error derivative $\dot{\mathbf{e}}_{p2}(t)$ (4.21) is bounded uniformly in t for all $t \geq 0$. Therefore, $\ddot{V}(t) = -\mathbf{e}_{p2}^T(t) \mathbf{Q} \dot{\mathbf{e}}_{p2}(t)$ is also bounded uniformly in t for all $t \geq 0$. Then, $\dot{V}(\mathbf{e}_{p2}(t))$ is uniformly continuous for $t \geq 0$. Using Lemma B.9 (Barbalat) it follows that $\dot{V}(\mathbf{e}_{p2}(t)) \to 0$ and hence $\mathbf{e}_{p2}(t) \to \mathbf{0}$ for $t \to \infty$.

By subtracting the reference model dynamics (3.2) from the closed-loop predictor dynamics (4.20) it follows that

$$\dot{\mathbf{x}}_m(t) - \dot{\hat{\boldsymbol{\chi}}}(t) = \mathbf{A}_m(\mathbf{x}_m(t) - \hat{\boldsymbol{\chi}}(t)) - \mathbf{H}\mathbf{e}_{p2}(t).$$

Because $\mathbf{A}_m$ is Hurwitz it can be further concluded that the error $\hat{\boldsymbol{\chi}}(t) - \mathbf{x}_m(t)$ is uniformly bounded and $(\hat{\boldsymbol{\chi}}(t) - \mathbf{x}_m(t)) \to \mathbf{0}$ for $t \to \infty$. Finally, it holds that the tracking error $\mathbf{e}_c(t) = \mathbf{x}_p(t) - \mathbf{x}_m(t) = (\mathbf{x}_p(t) - \hat{\boldsymbol{\chi}}(t)) + (\hat{\boldsymbol{\chi}}(t) - \mathbf{x}_m(t)) \to \mathbf{0}$ for $t \to \infty$. The presented result is summarized in the next theorem and the structure of the controller can be seen in Figure 4.2.

Theorem 4.7 (Reduced Order P-ACA) *Consider the plant dynamics* (4.14)*, the reference model* (3.2)*, the reduced order state predictor* (4.17)*, the virtual control law* (4.8) *and the update laws* (4.23)*. Furthermore let Assumptions 4.6 (Matching Condition - Reduced Order P-ACA) and 4.3 (Control Allocation) hold. Then, the equilibrium* $(\mathbf{e}_{p2}, \tilde{\boldsymbol{\Theta}}_\nu, \tilde{\boldsymbol{\Lambda}}_{a,r}) = (\mathbf{0}, \mathbf{0}, \mathbf{0})$ *is uniformly stable and* $\mathbf{e}_{p2}(t), \mathbf{e}_c(t) \to \mathbf{0}$ *for* $t \to \infty$*.* ■

Example 4.3 (Application of Reduced Order P-ACA) *In order to illustrate the results in this section, consider the plant* (3.6) *given in Example 3.1, insert the virtual control definition*

of Example 4.1 and rearrange the state's order such that

$$\underbrace{\begin{pmatrix}\dot{\Phi}(t)\\ \dot{p}(t)\end{pmatrix}}_{\dot{\mathbf{x}}_p} = \underbrace{\begin{bmatrix}0 & 1\\ 0 & L_p\end{bmatrix}}_{\mathbf{A}_p}\underbrace{\begin{pmatrix}\Phi(t)\\ p(t)\end{pmatrix}}_{\mathbf{x}_p} + \underbrace{\begin{bmatrix}0\\ L_\xi\end{bmatrix}}_{\mathbf{B}_\nu}\nu\left(t\right), \tag{4.24a}$$

$$\nu\left(t\right) = \underbrace{\begin{bmatrix}1 & 1\end{bmatrix}}_{\mathbf{B}_a}\underbrace{\begin{bmatrix}\lambda_1 & 0\\ 0 & \lambda_2\end{bmatrix}}_{\mathbf{\Lambda}}\underbrace{\begin{pmatrix}\xi_1(t)\\ \xi_2(t)\end{pmatrix}}_{\mathbf{u}}. \tag{4.24b}$$

This system has the structure of the system (4.14)*, where* $x_{p1}\left(t\right) = \Phi\left(t\right)$ *is the kinematic state and* $x_{p2}\left(t\right) = p\left(t\right)$ *is the dynamic state. The dynamic matrix partitions are then* $A_{p11} = 0$*,* $A_{p12} = 1$*,* $A_{p21} = 0$ *and* $A_{p22} = L_p$*. The input submatrix is* $B_{\nu 2} = L_\xi$*. The reference model* (3.7) *is rearranged in a similar manner to get*

$$\underbrace{\begin{pmatrix}\dot{\Phi}_m(t)\\ \dot{p}_m(t)\end{pmatrix}}_{\mathbf{x}_m} = \underbrace{\begin{bmatrix}0 & 1\\ -kp & -kd\end{bmatrix}}_{\mathbf{A}_m}\underbrace{\begin{pmatrix}\Phi_m(t)\\ p_m(t)\end{pmatrix}}_{\mathbf{x}_m} + \underbrace{\begin{bmatrix}0\\ kp\end{bmatrix}}_{\mathbf{B}_m}\underbrace{\Phi_r(t)}_{\mathbf{r}}.$$

In order to apply PMRAC, the plant dynamics (4.24) *are reparameterized to the form* (4.15)*. In this example, there is no constant disturbance and no nonlinearity in the plant dynamics. Hence, the regressor vector is* $\boldsymbol{\omega}\left(t\right) = \mathbf{x}_p\left(t\right) \in \mathbb{R}^2$ *and the unknown parameters are* $\mathbf{\Theta}_\nu \in \mathbb{R}^{2\times 2}$*. The plant is then*

$$\begin{aligned}\begin{pmatrix}\dot{\Phi}(t)\\ \dot{p}(t)\end{pmatrix} &= \mathbf{A}_m\begin{pmatrix}\Phi(t)\\ p(t)\end{pmatrix} + \mathbf{B}_\nu\left(\nu\left(t\right) + \mathbf{\Theta}_\nu\mathbf{x}_p\right),\\ \nu\left(t\right) &= \sqrt{2}\mathbf{\Lambda}_{a,r}\mathbf{u}\left(t\right).\end{aligned}$$

Here, the control efficiency matrix has been reduced to $\mathbf{\Lambda}_{a,r} = \mathbf{V}_{a,r}^T\mathbf{\Lambda}$ *as in Example 4.1. The matching condition* (4.16) *translates to*

$$\underbrace{\begin{bmatrix}0 & 0\\ kp & L_p + kd\end{bmatrix}}_{\mathbf{A}_p - \mathbf{A}_m} = \underbrace{\begin{bmatrix}0\\ L_\xi\end{bmatrix}}_{\mathbf{B}_\nu}\mathbf{\Theta}_\nu.$$

The reduced order state predictor (4.17) *is then given as*

$$\begin{aligned}\dot{\hat{p}}(t) &= \begin{bmatrix}-kp & -kd\end{bmatrix}\begin{pmatrix}\Phi(t)\\ \hat{p}(t)\end{pmatrix} + L_\xi\left(\hat{\nu}\left(t\right) + \hat{\mathbf{\Theta}}_\nu\left(t\right)\mathbf{x}_p\left(t\right)\right) - L\cdot e_{p2}\left(t\right),\\ \hat{\nu}\left(t\right) &= \sqrt{2}\hat{\mathbf{\Lambda}}_{a,r}\left(t\right)\mathbf{u}\left(t\right),\end{aligned} \tag{4.25}$$

with the state estimate $\hat{p}\left(t\right) \in \mathbb{R}$*, the parameter estimates* $\hat{\mathbf{\Lambda}}_{a,r}\left(t\right), \hat{\mathbf{\Theta}}_\nu\left(t\right) \in \mathbb{R}^{1\times 2}$*, the identification error* $e_{p2}\left(t\right) = \hat{p}\left(t\right) - p\left(t\right)$ *and the Luenberger gain* L*. Compared to P-ACA as presented in Example 4.1, this predictor has order 1 instead of 2 because it only considers the differential equations which have uncertain parameters. The reduced order predictor is the one that is implemented but for analysis and derivation of the control law, the following full state predictor* (4.19) *is used*

$$\begin{aligned}\underbrace{\begin{pmatrix}\dot{\Phi}(t)\\ \dot{\hat{p}}(t)\end{pmatrix}}_{\dot{\hat{\boldsymbol{x}}}} &= \underbrace{\begin{bmatrix}0 & 1\\ -kp & -kd\end{bmatrix}}_{\mathbf{A}_m}\underbrace{\begin{pmatrix}\Phi(t)\\ \hat{p}(t)\end{pmatrix}}_{\hat{\boldsymbol{x}}} + \mathbf{B}_\nu\left(\hat{\nu}\left(t\right) + \hat{\mathbf{\Theta}}_\nu\left(t\right)\mathbf{x}_p\left(t\right)\right) - \begin{bmatrix}1\\ L\end{bmatrix}e_{p2}\left(t\right),\\ \hat{\nu}\left(t\right) &= \sqrt{2}\hat{\mathbf{\Lambda}}_{a,r}\left(t\right)\mathbf{u}\left(t\right).\end{aligned}$$

The virtual control law (4.8) *is the same as in Example 4.1*

$$\nu_d(t) = \underbrace{\frac{kp}{L_\xi}}_{\mathbf{K}_r} r(t) - \hat{\mathbf{\Theta}}_\nu(t)\,\mathbf{x}_p(t). \tag{4.26}$$

and leads to the following closed-loop predictor dynamics

$$\begin{pmatrix}\dot{\Phi}(t)\\ \dot{\hat{p}}(t)\end{pmatrix} = \begin{bmatrix}0 & 1\\ -kp & -kd\end{bmatrix}\begin{pmatrix}\Phi(t)\\ \hat{p}(t)\end{pmatrix} + \begin{bmatrix}0\\ k_p\end{bmatrix} r - \begin{bmatrix}1\\ L\end{bmatrix} e_{p2}(t),$$

which correspond to the reference model dynamics (3.7) *plus an identification error disturbance term. The update laws* (4.23) *are given by*

$$\begin{aligned}\dot{\hat{\mathbf{\Theta}}}_\nu(t) &= -L_\xi P_L e_{p,2}(t)\,\mathbf{x}_p^T(t)\,\mathbf{\Gamma}_\theta,\\ \dot{\hat{\mathbf{\Lambda}}}_{a,r}(t) &= -\sqrt{2}L_\xi P_L e_{p,2}(t)\,\mathbf{u}^T(t)\,\mathbf{\Gamma}_\Lambda,\end{aligned} \tag{4.27}$$

where $P_L > 0 \in \mathbb{R}$ *satisfies Lyapunov equation*

$$P_L(-kd - L) + (-kd - L)P_L = -Q_L$$

for a positive $Q_L > 0 \in \mathbb{R}$*. For implementation, the predictor* (4.25)*, the control law* (4.26) *and the update laws* (4.27) *are used. Using Theorem 4.7 (Reduced Order P-ACA), it is proven that the equilibrium* $(e_{p2}, \hat{\mathbf{\Theta}}_\nu, \hat{\mathbf{\Lambda}}_{a,r}) = (0, \mathbf{0}, \mathbf{0})$ *is uniformly stable and* $e_{p2}(t), \mathbf{e}_c(t) \to \mathbf{0}$ *for* $t \to \infty$*. The tracking error is defined as* $\mathbf{e}_c(t) = \mathbf{x}_m(t) - \mathbf{x}_p(t)$*. The step that is missing here is the control allocation design which selects the inputs* $\mathbf{u}(t)$ *such that* $\nu_d(t) = \sqrt{2}\hat{\mathbf{\Lambda}}_{a,r}(t)\,\mathbf{u}(t)$*, i.e. such that Assumption 4.3 (Control Allocation) holds.*

In this section Reduced Order P-ACA has been introduced an. The main property of this approach is that the state predictor has a reduced order restricted to the dynamic states, i.e. considering only the differential equations containing uncertain parameters. Compared to P-ACA, the update laws have changed but the virtual control law and the control allocation are the same. Table 4.1 shows a comparison between P-ACA and Reduced Order P-ACA. The main advantage refers to the implemented algorithms at the expense of a more elaborated stability assessment. In the next chapter, control allocation approaches that comply with Assumption 4.3 (Control Allocation) are introduced.

P-ACA		
Plant	$\dot{\mathbf{x}}_p(t) = \mathbf{A}_p\mathbf{x}_p(t) + \mathbf{B}_\nu\boldsymbol{\nu}(t) + \mathbf{K}_\phi\boldsymbol{\phi}(\mathbf{x}_p(t)) + \mathbf{d}$ $\boldsymbol{\nu}(t) = \mathbf{B}_a\boldsymbol{\Lambda}\mathbf{u}(t)$	(4.1)
Reference Model	$\dot{\mathbf{x}}_m(t) = \mathbf{A}_m\mathbf{x}_m(t) + \mathbf{B}_m\mathbf{r}(t)$	(3.2)
Predictor Model	$\dot{\hat{\mathbf{x}}}_p(t) = \mathbf{A}_m\hat{\mathbf{x}}_p(t) + \mathbf{B}_\nu\left(\hat{\boldsymbol{\nu}}(t) + \hat{\boldsymbol{\Theta}}_\nu(t)\boldsymbol{\omega}(t)\right)$ $\hat{\boldsymbol{\nu}}(t) = \mathbf{U}_{a,r}\mathbf{D}_a\hat{\boldsymbol{\Lambda}}_{a,r}(t)\mathbf{u}(t)$	(4.7)
Virtual Control Law	$\boldsymbol{\nu}_d(t) = \mathbf{K}_r\mathbf{r}(t) - \hat{\boldsymbol{\Theta}}_\nu(t)\boldsymbol{\omega}(t)$	(4.8)
Matching Cond.	$\mathbf{Y} := \left[\mathbf{A}_p - \mathbf{A}_m,\ \ \mathbf{K}_\phi,\ \ \mathbf{d}\right] = \mathbf{B}_\nu\boldsymbol{\Theta}_\nu$	(4.6)
Error Dynamics	$\dot{\mathbf{e}}_p(t) = \mathbf{A}_m\mathbf{e}_p(t) + \mathbf{B}_\nu\left(\tilde{\boldsymbol{\nu}}(t) + \tilde{\boldsymbol{\Theta}}_\nu(t)\boldsymbol{\omega}(t)\right)$ $\tilde{\boldsymbol{\nu}}(t) = \mathbf{U}_{a,r}\mathbf{D}_a\tilde{\boldsymbol{\Lambda}}_{a,r}(t)\mathbf{u}(t)$	(4.11)
Update Laws	$\dot{\hat{\boldsymbol{\Theta}}}_\nu(t) = -\mathbf{B}_\nu^T\mathbf{P}\mathbf{e}_p(t)\boldsymbol{\omega}^T(t)\boldsymbol{\Gamma}_\theta$ $\dot{\hat{\boldsymbol{\Lambda}}}_{a,r}(t) = -\mathbf{D}_a\mathbf{U}_{a,r}^T\mathbf{B}_\nu^T\mathbf{P}\mathbf{e}_p(t)\mathbf{u}^T(t)\boldsymbol{\Gamma}_\Lambda$	(4.13)
Control Authority	Control Allocation	Asm. 4.3
Reduced-Order P-ACA		
Plant	$\dot{\mathbf{x}}_p(t) = \mathbf{A}_p\mathbf{x}_p(t) + \begin{bmatrix}\mathbf{0}\\ \mathbf{B}_{\nu 2}\end{bmatrix}\boldsymbol{\nu}(t) + \begin{bmatrix}\mathbf{0}\\ \mathbf{K}_{\phi 2}\end{bmatrix}\boldsymbol{\phi}(\mathbf{x}_p(t)) + \begin{bmatrix}\mathbf{0}\\ \mathbf{d}_2\end{bmatrix}$ $\boldsymbol{\nu}(t) = \mathbf{B}_a\boldsymbol{\Lambda}\mathbf{u}(t)$	(4.14)
Reference Model	$\dot{\mathbf{x}}_m(t) = \mathbf{A}_m\mathbf{x}_m(t) + \mathbf{B}_m\mathbf{r}(t)$	(3.2)
Reduced Predictor Model	$\dot{\hat{\mathbf{x}}}_{p2}(t) = \begin{bmatrix}\mathbf{A}_{m21} & \mathbf{A}_{m22}\end{bmatrix}\hat{\boldsymbol{\chi}}(t) - \mathbf{L}\mathbf{e}_{p2}(t) + \mathbf{B}_{\nu 2}\left(\hat{\boldsymbol{\nu}}(t) + \hat{\boldsymbol{\Theta}}_\nu(t)\boldsymbol{\omega}(t)\right)$ $\hat{\boldsymbol{\nu}}(t) = \mathbf{U}_{a,r}\mathbf{D}_a\hat{\boldsymbol{\Lambda}}_{a,r}(t)\mathbf{u}(t)$	(4.17)
Virtual Control Law	$\boldsymbol{\nu}_d(t) = \mathbf{K}_r\mathbf{r}(t) - \hat{\boldsymbol{\Theta}}_\nu(t)\boldsymbol{\omega}(t)$	(4.8)
Matching Cond.	$\mathbf{Y} = \left[\mathbf{A}_p - \mathbf{A}_m,\ \ \begin{bmatrix}\mathbf{0}\\ \mathbf{K}_{\phi 2}\end{bmatrix},\ \ \begin{bmatrix}\mathbf{0}\\ \mathbf{d}_2\end{bmatrix}\right] = \begin{bmatrix}\mathbf{0}\\ \mathbf{B}_{\nu 2}\end{bmatrix}\boldsymbol{\Theta}_\nu$	(4.16)
Error Dynamics	$\dot{\mathbf{e}}_{p2}(t) = (\mathbf{A}_{m22} - \mathbf{L})\mathbf{e}_{p2}(t) + \mathbf{B}_{\nu 2}\left(\tilde{\boldsymbol{\nu}}(t) + \tilde{\boldsymbol{\Theta}}_\nu(t)\boldsymbol{\omega}(t)\right)$ $\tilde{\boldsymbol{\nu}}(t) = \mathbf{U}_{a,r}\mathbf{D}_a\tilde{\boldsymbol{\Lambda}}_{a,r}(t)\mathbf{u}(t)$	(4.21)
Update Laws	$\dot{\hat{\boldsymbol{\Theta}}}_\nu(t) = -\mathbf{B}_{\nu 2}^T\mathbf{P}_L\mathbf{e}_{p2}(t)\boldsymbol{\omega}^T(t)\boldsymbol{\Gamma}_\theta$ $\dot{\hat{\boldsymbol{\Lambda}}}_{a,r}(t) = -\mathbf{D}_a\mathbf{U}_{a,r}^T\mathbf{B}_{\nu 2}^T\mathbf{P}_L\mathbf{e}_{p2}(t)\mathbf{u}^T(t)\boldsymbol{\Gamma}_\Lambda$	(4.23)
Control Authority	Control Allocation	Asm. 4.3

Table 4.1: Comparison between P-ACA and Reduced Order P-ACA

4.3 Prioritizing Control Allocation Strategy (PRIO CA)

Given the framework presented in the last two sections, the remaining task to solve is the control allocation of the desired virtual controls $\boldsymbol{\nu}_d(t)$ to the actuator inputs $\mathbf{u}(t)$ which corresponds to Assumption 4.3 (Control Allocation). The main design drivers correspond to Table 1.1. The actuator model (4.9) is linear, static and time-varying and input magnitude limits (4.3) need to be taken into account. The main objectives are achieving attainable virtual controls and handling unattainable ones. Finally, the maximum execution time needs to be taken into account to allow for running the algorithm in real time.

4.3.1 Motivation

The presented algorithm corresponds to a prioritizing control allocation strategy which is motivated from the application scenario of a VTOL aircraft and specifically the hexacopter system described in Chapter 2 (Multirotor Dynamics Model). The main reason is that the prioritizing control allocation provides means of handling unattainable desired virtual controls $\boldsymbol{\nu}_d(t)$. This is important since for the two hexacopter configurations presented in Chapter 2 there is always a single failure case that leads to a degraded control situation. The second reason is that the hexacopter system has an inherent decoupling of position and yaw motions as derived in Section 2.2.5. Therefore, a clear prioritization of virtual controls can be done from high to low as:

- Roll and pitch moments L/M,
- thrust T,
- and yaw moment N.

Here, the roll and pitch moments L/M have been prioritized over the thrust T since the direction of the exerted propulsion force is more important than its magnitude. If the direction of the thrust vector $-\mathbf{z}_B(t)$ cannot be tracked, the magnitude T does not provide meaningful means of control.

Since the presented algorithm can be used in a context different from adaptive control, the following task formulation is introduced. The connection between the Control Task 4.2 (Prioritizing Control Allocation) and Assumption 4.3 (Control Allocation) is described in Section 4.3.3.

Control Task 4.2 (Prioritizing Control Allocation) *Given a desired virtual control input $\boldsymbol{\nu}_d(t) \in \mathbb{R}^r$, compute the control inputs $\mathbf{u}(\mathbf{B}_{\nu u}(t), \boldsymbol{\nu}_d(t)) \in \mathbb{R}^m$ such that*

$$\boldsymbol{\nu}_d(t) = \mathbf{B}_{\nu u}(t)\,\mathbf{u}(t) \tag{4.28}$$

with $\mathbf{B}_{\nu u}(t) \in \mathbb{R}^{r\times m}$ and such that the input lies within the input set $\mathcal{U} \subset \mathbb{R}^m$ defined as

$$\mathcal{U} := \{\mathbf{u} \in \mathbb{R}^m \mid -1 \leq u_i \leq 1, \forall i = 1...m\}, \tag{4.29}$$

where u_i is the i-th element of $\mathbf{u}(t)$. The control input set $\mathcal{U}$ is an m-dimensional hypercube.

If the desired virtual control input $\boldsymbol{\nu}_d(t)$ is unattainable, prioritize the constraints

$$\nu_{d,j}(t) = \mathbf{B}_{\nu u,rj}(t)\,\mathbf{u}(t) \tag{4.30}$$

according to a selected virtual control order. Here, $\mathbf{B}_{\nu u,rj}(t) \in \mathbb{R}^{1\times m}$ is the j-th row of $\mathbf{B}_{\nu u}(t)$ and $\nu_{d,j}(t) \in \mathbb{R}$ is the j-th element of $\boldsymbol{\nu}_d(t)$.

The control allocation presented in this section is based on the knowledge of the SVD decomposition of the matrix $\mathbf{B}_{u\nu}(t)$ and follows a prioritization of virtual control inputs. The main idea is to allocate the virtual controls sequentially from the highest to the lowest priority while always maintaining an attainable input command, that is while complying with input constraints. The SVD is computed using an SVD update algorithm originally developed in [69] to deal with the time-varying input matrix and maintain the real-time properties.

4.3.2 Input Normalization

In the Control Task 4.2 it has been assumed that the control inputs $\mathbf{u}(t) \in \mathbb{R}^m$ are normalized such that (4.29) holds. In general, the control inputs $\overline{\mathbf{u}}(t) \in \mathbb{R}^m$ are not normalized but are limited by

$$\overline{u}_{i,min} \le \overline{u}_i(t) \le \overline{u}_{i,max}, \qquad \text{for } i = 1\dots m,$$

where $\overline{u}_{i,min}, \overline{u}_{i,max} \in \mathbb{R}$ are the minimum and maximum limit of each element of $\overline{\mathbf{u}}(t)$. These inputs $\overline{\mathbf{u}}(t)$ can always be transformed such that (4.29) holds through the following linear transformation

$$u_i(t) = 2 \cdot \frac{\overline{u}_i(t) - \overline{u}_{i,min}}{\overline{u}_{i,max} - \overline{u}_{i,min}} - 1 = u_{i,off} + M_i \cdot \overline{u}_i(t), \tag{4.31}$$

with the offset $u_{i,off} \in \mathbb{R}$ and the slope $M_i \in \mathbb{R}$ defined as

$$u_{i,off} := \frac{-\overline{u}_{i,max} - \overline{u}_{i,min}}{\overline{u}_{i,max} - \overline{u}_{i,min}}, \qquad M_i := \frac{2}{\overline{u}_{i,max} - \overline{u}_{i,min}}.$$

Similarly, the virtual controls $\overline{\boldsymbol{\nu}}(t)$ and the input matrix $\overline{\mathbf{B}}_{\nu u}(t)$ also need to be transformed. The input equation

$$\overline{\boldsymbol{\nu}}(t) = \overline{\mathbf{B}}_{\nu u}(t) \cdot \overline{\mathbf{u}}(t) \tag{4.32}$$

can be transformed by using the vector of stacked offsets $\mathbf{u}_{off} \in \mathbb{R}^m$ and the diagonal matrix $\mathbf{M} \in \mathbb{R}^{m\times m}$ defined as

$$\mathbf{u}_{off} = \begin{pmatrix} u_{1,off} \\ \vdots \\ u_{m,off} \end{pmatrix}, \qquad \mathbf{M} = \begin{bmatrix} M_1 & & 0 \\ & \ddots & \\ 0 & & M_m \end{bmatrix}.$$

By inverting (4.31) $\overline{\mathbf{u}} = \mathbf{M}^{-1} \cdot (\mathbf{u}(t) - \mathbf{u}_{off})$ and inserting it into (4.32), it leads to

$$\overline{\boldsymbol{\nu}}(t) = \overline{\mathbf{B}}_{\nu u}(t) \cdot \mathbf{M}^{-1} \cdot (\mathbf{u}(t) - \mathbf{u}_{off})$$

and hence the representation (4.28) follows from

$$\mathbf{B}_{\nu u}(t) := \overline{\mathbf{B}}_{\nu u}(t) \cdot \mathbf{M}^{-1}, \tag{4.33a}$$

$$\boldsymbol{\nu}(t) := \overline{\boldsymbol{\nu}}(t) + \mathbf{B}_{\nu u}(t) \cdot \mathbf{u}_{off}. \tag{4.33b}$$

Example 4.4 (Hexacopter Input Normalization) *For the hexacopter system with actuator configuration 1 as in Figure 2.1a, the nominal control input matrix* $\overline{\mathbf{B}}_{\nu u} = \mathbf{B}_a$ *is given by* (2.9) *and by the parameters in Table 2.3. By setting*

$$\overline{u}_{i,min} = (\omega_{P,min})^2, \qquad \overline{u}_{i,max} = (\omega_{P,max})^2, \qquad \textit{for } i = 1 \dots 6,$$

the inputs can be normalized using the offset $u_{i,off} = -1.0304$ *and the slope* $M_i = 1.7452 \cdot 10^{-6}$ *for* $i = 1 \dots 6$ *computed as in* (4.31).

Considering the estimated actuator control effectiveness matrix $\hat{\mathbf{\Lambda}}(t) \in \mathbb{R}^{6\times 6}$*, the input matrix* $\mathbf{B}_{\nu u}(t)$ *is computed as*

$$\mathbf{B}_{\nu u}(t) = diag(\mathbf{s}_v) \cdot \mathbf{B}_a \cdot \mathbf{M}^{-1} \cdot \hat{\mathbf{\Lambda}}(t).$$

For numerical reasons a scaling/normalization of the virtual controls is done using the inverse of the maximum achievable values of the virtual controls

$$\mathbf{s}_v = \left(24.637^{-1}, \;\; 1.7393^{-1}, \;\; 1.5063^{-1}, \;\; 0.4423^{-1}\right)^T.$$

This leads to the following normalized input matrix

$$\mathbf{B}_{\nu u}(t) = \begin{bmatrix} 0.0821 & 0.0821 & 0.0821 & 0.0821 & 0.0821 & 0.0821 \\ -0.1250 & -0.2500 & -0.1250 & 0.1250 & 0.2500 & 0.1250 \\ 0.2500 & 0.0000 & -0.2500 & -0.2500 & -0.0000 & 0.2500 \\ -0.1667 & 0.1667 & -0.1667 & 0.1667 & -0.1667 & 0.1667 \end{bmatrix} \cdot \hat{\mathbf{\Lambda}}(t)$$

and the virtual control

$$\boldsymbol{\nu}(t) = diag(\mathbf{s}_v) \cdot \overline{\boldsymbol{\nu}}(t) + \mathbf{B}_{\nu u}(t) \cdot \mathbf{u}_{off}. \tag{4.34}$$

4.3.3 Relation to Adaptive Control

In this thesis, the adaptive control approaches from Sections 4.1 (Predictor-Based Adaptive Control Allocation) and 4.2 (Reduced Order P-ACA) are combined with the control allocation approach from this section. They relate directly through the Assumption 4.3 (Control Allocation) and the Control Task 4.2 (Prioritizing Control Allocation). The nomenclature is not exactly the same mainly for two reasons. On the one hand, the Control Task 4.2 is not the only one way of solving the originally stated Assumption 4.3 . On the other hand, the Control Task 4.2 does not necessarily need to be integrated within an adaptive control concept. It can be generally used for linear control allocation problem solving. Lastly, the selected nomenclature serves the clear explanation of each of the approaches.

The relation between the two approaches is given by

$$\begin{aligned} \mathbf{B}_{\nu u}(t) = \mathbf{B}_a \hat{\mathbf{\Lambda}}(t) &= \mathbf{U}_{a,r} \mathbf{D}_a \hat{\mathbf{\Lambda}}_{a,r}(t), \\ u_{i,min} &= -1, \qquad \text{for } i = 1 \dots m, \\ u_{i,max} &= 1, \qquad \text{for } i = 1 \dots m, \end{aligned}$$

where the normalization presented in Section 4.3.2 is used for matching the input limits. Here, the SVD of $\mathbf{B}_a = \mathbf{U}_{a,r}\mathbf{D}_a\mathbf{V}_{a,r}^T$ (4.4) and the definition of the reduced effectiveness matrix $\mathbf{\Lambda}_{a,r} = \mathbf{V}_{a,r}^T\mathbf{\Lambda}$ (4.5) have been used. Note that the normalization input mapping is constant for constant input limits and therefore it remains constant for varying estimate $\hat{\mathbf{\Lambda}}(t)$. Furthermore, note that $\operatorname{rank}(B_a) = r$ by definition of the plant (4.1) and that therefore $\operatorname{rank}\mathbf{B}_{\nu u}(t) \leq r$.

4.3.4 Main Algorithm

In this section the time dependency is dropped for the sake of clarity. Furthermore, the following two assumptions are made.

Assumption 4.8 (Input Matrix Rank) *The matrix $\mathbf{B}_{\nu u}$ has a full row rank, i.e. it has a rank of r.*

Assumption 4.9 (Input Matrix SVD) *The singular value decomposition (SVD) of the matrix $\mathbf{B}_{\nu u} \in \mathbb{R}^{r\times m}$ is known and it is given by*

$$\mathbf{B}_{\nu u} = \mathbf{U}_b \cdot \mathbf{\Sigma}_b \cdot \mathbf{V}_b^T, \tag{4.35}$$

where $\mathbf{U}_b \in \mathbb{R}^{r\times r}, \mathbf{\Sigma}_b \in \mathbb{R}^{r\times m}, \mathbf{V}_b \in \mathbb{R}^{m\times m}$. $\mathbf{U}_b, \mathbf{V}_b$ are orthogonal matrices, i.e. $\mathbf{U}_b\mathbf{U}_b^T = \mathbf{I}$ and $\mathbf{V}_b\mathbf{V}_b^T = \mathbf{I}$. $\mathbf{\Sigma}_b$ is a rectangular diagonal matrix with non-negative entries on the diagonal which can be written as

$$\mathbf{\Sigma}_b = \begin{bmatrix}\mathbf{D}_b & \mathbf{0}_{r\times m-r}\end{bmatrix}.$$

$\mathbf{D}_b \in \mathbb{R}^{r\times r}$ is a diagonal positive definite matrix given Assumption 4.8. The diagonal elements are the singular values of $\mathbf{B}_{\nu u}$ and have a decreasing order. Taking this structure into account, it is possible to partition $\mathbf{V}_b = [\mathbf{V}_{b,r}, \mathbf{V}_{b,n}]$, such that $\mathbf{V}_{b,r} \in \mathbb{R}^{m\times r}$ and $\mathbf{V}_{b,n} \in \mathbb{R}^{m\times(m-r)}$. The matrix $\mathbf{B}_{\nu u}$ can be alternatively written as

$$\mathbf{B}_{\nu u} = \mathbf{U}_b \cdot \mathbf{D}_b \cdot \mathbf{V}_{b,r}^T. \tag{4.36}$$

Assumption 4.8 (Input Matrix Rank) is not a hard restriction but it facilitates the derivation of the control allocation algorithm and fully covers the single failure case for a system with redundant actuators. In Section 4.4, a SVD update algorithm is presented such that Assumption 4.9 (Input Matrix SVD) holds within the adaptive control concept.

If a solution of Control Task 4.2 (Prioritizing Control Allocation) exists, it is always a subset of the solution of its unconstrained version. The solution set of the unconstrained linear control allocation problem can be characterized by the SVD and the Moore-Penrose inverse of the matrix $\mathbf{B}_{u\nu}$ (4.35). Assumption 4.8 (Input Matrix Rank), the Moore-Penrose inverse of the matrix $\mathbf{B}_{u\nu}$ is given by

$$\mathbf{B}_{\nu u}^{+} = \mathbf{V}_{b,r} \cdot \mathbf{D}_b^{-1} \cdot \mathbf{U}_b^T$$

Given Assumptions 4.9 (Input Matrix SVD) and 4.8 (Input Matrix Rank), every solution to the unconstrained linear control allocation problem (4.28) has m degrees of freedom and can be written as [14, 59]

$$\mathbf{u} = \mathbf{B}_{\nu u}^{+} \cdot \boldsymbol{\nu} + \mathbf{V}_{b,n} \cdot \mathbf{w} = \sum_{i=1}^{r} \mathbf{p}_i \cdot \nu_i + \sum_{i=1}^{m-r} \mathbf{v}_{b,n,i} \cdot w_i, \tag{4.37}$$

where $\mathbf{w} \in \mathbb{R}^{m-r}$ are the null space variables, $\mathbf{p}_i \in \mathbb{R}^m$ is the i-th column of the pseudo inverse $\mathbf{B}_{\nu u}^{+}$ and $\mathbf{v}_{b,n,i} \in \mathbb{R}^m$ is the i-th column of the null space matrix $\mathbf{V}_{b,n}$. An important characteristic is that (4.37) is a one-to-one mapping of the input space. Hence,

given the virtual control vector $\boldsymbol{\nu}$ and the null space variable $\mathbf{w}$, the input vector $\mathbf{u}$ is uniquely defined and vice versa. This follows from the fact that the linear mapping

$$\begin{pmatrix} \boldsymbol{\nu} \\ \mathbf{w} \end{pmatrix} = \mathbf{T} \cdot \mathbf{u}, \qquad \mathbf{T} = \begin{bmatrix} \mathbf{U}_b \cdot \mathbf{D}_b \cdot \mathbf{V}_{b,r}^T \\ \mathbf{V}_{b,n}^T \end{bmatrix} \in \mathbb{R}^{m \times m},$$

has an inverse given by

$$\mathbf{T}^{-1} = \begin{bmatrix} \mathbf{V}_{b,r} \cdot \mathbf{D}_b^{-1} \cdot \mathbf{U}_b^T, & \mathbf{V}_{b,n} \end{bmatrix} \in \mathbb{R}^{m \times m}.$$

The detailed derivation is given in Lemma B.11 (Transformation Matrix Inverse) in Appendix B. If Assumption 4.8 (Input Matrix Rank) does not hold, an analog representation of (4.37) is needed and can be found by adjusting the dimensions of $\boldsymbol{\nu}$ and $\mathbf{w}$ to the rank of the input matrix.

The prioritizing control allocation (PRIO CA) utilizes the known directions $\mathbf{p}_i$ for $i = 1 \dots r$ of the virtual controls and the directions $\mathbf{v}_{n,i}$ for $i = 1 \dots (m - r)$ of the null space to simultaneously achieve the desired virtual controls and to handle control input saturation. The algorithm consists of a sequence of updates to a previously computed solution $\mathbf{u}_k$, starting at $\mathbf{u}_0 = \mathbf{0} \in \mathcal{U}$, given by

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \mathbf{c}_k \cdot \Delta_k, \tag{4.38}$$

where $k \in \mathbb{N}_0$ is the update count, $\mathbf{c}_k \in \mathbb{R}^m$ is the update direction and $\Delta_k \geq 0 \in \mathbb{R}$ is the step length. The main idea is to sequentially reduce the difference of the current virtual controls $\boldsymbol{\nu}_k := \mathbf{B}_{\nu u} \cdot \mathbf{u}_k$ and the desired virtual controls $\boldsymbol{\nu}_d$, defined as $\widetilde{\boldsymbol{\nu}}_k = \boldsymbol{\nu}_d - \boldsymbol{\nu}_k$, while maintaining the control input $\mathbf{u}$ within its constraints. This is achieved via a special selection of $\mathbf{c}_k$ and Δ_k.

In the unconstrained case, the selection of the update direction $\mathbf{c}_k$ and the step length Δ_k is fairly simple. They can for example be chosen as

$$\mathbf{c}_k = \operatorname{sgn}(\nu_{d,k}) \cdot \mathbf{p}_k, \tag{4.39a}$$

$$\Delta_k = |\nu_{d,k}|, \tag{4.39b}$$

for $k = 1 \dots r$ and $\mathbf{u}_0 = \mathbf{0}$. In this case the virtual control error $\widetilde{\boldsymbol{\nu}}$ is driven to zero by changing one virtual control per iteration. This leads to the pseudo-inverse solution $\mathbf{u} = \mathbf{B}_{\nu u}^{+} \cdot \boldsymbol{\nu}_d$ after r updates.

The constrained case mainly implies two changes to the strategy in order to avoid input commands outside their constraints. The step length Δ_k is limited from above and the update direction $\mathbf{c}_k$ needs to be changed every time a control input hits a saturation limit. The update direction $\mathbf{c}_k$ is chosen such that the virtual control can be improved and the saturated inputs do not go outside the boundaries. This concept is depicted in Figure 4.3 using the matrix $\mathbf{B}_{\nu u} = [1, 0.3]$ as in the following example.

Example 4.5 (Simple Input Matrix) *Consider the simple input matrix $\mathbf{B}_{\nu u} = [1, 0.3]$. From its dimensions, it is known that 2 inputs are available and that the null space has the dimension 1. Its pseudo inverse is given by*

$$\mathbf{B}_{\nu u}^{+} = \begin{bmatrix} 0.92 \\ 0.28 \end{bmatrix}$$

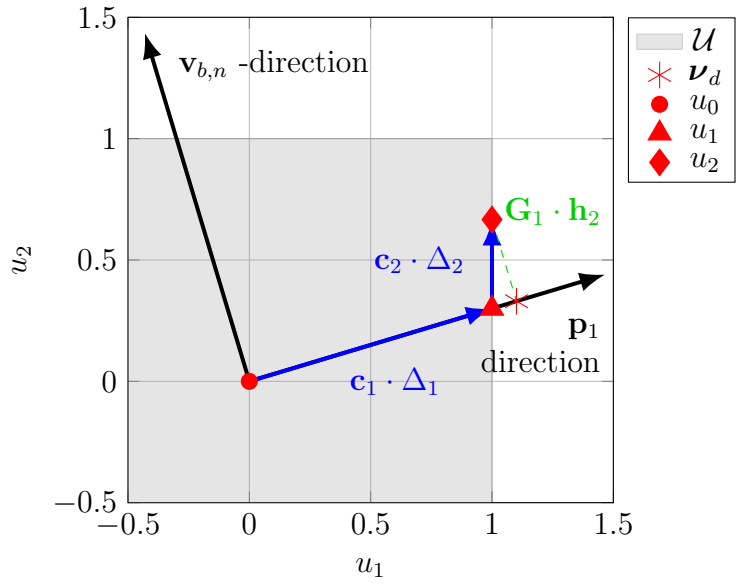


Figure 4.3: Prioritizing Control Allocation (PRIO CA) - Presentation of the Main Idea using $\mathbf{B}_{\nu u} = [1, 0.3]$ from Example 4.5

and its null space is spanned by the vector

$$\mathbf{v}_{b,n} = \begin{bmatrix} -0.29 \\ 0.96 \end{bmatrix}.$$

Every input is fully defined by the equation (4.37)

$$\mathbf{u} = \mathbf{p}_1 \cdot \nu_1 + \mathbf{v}_{b,n} \cdot w_1,$$

where $\mathbf{p}_1 = \mathbf{B}_{\nu u}^{+}$ *and* $w_1 \in \mathbb{R}$ *is the null space variable. The allocation process for* $\nu_d = 1.2$ *is depicted in Figure 4.3. The specific steps are described in the subsequent sections.*

In order to facilitate the description of the algorithm the following definitions are introduced. Without loss of generality, assume that the virtual controls are ordered from highest to lowest priority, i.e. ν_1 has the highest and ν_r has the lowest priority. The set of inputs that are saturated is defined as

$$\mathcal{S} := \{i \in \mathbb{N} \mid |u_i| \geq 1, \forall i = 1, \ldots, m\}, \tag{4.40}$$

and the complementary set is defined as

$$\mathcal{F} := \{i \in \mathbb{N} \mid |u_i| < 1, \forall i = 1, \ldots, m\}. \tag{4.41}$$

By stacking control inputs corresponding to $\mathcal{S}$ and $\mathcal{F}$ into a vector form, the saturated inputs $\mathbf{u}_s$ and the free control inputs $\mathbf{u}_f$ are defined:

$$\mathbf{u}_s = col(u_i) \in \mathbb{R}^s, i \in \mathcal{S},$$
$$\mathbf{u}_f = col(u_i) \in \mathbb{R}^f, i \in \mathcal{F},$$

where it holds that $m = s + f$. The function $col(\cdot)$ stacks the scalars into a column vector.

Update Direction $\mathbf{c}_k$ Computation

The algorithm starts from the attainable solution $\mathbf{u}_0 = \mathbf{0}$ corresponding to the virtual control error $\widetilde{\boldsymbol{\nu}}_0 = \boldsymbol{\nu}_d$ and tries to allocate the desired value of the virtual control with highest priority $\nu_{d,j}$. The counter $j > 0 \in \mathbb{R}$ is used for selecting the current virtual control with the highest priority, i.e. $j = 1$ in the first step. In the unconstrained case, the update desired update direction at any given step of the algorithm is given by (4.39a). If saturation occurs, the virtual control prioritization comes into play. The chosen update direction $\mathbf{c}_k$ must meet two characteristics. One is that it reduces the error $|\tilde{\nu}_{k,j}|$ and the other one is that the elements corresponding to saturated inputs do not go outside the limits, that is

$$c_{k,i} \cdot \operatorname{sgn}(u_i) \leq 0, \qquad \forall i \in \mathcal{S}. \tag{4.42}$$

Considering the form of the control allocation solution (4.37), the directions $\mathbf{p}_i$ with $i = j+1 \dots r$ and $\mathbf{v}_{n,i}$ with $i = 1 \dots m-r$ can be used in order to keep the control inputs within their bounds. Hence, the corresponding directions are combined into the matrix

$$\mathbf{G}_j = \begin{bmatrix} \mathbf{p}_{j+1}, & \dots, & \mathbf{p}_r, & \mathbf{v}_{b,n,1}, & \dots, & \mathbf{v}_{b,n,m-r} \end{bmatrix} \in \mathbb{R}^{m \times (m-j)} \tag{4.43}$$

and the update direction (4.39a) is extended to the form

$$\mathbf{c}_k = \operatorname{sgn}(\tilde{\nu}_{k,j}) \cdot \mathbf{p}_j + \mathbf{G}_j \cdot \mathbf{h}_k. \tag{4.44}$$

Here, $\mathbf{h}_k \in \mathbb{R}^{m-j}$ is a help variable at the k-th iteration of the algorithm whose dimension depends on the currently improving virtual control j. This help variable $\mathbf{h}_k$ corresponds to scaled changes to be made to the vector

$$\begin{pmatrix} \nu_{j+1}, & \dots, & \nu_r, & w_1, & \dots, & w_{m-r} \end{pmatrix}^T \in \mathbb{R}^{m-j}.$$

The strategy is to use the virtual controls with less priority and null space variables as degrees of freedom in order to generate an update direction $\mathbf{c}_k$ that maintains the control inputs within their boundaries. In the multirotor example, this means that in order to allocate roll and pitch moments L/M while maintaining the inputs within their boundaries, the thrust T, the yaw moment N and the null space variables are utilized as degrees of freedom. Furthermore, once allocated, roll and pitch moments L/M will not be changed during the allocation of the thrust T or the yaw moment N. In Figure 4.3 corresponding to Example 4.5 this means that in order to allocate ν_1 the null space variable w_1 is used.

First, consider only the saturated input subsystem

$$\mathbf{c}_{k,s} = \operatorname{sgn}(\tilde{\nu}_{k,j}) \cdot \mathbf{p}_{j,s} + \mathbf{G}_{j,s} \cdot \mathbf{h}_k, \tag{4.45}$$

where $\mathbf{c}_{k,s}$, $\mathbf{p}_{j,s} \in \mathbb{R}^s$ and $\mathbf{G}_{j,s} \in \mathbb{R}^{s \times (m-j)}$ correspond to the rows of the saturated inputs $i \in \mathcal{S}$ of $\mathbf{c}_k$, $\mathbf{p}_j$ and $\mathbf{G}_j$. In order to avoid input saturation, the following conditions must hold

$$\operatorname{sgn}(u_i) \cdot c_{k,i} \leq 0, \qquad \forall i \in \mathcal{S}. \tag{4.46}$$

Following the ideas presented in [142], $\mathbf{h}_k$ is selected such that $\mathbf{c}_{k,s} = \mathbf{0}$ as

$$\mathbf{h}_k = \operatorname{sgn}(\tilde{\nu}_{k,j}) \cdot \mathbf{G}_{j,s}^{+} \cdot (-\mathbf{p}_{j,s}). \tag{4.47}$$

In the case that $\mathbf{G}_{j,s}$ has a full row rank, this computation is possible as long as $s \leq (m-j)$, i.e. as long as the number of saturated inputs s is not greater than the degrees of freedom $(m-j)$. Given the prioritization strategy, this leads to a reducing number of permitted saturated inputs for low priority virtual controls. For the last virtual control to be allocated $j = r$, the matrix $\mathbf{G}_r = \mathbf{V}_n$ and it means that only the null space degrees of freedom $m - r$ can be used to avoid unachievable control inputs. The case of $\mathbf{G}_{j,s}$ without a full row rank is discussed in the next section. Finally, this selection of $\mathbf{h}_k$ (4.47) inserted in (4.44) leads to an update

$$\mathbf{c}_k = \operatorname{sgn}(\tilde{\nu}_{k,j}) \cdot \mathbf{p}_j + \operatorname{sgn}(\tilde{\nu}_{k,j}) \cdot \mathbf{G}_j \cdot \mathbf{G}_{j,s}^{+} \cdot (-\mathbf{p}_{j,s}). \tag{4.48}$$

Using this update direction, it is possible to further reduce the error $\tilde{\nu}_{k,j}$ while remaining within the input boundaries through a variation of less important virtual controls and null space variables. An example of this is depicted in Figure 4.3 where in the second update the null space variable w_1 is used to compute a valid update direction c_2.

Step Length Δ_k Computation

Once the update direction $\mathbf{c}_k$ has been selected, a maximum step length $\Delta_{k,max}$ is computed such that it corresponds the smallest Δ_k that leads to a change of the sets $\mathcal{S}$ (4.40) and $\mathcal{F}$ (4.41). For each element i of the input vector, such a step candidate can be computed by inserting the limit $u_{k+1} = 1 \cdot \operatorname{sgn}(c_{k,i})$ into (4.38)

$$1 \cdot \operatorname{sgn}(c_{k,i}) = u_{k,i} + c_{k,i} \cdot \Delta_{k,i}, \tag{4.49a}$$

$$\Delta_{k,i} = \frac{\operatorname{sgn}(c_{k,i}) - u_{k,i}}{c_{k,i}}. \tag{4.49b}$$

Here, $c_{k,i}, u_{k,i} \in \mathbb{R}$ are the i-th elements of $\mathbf{c}_k$ and $\mathbf{u}_k$ respectively and $\Delta_{k,i}$ is the smallest step length that leads to a saturation of the i-th input, i.e. either $u_i = 1$ or $u_i = -1$.

To robustly compute the step length (4.49), only the inputs corresponding to the set $\mathcal{D}$ are checked. The set $\mathcal{D}$ is defined as

$$\mathcal{D} := \left\{ i \in \mathbb{N} \;\middle|\; \begin{array}{l} i \in \mathcal{F} \quad \text{or} \\ (i \in \mathcal{S} \text{ and } c_{k,i} \cdot u_i < 0) \end{array}, \; |c_{k,i}| \geq c_{min}, \forall i = 1, \ldots, m \right\}, \tag{4.50}$$

The limit $|c_{k,i}| \geq c_{min}$ has numerical reasons and in our application it is selected as $c_{min} = 10^{-5}$. If $c_{k,i} \approx 0$, (4.49) cannot be computed. Nevertheless, in this case also $\Delta_{k,i}$ does not have a limit for this specific i. The condition $i \in \mathcal{F}$ means that the inputs that are not saturated are tested. From the saturated inputs $i \in \mathcal{S}$, only the ones that are updated towards the other limit $c_{k,i} \cdot u_i < 0$ are taken into account. The case $i \in \mathcal{S}$ and $c_{k,i} \cdot u_i > 0$, i.e. an increase of magnitude of a saturated input, is excluded by the correct selection of the update vector $\mathbf{c}_k$ as will be derived in the following. This case is excluded from $\mathcal{D}$ since $c_{k,i} \cdot u_i > 0$ can happen due to limited numerical accuracy and would only lead to an artificial upper limit of Δ_k.

The maximum step length corresponds to

$$\Delta_{k,max} = \min_{i \in \mathcal{D}} \Delta_{k,i} \tag{4.51}$$

and the step length is finally selected as

$$\Delta_k = \min\left(\Delta_{k,max}, |\tilde{\nu}_{k,j}|\right). \tag{4.52}$$

In other words, the inputs are updated in a direction $\mathbf{c}_k$ (4.39a) that reduces the difference $\tilde{\nu}_{k,j}$ with two possible results. Either $\tilde{\nu}_{k+1,j} = 0$ for $\Delta_k = |\tilde{\nu}_{k,j}|$, or for $\Delta_k = \Delta_{k,max}$ at least one input saturates before reaching that point and an error remains $|\tilde{\nu}_{k+1,j}| > 0$.

$$\tilde{\nu}_{k+1,j} = \tilde{\nu}_{k,j} - \operatorname{sgn}\left(\tilde{\nu}_{k,j}\right) \cdot \Delta_k.$$

If no input saturation occurs, this process can be repeated for all virtual controls and the result would lead to $\mathbf{u} = \mathbf{B}_{\nu u}^{+} \cdot \boldsymbol{\nu}_d$.

The presented process can be repeated until all virtual controls have been allocated or the number of saturated inputs s exceeds the number degrees of freedom $m - j$. The control allocation algorithm is depicted in Figure 4.4. The algorithm starts by setting $\mathbf{u}_0 = \mathbf{0}$, $\widetilde{\boldsymbol{\nu}}_0 = \boldsymbol{\nu}_d, k = 1$ and $j = 1$. The direction of the current virtual control $\mathbf{p}_j$ (j-th column of $\mathbf{B}_{\nu u}^{+}$) and the directions of the help variables $\mathbf{G}_j$ (4.43) are then updated. The sets $\mathcal{S}$ (4.40) and $\mathcal{F}$ (4.41) are identified and based on them the update direction $\mathbf{c}_k$ (4.48) is computed. If no update direction $\mathbf{c}_k$ can be found, the next virtual control $j + 1$ is allocated, otherwise the step length is calculated from (4.49), (4.51) and (4.52). Finally, the input $\mathbf{u}_k$ is updated (4.38) and the current virtual control error $\widetilde{\boldsymbol{\nu}}_{k+1} = \boldsymbol{\nu}_d - \mathbf{B}_{\nu u} \cdot \mathbf{u}_{k+1}$ is computed. Note that if an update direction $\mathbf{c}_k$ is not found, there is no update, otherwise the error $|\tilde{\nu}_{k,j}|$ is always reduced.

The algorithm has two stop conditions: either $j > r$ or $s > m - j$. In order to get an impression of the steps needed for the algorithm to finish, Figure 4.5 depicts the update process using the hexacopter example. One axis indicates the virtual control j that is being allocated, which are 4 in total, and the other axis shows how many inputs are saturated s, with a maximum of 6. Both stop conditions are marked as thick lines within the figure. In every step of the algorithm, an update vector $\mathbf{c}_k$ is searched. If it is not found, then the next virtual control is allocated, i.e. $j = j + 1$. If an update vector $\mathbf{c}_k$ is found, the input update (4.38) either leads to $|\tilde{\nu}_{k+1,j}| = 0$ and hence to $j = j + 1$ or input saturation occurs and s increases. Hence, with every iteration either j or s increase and the stop condition $s > m - j$ gets a step closer. If j increases, the stop condition $j > r$ also gets closer. The algorithm always starts in the left upper corner of the figure and from there it can only move downwards (larger j) and/or to the right (larger s). Downwards only a single step per iteration is possible while to the right s can increase with larger values. In the worst case, m update steps are needed for the algorithm to stop and return a solution. In the best case r steps are needed.

In Figure 4.5 two possible runs of the algorithm are shown and the following prioritization is chosen: $\nu_1 = L$, $\nu_2 = M$, $\nu_3 = T$ and $\nu_4 = N$. The red sequence shows an option where five updates are done and the algorithm stops due to condition $j > r$. The gray sequence shows an option where ν_3 can not be allocated because $s = 4 > m - j = 6 - 3 = 3$ in the last step. Furthermore, in this sequence the allocation of ν_4 does not take place. These two examples of the control allocation process are subsequently analyzed. One of them corresponds to an attainable desired virtual control $\boldsymbol{\nu}_d$ for a nominal hexacopter configuration and the other one to an unattainable desired virtual control $\boldsymbol{\nu}_d$ for a faulty hexacopter configuration.

The control allocation presented in this section is referred to Nominal Prioritizing Control Allocation (NOM PRIO CA) to differentiate it from its embedded software ver-

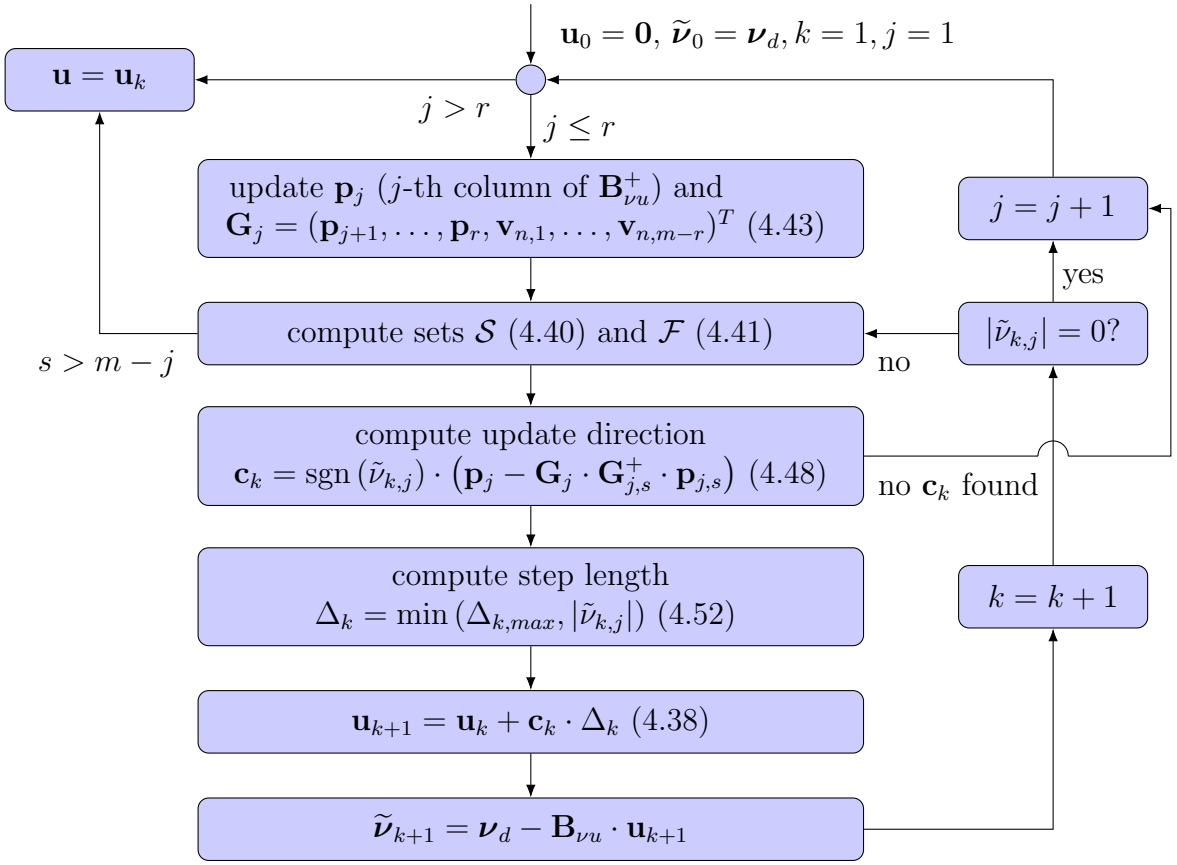


Figure 4.4: Flow Chart of Nominal Prioritized Control Allocation (NOM PRIO CA)

sion which includes some modifications. In order assess the performance, the Pseudo-Inverse Control Allocation (PINV CA) is introduced as an alternative approach.

Example 4.6 (Pseudo-Inverse Control Allocation (PINV CA)) *Given its popularity and its simplicity, the Pseudo-Inverse Control Allocation (PINV CA) is used for performance comparison. Given the Control Task 4.2 (Prioritizing Control Allocation), the PINV CA consists of two steps:*

$$\mathbf{u}_0 = \mathbf{B}_{\nu u}^{+} \cdot \boldsymbol{\nu}_d \tag{4.53a}$$

$$u_i = \max(\min(u_{0,i}, 1), -1), \qquad \forall i. \tag{4.53b}$$

The first step (4.53a) *computes the solution using the pseudo inverse* $\mathbf{B}_{\nu u}^{+}$ *such that* $\boldsymbol{\nu}_d = \mathbf{B}_{\nu u} \cdot \mathbf{u}_0$. *The second step* (4.53b) *limits the output to* $[-1, 1]$. *An important characteristic of this control allocation solution is that it corresponds to the minimum 2-norm solution if no input saturates. If in the first step an input saturates* $|u_{0,i}| > 1$, *it leads to an error in the allocated virtual controls. Furthermore, this control allocation does not consider the virtual control prioritization.*

Example 4.7 (Control Allocation of an Attainable Virtual Control) *Consider the hexacopter system with actuator configuration 1 as in Figure 2.1a corresponding to the Example 4.4 (Hexacopter Input Normalization). Consider its normalized input matrix* $\mathbf{B}_{\nu u}$ *as in* (4.4) *with the control effectiveness matrix* $\boldsymbol{\Lambda} = \hat{\boldsymbol{\Lambda}} = \mathbf{I}$. *The control allocation process can be analyzed*

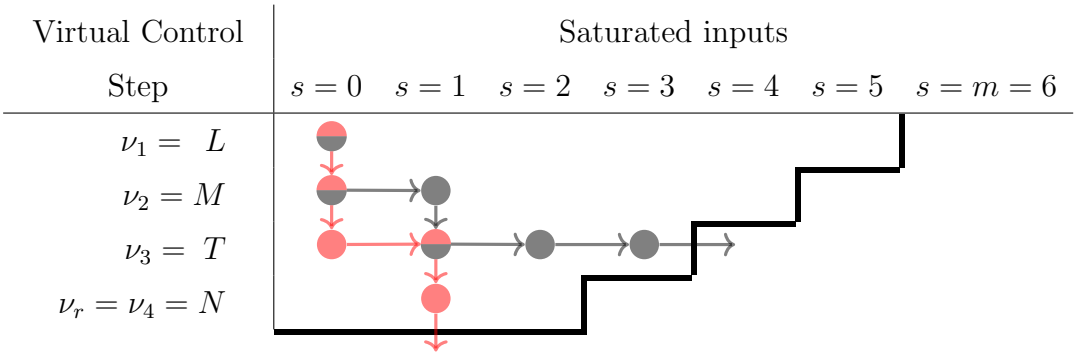


Figure 4.5: Control Allocation Process - Iteration Overview

by taking a look at the virtual controls $\boldsymbol{\nu}$ and the null space variables $\mathbf{w}$ in Figure 4.6 and the control inputs $\mathbf{u}$ in Figure 4.7. In this example, the desired virtual control $\boldsymbol{\nu}_d$ is attainable and depicted in Figure 4.6. For comparison purposes, the process of the Nominal Prioritizing Control Allocation (NOM PRIO CA) is displayed along with the results the Pseudo-Inverse Control Allocation (PINV CA) implemented as in Example 4.6. The NOM PRIO CA algorithm allocates ν_1 and ν_2 in the first and second iteration respectively. In the third iteration, $\tilde{\nu}_3$ is improved but before it reaches zero, the input u_3 saturates. Hence, in iteration 4, the variables ν_4, w_1 and w_2 are used to find the next update direction that drives $\tilde{\nu}_3$ to zero. In the last iteration ν_4 is allocated using w_1 and w_2 in order to maintain the inputs within their boundaries. This example corresponds to the red sequence in Figure 4.5.

The final results can be compared in the virtual control space plot in Figure 4.8. Here, the desired virtual control ν_d and the respective final results from the NOM PRIO CA and the PINV CA are depicted after inverting the transformation (4.34) from Example 4.4 (Hexacopter Input Normalization). Furthermore, in each of the two 2-dimensional views, cuts of the attainable control set (ACS) as presented in Chapter 2 are plotted. The 4-dimensional ACS depends only on the input matrix $\mathbf{B}_{\nu u}$ and the limits of the control inputs. On the other hand, the 2-dimensional cuts depend on the virtual controls computed by the control allocation approaches. Each 2-dimensional ACS cut corresponds to the values of the virtual controls that are not displayed in the plot. For example the ACS NOM PRIO in the T/N- plot corresponds to the cut of the 4-dimensional ACS at the values of L and M achieved by the NOM PRIO CA. Hence, it can be interpreted as the set of possible virtual controls T and N corresponding to the fixed values of L and M. This holds analogously for the ACS cuts in the L/M plane and the ACS cuts corresponding to the PINV CA.

In the presented case, it can be seen that NOM PRIO CA achieves the desired virtual control $\boldsymbol{\nu}_d$ and that a small error is present if the PINV CA is used. This is due to input saturation. Because the virtual controls of the two control allocations are slightly different, the corresponding ACS cuts exhibit also some differences. From this example, it can be seen that the NOM PRIO CA is able to improve the performance compared to the PINV CA in cases where input saturation occurs, even if no faults are present in the system.

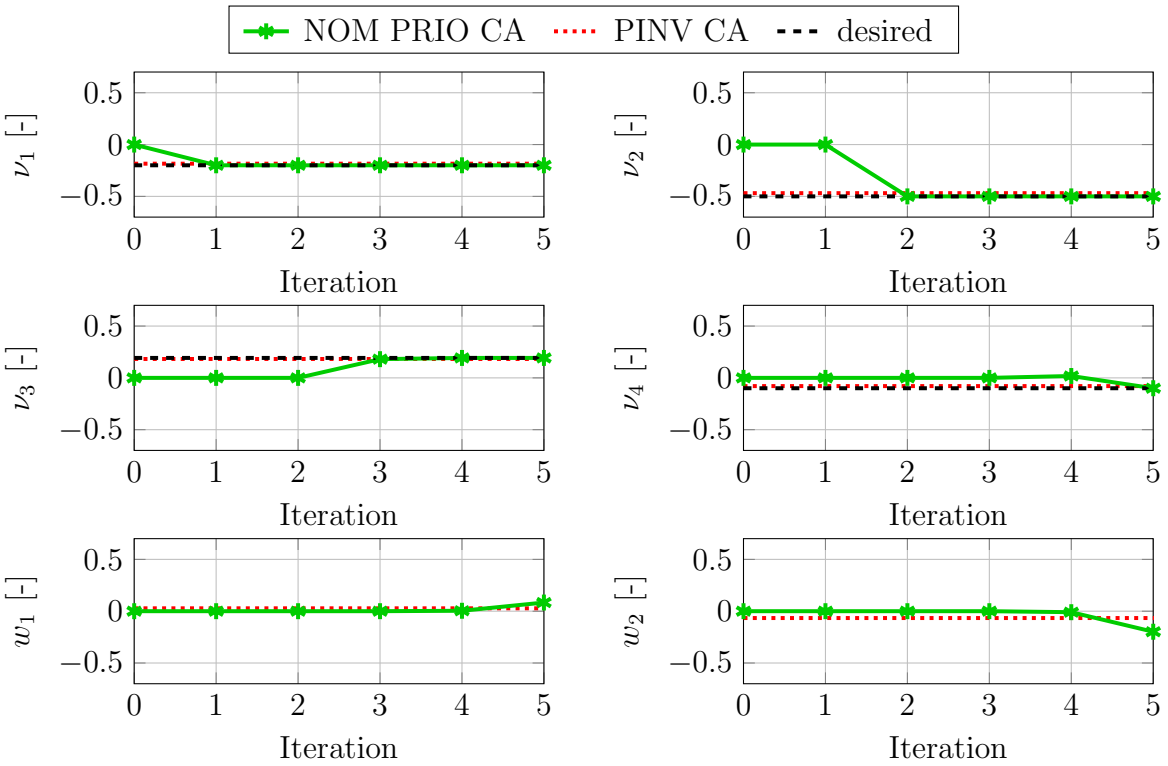


Figure 4.6: Control Allocation Process - Attainable Solution: virtual controls $\boldsymbol{\nu}$ and null space variables $\mathbf{w}$ corresponding to the Nominal Prioritizing Control Allocation (NOM PRIO CA) and the Pseudo-Inverse Control Allocation (PINV CA)

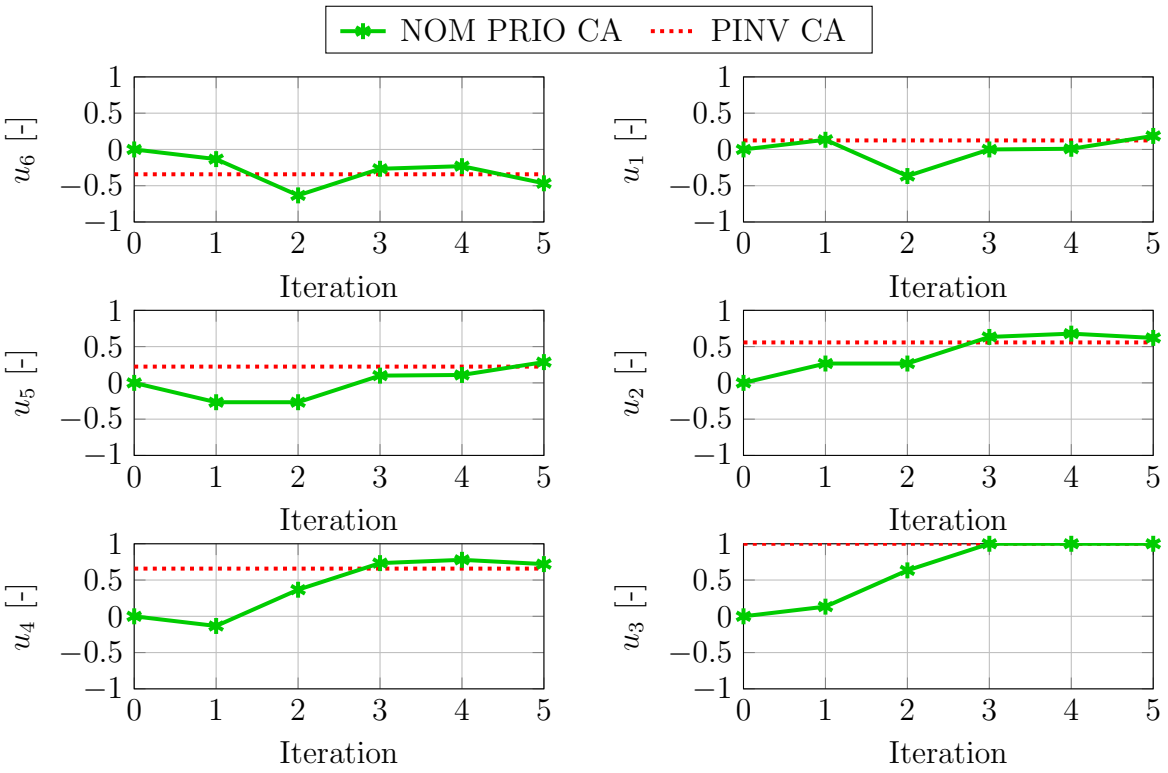


Figure 4.7: Control Allocation Process - Attainable Solution: control inputs $\mathbf{u}$ corresponding to the Nominal Prioritizing Control Allocation (NOM PRIO CA) and the Pseudo-Inverse Control Allocation (PINV CA)

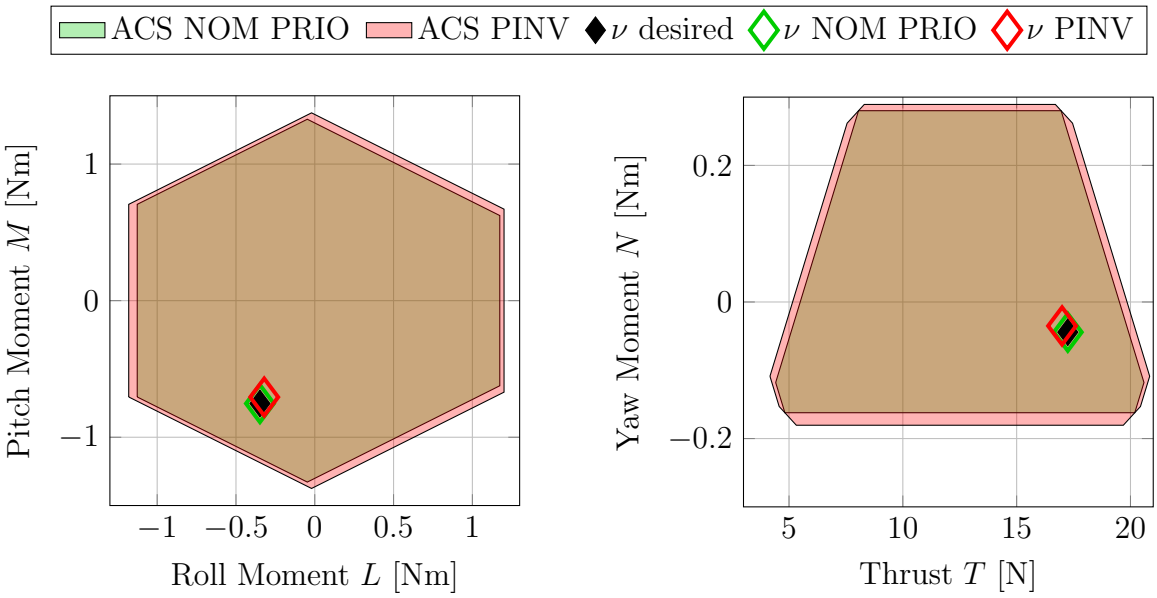


Figure 4.8: Control Allocation Process - Attainable Solution: virtual control space corresponding to the Nominal Prioritizing Control Allocation (NOM PRIO CA) and the Pseudo-Inverse Control Allocation (PINV CA)

Example 4.8 (Control Allocation during an Actuator Fault Case) *The studied system is hexacopter with actuator configuration 1 as in Figure 2.1a corresponding to the Example 4.4 (Hexacopter Input Normalization). Consider its normalized input matrix* $\mathbf{B}_{\nu u}$ *as in* (4.4) *with the control effectiveness matrix*

$$\mathbf{\Lambda} = \hat{\mathbf{\Lambda}} = diag([1, 1, 0.5, 1, 1, 1]).$$

The third actuator has a fault and its effectiveness is given by $\lambda_3 = \hat{\lambda}_3 = 0.5$. *The virtual controls* $\boldsymbol{\nu}$ *and the null space variables* $\mathbf{w}$ *are depicted in Figure 4.10 and the control inputs* $\mathbf{u}$ *in Figure 4.11. The desired virtual control* $\boldsymbol{\nu}_d$ *is unattainable in this case and depicted in Figure 4.10. For comparison purposes, the process of the Nominal Prioritizing Control Allocation (NOM PRIO CA) is displayed along with the results the Pseudo-Inverse Control Allocation (PINV CA) implemented as in Example 4.6. The NOM PRIO CA algorithm allocates* ν_1 *in the first iteration. In the second iteration* $\tilde{\nu}_2$ *is improved but before it reaches zero, the input* u_6 *saturates. Hence, in the third iteration, the variables* ν_3, ν_4, w_1 *and* w_2 *are used to find the next update direction and* $\tilde{\nu}_2$ *is driven to zero. In the iterations 4 to 6, the error* $\tilde{\nu}_3$ *is improved but it never reaches zero. The reason is that the inputs* u_3, u_4 *and* u_2 *saturate one per iteration. Hence, after iteration 6 the number of saturated inputs is* $s = 4$ *and no more updates are made. This example corresponds to the gray sequence in Figure 4.5.*

The final results can be compared in the virtual control space plot in Figure 4.8. Here, the desired virtual control ν_d *and the respective final results from the NOM PRIO CA and the PINV CA are depicted after inverting the transformation* (4.34) *from Example 4.4 (Hexacopter Input Normalization). Furthermore, in each of the two 2-dimensional views, cuts of the attainable control set (ACS) are plotted as explained in Example 4.7 (Control Allocation of an Attainable Virtual Control). Because the desired virtual control* $\boldsymbol{\nu}_d$ *is unattainable, it can be seen that both NOM PRIO CA and PINV CA show virtual control errors. In the case of the PINV CA small errors in* ν_3 *and* ν_4 *and larger errors in* ν_1 *and* ν_2 *arise. In the PINV CA, there*

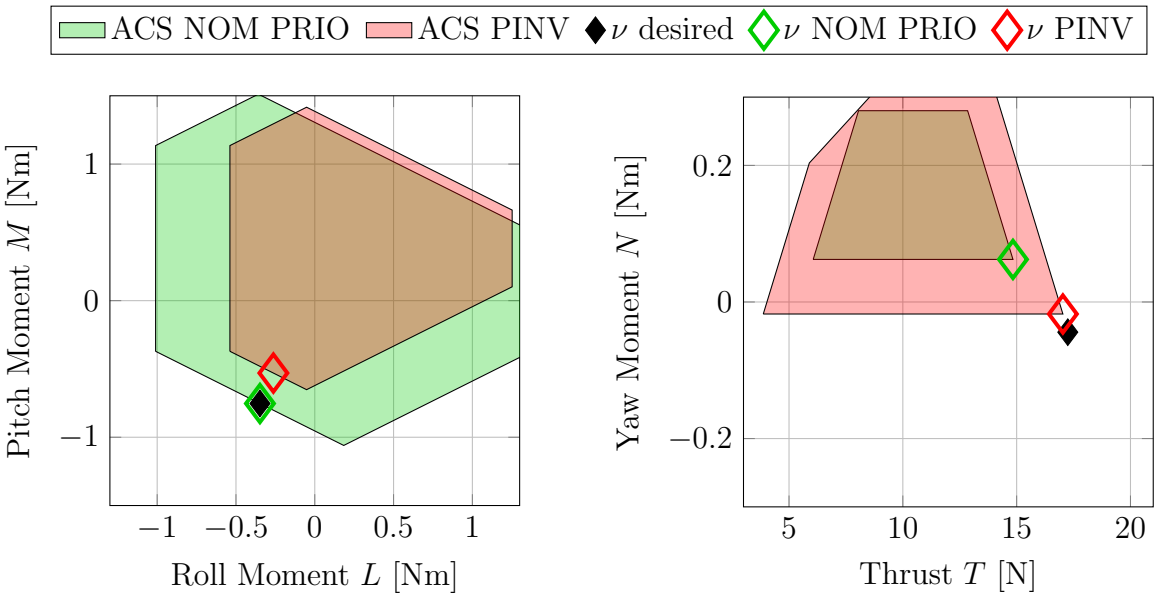


Figure 4.9: Control Allocation Process - Fault Case $\lambda_3 = 0.5$: virtual control space corresponding to the Prioritizing Control Allocation (PRIO CA) and the Pseudo-Inverse Control Allocation (PINV CA)

is no virtual control priority and hence a trade-off is not considered. In contrast, in the case of the NOM PRIO CA, the virtual control errors with higher priority are driven to zero, i.e. ν_1 and ν_2. This is achieved at the cost of a considerably larger error in the low priority virtual controls ν_3 and ν_4 as compared with the PINV CA. The ACS cuts further show that this trade-off is not an artificial characteristic of the presented algorithms but rather a physical limitation of the system. Specifically, it can be seen in the T/N plane that for the given desired L and M values (corresponding to NOM PRIO CA), the NOM PRIO CA computes the best possible T and N within the ACS. The same holds for the PINV CA solution but with the use of different values for L and M. This example shows the characteristic features of the NOM PRIO CA.

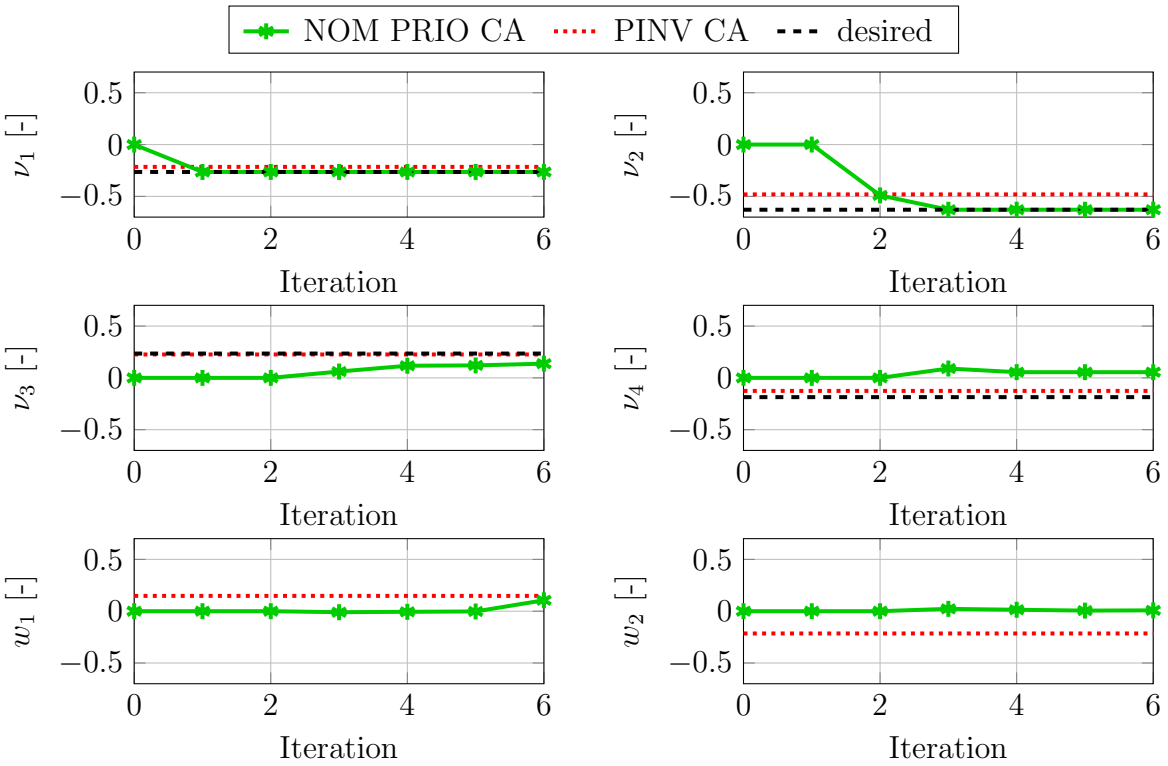


Figure 4.10: Control Allocation Process - Fault Case $\lambda_3 = 0.5$: virtual controls $\boldsymbol{\nu}$ and null space variables $\mathbf{w}$ corresponding to the Nominal Prioritizing Control Allocation (NOM PRIO CA) and the Pseudo-Inverse Control Allocation (PINV CA)

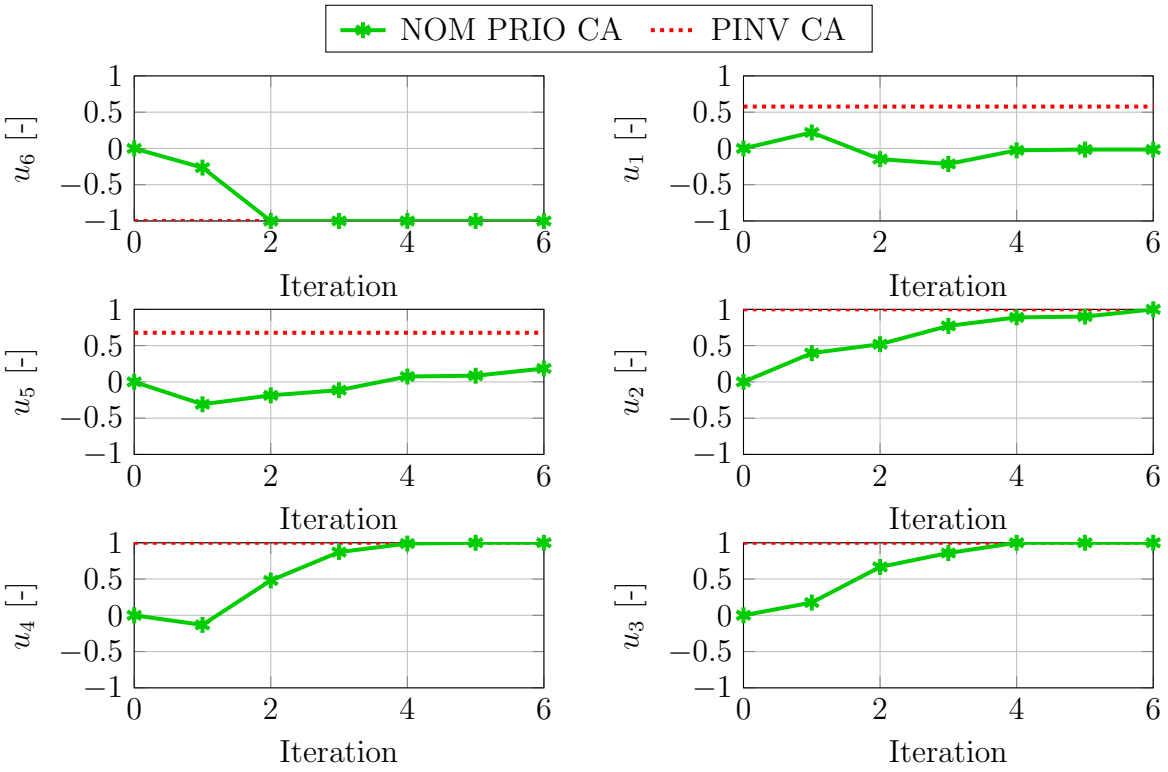


Figure 4.11: Control Allocation Process - Fault Case $\lambda_3 = 0.5$: control inputs $\mathbf{u}$ corresponding to the Nominal Prioritizing Control Allocation (NOM PRIO CA) and the Pseudo-Inverse Control Allocation (PINV CA)

Example 4.9 (Control Allocation during an Actuator Failure Case) *Consider the hexacopter system with actuator configuration 1 as in Figure 2.1a corresponding to the Example 4.4 (Hexacopter Input Normalization). Consider its normalized input matrix* $\mathbf{B}_{\nu u}$ *as in* (4.4) *with the control effectiveness matrix*

$$\mathbf{\Lambda} = \hat{\mathbf{\Lambda}} = diag([1, 1, 0, 1, 1, 1]).$$

The third actuator has completely failed and its effectiveness is given by $\lambda_3 = \hat{\lambda}_3 = 0$*. The virtual controls* $\boldsymbol{\nu}$ *and the null space variables* $\mathbf{w}$ *are depicted in Figure 4.12 and the control inputs* $\mathbf{u}$ *in Figure 4.13. The desired virtual control* $\boldsymbol{\nu}_d$ *is unattainable in this case and depicted in Figure 4.12. For comparison purposes, the process of the Nominal Prioritizing Control Allocation (NOM PRIO CA) is displayed along with the results the Pseudo-Inverse Control Allocation (PINV CA) implemented as in Example 4.6. The NOM PRIO CA algorithm allocates* ν_1*,* ν_2 *and* ν_3 *in the first three iterations. In the fourth iteration* $\tilde{\nu}_4$ *is improved but before it reaches zero, the input* u_6 *saturates. In the next step the algorithm fails to find a new update direction* $\mathbf{c}_k$*, the index is increased* $j = j + 1$ *and the algorithm stops due to the stop condition* $j > r$*.*

The final results can be compared in the virtual control space plot in Figure 4.14. Here, the desired virtual control ν_d *and the respective final results from the NOM PRIO CA and the PINV CA are depicted after inverting the transformation* (4.34) *from Example 4.4 (Hexacopter Input Normalization). Furthermore, in each of the two 2-dimensional views, cuts of the attainable control set (ACS) are plotted as explained in Example 4.7 (Control Allocation of an Attainable Virtual Control). Because the desired virtual control* $\boldsymbol{\nu}_d$ *is unattainable, it can be seen that both NOM PRIO CA and PINV CA show virtual control errors. In this example, the errors of* ν_1*,* ν_2 *and* ν_3 *are driven to zero by the NOM PRIO CA and only the error of* ν_4 *is larger than the one corresponding to the PINV CA. From the ACS NOM PRIO in the* T/N *plane it can be seen that under the constraints of allocating* L *and* M *correctly the best possible* N *is achieved. Hence, it can be concluded that the algorithm failed to find a new update direction* $\mathbf{c}_k$ *because of the system's physical limitation and that it computed the best possible result.*

This example shows a very interesting characteristic of the hexacopter system which is one of the main drivers for the development of the NOM PRIO CA. It can be seen that in a failure case, the desired virtual control $L_d = M_d = 0$ is not achieved by the PINV CA. This is problematic since it would lead to a deviation from e.g. a hover state even in the presented case of a known input matrix. This is an undesirable effect and one may try to compensate it by an integrator or an adaptive controller, but this would only work for static flight states and not for dynamic maneuvers since the control allocation error of the PINV CA changes with varying desired virtual controls.

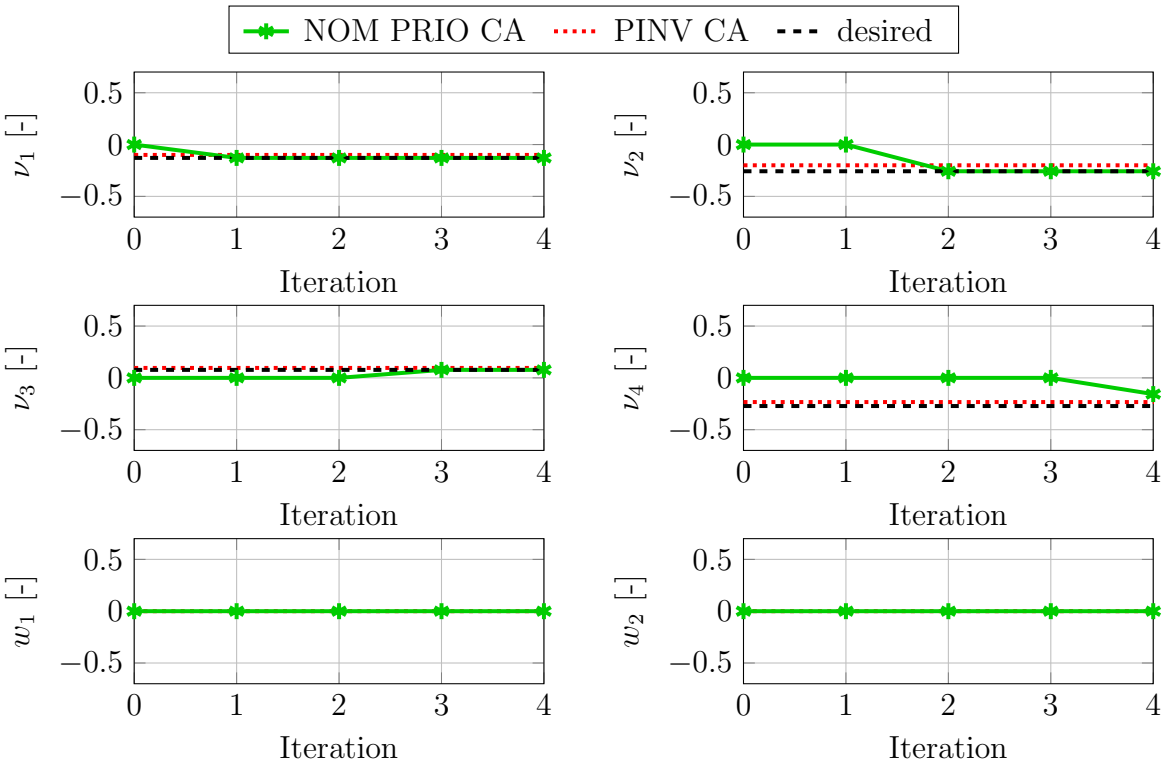


Figure 4.12: Control Allocation Process - Failure Case $\lambda_3 = 0$: virtual controls $\boldsymbol{\nu}$ and null space variables $\mathbf{w}$ corresponding to the Nominal Prioritizing Control Allocation (NOM PRIO CA) and the Pseudo-Inverse Control Allocation (PINV CA)

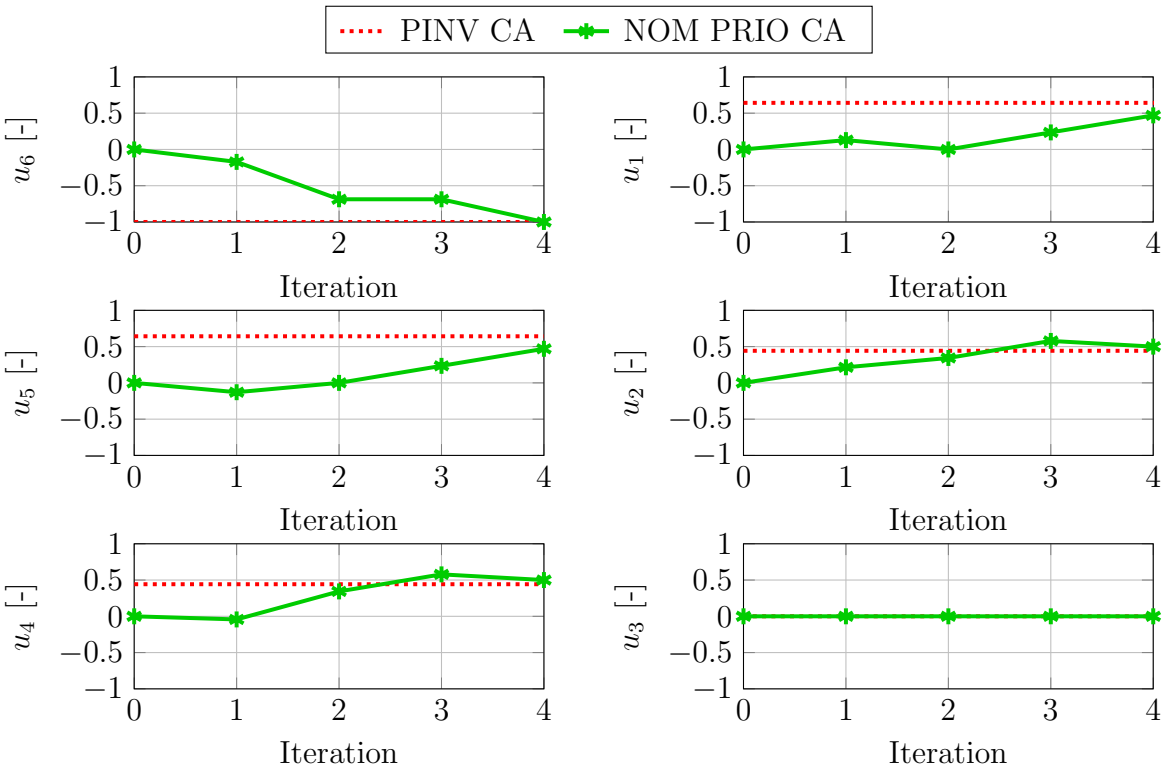


Figure 4.13: Control Allocation Process - Failure Case $\lambda_3 = 0$: control inputs $\mathbf{u}$ corresponding to the Nominal Prioritizing Control Allocation (NOM PRIO CA) and the Pseudo-Inverse Control Allocation (PINV CA)

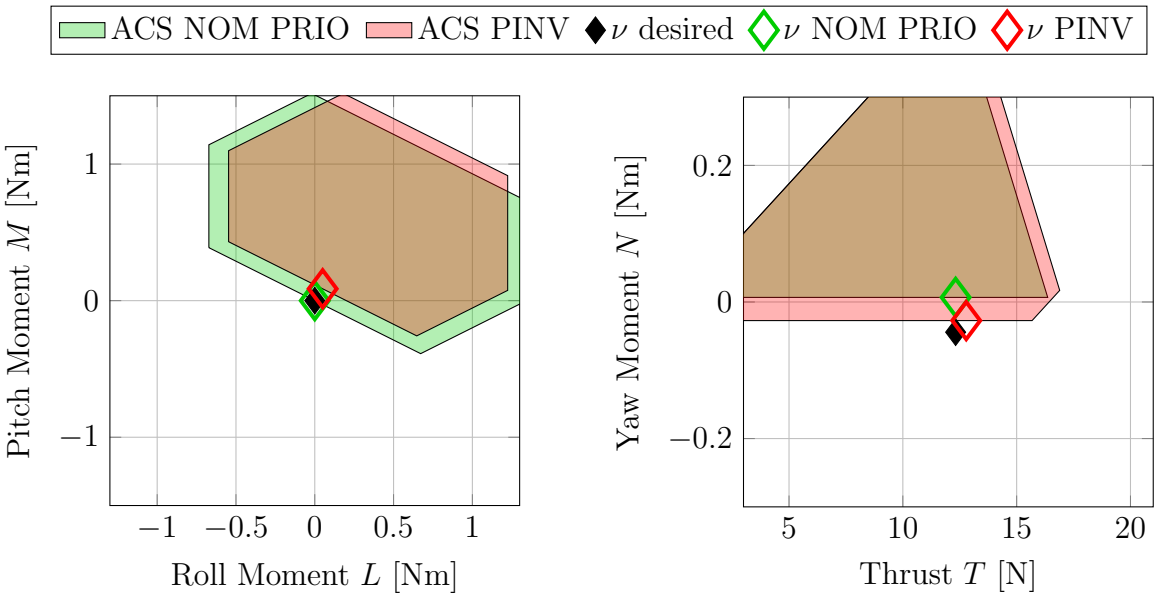


Figure 4.14: Control Allocation Process - Failure Case $\lambda_3 = 0$: virtual control space corresponding to the Prioritizing Control Allocation (PRIO CA) and the Pseudo-Inverse Control Allocation (PINV CA)

4.3.5 Update Direction - Singular Cases

As noted in the last section, the update direction $\mathbf{c}_k$ (4.48) can be found if $\mathbf{G}_{j,s} \in \mathbb{R}^{s\times(m-j)}$ has full row rank using (4.47). This leads to the solution of the equality condition

$$\operatorname{sgn}(u_i) \cdot \mathbf{c}_{k,i} = 0, \qquad \forall i \in \mathcal{S}.$$

This is not a necessary condition and therefore other solutions to the update direction $\mathbf{c}_k$ can be found. The necessary condition is (4.46) such that the update direction of the saturated inputs $\mathbf{c}_{k,s}$ does not point in the same direction as the input itself. By inserting (4.45) into (4.46) it follows that

$$\operatorname{sgn}(u_i) \cdot c_{k,i} = \operatorname{sgn}(u_i) \cdot \operatorname{sgn}(\tilde{\nu}_{k,j}) \cdot p_{j,i} + \operatorname{sgn}(u_i) \cdot \mathbf{G}_{j,i} \cdot \mathbf{h}_k \leq 0, \qquad \forall i \in \mathcal{S},$$

which can be written in matrix form as

$$\mathbf{A} \cdot \mathbf{x} \leq \mathbf{b}, \tag{4.54}$$

with the following definitions

$$\begin{aligned}
\mathbf{x} &= \mathbf{h}_k \in \mathbb{R}^{m-j}, \\
\mathbf{A} &= diag(\operatorname{sgn}(\mathbf{u}_s)) \cdot \mathbf{G}_{j,s} \in \mathbb{R}^{s\times(m-j)}, \\
\mathbf{b} &= -\operatorname{sgn}(\tilde{\nu}_{k,j}) \cdot diag(\operatorname{sgn}(\mathbf{u}_s)) \cdot \mathbf{p}_{j,s} \in \mathbb{R}^{s}.
\end{aligned}$$

From this representation, it is clear that the system of linear inequalities $\mathbf{A} \cdot \mathbf{x} \leq \mathbf{b}$ has in general more solutions than the system of linear equalities $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$. The singular case occurs whenever $\operatorname{rank}(\mathbf{G}_{j,s}) = \operatorname{rank}(\mathbf{A}) < s$, that is whenever we have more constraints than degrees of freedom. In the presented control allocation, no attempt will be made to solve the linear inequality system in general but some specific low

dimensional cases will be addressed. It is important to note that the main algorithm presented in the last section, has the characteristic of improving the result every update step (4.38). By adding the following singularity cases, the control allocation is improved in cases where otherwise no update vector $\mathbf{c}_k$ would have been found (see Figure 4.4).

The considered singular cases are the following:

- Case $s = 1$: $\mathbf{A} \in \mathbb{R}^{1\times(m-j)}$ is a row vector and $\operatorname{rank}(\mathbf{A}) < 1$ can only happen if $\mathbf{A}$ is a zero vector. In this case, the system (4.54) can only be solved if $0 \leq b$ and $\mathbf{x}$ can be chosen as $\mathbf{x} = \mathbf{h}_k = \mathbf{0}$. By inserting it into (4.44), the update vector becomes $\mathbf{c}_k = \operatorname{sgn}(\tilde{\nu}_{k,j}) \cdot \mathbf{p}_j$.
- Case $s = 2$: $\mathbf{A} \in \mathbb{R}^{2\times(m-j)}$ and $\operatorname{rank}(\mathbf{A}) < 2$ can be divided into several sub-cases:
  - $\mathbf{A} = \mathbf{0}$: the system (4.54) can only be solved if $\mathbf{0} \leq \mathbf{b}$ and $\mathbf{x}$ can be chosen as $\mathbf{x} = \mathbf{h}_k = \mathbf{0}$. By inserting it into (4.44), the update vector becomes $\mathbf{c}_k = \operatorname{sgn}(\tilde{\nu}_{k,j}) \cdot \mathbf{p}_j$.
  - The first row $\mathbf{A}_{r1} \in \mathbb{R}^{1\times(m-j)}$ of $\mathbf{A}$ is a zero matrix: there is only a solution if $0 \leq b_1$. In this case, the update vector $\mathbf{c}_k$ can be computed by selecting $\mathbf{x}$ such that $\mathbf{A}_{r2} \cdot \mathbf{x} = b_2$. This is equivalent to solving the $s = 1$ case for a full row rank $\mathbf{G}_{j,s}$ with (4.47) and (4.48).
  - The second row $\mathbf{A}_{r2} \in \mathbb{R}^{1\times(m-j)}$ of $\mathbf{A}$ is a zero matrix: this is analogous to the last case.
  - The rows $\mathbf{A}_{r1}$ and $\mathbf{A}_{r2}$ are non-zero and linear dependent (parallel): If there is a solution, either $\mathbf{A}_{r1} \cdot \mathbf{x} = b_1$ or $\mathbf{A}_{r2} \cdot \mathbf{x} = b_2$ solve the problem. Hence, $\mathbf{x}$ can be computed to solve for one equation and the second inequality is tested afterwards. In Appendix C (Control Allocation) it is shown that this approach leads to a solution if it exists.

The presented cases can be divided into two groups. One is the zero matrix case $\mathbf{A} = \mathbf{0}$ and the other one is the dimension reduction case. In the case that $\mathbf{A} = \mathbf{0}$, the selection of the help variables does not affect the update vector $\mathbf{c}_k$ along the saturated inputs. Therefore a solution only exists if the desired update direction p_j does not increase the magnitude of the saturated inputs. The second case corresponds to the last three sub cases for $s = 2$. A solution can be searched by solving the $s = 1$ problem for the two possible cases and subsequently checking the satisfaction of the other inequality.

This can be analogously applied for higher order cases, but the amount of cases to be tested increases. For example in the case $s = 3$ there are three possible two dimensional solution candidates and each of them could be singular itself. In the following, an algorithm that searches for all possible reduced-order solutions is presented. Although it is not intended for running on the embedded target, it serves to demonstrate the theoretical achievable performance of the PRIO CA algorithm. Therefore it is called Ideal Prioritizing Control Allocation (ID PRIO CA) algorithm.

Let $e \in \mathbb{N}_0$ be the number of the considered equation constraints that are used for computing the update vector $\mathbf{c}_k$ using the equality solution as in (4.48). In a first step, all constraints are considered and hence $e = s = |\mathcal{S}|$, where $|\mathcal{S}|$ is the cardinality of the set $\mathcal{S}$. In the following steps, always less constraints are considered in order to

cover singularity cases and e is reduced. For every e, there exist $\binom{s}{e}$ possible unordered subsets $\mathcal{E}_q$ of $\mathcal{S}$ with the set counter $q = 1 \dots \binom{s}{e}$. Each set $\mathcal{E}_q$ contains e elements. For every set $\mathcal{E}_q$, the sub vector of the saturated inputs $\mathbf{u}_s$ can be defined by stacking the control inputs as

$$\mathbf{u}_q = col(u_i) \in \mathbb{R}^e, i \in \mathcal{E}_q.$$

The function $col(\cdot)$ stacks the scalars into a column vector. The idea is to compute an update direction candidate $\overline{\mathbf{c}}_k$ considering only the equation constraints corresponding to the set $\mathcal{E}_q$ in the first step and then checking the conditions

$$\operatorname{sgn}(u_i) \cdot \overline{\mathbf{c}}_{k,i} \leq 0, \qquad \forall i \in \mathcal{S}, \notin \mathcal{E}_q. \tag{4.55}$$

This idea translates to the following algorithm. First, consider only the subsystem corresponding to the reduced set of saturated inputs

$$\mathbf{c}_{k,q} = \operatorname{sgn}(\tilde{\nu}_{k,j}) \cdot \mathbf{p}_{j,q} + \mathbf{G}_{j,q} \cdot \mathbf{h}_k,$$

where $\mathbf{c}_{k,q}$, $\mathbf{p}_{j,q} \in \mathbb{R}^e$ and $\mathbf{G}_{j,q} \in \mathbb{R}^{e \times (m-j)}$ correspond to the rows of the saturated inputs $i \in \mathcal{E}_q$. Following the ideas presented in Section 4.3.4, $\mathbf{h}_k$ is selected such that $\mathbf{c}_{k,q} = \mathbf{0}$ as

$$\mathbf{h}_k = \operatorname{sgn}(\tilde{\nu}_{k,j}) \cdot \mathbf{G}_{j,q}^{+} \cdot (-\mathbf{p}_{j,q}).$$

This leads to the update direction candidate

$$\overline{\mathbf{c}}_k = \operatorname{sgn}(\tilde{\nu}_{k,j}) \cdot \mathbf{p}_j + \operatorname{sgn}(\tilde{\nu}_{k,j}) \cdot \mathbf{G}_j \cdot \mathbf{G}_{j,q}^{+} \cdot (-\mathbf{p}_{j,q}). \tag{4.56}$$

If the conditions (4.55) are satisfied, an update direction has been found, otherwise it is discarded.

The complete algorithm for computing the update direction $\mathbf{c}_k$ is depicted in Figure 4.15. The algorithm is initialized with $e = s$ which corresponds to the non-singular case presented in the last section. In the first step, the unordered sets $\mathcal{E}_q$ of the considered equations systems are computed. For each of these sets, a candidate update direction $\overline{\mathbf{c}}_k$ is computed as in (4.56) and the conditions (4.55) tested. If the conditions are met, an update direction is found, otherwise the counter $q = q + 1$ is increased and the next set $\mathcal{E}_q$ is checked. If all the sets $\mathcal{E}_q$ for a given number of considered equations e have been tested, this number is reduced $e = e - 1$. In this way the algorithm covers the non-singular case first and step by step tries to solve the dimension reduction cases. The Ideal Prioritizing Control Allocation (ID PRIO CA) corresponds to the NOM PRIO CA algorithm as in Figure 4.4 where the computation of the update direction $\mathbf{c}_k$ is exchanged by the presented algorithm corresponding to Figure 4.15.

4.3.6 Implementation Aspects

This section addresses aspects of the implementation of the PRIO CA algorithm within the embedded computing unit of the aircraft, i.e. it addresses objective 5 of this thesis. Even the best control allocation is not very useful if it exceeds the limitations of the given computing unit. In this case the focus is real-time operation, which means that the whole computation of the control law including the control allocation should not exceed the selected sampling time. Therefore, several changes to the presented algorithm are introduced in order to limit the maximum execution time of the control allocation solution. The changes are

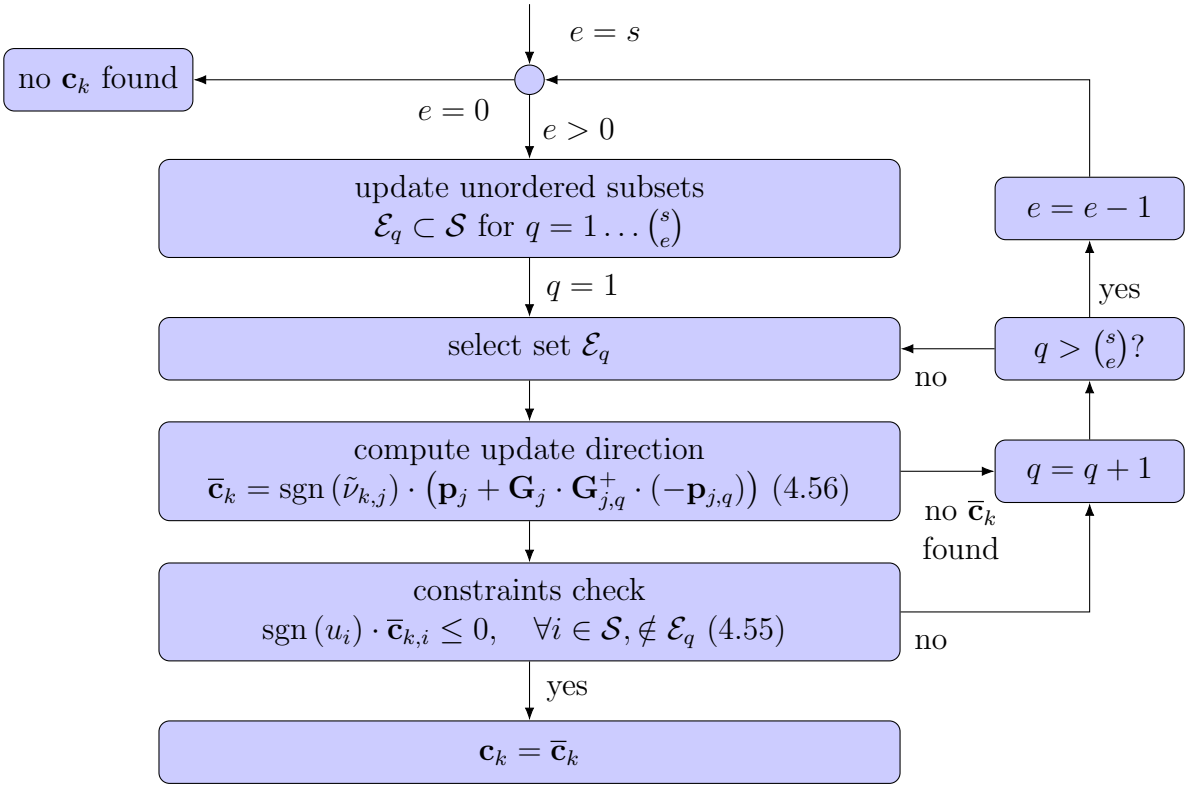


Figure 4.15: Flow Chart of Update Direction Computation - Ideal Prioritizing Control Allocation (ID PRIO CA)

- A pseudo-inverse dimension limitation,
- A higher dimensional alternative for the computation of the update direction $\mathbf{c}_k$,
- And a modification for virtual controls with the same priority.

The Prioritizing Control Allocation (PRIO CA) implemented on the multirotor system includes all the modifications and is depicted in Figure 4.16. Its performance is subsequently compared with the Nominal Prioritizing Control Allocation (PRIO CA) presented in Section 4.3.4 and the Ideal Prioritizing Control Allocation (PRIO CA) presented in Section 4.3.5.

Pseudo-Inverse Dimension Limitation

The computation of the pseudo inverse $\mathbf{G}^+_{j,s}$ in (4.48) needs in general numerical iterative algorithms and a maximum execution time cannot be directly given. Therefore, in this thesis this operation is limited to a maximum of two saturated inputs $s \leq 2$, i.e. limited to 2 rows of $\mathbf{G}_{j,s}$. If $\mathbf{G}_{j,s}$ has a full row rank, its pseudo inverse can be computed as

$$\mathbf{G}^+_{j,s} = \mathbf{G}^T_{j,s} \cdot \left(\mathbf{G}_{j,s} \cdot \mathbf{G}^T_{j,s}\right)^{-1}.$$

In this case the inverse of a $\mathbb{R}^{2\times 2}$ matrix or a scalar is used without an iterative computation. If $\mathbf{G}_{j,s}$ does not have a full rank, the algorithm continues with lower dimension solutions as presented in the last section. In this case $\mathbf{G}^+_{j,q} \in \mathbb{R}^{(m-j)\times e}$ in (4.56) is a row

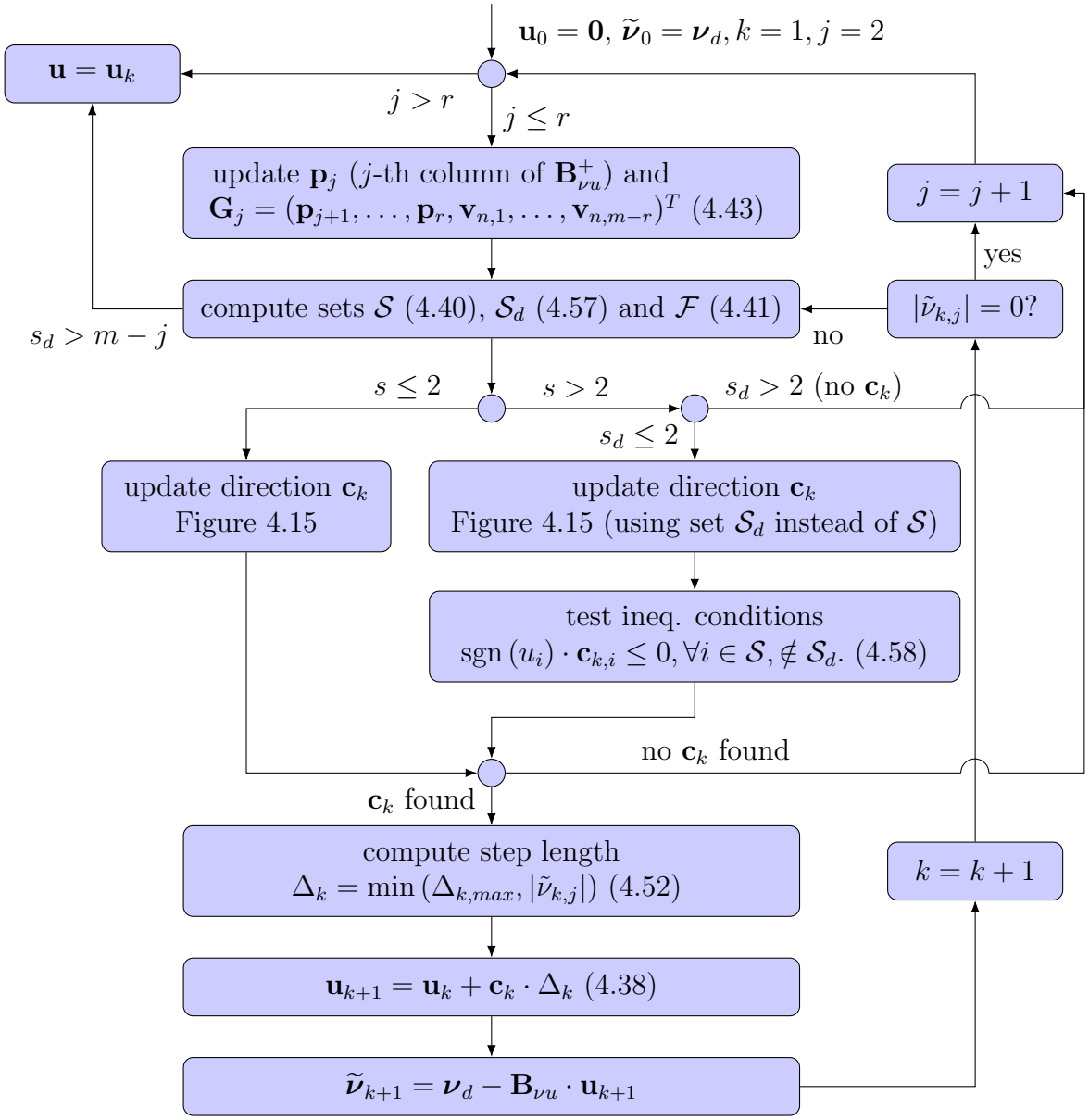


Figure 4.16: Flow Chart of Prioritized Control Allocation (PRIO CA)

vector with $e = 1$. The presented case corresponds to the left branch of the program flow in Figure 4.16.

The limitation to a maximum of two saturated inputs $s \leq 2$ presented in this section supposes a reduction of the control allocation performance but it permits the real-time implementation of the algorithm. The performance loss will be subsequently evaluated.

Update Direction - Higher Dimensional Alternative

For cases in which the number of saturated inputs is $s \geq 3$, the insights of the singular cases analysis in Section 4.3.5 are used. In order to improve the basic algorithm but avoid resorting to iterative numerical methods, the desired update direction $\mathbf{p}_j$ is taken

into account. Let the set of inputs that are directionally saturated be defined as

$$\mathcal{S}_d := \{i \in \mathbb{N} \mid |u_i| \geq 1 \text{ and } \operatorname{sgn}(u_i) \cdot \operatorname{sgn}(\tilde{\nu}_{k,j}) \cdot p_{j,i} > 0, \forall i = 1, \ldots, m\}. \tag{4.57}$$

The set $\mathcal{S}_d$ is a subset of $\mathcal{S}$ which excludes input indices whose desired update direction $\operatorname{sgn}(\tilde{\nu}_{k,j}) \cdot p_{j,i}$ do not increase the input's magnitude. By stacking control inputs corresponding to $\mathcal{S}_d$ into vector form, the directionally saturated inputs $\mathbf{u}_{s_d}$ are defined as

$$\mathbf{u}_{s_d} = col(u_i) \in \mathbb{R}^{s_d}, i \in \mathcal{S}_d.$$

The idea is to solve the update direction computation $\mathbf{c}_k$ considering only the set $\mathcal{S}_d$ in the first step and then checking the conditions

$$\operatorname{sgn}(u_i) \cdot \mathbf{c}_{k,i} \leq 0, \qquad \forall i \in \mathcal{S}, \notin \mathcal{S}_d. \tag{4.58}$$

In this way, it is possible to search an update direction in a lower dimensional space. The existence of a solution is not always guaranteed but it increases the possibilities of finding one. This idea translates to the following algorithm. First, consider only the subsystem corresponding to the directionally saturated inputs

$$\mathbf{c}_{k,s_d} = \operatorname{sgn}(\tilde{\nu}_{k,j}) \cdot \mathbf{p}_{j,s_d} + \mathbf{G}_{j,s_d} \cdot \mathbf{h}_k,$$

where $\mathbf{c}_{k,s_d}$, $\mathbf{p}_{j,s_d} \in \mathbb{R}^{s_d}$ and $\mathbf{G}_{j,s_d} \in \mathbb{R}^{s_d \times (m-j)}$ correspond to the rows of the saturated inputs $i \in \mathcal{S}_d$. The help variable is computed then as in Section 4.3.5 (Update Direction - Singular Cases) using the set $\mathcal{S}_d$ instead of the $\mathcal{S}$. For the non-singular case this corresponds to the selection derived in Section 4.3.4 (Main Algorithm). Hence, the help variable is selected as

$$\mathbf{h}_k = \operatorname{sgn}(\tilde{\nu}_{k,j}) \cdot \mathbf{G}^+_{j,s_d} \cdot (-\mathbf{p}_{j,s_d}),$$

which leads to the candidate update direction

$$\mathbf{c}_k = \operatorname{sgn}(\tilde{\nu}_{k,j}) \cdot \mathbf{p}_j + \operatorname{sgn}(\tilde{\nu}_{k,j}) \cdot \mathbf{G}_j \cdot \mathbf{G}^+_{j,s_d} \cdot (-\mathbf{p}_{j,s_d}). \tag{4.59}$$

If the conditions (4.58) are satisfied, an update direction is found, otherwise it is discarded.

This approach allows for searching an update direction for cases where $s \geq 3$ and $s_d \leq 2$ while using a pseudo-inverse of a lower dimension s_d compared to the dimension of the original algorithm s. For cases where $s_d \geq 3$, no attempt of searching an update direction is made. The presented case corresponds to the right branch of the program flow in Figure 4.16.

Virtual Controls with same Priority

For the hexarotor and for most symmetric multirotor systems, roll and pitch moments have the same priority. Therefore, a small modification is made to the NOM PRIO CA algorithm. Let $\mathbf{p}_L \in \mathbb{R}^m$ and $\mathbf{p}_M \in \mathbb{R}^m$ be the columns of the pseudo inverse $\mathbf{B}^+_{\nu u}$ that correspond to the roll moment L and the pitch moment M. Instead of allocating them sequentially, the following desired update direction is used

$$\mathbf{p}_{LM} = \begin{bmatrix} \mathbf{p}_L, & \mathbf{p}_M \end{bmatrix} \cdot \begin{bmatrix} L_d \\ M_d \end{bmatrix} \tag{4.60}$$

with the desired magnitude of $\nu_{LM,d} = 1$. This approach can be generally used for any virtual controls that have the same priority. The advantage is that the amount of update steps is reduced. In (4.60) two virtual controls are put together and therefore the iterations are reduced by one. This can be implemented by setting the initial value $j = 2$, the desired update direction as $\mathbf{p}_2 = \mathbf{P}_{LM}$ and the desired magnitude as $\nu_2 = 1$. The step $j = 1$ is not performed.

Comparison with Nominal Prioritizing Control Allocation

The Prioritizing Control Allocation (PRIO CA) includes all the presented modifications in Section 4.3.6 and is the version of the algorithm that runs in the embedded computer platform which used for the presented flight experiments. In the following, its performance is compared with the Nominal Prioritizing Control Allocation (PRIO CA) presented in Section 4.3.4 and the Ideal Prioritizing Control Allocation (PRIO CA) presented in Section 4.3.5.

Example 4.10 (Control Allocation of an Unattainable Virtual Control) *The studied system is the hexacopter with actuator configuration 1 as in Figure 2.1a corresponding to the Example 4.4 (Hexacopter Input Normalization). Consider its normalized input matrix $\mathbf{B}_{\nu u}$ as in (4.4) with the control effectiveness matrix $\hat{\mathbf{\Lambda}} = \mathbf{\Lambda} = \mathbf{I}$. The considered desired virtual control $\boldsymbol{\nu}_d$ is unattainable. The virtual controls $\boldsymbol{\nu}$, the null space variables $\mathbf{w}$ and the control inputs $\mathbf{u}$ corresponding to the Nominal Prioritizing Control Allocation (NOM PRIO CA) and the Prioritizing Control Allocation (PRIO CA) are depicted in Figures 4.18 and 4.19.*

The NOM PRIO CA algorithm allocates ν_1 and ν_2 in the first and second iteration respectively. In the following 4 iterations, $\tilde{\nu}_3$ is improved and only reaches zero at the last iteration. In the previous iterations the control inputs u_5, u_4 and u_6 saturate one per iteration. Since at this last step the number of saturated inputs is $s = 3 > m - j = 6 - 4 = 2$, no update is made to improve the virtual control ν_4. The PRIO CA algorithm allocates ν_1 and ν_2 simultaneously in the first iteration as described in Section 'Virtual Controls with same Priority'. In the iterations 2,3 and 4, $\tilde{\nu}_3$ is improved and the control inputs u_5, u_4 and u_6 saturate one per iteration in the similar way as for the NOM PRIO CA. At this point, $|\tilde{\nu}_3| > 0$ and 3 inputs are saturated. Hence, the nominal computation of the update direction $\mathbf{c}_k$ (4.48) is not used due to the pseudo-inverse dimension limitation and the approach described in Section 'Update Direction - Higher Dimensional Alternative' is used instead. No further improvements are made for $\tilde{\nu}_3$ because the inverse of a $\mathbb{R}^{3\times3}$ matrix is needed for computing a valid update direction $\mathbf{c}_k$. In the last step, ν_4 is allocated and an update direction $\mathbf{c}_k$ using (4.59) is found without the inverse of a $\mathbb{R}^{3\times3}$ matrix such that $\tilde{\nu}_4$ is driven to zero. This is possible because the linear inequality solution is considered instead of the linear equality as explained in Section 4.3.5.

The final results can be compared in the virtual control space plot in Figure 4.17. Here, the desired virtual control ν_d and the respective final results from the NOM PRIO CA and the PRIO CA are depicted after inverting the transformation (4.34) from Example 4.4 (Hexacopter Input Normalization). Furthermore, in each of the two 2-dimensional views, cuts of the attainable control set (ACS) are plotted as explained in Example 4.7 (Control Allocation of an Attainable Virtual Control). It can be seen that both, the PRIO CA and the NOM PRIO CA, satisfy the L and M commands. Therefore, the attainable control set (ACS) in the T/N plane is equal for both approaches. The NOM PRIO CA achieves the commanded thrust at the cost of a larger N error. The PRIO CA cannot reduce the thrust allocation error to zero, since the matrix inverse is limited to a $\mathbb{R}^{2\times2}$ matrix and 3 inputs are saturated. This control allocation

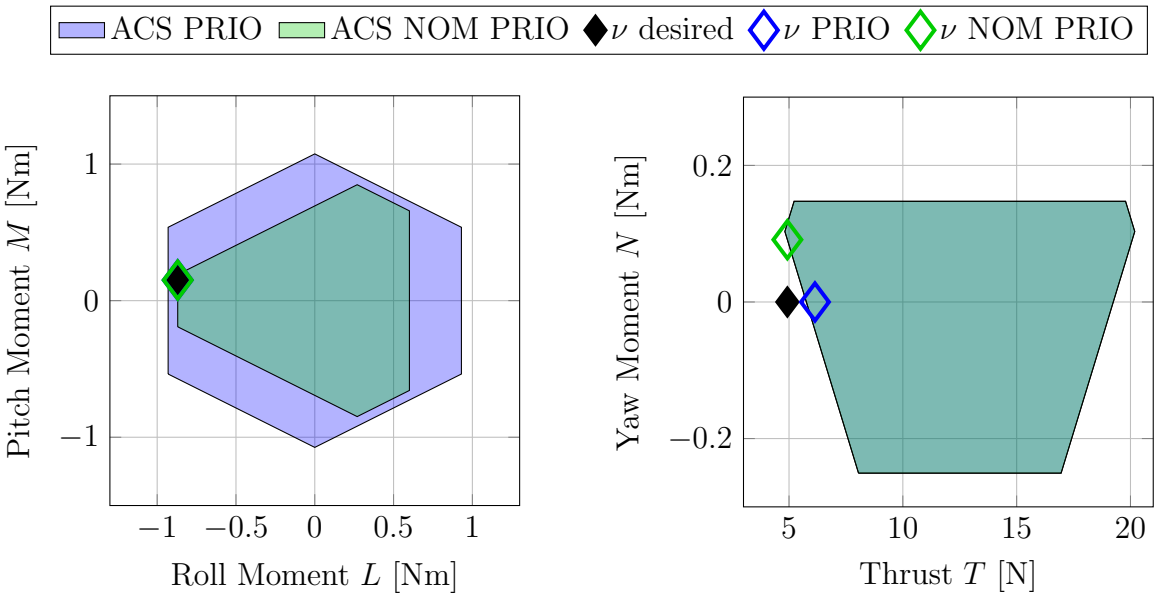


Figure 4.17: Control Allocation Process - Unattainable Solution: virtual control space corresponding to the Prioritizing Control Allocation (PRIO CA) and the Nominal Prioritizing Control Allocation (NOM PRIO CA)

performance reduction is accepted in exchange of avoiding the computation of high order matrix inverses. Because of this missing thrust allocation step, PRIO CA is able to reduce the N error in the following step.

Example 4.10 shows that the trade-off between the control allocation performance and the needed computational resources can be steered by the modifications presented in Section 4.3.6. For different system characteristics, the presented modifications could be adapted to achieve the desired results. Furthermore, for the cases presented in the Examples 4.7 (Control Allocation of an Attainable Virtual Control), 4.8 (Control Allocation during an Actuator Fault Case) and 4.9 (Control Allocation during an Actuator Failure Case), the maximum difference of the computed control inputs between the PRIO CA and the NOM PRIO CA is $1.2 \cdot 10^{-3}$ and therefore no comparison is presented.

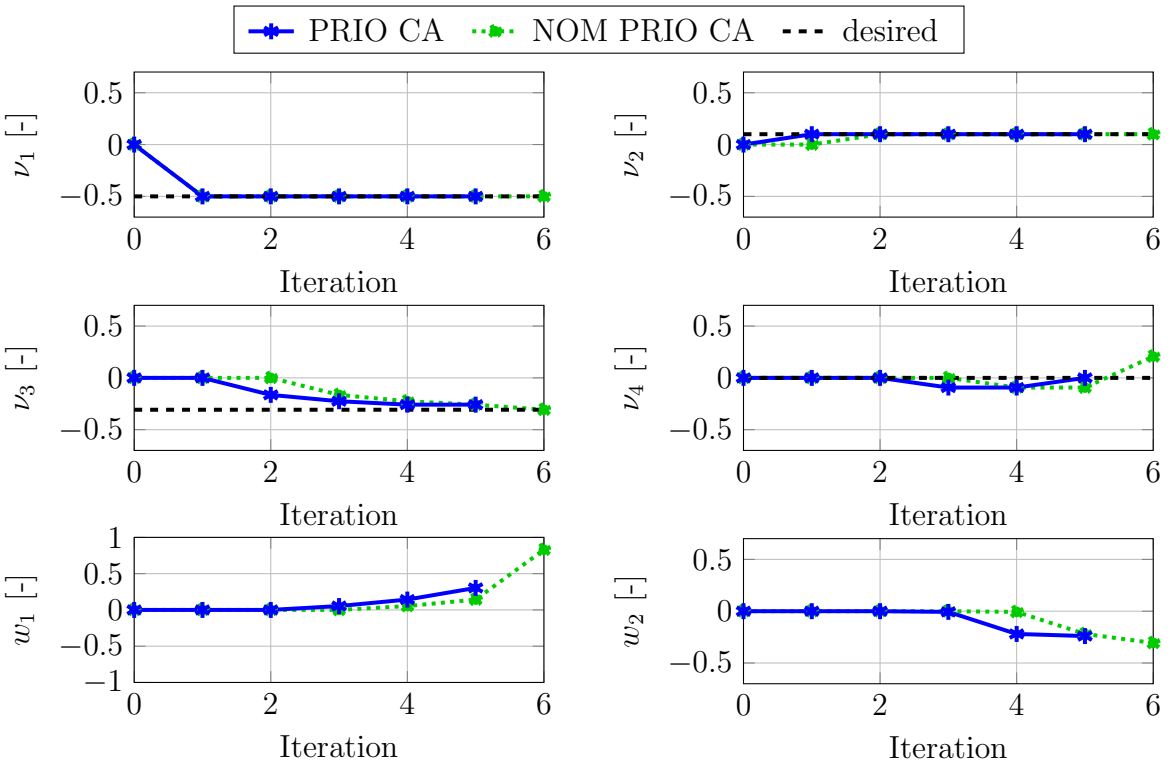


Figure 4.18: Control Allocation Process - Unattainable Solution: virtual controls $\boldsymbol{\nu}$ and null space variables $\mathbf{w}$ corresponding to the Nominal Prioritizing Control Allocation (NOM PRIO CA) and the Prioritizing Control Allocation (PRIO CA)

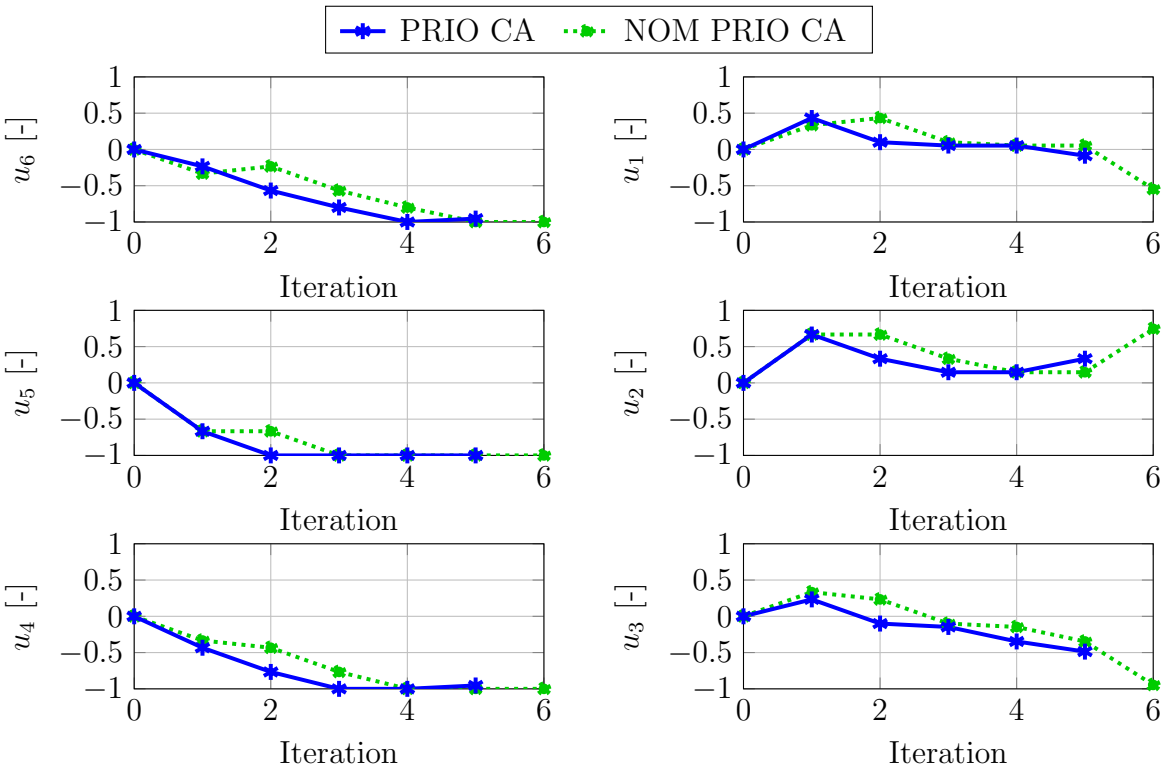


Figure 4.19: Control Allocation Process - Unattainable Solution: control inputs $\mathbf{u}$ corresponding to the Nominal Prioritizing Control Allocation (NOM PRIO CA) and the Prioritizing Control Allocation (PRIO CA)

Numerical Comparison for a Large Set of Test Points

In the following, the control allocation performance is compared for the two hexacopter configurations depicted in Figure 2.1 for different failure cases. The results are shown in Figures 4.20 and 4.21. For each of the configurations, a 4-dimensional grid of desired virtual controls is created. The grid along each virtual control has 100 points equally spaced between the minimum and maximum achievable value in a fault-free configuration. Out of this initial grid, only the attainable virtual controls of the nominal configuration are taken into account and gives a total of approximately $12 \cdot 10^6$ test points (or 12% of the original points). The grid has been selected this way because it significantly reduces the test points and because it is expected that the controller is designed such that the desired virtual controls lie within the nominal attainable control set (ACS). The attainability check is done as a point inclusion test within the ACS using the Toolbox MPT 3.0 [68]. For every point in the grid, the compared control allocation methods are evaluated in both hexacopter configurations in different failure scenarios. In configuration 1, a failure of actuator 3 (rear-right) and of actuator 5 (center-left) are considered. In configuration 2, a failure of actuator 1 (front-right), of actuator 3 (rear-right) and of actuator 5 (center-left) are considered. Due to symmetry considerations, the selected scenarios are representative for all possible single actuator failures.

The selected performance metrics correspond to the percentage of correctly allocated virtual controls for the following three groups:

- *LMTN*: The four virtual controls are simultaneously correctly allocated, i.e. thrust (T), roll (L), pitch (M) and yaw moments (N).
- *LMT*: Thrust (T), roll (L) and pitch (M) moments are simultaneously correctly allocated.
- *LM*: Roll (L) and pitch (M) moments are simultaneously correctly allocated.

A virtual control is considered correctly allocated if the magnitude of the error is less or equal the tolerance of 10^{-6}.

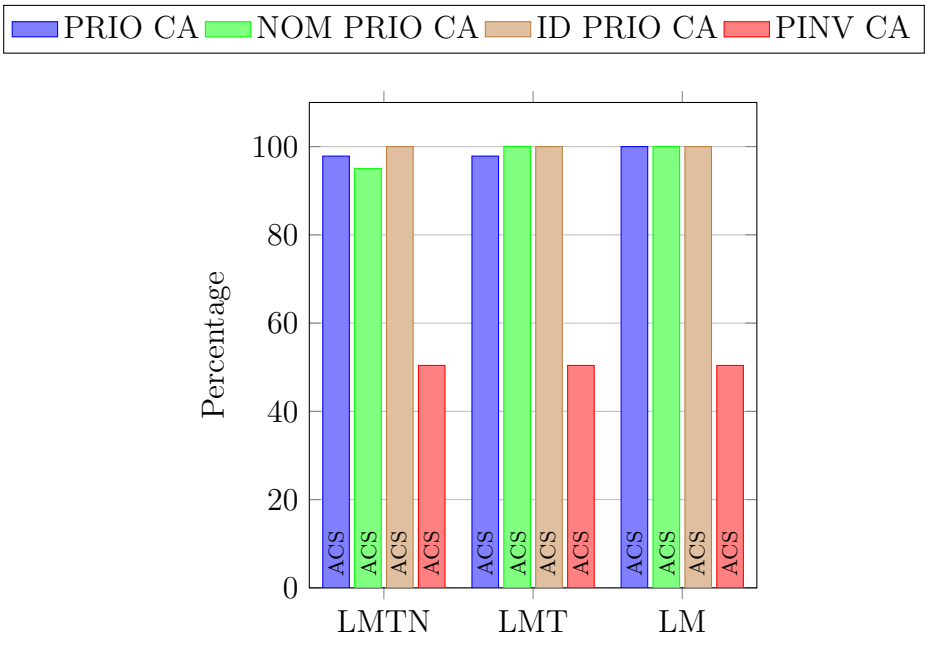
In Figure 4.20, the results for hexacopter configuration 1 are depicted. In Figure 4.20a, it can be clearly seen that in the nominal case, the prioritizing control allocation approaches are superior than the pseudo-inverse control allocation (PINV CA). It can be seen that the Ideal Prioritizing Control Allocation (ID PRIO CA) is able to allocate all the test points correctly. The Nominal Prioritizing Control Allocation (NOM PRIO CA) is able to allocate thrust (T), roll (L) and pitch (M) moments for all the test points. The yaw moment (N) is not always correctly allocated. The difference between these two approaches is the consideration of the linear inequality solution instead of only the linear equality for computing the update direction $\mathbf{c}_k$ as explained in Section 4.3.5. The Prioritizing Control Allocation (PRIO CA) as implemented for the flight tests is able to allocate roll (L) and pitch (M) moments for all the test points. Thrust (T) and yaw moments (N) are not always correctly allocated. This limitation corresponds to the previously explained trade-off between computational effort and control allocation performance. Finally, in the case of PINV CA all three metrics have the same value. This is mainly because if saturation occurs, a deviation from the desired virtual control is unavoidable using the PINV CA. In the example of the hexacopter, this gets aggravated by the fact that it is a highly coupled system in the sense that every control input directly affects at least three virtual controls.

In Figure 4.20b, the failure of actuator 3 is addressed. In this case 32.10% of the test points are attainable. Under these conditions, the main characteristics of the PRIO CA can be seen. For all three implementation variants it holds that high priority virtual controls get correctly allocated in a higher percentage than low priority virtual controls. The performance difference between the PINV CA and the priority-based control allocations is smaller in the *LMTN* case than in the other cases. This is expected since the PINV CA does not take the virtual control priority into account. In the *LMT* and the *LM* cases, horizontal lines mark the limit between the cases that are inside the ACS (underneath) or outside the ACS (above). The nomenclature unattainable control set (UCS) is used in the figure to denote being outside of the ACS. For the *LMTN* case, there is no line since the virtual controls can only be correctly allocated if they are attainable and hence all lie within the ACS. For the actuator 5 failure case very similar results can be seen in Figure 4.20c. In this case 32.23% of the test points are attainable.

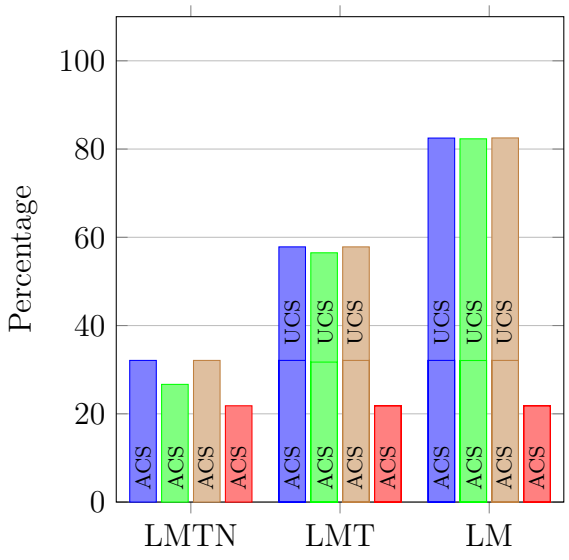
In Figure 4.21, the results for hexacopter configuration 2 are depicted. Figure 4.21a shows the nominal case, Figure 4.21b shows the actuator 1 failure case with 18.71% attainable test points, Figure 4.21c shows the actuator 3 failure case with 38.95% attainable test points and Figure 4.21d shows the actuator 5 failure case with 38.80% attainable test points. In general terms, the results are similar to the hexacopter configuration 1 case. Nevertheless, there are several differences. As expected from the discussion in Section 2.3 and the comparison of the T/N plane of the ACS in Figures 2.4 and 2.5, the percentage of test points within the ACS compared to hexacopter configuration 1 is smaller for actuators the front actuators (1 and 6) and larger otherwise. Furthermore, in the nominal case, a larger difference between NOM PRIO CA and ID PRIO CA can be seen. This is due to a higher occurrence of singular cases in the update direction $\mathbf{c}_k$ computation (4.48) as explained in Section 4.3.5.

The main advantage of the PRIO CA algorithm can be seen in Figures 4.20 and 4.21. It enables the utilization of the unattainable control set (UCS) for improving the allocation of high priority virtual controls. In our specific context, for the LMT case it allows for the utilization of at least 50% of the nominal ACS and thus enabling controlled flight during failure scenarios. For the LMT case even 75% of the nominal ACS is correctly allocated. Furthermore, the PRIO CA is clearly superior than the PINV CA in nominal as well as in failure scenarios. For all the failure scenarios, in the LMT case the PRIO CA correctly allocates at least twice as much test points as the PINV CA and in the LM case at least 3 times as much.

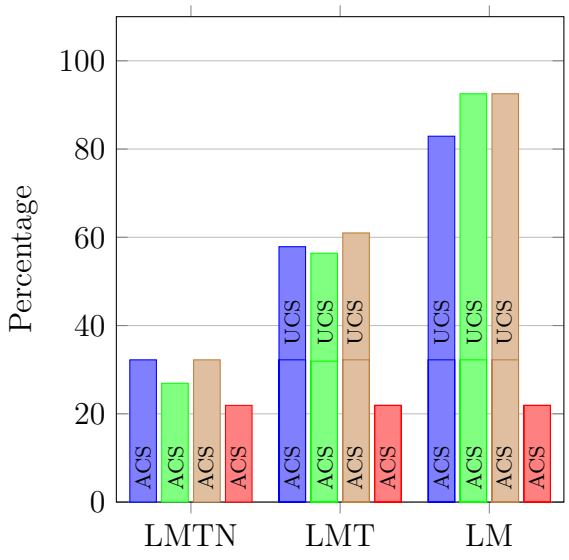
The PRIO CA algorithm has a fixed maximum number of iterations before returning a result. It depends on the number of virtual controls r and the number of inputs m. The PRIO CA sub-optimality depends mainly on the possibility of finding an update direction $\mathbf{c}_k$ which is directly coupled to the number of saturated inputs. If a maximum of two inputs saturate $s \leq 2$, the PRIO CA returns the optimal solution with respect to the prioritizing strategy.



(a) Nominal Case

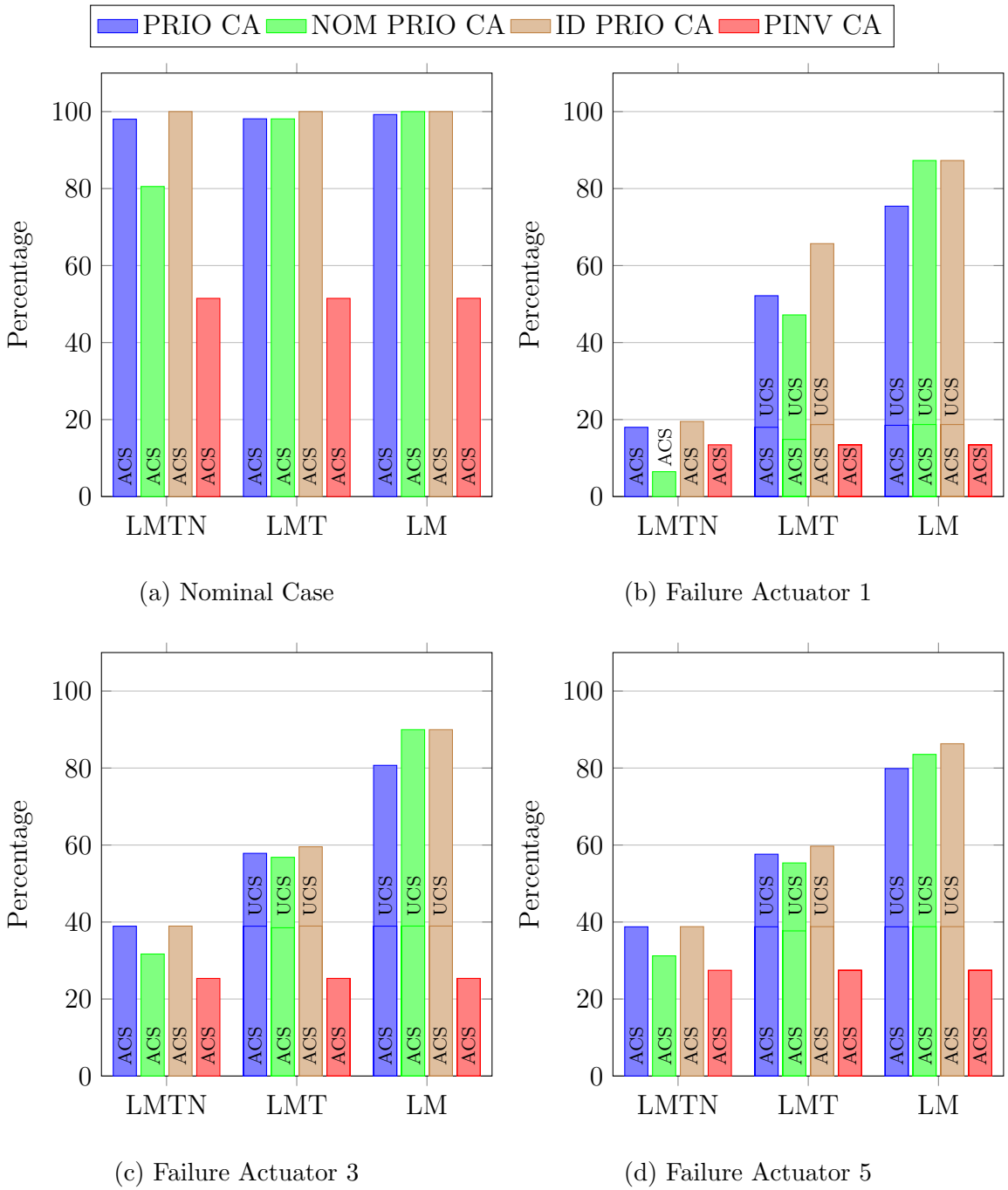


(b) Failure Actuator 3



(c) Failure Actuator 5

Figure 4.20: Control Allocation General Comparison Results: Hexacopter System with Actuator Configuration 1 as in Figure 2.1a



(a) Nominal Case

(b) Failure Actuator 1

(c) Failure Actuator 3

(d) Failure Actuator 5

Figure 4.21: Control Allocation General Comparison Results: Hexacopter System with Actuator Configuration 2 as in Figure 2.1b

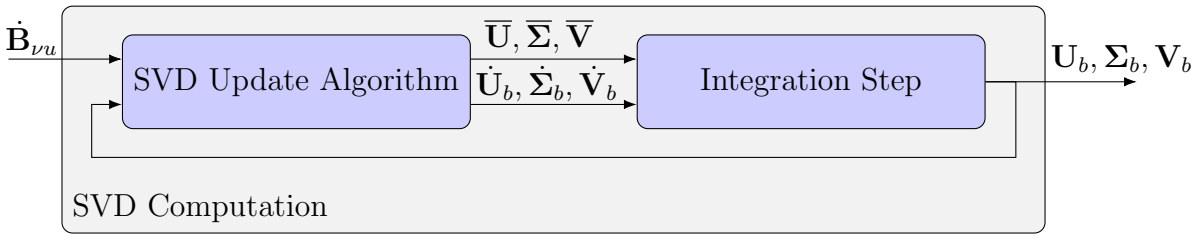


Figure 4.22: Diagram of the SVD Computation

4.4 SVD Update Algorithm

In order to implement the prioritizing control allocation (PRIO CA), the singular value decomposition (SVD) of the time-varying matrix $\mathbf{B}_{\nu u}(t) \in \mathbb{R}^{r \times m}$ needs to be known as stated in Assumption 4.9 (Input Matrix SVD). The main challenge is that standard iterative numerical methods that do not guarantee a maximum execution time. In this section, an alternative algorithm is proposed in order to update the singular value decomposition of a matrix $\mathbf{B}_{\nu u}(t)$ given its time derivative $\dot{\mathbf{B}}_{\nu u}(t)$ and an offline computed SVD of the initial input matrix $\mathbf{B}_{\nu u}(t = 0)$.

4.4.1 Main Algorithm

This algorithm is an extension of the SVD update algorithm developed by Höcht [69, Sec. 4.7]. In [69], this algorithm is applied to square matrices to limit the minimum eigenvalue from below such that zero eigenvalues are avoided. This section presents an extension of the SVD update algorithm to non-square matrices. Without loss of generality, the derivation is made for a matrix with more columns than rows corresponding to a system with redundant actuators. In the following, the time dependency is omitted for readability since all the matrices vary in time. Given are the matrix $\mathbf{B}_{\nu u} \in \mathbb{R}^{r \times m}$ with $r \leq m$ and its singular value decomposition

$$\mathbf{B}_{\nu u} = \mathbf{U}_b \cdot \mathbf{\Sigma}_b \cdot \mathbf{V}_b^T \tag{4.61}$$

where $\mathbf{U}_b \in \mathbb{R}^{r \times r}, \mathbf{\Sigma}_b \in \mathbb{R}^{r \times m}, \mathbf{V}_b \in \mathbb{R}^{m \times m}$. $\mathbf{U}_b, \mathbf{V}_b$ are orthogonal matrices, i.e. $\mathbf{U}_b \mathbf{U}_b^T = \mathbf{I}$ and $\mathbf{V}_b \mathbf{V}_b^T = \mathbf{I}$. $\mathbf{\Sigma}_b$ is a rectangular diagonal matrix which can be written as

$$\mathbf{\Sigma}_b = \begin{bmatrix} \mathbf{D}_b & \mathbf{0}_{r \times m-r} \end{bmatrix},$$

where $\mathbf{D}_b \in \mathbb{R}^{r \times r}$ is a diagonal matrix with non-negative entries. The diagonal elements are the singular values of $\mathbf{B}_{\nu u}$ and have a decreasing order. They are written as $\sigma_i \geq 0 \in \mathbb{R}$ for $i = 1 \ldots r$.

The goal of the presented algorithm is to compute the derivatives $\dot{\mathbf{U}}_b, \dot{\mathbf{\Sigma}}_b$ and $\dot{\mathbf{V}}_b$ of the SVD given the derivative of the matrix $\dot{\mathbf{B}}_{\nu u}$ and the current SVD composed by $\mathbf{U}_b, \mathbf{\Sigma}_b, \mathbf{V}_b$. The SVD of the matrix $\mathbf{B}_{\nu u}$ can then be continuously updated as depicted in Figure 4.22. In the figure, the matrices $\overline{\mathbf{U}}, \overline{\mathbf{\Sigma}}, \overline{\mathbf{V}}$ are an alternative singular value decomposition of $\mathbf{B}_{\nu u} = \overline{\mathbf{U}} \cdot \overline{\mathbf{\Sigma}} \cdot \overline{\mathbf{V}}^T$ such that it is compatible with the given derivative $\dot{\mathbf{B}}_{\nu u}$. The computation of an alternative SVD is only necessary in cases where the singular vectors are not uniquely defined. The ambiguity occurs in two cases: 1) Two or

more singular values are equal, 2) One or more singular values are zero and the system is overactuated, i.e. $m > r$. In the following, the case of different and positive singular values σ_i is analyzed and the ambiguity of singular vectors is studied in the next section. Taking the derivative of (4.61) leads to

$$\dot{\mathbf{B}}_{\nu u} = \mathbf{U}_b \cdot \dot{\mathbf{\Sigma}}_b \cdot \mathbf{V}_b^T + \dot{\mathbf{U}}_b \cdot \mathbf{\Sigma}_b \cdot \mathbf{V}_b^T + \mathbf{U}_b \cdot \mathbf{\Sigma}_b \cdot \dot{\mathbf{V}}_b^T. \tag{4.62}$$

By multiplying by $\mathbf{U}_b^T$ from the left and by $\mathbf{V}_b$ from the right, and using their orthogonality property, it follows that

$$\mathbf{U}_b^T \cdot \dot{\mathbf{B}}_{\nu u} \cdot \mathbf{V}_b = \dot{\mathbf{\Sigma}}_b + \mathbf{U}_b^T \cdot \dot{\mathbf{U}}_b \cdot \mathbf{\Sigma}_b + \mathbf{\Sigma}_b \cdot \dot{\mathbf{V}}_b^T \cdot \mathbf{V}_b. \tag{4.63}$$

In the next step, the derivative of the orthogonality constraint is taken to get the following equivalence

$$\mathbf{U}_b^T \cdot \mathbf{U}_b = \mathbf{I} \quad \Leftrightarrow \quad \dot{\mathbf{U}}_b^T \cdot \mathbf{U}_b + \mathbf{U}_b^T \cdot \dot{\mathbf{U}}_b = \mathbf{0}, \tag{4.64a}$$

$$\mathbf{V}_b^T \cdot \mathbf{V}_b = \mathbf{I} \quad \Leftrightarrow \quad \dot{\mathbf{V}}_b^T \cdot \mathbf{V}_b + \mathbf{V}_b^T \cdot \dot{\mathbf{V}}_b = \mathbf{0}. \tag{4.64b}$$

Then, the following invertible parameter transformation of the derivatives $\dot{\mathbf{U}}_b$ and $\dot{\mathbf{V}}_b$ is defined

$$\mathbf{C}_u = \mathbf{U}_b^T \cdot \dot{\mathbf{U}}_b, \tag{4.65a}$$

$$\mathbf{C}_v = \mathbf{V}_b^T \cdot \dot{\mathbf{V}}_b. \tag{4.65b}$$

In this way, the orthogonality conditions (4.64) are equivalent to stating that the matrices $\mathbf{C}_u \in \mathbb{R}^{r \times r}$ and $\mathbf{C}_v \in \mathbb{R}^{m \times m}$ are skew symmetric

$$\mathbf{C}_v^T = -\mathbf{C}_v,$$
$$\mathbf{C}_u^T = -\mathbf{C}_u.$$

By inserting the definitions (4.65) in the derivative (4.63) it follows

$$\mathbf{Y} := \mathbf{U}_b^T \cdot \dot{\mathbf{B}}_{\nu u} \cdot \mathbf{V}_b = \dot{\mathbf{\Sigma}}_b + \mathbf{C}_u \cdot \mathbf{\Sigma}_b + \mathbf{\Sigma}_b \cdot \mathbf{C}_v^T, \tag{4.66}$$

where $\mathbf{Y} \in \mathbb{R}^{r \times m}$ and $\mathbf{\Sigma}_b$ are known matrices and $\dot{\mathbf{\Sigma}}_b$, $\mathbf{C}_u$ and $\mathbf{C}_v$ are to be calculated.

In order to illustrate which elements of $\mathbf{Y}$ are needed for computing the unknown variables $\dot{\mathbf{\Sigma}}_b$, $\mathbf{C}_u$ and $\mathbf{C}_v$, the matrix $\mathbf{Y}$ (4.66) can be separated in three submatrices as

$$\mathbf{Y} = \begin{pmatrix} \mathbf{Y}_{11} & \cdots & \mathbf{Y}_{1r} & \cdots & \mathbf{Y}_{1m} \\ \vdots & \ddots & \vdots & \cdots & \vdots \\ \mathbf{Y}_{r1} & \cdots & \mathbf{Y}_{rr} & \cdots & \mathbf{Y}_{rm} \end{pmatrix}. \tag{4.67}$$

This separation is motivated by the structure of the last two terms in (4.66):

$$\mathbf{C}_u \cdot \mathbf{\Sigma}_b = \mathbf{C}_u \cdot \begin{bmatrix} \mathbf{D}_b, & \mathbf{0}_{r \times m-r} \end{bmatrix} = \begin{bmatrix} \sigma_1 \cdot \mathbf{C}_{u,c,1}, \ldots, \sigma_r \cdot \mathbf{C}_{u,c,r}, & \mathbf{0}_{r \times m-r} \end{bmatrix}, \tag{4.68}$$

$$\mathbf{\Sigma}_b \cdot \mathbf{C}_v^T = \begin{bmatrix} \mathbf{D}_b, & \mathbf{0}_{r \times m-r} \end{bmatrix} \cdot \mathbf{C}_v^T = \begin{bmatrix} \sigma_1 \cdot \mathbf{C}_{v,c,1}^T \\ \vdots \\ \sigma_r \cdot \mathbf{C}_{v,c,r}^T \end{bmatrix}. \tag{4.69}$$

Here, $\mathbf{C}_{u,c,i} \in \mathbb{R}^r$ and $\mathbf{C}_{v,c,i} \in \mathbb{R}^m$ are the i-th columns of $\mathbf{C}_u$ and $\mathbf{C}_v$ accordingly. Because $\mathbf{C}_v$ and $\mathbf{C}_u$ are skew symmetric, the i-th elements of $\mathbf{C}_{u,c,i}$ and $\mathbf{C}_{v,c,i}$ equal zero and therefore the diagonal elements of $\mathbf{C}_u \cdot \mathbf{\Sigma}_b$ and $\mathbf{\Sigma}_b \cdot \mathbf{C}_v^T$ equal zero. The equation (4.66) can then be rewritten as

$$Y_{ii} = \dot{\sigma}_i, \qquad \text{for } i = 1 \dots r, \tag{4.70a}$$
$$Y_{ij} = \sigma_j \cdot C_{u,ij} + \sigma_i \cdot C_{v,ji}, \qquad \text{for } i = 1 \dots r, j = 1 \dots r, i \neq j, \tag{4.70b}$$
$$Y_{ij} = \sigma_i \cdot C_{v,ji}, \qquad \text{for } i = 1 \dots r, j = r+1 \dots m. \tag{4.70c}$$

Here, Y_{ij}, $C_{u,ij}$ and $C_{v,ij} \in \mathbb{R}$ are the elements corresponding to the i-th row and j-th column of the matrices $\mathbf{Y}$, $\mathbf{C}_u$ and $\mathbf{C}_v$ respectively. The diagonal terms of the matrix Y_{ii} for $i = 1 \dots r$ define the derivative $\dot{\mathbf{\Sigma}}_b$, i.e. the yellow submatrix in (4.67). The red submatrix in (4.67) corresponds to the off-diagonal elements of the matrix Y_{ij} for $i, j = 1 \dots r$ and $i \neq j$ which define the matrix $\mathbf{C}_u$ and the elements $C_{v,ij}$ for $i, j = 1 \dots r$ of the matrix $\mathbf{C}_v$. The blue submatrix in (4.67) corresponds to the elements of the right submatrix Y_{ij} for $i = 1 \dots r, j = r+1 \dots m$ which define the elements $C_{v,ij}$ for $i = 1 \dots r, j = r+1 \dots m$ and for $j = 1 \dots r, i = r+1 \dots m$ of the matrix $\mathbf{C}_v$. Using the same color mapping, this can be illustrated as

$$\dot{\Sigma}, \quad \mathbf{C}_u, \quad \mathbf{C}_v = \begin{pmatrix} 0 & \cdots & \mathbf{C}_{v,1r} & \cdots & \mathbf{C}_{v,1m} \\ \vdots & \ddots & \vdots & & \vdots \\ \mathbf{C}_{v,r1} & \cdots & 0 & \cdots & \mathbf{C}_{v,rm} \\ \vdots & & \vdots & \ddots & \vdots \\ \mathbf{C}_{v,m1} & \cdots & \mathbf{C}_{v,mr} & \cdots & 0 \end{pmatrix}. \tag{4.71}$$

Notice that the elements $C_{v,ij}$ for $i = r+1 \dots m, j = r+1 \dots m$ are not defined by the constraint (4.66). These terms describe the rotation of the right singular vectors corresponding to the null space of the matrix with respect to each other. Hence, this rotation has no effect on the derivative of the matrix $\dot{\mathbf{B}}_{\nu u}$. Therefore, the elements $C_{v,ij}$ for $i = r+1 \dots m, j = r+1 \dots m$ can be set to zero.

In a first step, the derivative of $\dot{\mathbf{\Sigma}}_b$ can be directly computed from (4.70a) since all the non-diagonal elements of $\dot{\mathbf{\Sigma}}_b$ are zero. In the next step, the derivatives $\mathbf{C}_u$ and $\mathbf{C}_v$ are computed from the constraints (4.70b) corresponding to the red submatrix in (4.67) and (4.71). Each equation has two unknown variables $C_{u,ij}$ and $C_{v,ji}$. By taking two elements of $\mathbf{Y}$ with exchanged indexes, a solvable system of equations is build as

$$Y_{ij} = \sigma_j \cdot C_{u,ij} + \sigma_i \cdot C_{v,ji},$$
$$Y_{ji} = \sigma_i \cdot C_{u,ji} + \sigma_j \cdot C_{v,ij}.$$

Because the matrices $\mathbf{C}_u$ and $\mathbf{C}_v$ are skew-symmetric it holds that

$$Y_{ij} = \sigma_j \cdot C_{u,ij} + \sigma_i \cdot C_{v,ji},$$
$$Y_{ji} = -\sigma_i \cdot C_{u,ij} - \sigma_j \cdot C_{v,ji},$$

which can be brought to the matrix form

$$\begin{bmatrix} Y_{ij} \\ Y_{ji} \end{bmatrix} = \begin{bmatrix} \sigma_j & \sigma_i \\ -\sigma_i & -\sigma_j \end{bmatrix} \begin{bmatrix} C_{u,ij} \\ C_{v,ji} \end{bmatrix}. \tag{4.72}$$

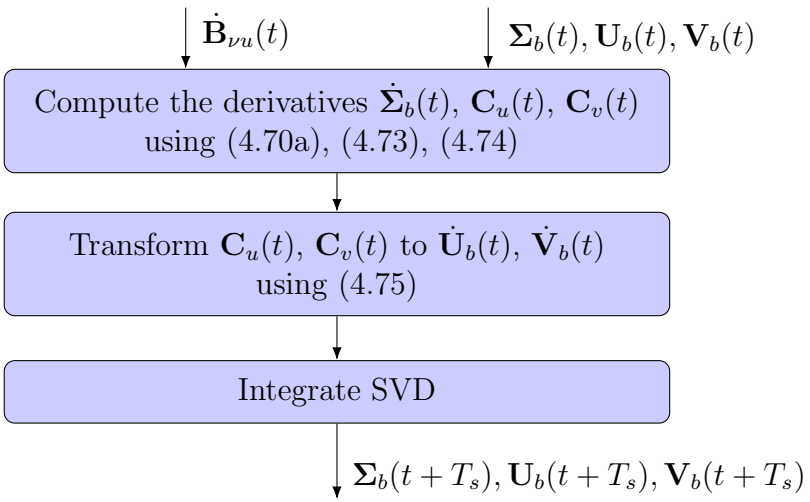


Figure 4.23: Diagram of the SVD Update Algorithm - Unique and Positive Singular Values

In order to compute the elements $C_{u,ij}$ and $C_{v,ji}$, the matrix containing the singular values needs to be inverted. This is possible as long as $0 \neq -\sigma_j^2 + \sigma_i^2$, i.e. for different singular values. In this case, the inverse can be analytically computed as

$$\begin{bmatrix} \sigma_j & \sigma_i \\ -\sigma_i & -\sigma_j \end{bmatrix}^{-1} = \frac{1}{-\sigma_j^2 + \sigma_i^2} \begin{bmatrix} -\sigma_j & -\sigma_i \\ \sigma_i & \sigma_j \end{bmatrix},$$

and the elements $C_{u,ij}$ and $C_{v,ji}$ are computed as

$$\begin{bmatrix} C_{u,ij} \\ C_{v,ji} \end{bmatrix} = \frac{1}{-\sigma_j^2 + \sigma_i^2} \begin{bmatrix} -\sigma_j & -\sigma_i \\ \sigma_i & \sigma_j \end{bmatrix} \begin{bmatrix} Y_{ij} \\ Y_{ji} \end{bmatrix}, \quad \text{for } i = 1 \dots r, j = 1 \dots r, i \neq j. \tag{4.73}$$

The constraints (4.70b), corresponding to the blue submatrix in (4.67) and (4.71), have only one unknown variable per constraint. The solution can be directly computed as

$$C_{v,ji} = \frac{Y_{ij}}{\sigma_i}, \qquad \text{for } i = 1 \dots r, j = r+1 \dots m. \tag{4.74}$$

This computation is possible as long as the singular value σ_i is greater than zero. Finally, the derivatives $\dot{\mathbf{U}}_b$ and $\dot{\mathbf{V}}_b$ follow from the inverse transformation of (4.65)

$$\dot{\mathbf{U}}_b = \mathbf{U}_b \cdot \dot{\mathbf{C}}_u, \tag{4.75a}$$

$$\dot{\mathbf{V}}_b = \mathbf{V}_b \cdot \dot{\mathbf{C}}_v. \tag{4.75b}$$

Hence, in this section it has been shown that the matrices $\dot{\mathbf{\Sigma}}_b$, $\dot{\mathbf{U}}_b$ and $\dot{\mathbf{V}}_b$ can be computed for the case of different non-zero singular values using (4.70a), (4.73), (4.74), and (4.75). The algorithm overview is depicted in Figure 4.23. In the next section the case of equal singular values is addressed.

4.4.2 Non-Unique Singular Vectors

The non-uniqueness of singular vectors allows us for solving the cases where: 1) Two or more singular values are equal, 2) One or more singular values are zero and the

system is overactuated, i.e. $m > r$. In this cases, the solutions (4.73) and (4.74) cannot be applied. Case 1 is treated in this section. Case 2 can be handled but it increases the algorithm complexity. In our application this is avoided by limiting the singular values from below as in [69]. For the hexacopter systems presented in Chapter 2, this limitation does not activate during any of the presented test scenarios. A minimum of 2 actuator failures need to simultaneously occur to force one of the singular values to zero.

For equal singular values, the solution presented in [69] is used. For the sake of completeness, the approach is described in the following. A solution of equation (4.72) exists only if $[Y_{ij}, Y_{ji}]^T$ and $[C_{u,ij}, C_{v,ji}]^T$ are linearly dependent. A solution can always be found by changing $\mathbf{C}_u$ and $\mathbf{C}_v$ through an orthogonal transformation of the singular vectors corresponding to the group of equal singular values. The transformed singular vectors still correspond to a valid SVD and therefore the SVD derivative can be found. First, in the next lemma it is shown that an alternative SVD exists for equal singular values.

Lemma 4.10 (Non-Uniqueness of Singular Vectors for equal Singular Values) *The set of indexes corresponding to a group of equal singular values is defined as the set*

$$\mathcal{G} = \{i | \sigma_i = \overline{\sigma}\} \tag{4.76}$$

and it is assumed that the number of equal singular values is $g \geq 2 \in \mathbb{R}$ without loss of generality. Then, the concatenation of left and right singular vectors corresponding to the group are $\mathbf{U}_g \in \mathbb{R}^{r \times g}$ and $\mathbf{V}_g \in \mathbb{R}^{m \times g}$. Let $\overline{\mathbf{U}}_g \in \mathbb{R}^{r \times g}$ and $\overline{\mathbf{V}}_g \in \mathbb{R}^{m \times g}$ be defined as alternative singular vectors of $\mathbf{U}_g$ and $\mathbf{V}_g$ computed by the orthogonal transformation

$$\overline{\mathbf{U}}_g = \mathbf{U}_g \mathbf{Q}, \qquad \overline{\mathbf{V}}_g = \mathbf{V}_g \mathbf{Q}, \tag{4.77}$$

where $\mathbf{Q} \in \mathbb{R}^{g \times g}$ is an orthogonal matrix.

In the case of equal singular values, every orthogonal transformation (4.77) of the corresponding singular vectors $\mathbf{U}_g$ and $\mathbf{V}_g$ leads to a valid singular value decomposition

$$\mathbf{B}_{\nu u} = \overline{\mathbf{U}} \mathbf{\Sigma}_b \overline{\mathbf{V}}^T = \mathbf{U}_b \mathbf{\Sigma}_b \mathbf{V}_b^T, \tag{4.78}$$

where the new matrices $\overline{\mathbf{U}}$ and $\overline{\mathbf{V}}$ correspond to the original matrices $\mathbf{U}_b$ and $\mathbf{V}_b$ with exchanged singular vectors $\mathbf{U}_g$ and $\mathbf{V}_g$ by the alternative singular vectors $\overline{\mathbf{U}}_g$ and $\overline{\mathbf{V}}_g$.

Proof: From Lemma B.1 (Matrix Product) in the Appendix B the SVD of $\mathbf{B}_{\nu u}$ (4.61) can be alternatively expressed as

$$\mathbf{B}_{\nu u} = \sum_{i=1}^{r} \sigma_i \cdot \mathbf{u}_{b,c,i} \mathbf{v}_{b,c,i}^T,$$

with the i-th left singular vector $\mathbf{u}_{b,c,i} \in \mathbb{R}^r$ and the i-th right singular vector $\mathbf{v}_{b,c,i} \in \mathbb{R}^m$ being the i-th columns of $\mathbf{U}_b$ and $\mathbf{V}_b$ respectively. For equal singular values, the sum can be divided in two groups as

$$\mathbf{B}_{\nu u} = \sum_{i \notin \mathcal{G}} \sigma_i \cdot \mathbf{u}_{b,c,i} \mathbf{v}_{b,c,i}^T + \overline{\sigma} \cdot \sum_{i \in \mathcal{G}} \mathbf{u}_{b,c,i} \mathbf{v}_{b,c,i}^T.$$

The second term can expressed using the transformed versions of the corresponding singular vectors. Let $\overline{\mathbf{u}}_{c,i}$, $\overline{\mathbf{v}}_{c,i}$ and $\mathbf{q}_{c,i}$ be the i-th column vectors of the matrices $\overline{\mathbf{U}}$, $\overline{\mathbf{V}}$, and $\mathbf{Q}$ respectively. Using the transformation (4.77) and Lemma B.1 (Matrix Product), it follows that

$$\begin{aligned}
\overline{\sigma} \cdot \sum_{i \in \mathcal{G}} \overline{\mathbf{u}}_{c,i} \overline{\mathbf{v}}_{c,i}^T &= \overline{\sigma} \cdot \sum_{i \in \mathcal{G}} \left(\mathbf{U}_g \mathbf{q}_{c,i}\right) \left(\mathbf{V}_g \mathbf{q}_{c,i}\right)^T, \\
&= \overline{\sigma} \cdot \mathbf{U}_g \cdot \left(\sum_{i \in \mathcal{G}} \mathbf{q}_{c,i} \mathbf{q}_{c,i}^T \right) \cdot \mathbf{V}_g^T, \\
&= \overline{\sigma} \cdot \mathbf{U}_g \cdot \mathbf{Q} \cdot \mathbf{Q}^T \cdot \mathbf{V}_g^T, \\
&= \overline{\sigma} \cdot \mathbf{U}_g \cdot \mathbf{V}_g^T, \\
&= \overline{\sigma} \cdot \sum_{i \in \mathcal{G}} \mathbf{u}_{b,c,i} \mathbf{v}_{b,c,i}^T.
\end{aligned}$$

Hence, the matrix $\mathbf{B}_{\nu u}$ can be equivalently expressed by the transformed singular vectors

$$\mathbf{B}_{\nu u} = \sum_{i \notin \mathcal{G}} \sigma_i \cdot \mathbf{u}_{b,c,i} \mathbf{v}_{b,c,i}^T + \overline{\sigma} \cdot \sum_{i \in \mathcal{G}} \overline{\mathbf{u}}_{c,i} \overline{\mathbf{v}}_{c,i}^T.$$

In order for $\overline{\mathbf{U}}$ and $\overline{\mathbf{V}}$ to be valid singular vectors, orthogonality is still to be proved. First, using the definition of $\overline{\mathbf{U}}_g$ (4.77) it is shown that all the transformed singular vectors are perpendicular to the singular vectors that were not transformed

$$\overline{\mathbf{u}}_{c,i}^T \mathbf{u}_{b,c,j} = \mathbf{q}_{c,i}^T \mathbf{U}_g^T \mathbf{u}_{b,c,j} = 0, \quad \text{for } i \in \mathcal{G}, j \notin \mathcal{G}.$$

Here, the orthogonality of the original matrix $\mathbf{U}_b$ and Lemma B.10 (Orthogonal Matrix) have been used. Second, note that all the transformed vectors are perpendicular to each other and have unit length

$$\begin{aligned}
\overline{\mathbf{U}}_g^T \overline{\mathbf{U}}_g &= \mathbf{Q}^T \mathbf{U}_g^T \mathbf{U}_g \mathbf{Q}, \\
&= \mathbf{Q}^T \mathbf{Q}, \\
&= \mathbf{I}.
\end{aligned}$$

Here, the orthogonality of the original matrix $\mathbf{U}_b$ and of the transformation matrix $\mathbf{Q}$ and Lemma B.10 (Orthogonal Matrix) have been used. Since the same argumentation can be done for the right singular vectors, it follows that $\overline{\mathbf{U}}^T \overline{\mathbf{U}} = \mathbf{I}$ and $\overline{\mathbf{V}}^T \overline{\mathbf{V}} = \mathbf{I}$. Furthermore, the same argumentation can iteratively be applied if more than one group of equal singular values exist. ■

From Lemma 4.10, it is known that the singular vectors can be transformed in case of equal singular values. This fact is used in the following to solve the constraint (4.72) corresponding to the red submatrix in (4.67) and (4.71). If two singular values are equal, (4.72) becomes

$$\begin{bmatrix} Y_{ij} \\ Y_{ji} \end{bmatrix} = \sigma_i \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} C_{u,ij} \\ C_{v,ji} \end{bmatrix}. \tag{4.79}$$

This system of equations is only solvable if $Y_{ij} = -Y_{ji}$ or $\sigma_i = 0$. If $Y_{ij} = -Y_{ji}$ and $\sigma_i > 0$, $C_{u,ij}$ and $C_{v,ji}$ have an infinite number of solutions. One of them is $C_{u,ij} = C_{v,ji}$ which leads to

$$\begin{aligned}
Y_{ij} &= 2 \cdot \sigma_i \cdot C_{u,ij}, \\
-Y_{ij} &= -2 \cdot \sigma_i \cdot C_{u,ij}.
\end{aligned}$$

Hence, $C_{u,ij}$ and $C_{v,ji}$ are computed as

$$C_{u,ij} = C_{v,ji} = \frac{1}{2 \cdot \sigma_i} Y_{ij}. \tag{4.80}$$

If $Y_{ij} \neq -Y_{ji}$, the singular vectors corresponding to the equal singular values need to be transformed using (4.77). Therefore, the singular vectors are selected such that the condition $Y_{ij} = -Y_{ji}$ is satisfied. From (4.66), this translates to

$$0 = Y_{ij} + Y_{ji} = \mathbf{u}_{b,c,i}^T \dot{\mathbf{B}}_{\nu u} \mathbf{v}_{b,c,j} + \mathbf{u}_{b,c,j}^T \dot{\mathbf{B}}_{\nu u} \mathbf{v}_{b,c,i}. \tag{4.81}$$

The condition (4.81) needs to hold for every pair of singular values that are equal.

The next step is to find the transformation matrix $\mathbf{Q}$ such that for every pair of equal singular values the condition (4.81) holds for the transformed singular vectors. For $i, j \in \mathcal{G}$, $i \neq j$ and using the definition of $\overline{\mathbf{U}}_g$ and $\overline{\mathbf{V}}_g$ (4.77) this leads to

$$\begin{aligned}
0 &= \left(\mathbf{U}_g \mathbf{q}_{c,i}\right)^T \dot{\mathbf{B}}_{\nu u} \left(\mathbf{V}_g \mathbf{q}_{c,j}\right) + \left(\mathbf{U}_g \mathbf{q}_{c,j}\right)^T \dot{\mathbf{B}}_{\nu u} \left(\mathbf{V}_g \mathbf{q}_{c,i}\right) \\
0 &= \mathbf{q}_{c,i}^T \mathbf{U}_g^T \dot{\mathbf{B}}_{\nu u} \mathbf{V}_g \mathbf{q}_{c,j} + \mathbf{q}_{c,j}^T \mathbf{U}_g^T \dot{\mathbf{B}}_{\nu u} \mathbf{V}_g \mathbf{q}_{c,i} \\
0 &= \mathbf{q}_{c,i}^T \mathbf{U}_g^T \dot{\mathbf{B}}_{\nu u} \mathbf{V}_g \mathbf{q}_{c,j} + \mathbf{q}_{c,i}^T \mathbf{V}_g^T \dot{\mathbf{B}}_{\nu u}^T \mathbf{U}_g \mathbf{q}_{c,j} \\
0 &= \mathbf{q}_{c,i}^T \left(\mathbf{U}_g^T \dot{\mathbf{B}}_{\nu u} \mathbf{V}_g + \mathbf{V}_g^T \dot{\mathbf{B}}_{\nu u}^T \mathbf{U}_g \right) \mathbf{q}_{c,j}
\end{aligned}$$

Since this needs to hold for all pairs $i, j \in \mathcal{G}$, the condition is written in matrix form as

$$\mathbf{Q}^T \left(\mathbf{U}_g^T \dot{\mathbf{B}}_{\nu u} \mathbf{V}_g + \mathbf{V}_g^T \dot{\mathbf{B}}_{\nu u}^T \mathbf{U}_g \right) \mathbf{Q} = \mathbf{Y}_g + \mathbf{Y}_g^T = 2 \cdot \dot{\mathbf{D}}_g \tag{4.82}$$

where $\mathbf{Y}_g \in \mathbb{R}^{g \times g}$ is composed of the elements Y_{ij} of $\mathbf{Y}$ such that $i, j \in \mathcal{G}$ and hence $\dot{\mathbf{D}}_g \in \mathbb{R}^{g \times g}$ is a diagonal matrix composed of the derivative of the singular values $\dot{\sigma}_i$ with $i \in \mathcal{G}$ corresponding to the studied group.

As can be seen in (4.82), the solution of the constraint (4.72) has been transformed to the diagonalization of the matrix $\mathbf{U}_g^T \dot{\mathbf{B}}_{\nu u} \mathbf{V}_g + \mathbf{V}_g^T \dot{\mathbf{B}}_{\nu u}^T \mathbf{U}_g$. Since it is a symmetric matrix, the diagonalization problem corresponds to solving the eigenvalue problem and the columns of the matrix $\mathbf{Q}$ are then the corresponding eigenvectors. After transforming the left and right singular vectors with (4.77) to get the new matrices $\overline{\mathbf{U}}$ and $\overline{\mathbf{V}}$, the matrix $\mathbf{Y}$ can be recomputed as

$$\overline{\mathbf{Y}} := \overline{\mathbf{U}}^T \dot{\mathbf{B}}_{\nu u} \overline{\mathbf{V}},$$

which guarantees that $\overline{\mathbf{Y}}_{ij} = -\overline{\mathbf{Y}}_{ji}$ holds for $i, j \in \mathcal{G}$. Hence, the corresponding values of $\mathbf{C}_u$ and $\mathbf{C}_v$ can be computed from (4.80).

Because in the procedure of solving for $\mathbf{C}_u$ and $\mathbf{C}_v$, the matrix $\mathbf{Y}$ has been transformed, it is necessary to handle the case of equal singular values at the beginning of the SVD update algorithm. Furthermore, it needs to be taken into account that the SVD derivatives correspond to the transformed singular vectors $\overline{\mathbf{U}}$ and $\overline{\mathbf{V}}$ and not to the original ones. The details of such an integration step can be found in [69].

4.4.3 Implementation Aspects

Regarding the implementation of the presented algorithm, three important considerations are discussed in the following.

Maximum Execution Time

The main reason for choosing the SVD update algorithm over numerical iterative algorithms is the possibility of having a maximum execution time. In the case of different positive singular values, this is the case. For equal singular values, this is only the case if the groups of singular values have a maximum of $g = 3$ elements. For groups of $g = 2$ and $g = 3$ elements, the eigenvalue problem can be solved using the quadratic or Cardano's formulas [69]. For $g > 3$ the use of iterative numerical methods is necessary.

Orthogonality of Singular Vectors

The orthogonality of the matrices $\mathbf{U}_b$ and $\mathbf{V}_b$ is theoretically maintained given the condition that $\mathbf{C}_u$ and $\mathbf{C}_v$ defined in (4.65) are skew symmetric. However, the implementation of the algorithm requires a discretization of the integration and with it small numerical errors lead to a deviation from the orthogonality. In order to correct this effect dynamically, an additional feedback within the derivatives (4.75) is proposed by [69]

$$\dot{\mathbf{U}}_b(t) = \mathbf{U}_b(t)\left(\mathbf{C}_u(t) - k_u\left(\mathbf{U}_b^T(t)\,\mathbf{U}_b(t) - \mathbf{I}\right)\right), \tag{4.83a}$$

$$\dot{\mathbf{V}}_b(t) = \mathbf{V}_b(t)\left(\mathbf{C}_v(t) - k_v\left(\mathbf{V}_b^T(t)\,\mathbf{V}_b(t) - \mathbf{I}\right)\right), \tag{4.83b}$$

with positive gains $k_u, k_v > 0 \in \mathbb{R}$. In order to understand the effect of the added feedback, consider the orthogonality error $\mathbf{E}_u(t) := \mathbf{U}_b^T(t)\,\mathbf{U}_b(t) - \mathbf{I} \in \mathbb{R}^{r\times r}$. Given that $\mathbf{C}_u(t)$ is skew-symmetric and that $\mathbf{U}_b(t)$ is orthogonal, the derivative of $\mathbf{E}_u(t)$ is given by

$$\begin{aligned}
\mathbf{E}_u(t) &= \mathbf{U}_b^T(t)\,\dot{\mathbf{U}}_b(t) + \dot{\mathbf{U}}_b^T(t)\,\mathbf{U}_b(t) \\
&= \left(\mathbf{C}_u(t) - k_u\left(\mathbf{U}_b^T(t)\,\mathbf{U}_b(t) - \mathbf{I}\right)\right) + \left(\mathbf{C}_u(t) - k_u\left(\mathbf{U}_b^T(t)\,\mathbf{U}_b(t) - \mathbf{I}\right)\right)^T \\
&= -2k_u\left(\mathbf{U}_b^T(t)\,\mathbf{U}_b(t) - \mathbf{I}\right) \\
&= -2k_u \cdot \mathbf{E}_u(t).
\end{aligned}$$

Since $-2k_u < 0$, the error dynamics $\mathbf{E}_u(t)$ are asymptotically stable and have their equilibrium at $\mathbf{E}_u = \mathbf{0}$ [100, Sec. 2.6], which corresponds to the orthogonality condition of $\mathbf{U}_b(t)$. Notice that the term $-k_u\left(\mathbf{U}_b^T(t)\,\mathbf{U}_b(t) - \mathbf{I}\right)$ is symmetric and therefore does not influence the skew-symmetric part of the derivative $\mathbf{C}_u(t)$. The same argumentation can be done for $\mathbf{C}_v(t)$. Maintaining the orthogonality of the matrices $\mathbf{U}_b(t)$ and $\mathbf{V}_b(t)$ is critical for the correct update of the singular value decomposition as presented in this section.

SVD Numerical Drift

Because the matrix $\mathbf{B}_{\nu u}(t)$ and the SVD $\mathbf{\Sigma}_b(t)$, $\mathbf{U}_b(t)$ and $\mathbf{V}_b(t)$ are integrated independently, small numerical computational errors can lead to a drift. In order to avoid it, an error correction term is added to the derivative of the input matrix $\mathbf{B}_{\nu u}(t)$ as follows

$$\mathbf{F}(t) = \dot{\mathbf{B}}_{\nu u}(t) - k\left(\mathbf{U}_b(t)\,\mathbf{\Sigma}_b(t)\,\mathbf{V}_b^T(t) - \mathbf{B}_{\nu u}(t)\right), \tag{4.84}$$

such that (4.62) is exchanged by

$$\mathbf{F}(t) = \dot{\mathbf{U}}_b(t) \cdot \mathbf{\Sigma}_b(t) \cdot \mathbf{V}_b^T(t) + \mathbf{U}_b(t) \cdot \dot{\mathbf{\Sigma}}_b(t) \cdot \mathbf{V}_b^T(t) + \mathbf{U}_b(t) \cdot \mathbf{\Sigma}_b(t) \cdot \dot{\mathbf{V}}_b^T(t). \quad (4.85)$$

The error between current SVD and input matrix is defined as

$$\mathbf{E}_b(t) := \mathbf{U}_b(t)\,\mathbf{\Sigma}_b(t)\,\mathbf{V}_b^T(t) - \dot{\mathbf{B}}_{\nu u}(t) \in \mathbb{R}^{n \times m}.$$

The error dynamics are given by

$$\dot{\mathbf{E}}_b(t) = \mathbf{F}(t) - \dot{\mathbf{B}}_{\nu u}(t) = -k \cdot \mathbf{E}_b(t)$$

and by selecting $k > 0 \in \mathbb{R}$ the equilibrium $\mathbf{E}_b = \mathbf{0}$ is asymptotically stable [100, Sec. 2.6].

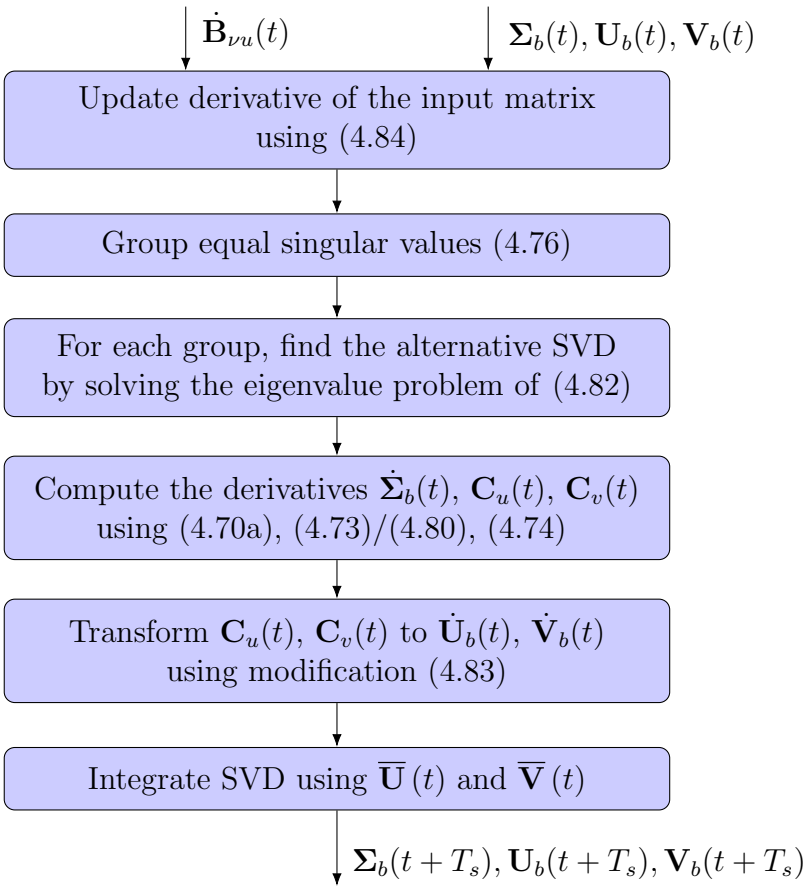


Figure 4.24: Diagram of the SVD Update Algorithm

The final SVD update algorithm including the robustness modifications is depicted in Figure 4.24. In a first step, the input matrix derivative $\dot{\mathbf{B}}_{\nu u}(t)$ is updated using the robustness modification (4.84). Subsequently, the transformation of the singular vectors corresponding to equal singular values is done as presented in Section 4.4.2. After grouping equal singular values (4.76), the eigenvalue problem of (4.82) is solved in order to find the transformation matrix $\mathbf{Q}$. This process is repeated for every group of singular values leading to the alternative singular vectors $\overline{\mathbf{U}}(t)$ and $\overline{\mathbf{V}}(t)$. The derivatives $\dot{\mathbf{\Sigma}}_b(t)$, $\mathbf{C}_u(t)$, $\mathbf{C}_v(t)$ are then computed using (4.70a), (4.73)/(4.80) and (4.74). The

derivatives $\dot{\mathbf{U}}_b(t)$ and $\dot{\mathbf{V}}_b(t)$ are calculated from $\mathbf{C}_u(t)$, $\mathbf{C}_v(t)$ using the robustness modification (4.83). Finally, the SVD matrices are integrated using the alternative singular vectors $\overline{\mathbf{U}}\,(t)$ and $\overline{\mathbf{V}}\,(t)$ as current states.

4.5 Bibliographical Remarks

In this section, a predictor-based adaptive control allocation (P-ACA) and a reduced-order version of it have been presented. Because of its indirect formulation, the control allocation is separated from the rigid body control and offers more flexibility than direct adaptive formulations [50, 98, 143, 144]. The formulation of P-ACA ensures that the rigid body control laws are not overparameterized and that actuator redundancy is fully handled within the control allocation. The Prioritizing Control Allocation (PRIO CA) is specially suitable for a degraded control situations, where the desired virtual control $\boldsymbol{\nu}_d$ is not attainable. It naturally disregards low priority virtual controls while satisfying the high priority virtual control commands. Furthermore, the algorithm has been developed such that a result is achieved within a fixed limited number of iterations.

The PRIO CA was motivated by the PAN approach presented in [17], where the pseudo inverse control allocation solution is modified along the null space of the control input matrix in order to effectively address input magnitude saturation. The main difference is the implemented strategy. In [17] and [142] the null space is used to minimize an error norm and ultimately implement direct allocation in the sense of Durham. The PRIO CA implements a prioritizing strategy and always maintains an attainable virtual control. The null space is used to find update directions such that the input limits are respected. In this work the inclusion of different update directions for singular cases has increased the efficacy of the algorithm.

A common alternative of implementing a prioritization strategy is the use of an optimization routine to minimize a function of weighted virtual control errors. The advantage of the PRIO CA is better prioritization of the virtual controls. The performance of the optimization routines largely depends on the correct selection of the weights which should be as different as possible for enforcing prioritization but as equal as possible to avoid ill-conditioned optimization problems. Further, for unattainable controls the optimal result of the weighted function necessarily leads to an allocation error of all the virtual controls, even if in different orders of magnitude.

In this work, the SVD update algorithm presented in [69] has been extended to non-square matrices. In this way, it is possible to apply the PRIO CA to overactuated systems and a time-varying input matrix. Furthermore, the numerical drift correction (4.84) has been introduced to increase robustness of the SVD update.

Compared to the previous work in [42, 43], the uncertainty parameterization and the control allocation strategy have been changed. The fault model (3.5) integrates not only multiplicative but also additive uncertainties such that the stuck actuator case can be taken into account. Furthermore, the control effectiveness matrix $\boldsymbol{\Lambda}$ is not limited to a diagonal matrix. The control allocation in [42,43] implements a prioritization strategy in form of a gradient optimization using a weighted error function.

Chapter 5

Multirotor Controllers

In Chapter 3 (Adaptive Fault Tolerant Control) the Parameter Reduction due to Overactuation (PRO) approach has been presented to solve fault tolerant control problems for systems with redundant actuators using adaptive control. Because of the importance of handling control input saturation, in Chapter 4 (Adaptive Control Allocation) the Predictor-Based Adaptive Control Allocation (P-ACA) has been introduced as a way of integrating control allocation approaches within the adaptive control framework. In this chapter, these methods are applied to the hexacopter system as presented in Chapter 2 (Multirotor Dynamics Model). The main goal is the design of an adaptive fault tolerant controller for a VTOL multicopter system that considers actuator redundancy, actuator limits and implements a strategy for degraded control authority as stated in Section 1.3. The capabilities of the resulting flight controller are demonstrated in flight tests and presented in Chapter 6 (Experimental Results).

The hexacopter system is an overactuated system and can be described by the rigid body dynamics (2.15) and the actuator model (2.16). The separation has been achieved by using the thrust $T(t)$ and the moments $\overrightarrow{\mathbf{M}}_P^R(t)$ generated by the propulsion system as virtual controls $\boldsymbol{\nu}(t) \in \mathbb{R}^4$. For the sake of clarity, the system description is summarized here

$$\left(\dot{\vec{\mathbf{r}}}^R\right)_N^N(t) = \left(\vec{\mathbf{v}}^R\right)_N^N(t), \tag{5.1a}$$

$$\left(\dot{\vec{\mathbf{v}}}^R\right)_N^{NN}(t) = \left(\vec{\mathbf{g}}^G\right)_N - \frac{1}{m}\left(\vec{\mathbf{z}}_B\right)_N(t) \cdot \left(T(t) + d_T\right), \tag{5.1b}$$

$$\dot{\mathbf{M}}_{NB}(t) = \mathbf{M}_{NB}(t) \cdot \boldsymbol{\Omega}\left(\left(\vec{\boldsymbol{\omega}}^{NB}\right)_B(t)\right), \tag{5.1c}$$

$$\left(\dot{\vec{\boldsymbol{\omega}}}^{NB}\right)_B^B(t) = \left(\mathbf{I}^R\right)_{BB}^{-1}\mathbf{f}_c\left(\left(\vec{\boldsymbol{\omega}}^{NB}\right)_B(t)\right) + \left(\mathbf{I}^R\right)_{BB}^{-1} \cdot \left(\left(\overrightarrow{\mathbf{M}}_P^R\right)_B(t) + \left(\vec{\mathbf{d}}_M\right)_B\right), \tag{5.1d}$$

$$\boldsymbol{\nu}(t) = \begin{bmatrix} T(t) \\ \left(\overrightarrow{\mathbf{M}}_P^R\right)_B(t) \end{bmatrix} = \mathbf{B}_a \boldsymbol{\Lambda} \mathbf{u}(t). \tag{5.1e}$$

The rigid body motion is described by the kinematic relations (5.1a) and (5.1c) and the dynamics (5.1b) and (5.1d). The actuator model is given by (5.1e). In order to model actuator faults, the control effectiveness matrix $\boldsymbol{\Lambda} \in \mathbb{R}^{m \times m}$ and the constant thrust and moment disturbances $d_T \in \mathbb{R}$ and $\vec{\mathbf{d}}_M \in \mathbb{R}^3$ are considered unknown parameters of the plant. The constant disturbances d_T and $\vec{\mathbf{d}}_M$ correspond to $\mathbf{d}_\nu$ in (2.16) and have been rearrange to match the system description (4.1). In (5.1d), the term $\left(\mathbf{I}^R\right)_{BB}^{-1} \cdot \left(\vec{\mathbf{d}}_M\right)_B$ corresponds to a constant unknown and the term $-\frac{1}{m}\left(\vec{\mathbf{z}}_B\right)_N(t) \cdot d_T$ in (5.1b) corresponds to

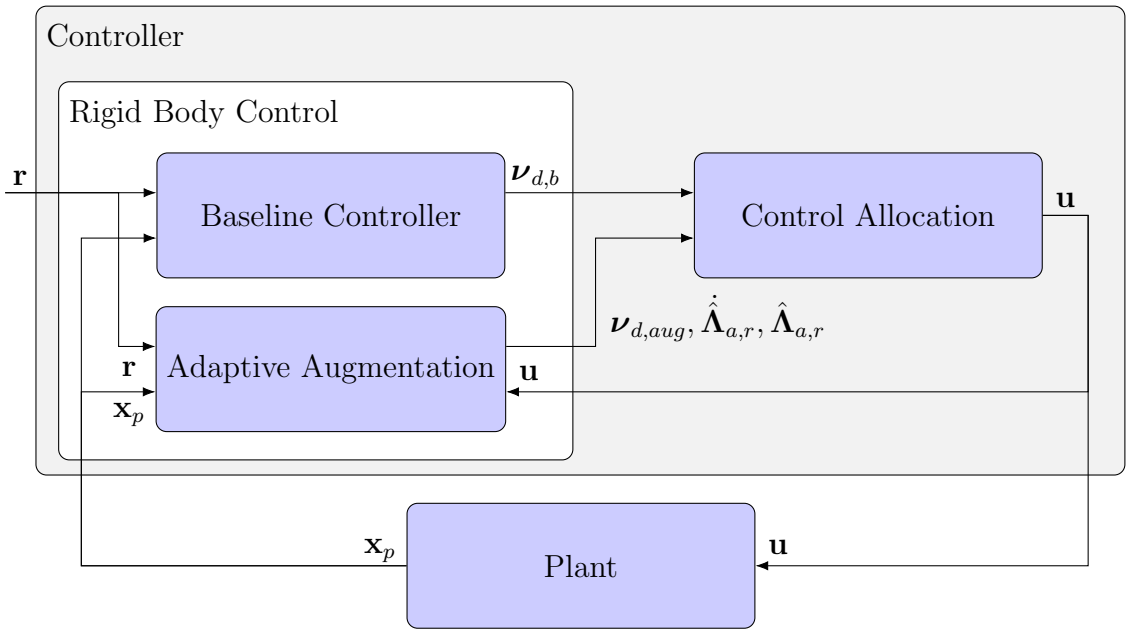


Figure 5.1: Control Overview Multirotor

a linearly parameterizable function of the state. The function $\boldsymbol{\Omega}(\cdot)$ mapping a vector to a skew-symmetric matrix is defined in (2.14). Furthermore, the following abbreviation for the Coriolis term has been used

$$\mathbf{f}_c\left(\left(\vec{\boldsymbol{\omega}}^{NB}\right)_B(t)\right) := -\left(\vec{\boldsymbol{\omega}}^{NB}\right)_B(t) \times \left(\left(\mathbf{I}^R\right)_{BB}\left(\vec{\boldsymbol{\omega}}^{NB}\right)_B(t)\right). \tag{5.2}$$

An overview of the general controller structure can be seen in Figure 5.1. There are mainly three elements that are combined to the final controller: the baseline controller, the adaptive augmentation and the control allocation. The baseline controller can be seen as a nominal non-adaptive controller of the plant, where uncertainties corresponding to normal operation are taken into account. In order to compensate for larger uncertainties like actuator faults, it is augmented by an adaptive controller. Following the P-ACA approach, the baseline controller and the adaptive augmentation are designed based on the rigid body dynamics (5.1a)-(5.1d) using the virtual control $\boldsymbol{\nu}(t)$ as the control input of the system. The mapping of the desired virtual controls to the actuators is done within the control allocation, which addresses the redundancy of the system and the input constraints. In the last chapters, no baseline controller has been considered due to the fact that it does neither affect the structure of the adaptive controller nor the stability proof [67]. In this chapter, it is included as a conceptual separation in order to highlight the changes due to the adaptive augmentation and to facilitate the selection of a lean, compact and unified controller structure (thesis objective 6).

Given that the virtual control $\boldsymbol{\nu}(t)$ of the system is 4-dimensional, 4 outputs can be independently commanded depending on the application scenario. The selected outputs in this work are:

- The reduced attitude vector command $\vec{\mathbf{z}}_c(t) \in \mathbb{S}^2$.
- The yaw rate command $\left(w_{c,z}\right)_B(t) \in \mathbb{R}$.

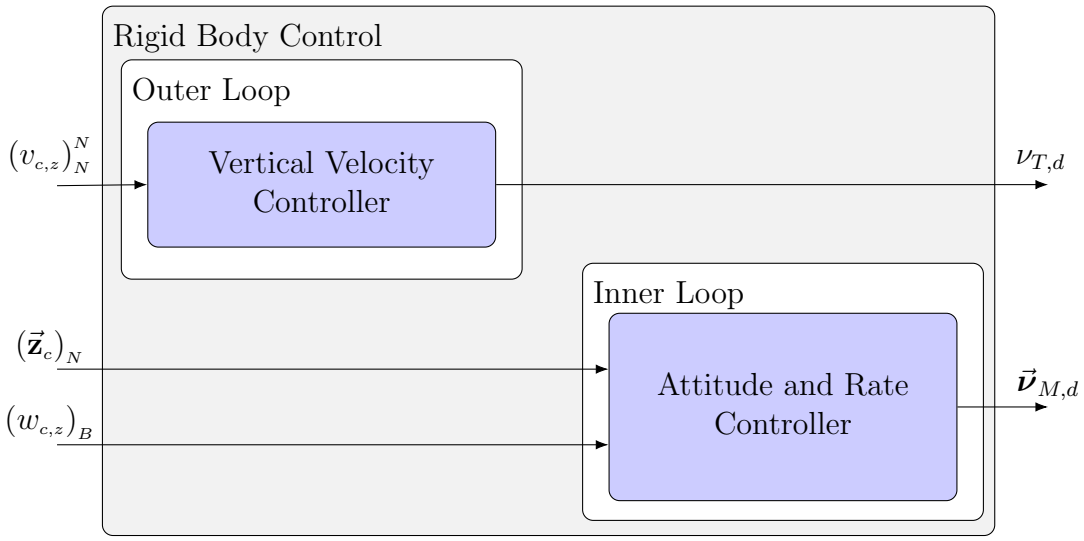


Figure 5.2: Baseline Controller Structure

- And the vertical velocity command $(v_{c,z})_N^N(t) \in \mathbb{R}$.

The rigid body controller structure is depicted in Figure 5.2. It is separated into an inner loop and an outer loop motivated by the inherently cascaded structure of the multirotor dynamics (5.1). On the one hand, the position $\vec{\mathbf{r}}^R(t)$ and velocity states $\vec{\mathbf{v}}^R(t)$ do not influence the attitude subsystem (5.1c)-(5.1d). On the other hand, the reduced attitude vector $\vec{\boldsymbol{z}}_B(t)$, a part of the rotation matrix $\mathbf{M}_{NB}(t)$ (2.12), and the thrust $T(t)$ can be seen as inputs of the position subsystem (5.1a)-(5.1b). Hence, the inner loop controller corresponds to an attitude and rate controller with the reduced attitude vector $\vec{\boldsymbol{z}}_c(t)$ and the yaw rate $(w_{c,z})_B(t)$ as input commands. Its output is the desired moment $\vec{\boldsymbol{\nu}}_{M,d}(t) \in \mathbb{R}^3$. The outer loop corresponds to the vertical velocity controller that handles the vertical velocity command $(v_{c,z})_N^N(t)$ by setting the desired total thrust $\nu_{T,d}(t)$. The overall outputs are the desired thrust $\nu_{T,d}(t)$ and the desired moment $\vec{\boldsymbol{\nu}}_{M,d}(t) \in \mathbb{R}^3$ which compose the desired virtual control $\boldsymbol{\nu}_d(t) \in \mathbb{R}^4$. The outer loop can further integrate position control in the horizontal plane as in [43] but in this thesis the scope is limited to the parts of the controller that are directly affected by actuator faults. Finally note that Figures 5.1 and 5.2 refer to the same rigid body controller but using different conceptual divisions.

The chapter is organized as follows. In Section 5.1, the Parameter Reduction due to Overactuation (PRO) is applied to the system. Based on the given parameterization, the control allocation is described in Section 5.2. Subsequently, the rigid body controllers corresponding to the baseline controller and the adaptive augmentation as in Figure 5.1 are derived. The baseline attitude and rate controller is treated in Section 5.3 and the corresponding adaptive augmentation in Section 5.4. The baseline vertical velocity controller and the corresponding augmentation are derived in Section 5.5.

5.1 Parameter Reduction due to Overactuation

The dynamics of the multirotor system have been separated into rigid body dynamics (5.1a)-(5.1d) and the actuator model (5.1e). To apply the Parameter Reduction due to

Overactuation (PRO) approach, a singular value decomposition of $\mathbf{B}_a$ (2.8)

$$\mathbf{B}_a = \mathbf{U}_a \mathbf{\Sigma}_a \mathbf{V}_a^T = \mathbf{U}_{a,r} \mathbf{D}_a \mathbf{V}_{a,r}^T \tag{5.3}$$

is used analogously to (4.4) and (3.48). Here, $\mathbf{U}_a \in \mathbb{R}^{4\times4}$ and $\mathbf{V}_a \in \mathbb{R}^{6\times6}$ are orthogonal matrices and $\mathbf{V}_{a,r} \in \mathbb{R}^{6\times4}$, $\mathbf{V}_{a,n} \in \mathbb{R}^{6\times2}$ are partitions of $\mathbf{V}_a = [\mathbf{V}_{a,r}, \mathbf{V}_{a,n}]$ such that $\mathbf{V}_{a,n}$ spans the null space of $\mathbf{B}_a$. Furthermore, $\mathbf{\Sigma}_a = [\mathbf{D}_a, \mathbf{0}] \in \mathbb{R}^{4\times6}$ and $\mathbf{D}_a \in \mathbb{R}^{4\times4}$ is a diagonal positive definite matrix. Since the matrix is full row rank, no division of $\mathbf{U}_a$ is needed and in this case it holds that $\mathbf{U}_a = \mathbf{U}_{a,r}$.

Example 5.1 (Hexacopter Input Matrix SVD) *For the hexacopter system with actuator configuration 1 as in Figure 2.1a, the nominal control input matrix $\mathbf{B}_a$ is given by (2.9) and by the parameters in Table 2.3. The normalized input matrix has been derived in Example 4.4 and has the following singular value decomposition (SVD)*

$$\mathbf{U}_a = \begin{bmatrix} 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix},$$

$$\mathbf{D}_a = \begin{bmatrix} 0.500 & 0 & 0 & 0 \\ 0 & 0.433 & 0 & 0 \\ 0 & 0 & 0.408 & 0 \\ 0 & 0 & 0 & 0.201 \end{bmatrix},$$

$$\mathbf{V}_{a,r} = \begin{bmatrix} 0.5000 & -0.2887 & 0.4082 & -0.4082 \\ -0.0000 & -0.5774 & -0.4082 & -0.4082 \\ -0.5000 & -0.2887 & 0.4082 & -0.4082 \\ -0.5000 & 0.2887 & -0.4082 & -0.4082 \\ 0.0000 & 0.5774 & 0.4082 & -0.4082 \\ 0.5000 & 0.2887 & -0.4082 & -0.4082 \end{bmatrix}, \quad \mathbf{V}_{a,n} = \begin{bmatrix} -0.3491 & -0.4599 \\ 0.5728 & -0.0724 \\ -0.2237 & 0.5322 \\ -0.3491 & -0.4599 \\ 0.5728 & -0.0724 \\ -0.2237 & 0.5322 \end{bmatrix}.$$

For the multirotor systems presented in Chapter 2, it is possible to partition the singular value decomposition into two independent parts corresponding to thrust and moments respectively. This separation can be exploited to divide the reduced control effectiveness matrix into two submatrices. This simplifies the controller design since the update laws can be independently derived. Without loss of generality, this process is exemplified based on Example 5.1. The matrix $\mathbf{U}_a$ in Example 5.1 can be partitioned into moment and thrust parts as follows

$$\mathbf{U}_a = \begin{bmatrix} \mathbf{0} & U_T \\ \mathbf{U}_M & \mathbf{0} \end{bmatrix}, \quad \mathbf{U}_M := \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \quad U_T := -1.$$

The singular values and right singular vectors can be analogously grouped by defining

$$\mathbf{D}_a = \begin{bmatrix} \mathbf{D}_m & \mathbf{0} \\ \mathbf{0} & D_T \end{bmatrix}, \qquad \mathbf{V}_{a,r} = \begin{bmatrix} \mathbf{V}_M, & \mathbf{V}_T \end{bmatrix},$$

where $\mathbf{D}_M \in \mathbb{R}^{3\times3}$, $D_T \in \mathbb{R}$, $\mathbf{V}_M \in \mathbb{R}^{6\times3}$, $\mathbf{V}_T \in \mathbb{R}^6$. Hence, the submatrices of $\mathbf{B}_a$ defined in (2.8) can be written as

$$\mathbf{B}_a = \begin{bmatrix} \mathbf{B}_{a,T} \\ \mathbf{B}_{a,M} \end{bmatrix} = \begin{bmatrix} U_T \cdot D_T \cdot \mathbf{V}_T^T \\ \mathbf{U}_M \cdot \mathbf{D}_M \cdot \mathbf{V}_M^T \end{bmatrix}. \tag{5.4}$$

Following the Parameter Reduction due to Overactuation (PRO) approach, the control effectiveness matrix $\mathbf{\Lambda}$ can be separated into $\mathbf{\Lambda}_{a,r} = \mathbf{V}_{a,r}^T \cdot \mathbf{\Lambda} \in \mathbb{R}^{4\times 6}$ and $\mathbf{\Lambda}_{a,n} = \mathbf{V}_{a,n}^T \cdot \mathbf{\Lambda} \in \mathbb{R}^{2\times 6}$. Due to the structure of the $\mathbf{B}_a$ matrix (5.4), $\mathbf{\Lambda}_{a,r}$ can be separated into thrust and moment parts as

$$\mathbf{\Lambda}_{a,r} = \begin{bmatrix} \mathbf{\Lambda}_M \\ \mathbf{\Lambda}_T \end{bmatrix}, \tag{5.5a}$$

$$\mathbf{\Lambda}_T = \mathbf{V}_T^T \mathbf{\Lambda} \in \mathbb{R}^{1\times 6}, \tag{5.5b}$$

$$\mathbf{\Lambda}_M = \mathbf{V}_M^T \mathbf{\Lambda} \in \mathbb{R}^{3\times 6}. \tag{5.5c}$$

Similarly to (3.51), the control effectiveness matrix can be transformed back by using the orthogonality of $\mathbf{V}$ as

$$\begin{bmatrix} \mathbf{\Lambda}_M \\ \mathbf{\Lambda}_L \\ \mathbf{\Lambda}_{a,n} \end{bmatrix} = \begin{bmatrix} \mathbf{V}_M^T \\ \mathbf{V}_T^T \\ \mathbf{V}_{a,n}^T \end{bmatrix} \cdot \mathbf{\Lambda} = \mathbf{V}^T \cdot \mathbf{\Lambda} \quad \Rightarrow \quad \mathbf{\Lambda} = \mathbf{V} \cdot \begin{bmatrix} \mathbf{\Lambda}_M \\ \mathbf{\Lambda}_L \\ \mathbf{\Lambda}_{a,n} \end{bmatrix}.$$

Using the SVD of $\mathbf{B}_a$ (5.3), the partitions $\mathbf{B}_{a,T}$ and $\mathbf{B}_{a,M}$ (5.4) and the partitions of the control effectiveness matrix $\mathbf{\Lambda}_T$ and $\mathbf{\Lambda}_M$ (5.5), the virtual control inputs can be separated into thrust and moments as

$$T(t) = U_T \cdot D_T \cdot \mathbf{\Lambda}_T \cdot \mathbf{u}(t), \tag{5.6a}$$

$$\left(\overrightarrow{\mathbf{M}}_P^R\right)_B (t) = \mathbf{U}_M \cdot \mathbf{D}_M \cdot \mathbf{\Lambda}_M \cdot \mathbf{u}(t). \tag{5.6b}$$

This corresponds to the actuator model (5.1e) with the reduced parameter version using PRO. The order of the singular values in $\mathbf{D}_a$ and therefore the order of the singular vectors in $\mathbf{U}_a$ and $\mathbf{V}_r$ depend on the parameters of the specific multicopter. Nevertheless, the separation between thrust and moments as in (5.6) is possible as long as all the rotor axes are aligned with the body-fixed axis $\mathbf{z}_B$.

5.2 Control Allocation

After normalizing the control inputs as in Section 4.3.2 and parameterizing the control effectiveness matrix as in (5.6), the control allocation problem to be solved is defined as follows.

Control Task 5.1 (Multirotor Control Allocation) *Given a desired virtual control input $\boldsymbol{\nu}_d(t) \in \mathbb{R}^4$, compute the control inputs $\mathbf{u}(\mathbf{B}_{\nu u}(t), \boldsymbol{\nu}_d(t)) \in \mathbb{R}^6$ such that*

$$\boldsymbol{\nu}_d(t) = \mathbf{B}_{\nu u}(t)\, \mathbf{u}(t), \tag{5.7}$$

with $\mathbf{B}_{\nu u}(t) \in \mathbb{R}^{4\times 6}$ and such that the input lies within the input set $\mathcal{U} \subset \mathbb{R}^6$ defined as

$$\mathcal{U} := \{\mathbf{u} \in \mathbb{R}^6 \mid -1 \leq u_i \leq 1, \forall i = 1...6\},$$

where u_i is the i-th element of $\mathbf{u}(t)$. Note that the control input set $\mathcal{U}$ is a 6-dimensional hypercube. The matrix $\mathbf{B}_{\nu u}(t)$ is given by the estimate

$$\mathbf{B}_{\nu u}(t) = \mathbf{B}_a \hat{\mathbf{\Lambda}}(t) = \mathbf{U}_{a,r} \mathbf{D}_a \hat{\mathbf{\Lambda}}_{a,r}(t) = \begin{bmatrix} U_T \cdot D_T \cdot \hat{\mathbf{\Lambda}}_T(t) \\ \mathbf{U}_M \cdot \mathbf{D}_M \cdot \hat{\mathbf{\Lambda}}_M(t) \end{bmatrix} \tag{5.8}$$

using the definition (5.4)*. Here,* $\hat{\mathbf{\Lambda}}_T(t) \in \mathbb{R}^{1\times 6}$ *and* $\hat{\mathbf{\Lambda}}_M(t) \in \mathbb{R}^{3\times 6}$ *are the estimates of* $\mathbf{\Lambda}_T$ *and* $\mathbf{\Lambda}_M$ *respectively. The desired virtual control can be separated as*

$$\boldsymbol{\nu}_d(t) = \begin{pmatrix} \nu_{T,d}(t) \\ \vec{\boldsymbol{\nu}}_{M,d}(t) \end{pmatrix} \tag{5.9}$$

into the desired thrust $\nu_{T,d}(t) \in \mathbb{R}$ *and the desired moment* $\vec{\boldsymbol{\nu}}_{M,d}(t) \in \mathbb{R}^3$*.*

If the desired virtual control input $\boldsymbol{\nu}_d(t)$ *is unattainable, prioritize the constraints*

$$\nu_{d,j}(t) = \mathbf{B}_{\nu u,rj}(t)\,\mathbf{u}(t)$$

according to the selected order:

1. *Roll and pitch moments* $L_d(t)$ *and* $M_d(t)$*.*
2. *Total thrust* $T_d(t)$*.*
3. *Yaw moment* $N_d(t)$*.*

Here, $\mathbf{B}_{\nu u,rj}(t) \in \mathbb{R}^{1\times 6}$ *is the* j*-th row of* $\mathbf{B}_{\nu u}(t)$*.*

The formulation of Control Task 5.1 allows for a separated design of the desired thrust $\nu_{T,d}(t)$ and the estimate $\hat{\mathbf{\Lambda}}_T(t)$ from the design of the desired moment $\vec{\boldsymbol{\nu}}_{M,d}(t)$ and the estimate $\hat{\mathbf{\Lambda}}_M(t)$. A coupling between the different subsystems will only exist if input saturation does not permit an independent selection of the 4 virtual controls $\boldsymbol{\nu}_d(t)$. This separation is only important for the subsequent derivation of the rigid body controllers and has no influence on the control allocation itself.

The Control Task 5.1 is fulfilled using the Prioritizing Control Allocation Strategy (PRIO CA) as presented in Section 4.3 given that the SVD of the input matrix $\mathbf{B}_{\nu u}(t)$ is known. The SVD of the input matrix $\mathbf{B}_{\nu u}(t)$ can be continuously computed given its derivative

$$\dot{\mathbf{B}}_{\nu u}(t) = \begin{bmatrix} U_T \cdot D_T \cdot \dot{\hat{\mathbf{\Lambda}}}_T(t) \\ \mathbf{U}_M \cdot \mathbf{D}_M \cdot \dot{\hat{\mathbf{\Lambda}}}_M(t) \end{bmatrix}$$

using the SVD update algorithm presented in Section 4.4. The derivatives $\dot{\hat{\mathbf{\Lambda}}}_T(t) \in \mathbb{R}^{1\times 6}$ and $\dot{\hat{\mathbf{\Lambda}}}_M(t) \in \mathbb{R}^{3\times 6}$ are known since they correspond to the P-ACA adaptive update laws. Having addressed the control allocation, the following sections deal with the rigid body controllers composed of a baseline controller and an adaptive augmentation as shown in Figure 5.1. For this derivation, the virtual controls $\nu_{T,d}(t)$ and $\vec{\boldsymbol{\nu}}_{M,d}(t)$ are considered to be the control inputs of the rigid body dynamics.

5.3 Attitude and Rate Controller

In this section, the inner loop baseline controller as in Figure 5.2 is designed. The control problem is formulated as a reduced attitude tracking problem. Reduced attitude instead of full attitude control has been selected because of the multirotor system characteristics. As discussed in Chapter 2, this choice facilitates the controller tuning and

enables better performance by avoiding coupling of motions with different time constants and noise characteristics. Furthermore, the problem is formulated directly in the attitude configuration space $SO(3)$. The main motivation is the maximization of the utilizable flying envelop which corresponds to the thesis objective 9.

In Section 5.3.1, the nonlinear reduced attitude control problem is addressed and error dynamics with an almost globally asymptotically stable and a locally exponentially stable equilibrium are derived. In Section 5.3.2, the decoupled control of the reduced attitude vector $(\vec{\mathbf{z}}_B)_N(t)$ and yaw rate $(w_{c,z})_B(t)$ is analyzed. This is important for the degraded control authority case, where the desired yaw moment might not be achieved due to the prioritization defined in Control Task 5.1 (Multirotor Control Allocation). Based on these results the baseline controller is derived. It is composed of an attitude reference model and the attitude tracking control law which are presented in Sections 5.3.3 and 5.3.4. The adaptive augmentation is presented in Section 5.4.

5.3.1 Nonlinear Attitude Control

The attitude dynamics are given by a second order differential equation corresponding to a subsystem of the rigid body dynamics $(5.1\mathrm{c})$-$(5.1\mathrm{d})$. For the derivation of the baseline controller, the propulsion moments $\overrightarrow{\mathbf{M}}_P^R(t)$ are considered as the control input and a zero moment disturbance $\vec{\mathbf{d}}_M$ is assumed. Hence, the attitude dynamics are given by

$$\dot{\mathbf{M}}_{NB}(t) = \mathbf{M}_{NB}(t) \cdot \boldsymbol{\Omega}\left(\left(\vec{\boldsymbol{\omega}}^{NB}\right)_B(t)\right), \tag{5.10a}$$

$$\left(\dot{\vec{\boldsymbol{\omega}}}^{NB}\right)_B^B(t) = \left(\mathbf{I}^R\right)_{BB}^{-1}\mathbf{f}_c\left(\left(\vec{\boldsymbol{\omega}}^{NB}\right)_B(t)\right) + \left(\mathbf{I}^R\right)_{BB}^{-1} \cdot \left(\overrightarrow{\mathbf{M}}_P^R\right)_B(t). \tag{5.10b}$$

Since this configuration space does not correspond to an Euclidean space, the attitude control problem is inherently nonlinear and has several characteristics worth noting. One of the most relevant differences to a linear system is that it is impossible to achieve global stabilization for an attitude equilibrium using continuous time-invariant feedback [10, 26]. This can be illustrated in the one dimensional case as in [26]. Consider the stabilization of the multicopter thrust vector for a planar motion as depicted in Figure 5.3. Here, $\vec{\mathbf{x}}_B$ and $\vec{\mathbf{z}}_B$ correspond to the body-fixed axes. Let the indicated point A be the desired equilibrium towards which the propulsion force vector $\vec{\mathbf{F}}_P^R$ should point. From any other attitude there are two paths to it: clockwise (positive) or anticlockwise (negative). A continuous attitude feedback translates to a continuous time-invariant moment field according to the desired spring moment. In this case another equilibrium is necessarily created. The reason is that the moment field always needs to change direction from clockwise to counterclockwise and therefore it crosses a second zero moment point (apart from desired the equilibrium). Because of the symmetry of the problem, this point usually corresponds to a $180\ [^\circ]$ attitude error, that is point B in Figure 5.3. This same concept applies for attitude stabilization problems in higher dimensions like the reduced or full attitude control problems. Furthermore, this principle is independent from the used attitude parameters and hence applies for control laws using rotation matrices, Euler angles and quaternions the same way.

In order to address this topological limitation of attitude control and to differentiate it from other weaker local stability results, the concept of almost-global stabilization has been introduced.

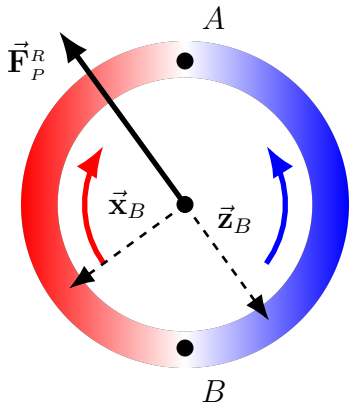


Figure 5.3: Continuous Time-Invariant Moment Vector Field

Definition 5.1 (Almost global asymptotic stability) *[113] An equilibrium $\mathbf{x}_d$ of the dynamic system $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t))$ is almost globally asymptotically stable if it is asymptotically stable with a region of attraction that is almost global. This means that the set outside the region of attraction has Lebesgue measure equal to zero.*

This definition is a natural extension of local asymptotic stability for systems that cannot achieve global asymptotic stability due to topological restrictions of the dynamical system. As previously discussed, this is the case for the attitude control problem using a time-invariant continuous control law since more than one equilibrium points exist. In order to design a controllers that achieve almost global asymptotic stability in the closed-loop, the following approach is used as in [25, 55, 94, 113].

Theorem 5.2 (Almost Global Asymptotic Stability) *An equilibrium $\mathbf{x}_e$ of the dynamic system $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t))$ is almost globally asymptotically stable if*

1. *all the solutions $\mathbf{x}(t)$ of the system are bounded and converge to an equilibrium point for $t \to \infty$,*
2. *the equilibrium $\mathbf{x}_e$ is locally asymptotically stable,*
3. *and all the other equilibrium points are unstable.*

The main fact used for this theorem is that the region of attraction of the desired equilibrium $\mathbf{x}_e$ excludes the stable manifolds of the unstable equilibria which have zero Lebesgue measure [94].

In the following, closed-loop error dynamics of the reduced attitude are derived such that the desired equilibrium is almost globally asymptotically stable and locally exponentially stable. Instead of tracking the full attitude in $SO(3)$, the idea is to track the reduced-attitude vector $\vec{\mathbf{z}}_B(t)$ defined in the 3-dimensional unit sphere $\mathbb{S}^2$, that is $\{\vec{\mathbf{z}}_B(t) \in \mathbb{R}^3, ||\vec{\mathbf{z}}_B(t)|| = 1\}$. The vector $\vec{\mathbf{z}}_B(t)$ points along the body-fixed z-axis and corresponds to the inverted direction of the total propulsion force. Hence, this reduce-attitude representation is directly used to control the position and is therefore a natural choice for a tracking variable of the inner loop controller.

For the derivation of the tracking controller, a twice continuously differentiable desired reduced-attitude vector $\vec{\mathbf{z}}_d(t) \in \mathbb{S}^2, \mathbb{C}^2$ is considered. Its first derivative has the form

$$\left(\dot{\vec{\mathbf{z}}}_d\right)_N^N(t) = \left(\vec{\boldsymbol{\omega}}^{ND}\right)_N(t) \times \left(\vec{\mathbf{z}}_d\right)_N(t). \tag{5.11}$$

Here, $\left(\vec{\boldsymbol{\omega}}^{ND}\right)_N(t) \in \mathbb{R}^3$ is the rotational rate of the desired vector $\left(\vec{\mathbf{z}}_d\right)_N$ with respect to the N-frame given in the N-frame. The kinematic constraint (5.11) matches the structure of the derivative $\left(\dot{\vec{\mathbf{z}}}_B\right)_N^N(t)$ (2.17). It arises from the fact that the vector is only allowed to rotate but not to change its length. This can be checked by computing its derivative along the direction of the vector $\left(\vec{\mathbf{z}}_d\right)_N(t)$ as

$$\begin{aligned}
\left(\vec{\mathbf{z}}_d\right)_N^T(t)\left(\dot{\vec{\mathbf{z}}}_d\right)_N^N(t) &= \left(\vec{\mathbf{z}}_d\right)_N^T(t)\left(\left(\vec{\boldsymbol{\omega}}^{ND}\right)_N(t) \times \left(\vec{\mathbf{z}}_d\right)_N(t)\right) \\
&= \left(\vec{\boldsymbol{\omega}}^{ND}\right)_N^T(t)\left(\left(\vec{\mathbf{z}}_d\right)_N(t) \times \left(\vec{\mathbf{z}}_d\right)_N(t)\right) \\
&= 0.
\end{aligned}$$

Here, the Lemmas D.1 (Cross Product) and D.2 (Scalar Triple Product) in Appendix D have been used. The tracking problem is defined in the following task.

Control Task 5.2 (Reduced Attitude Tracking Control) *Given the plant* (5.10)*, the body-fixed unit vector* $\vec{\mathbf{z}}_B(t) \in \mathbb{S}^2$ *and the desired unit vector* $\vec{\mathbf{z}}_d(t) \in \mathbb{S}^2$*, find a continuous control law* $\overrightarrow{\mathbf{M}}_P^R(t)$ *such that the equilibrium* $\vec{\mathbf{z}}_B(t) = \vec{\mathbf{z}}_d(t)$ *is almost globally asymptotically stable.*

In a first step, the attitude error kinematics are derived. This is done by deriving the desired vector in the body-fixed system $\left(\vec{\mathbf{z}}_d\right)_B(t) = \mathbf{M}_{BN}(t) \cdot \left(\vec{\mathbf{z}}_d\right)_N(t)$. In this frame, the desired equilibrium is given by $\left(\vec{\mathbf{z}}_d\right)_B(t) = \left(\vec{\mathbf{z}}_B\right)_B(t) = [0,0,1]^T$. In the following the derivative of $\mathbf{M}_{BN}$ is computed. Using Lemma D.1 (Cross Product), the attitude kinematics (5.10a) can be rewritten as

$$\begin{aligned}
\dot{\mathbf{M}}_{NB}(t) &= \mathbf{M}_{NB}(t) \cdot \boldsymbol{\Omega}\left(\left(\vec{\boldsymbol{\omega}}^{NB}\right)_B(t)\right), \\
&= \boldsymbol{\Omega}\left(\mathbf{M}_{NB}(t) \cdot \left(\vec{\boldsymbol{\omega}}^{NB}\right)_B(t)\right) \cdot \mathbf{M}_{NB}, \\
&= \boldsymbol{\Omega}\left(\left(\vec{\boldsymbol{\omega}}^{NB}\right)_N(t)\right) \cdot \mathbf{M}_{NB}.
\end{aligned}$$

Therefore, by noting that $\boldsymbol{\Omega}\left(\left(\vec{\boldsymbol{\omega}}^{NB}\right)_N(t)\right)$ is skew-symmetric it holds that

$$\begin{aligned}
\dot{\mathbf{M}}_{BN}(t) &= \dot{\mathbf{M}}_{NB}^T(t), \\
&= \mathbf{M}_{NB}^T \cdot \boldsymbol{\Omega}\left(\left(\vec{\boldsymbol{\omega}}^{NB}\right)_N(t)\right)^T, \\
&= -\mathbf{M}_{BN} \cdot \boldsymbol{\Omega}\left(\left(\vec{\boldsymbol{\omega}}^{NB}\right)_N(t)\right).
\end{aligned} \tag{5.12}$$

Finally, using the attitude kinematics (5.12), the desired attitude kinematics (5.11), the definition of $\boldsymbol{\Omega}(\cdot)$ (2.14), and Lemma D.1 (Cross Product), the rotation of the desired vector in the body-fixed system $\left(\vec{\mathbf{z}}_d\right)_B(t) = \mathbf{M}_{BN}(t) \cdot \left(\vec{\mathbf{z}}_d\right)_N(t)$ is described by

$$\begin{aligned}
\left(\dot{\vec{\mathbf{z}}}_d\right)_B^B(t) &= \dot{\mathbf{M}}_{BN}(t)\left(\vec{\mathbf{z}}_d\right)_N(t) + \mathbf{M}_{BN}(t)\left(\dot{\vec{\mathbf{z}}}_d\right)_N^N(t), \\
&= -\mathbf{M}_{BN}(t)\left[\left(\vec{\boldsymbol{\omega}}^{NB}\right)_N(t) \times \left(\vec{\mathbf{z}}_d\right)_N(t)\right] + \mathbf{M}_{BN}(t)\left[\left(\vec{\boldsymbol{\omega}}^{ND}\right)_N(t) \times \left(\vec{\mathbf{z}}_d\right)_N(t)\right], \\
&= -\mathbf{M}_{BN}(t)\left[\left(\vec{\boldsymbol{\omega}}^{NB}(t) - \vec{\boldsymbol{\omega}}^{ND}(t)\right) \times \left(\vec{\mathbf{z}}_d\right)_N(t)\right], \\
&= -\left(\vec{\boldsymbol{\omega}}^{DB}\right)_B(t) \times \left(\vec{\mathbf{z}}_d\right)_B(t).
\end{aligned}$$

Here, the error rate is defined as the difference between the body and the desired rotational rates $\vec{\boldsymbol{\omega}}^{DB}(t) := \vec{\boldsymbol{\omega}}^{NB}(t) - \vec{\boldsymbol{\omega}}^{ND}(t) \in \mathbb{R}^3$. Since the motion kinematics cannot be changed by the control law, the desired closed-loop dynamics need to be accordingly designed. In this work they are selected as

$$\left(\dot{\vec{\mathbf{z}}}_d\right)_B^B(t) = -\left(\vec{\boldsymbol{\omega}}^{DB}\right)_B(t) \times \left(\vec{\mathbf{z}}_d\right)_B(t), \tag{5.13a}$$

$$\left(\dot{\vec{\boldsymbol{\omega}}}^{DB}\right)_B^B(t) = -k_a\left(\left(\vec{\mathbf{z}}_d\right)_B(t) \times \left(\vec{\mathbf{z}}_B\right)_B\right) - \mathbf{K}_\omega\left(\vec{\boldsymbol{\omega}}^{DB}\right)_B(t), \tag{5.13b}$$

where $k_a \in \mathbb{R}$ is a positive spring constant and $\mathbf{K}_\omega \in \mathbb{R}^{3\times3}$ is a diagonal positive definite damping matrix. Further, note that $\left(\vec{\mathbf{z}}_B\right)_B(t) = [0,0,1]^T$. The first term in (5.13b) corresponds to an attitude dependent spring moment and the second term is a velocity dependent damping moment.

In order to understand the effect of the first term, let

$$\vec{\mathbf{e}}_a^{DB}(t) := \vec{\mathbf{z}}_d(t) \times \vec{\mathbf{z}}_B(t) \in \mathbb{R}^3 \tag{5.14}$$

be defined as the attitude error vector. This vector points in the direction of a rotation axis that rotates $\vec{\mathbf{z}}_d(t)$ towards $\vec{\mathbf{z}}_B(t)$ and its magnitude is given by

$$||\vec{\mathbf{e}}_a^{DB}(t)|| = ||\vec{\mathbf{z}}_d(t) \times \vec{\mathbf{z}}_B(t)|| = |\sin\left(\varphi_d\left(\vec{\mathbf{z}}_B(t), \vec{\mathbf{z}}_d(t)\right)\right)|, \tag{5.15}$$

where $\varphi_d(t) \in [0,\pi]$ is the angle between $\vec{\mathbf{z}}_d(t)$ and $\vec{\mathbf{z}}_B(t)$. The magnitude of the attitude error vector over the angle $\varphi_d(t)$ is depicted in Figure 5.4. Since the moment field corresponding to $-k_a \cdot \left(\vec{\mathbf{e}}_a^{DB}\right)_B(t)$ is continuous and only depends on the error state $\left(\vec{\mathbf{z}}_d\right)_B$ it is known that the closed-loop dynamics (5.13) have more than one equilibrium point. Therefore, almost global asymptotic stability of the desired equilibrium is the best stability result. In the following, it is shown that the dynamics (5.13) satisfy the Control Task 5.2 and in Section 5.3.4 the corresponding control law $\overrightarrow{\mathbf{M}}_P^R(t)$ is derived such that the closed-loop dynamics (5.13) are achieved. The equilibria of (5.13) are analyzed as in [26] in a first step.

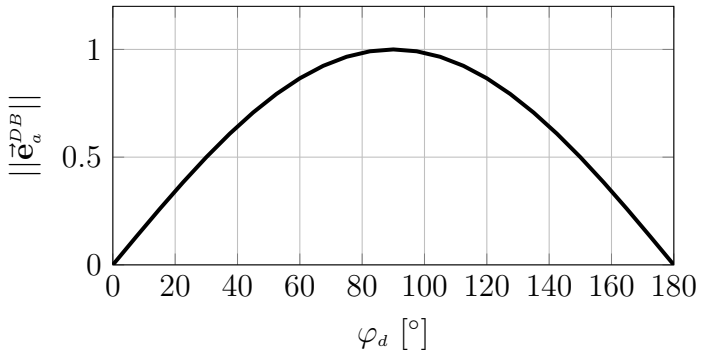


Figure 5.4: Attitude Error Vector Magnitude

Theorem 5.3 (Attitude Equilibrium Points) *Given the system dynamics* (5.13)*, the set of equilibrium points is given by* $\boldsymbol{\chi}_e = \left(\left(\vec{\mathbf{z}}_{d,e}\right)_B, \vec{\boldsymbol{\omega}}_e^{DB}\right) = \left(\pm\left(\vec{\mathbf{z}}_B\right)_B, \mathbf{0}\right) = \left(\pm[0,0,1]^T, \mathbf{0}\right)$*. The subscript* e *refers to the values at the equilibrium of the system.*

Proof: For the equilibria of the system (5.13) the following conditions hold

$$\mathbf{0} = -\left(\vec{\boldsymbol{\omega}}_e^{DB}\right)_B \times \left(\vec{\mathbf{z}}_{d,e}\right)_B, \tag{5.16a}$$

$$\mathbf{0} = -k_a \left(\left(\vec{\mathbf{z}}_{d,e}\right)_B \times \left(\vec{\mathbf{z}}_B\right)_B\right) - \mathbf{K}_\omega \left(\vec{\boldsymbol{\omega}}_e^{DB}\right)_B. \tag{5.16b}$$

By multiplying (5.16b) with $\left(\vec{\boldsymbol{\omega}}_e^{DB}\right)_B^T$ from the left, using Lemma D.2 (Scalar Triple Product) and inserting (5.16a) it follows that

$$\begin{aligned} 0 &= -k_a \left(\vec{\boldsymbol{\omega}}_e^{DB}\right)_B^T \left(\left(\vec{\mathbf{z}}_{d,e}\right)_B \times \left(\vec{\mathbf{z}}_B\right)_B\right) - \left(\vec{\boldsymbol{\omega}}_e^{DB}\right)_B^T \mathbf{K}_\omega \left(\vec{\boldsymbol{\omega}}_e^{DB}\right)_B, \\ &= -k_a \left(\vec{\mathbf{z}}_B\right)_B^T \left(\left(\vec{\boldsymbol{\omega}}_e^{DB}\right)_B^T \times \left(\vec{\mathbf{z}}_{d,e}\right)_B\right) - \left(\vec{\boldsymbol{\omega}}_e^{DB}\right)_B^T \mathbf{K}_\omega \left(\vec{\boldsymbol{\omega}}_e^{DB}\right)_B, \\ &= -\left(\vec{\boldsymbol{\omega}}_e^{DB}\right)_B^T \mathbf{K}_\omega \left(\vec{\boldsymbol{\omega}}_e^{DB}\right)_B. \end{aligned}$$

Because $\mathbf{K}_\omega$ is positive definite it follows that $\vec{\boldsymbol{\omega}}_e^{DB} = \mathbf{0}$. Inserting this result back into (5.16b) leads to

$$\mathbf{0} = -k_a \left(\left(\vec{\mathbf{z}}_{d,e}\right)_B \times \left(\vec{\mathbf{z}}_B\right)_B\right).$$

Because $k_a > 0$ it follows that $\vec{\mathbf{z}}_B(t)$ and $\vec{\mathbf{z}}_d(t)$ are parallel at the equilibrium points $\left(\vec{\mathbf{z}}_{d,e}\right)_B = \pm\left(\vec{\mathbf{z}}_B\right)_B$ with $\left(\vec{\mathbf{z}}_B\right)_B = [0,0,1]^T$. ■

Note that although both equilibria $\left(\vec{\mathbf{z}}_{d,e}\right)_B = \pm\left(\vec{\mathbf{z}}_B\right)_B$ correspond to a single point in the 3-dimensional unit sphere $\mathbb{S}^2$, in the full attitude configuration space $SO(3)$ they correspond to a set of attitudes. This is because the reduced attitude $\vec{\mathbf{z}}_B$ does not change through a rotation about the axis $\vec{\mathbf{z}}_B$. This is an intended effect, since it allows for decoupling the yaw motion from the reduced attitude and consequently the position dynamics. The following theorem is the main result of the section.

Theorem 5.4 (Reduced Attitude AGAS) *Given the system dynamics* (5.13)*, a positive gain* $k_a > 0$ *and a diagonal positive definite matrix* $\mathbf{K}_\omega$*, the desired equilibrium* $\boldsymbol{\chi}_{d,e} = \left(\left(\vec{\mathbf{z}}_B\right)_B, \mathbf{0}\right) = \left([0,0,1]^T, \mathbf{0}\right)$ *is almost globally asymptotically stable (AGAS). Furthermore, the undesired equilibrium* $\boldsymbol{\chi}_{u,e} = \left(-\left(\vec{\mathbf{z}}_B\right)_B, \mathbf{0}\right) = \left([0,0,-1]^T, \mathbf{0}\right)$ *is unstable.*

Proof: The proof is based on checking the three conditions of Theorem 5.2 (Almost Global Asymptotic Stability) using a Lyapunov approach. Since the attitude error between $\vec{\mathbf{z}}_d(t)$ and $\vec{\mathbf{z}}_B(t)$ can be measured by the angle between them $\varphi_d(t) \in [0, \pi]$, the following reduced attitude penalty function is proposed

$$V_\varphi\left(\varphi_d\left(\left(\vec{\mathbf{z}}_d\right)_B(t)\right)\right) = 1 - \left(\vec{\mathbf{z}}_B\right)_B^T \cdot \left(\vec{\mathbf{z}}_d\right)_B(t) = 1 - \cos\varphi_d(t). \tag{5.17}$$

This function is a strictly monotonically increasing function of the attitude error with $V_\varphi\left(\varphi_d(t)\right) = 0$ if and only if $\vec{\mathbf{z}}_B(t) = \vec{\mathbf{z}}_d(t)$. This function is depicted in Figure 5.5.

Using the reduced attitude penalty function $V_\varphi\left(\varphi_d\left(\left(\vec{\mathbf{z}}_d\right)_B(t)\right)\right)$ (5.17), the following positive definite Lyapunov candidate function is defined.

Lemma 5.5 (Attitude Lyapunov function) *Let the attitude Lyapunov function be*

$$\begin{aligned} V\left(\left(\vec{\mathbf{z}}_d\right)_B(t), \vec{\boldsymbol{\omega}}^{DB}(t)\right) = k_a \cdot V_\varphi\left(\varphi_d\left(\left(\vec{\mathbf{z}}_d\right)_B(t)\right)\right) + \frac{1}{2}\left(\vec{\boldsymbol{\omega}}^{DB}\right)_B^T(t)\left(\vec{\boldsymbol{\omega}}^{DB}\right)_B(t) \\ + c \cdot \left(\vec{\boldsymbol{\omega}}^{DB}\right)_B^T(t)\left(\vec{\mathbf{e}}_a^{DB}\right)_B(t), \end{aligned} \tag{5.18}$$

where $c \in \mathbb{R}$, $k_a > 0 \in \mathbb{R}$ *and* $\vec{\mathbf{e}}_a^{DB}(t) := \vec{\mathbf{z}}_d(t) \times \vec{\mathbf{z}}_B(t) \in \mathbb{R}^3$ *is defined as the attitude error vector. The Lyapunov function is defined in the configuration space* $\mathbb{S}^2 \times \mathbb{R}^3$ *and is positive definite for* $|c| \leq \sqrt{k_a}$*. Positive definite means that* $V = 0$ *if and only if* $\left(\varphi_d\left(\left(\vec{\mathbf{z}}_d\right)_B(t)\right), \vec{\boldsymbol{\omega}}^{DB}(t)\right) = (0, \mathbf{0})$*, otherwise it holds that* $V(t) > 0$*.*

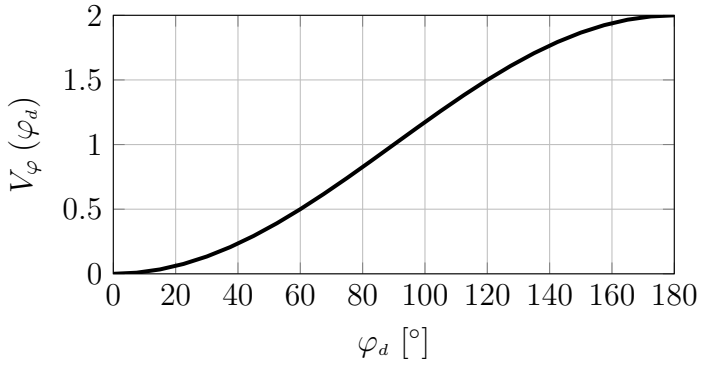


Figure 5.5: Reduced-Attitude Penalty Function

The proof is found in Lemma D.3 (Attitude Lyapunov function) in Appendix D. In order to analyze the stability properties of the system, the derivative of $V(t)$ (5.18) is needed. Using the kinematics (5.13a), the definition of $\vec{\mathbf{e}}_a^{DB}(t)$ (5.14), and Lemma D.2 (Scalar Triple Product), the derivative of the attitude penalty function (5.17) is computed as

$$\begin{aligned}\dot{V}_\varphi\left(\varphi_d\left(\left(\vec{\mathbf{z}}_d\right)_B(t)\right)\right) &= -\left(\vec{\mathbf{z}}_B\right)_B^T \cdot \left(\dot{\vec{\mathbf{z}}}_d\right)_B^B(t)\,, \\ &= \left(\vec{\mathbf{z}}_B\right)_B^T \cdot \left(\left(\vec{\boldsymbol{\omega}}^{DB}\right)_B(t) \times \left(\vec{\mathbf{z}}_d\right)_B(t)\right), \\ &= \left(\vec{\boldsymbol{\omega}}^{DB}\right)_B^T(t) \cdot \left(\left(\vec{\mathbf{z}}_d\right)_B(t) \times \left(\vec{\mathbf{z}}_B\right)_B\right), \\ &= \left(\vec{\mathbf{e}}_a^{DB}\right)_B^T(t) \cdot \left(\vec{\boldsymbol{\omega}}^{DB}\right)_B(t)\,.\end{aligned} \tag{5.19}$$

The derivative of the Lyapunov function $\dot{V}(t)$ is then given by

$$\begin{aligned}\dot{V} &= k_a \cdot \dot{V}_\varphi + \left(\vec{\boldsymbol{\omega}}^{DB}\right)_B^T \left(\dot{\vec{\boldsymbol{\omega}}}^{DB}\right)_B^B + c \cdot \left(\vec{\mathbf{e}}_a^{DB}\right)_B^T \left(\dot{\vec{\boldsymbol{\omega}}}^{DB}\right)_B^B + c \cdot \left(\vec{\boldsymbol{\omega}}^{DB}\right)_B^T \left(\dot{\vec{\mathbf{e}}}_a^{DB}\right)_B^B, \\ &= k_a \cdot \left(\vec{\mathbf{e}}_a^{DB}\right)_B^T \left(\vec{\boldsymbol{\omega}}^{DB}\right)_B + c \cdot \left(\vec{\boldsymbol{\omega}}^{DB}\right)_B^T \left(\dot{\vec{\mathbf{e}}}_a^{DB}\right)_B^B + \left(\left(\vec{\boldsymbol{\omega}}^{DB}\right)_B + c \cdot \left(\vec{\mathbf{e}}_a^{DB}\right)_B\right)^T \left(\dot{\vec{\boldsymbol{\omega}}}^{DB}\right)_B^B,\end{aligned} \tag{5.20}$$

where the time dependency has not been explicitly stated for the sake of readability. The attitude error derivative can be bounded as follows.

Lemma 5.6 (Attitude Error Derivative Bound) *The norm of the derivative of the attitude error vector $\vec{\mathbf{e}}_a^{DB}(t) = \vec{\mathbf{z}}_d(t) \times \vec{\mathbf{z}}_B(t)$ can be bounded by the norm of the error rotational rates*

$$\left|\left|\left(\dot{\vec{\mathbf{e}}}_a^{DB}\right)_B^B(t)\right|\right| \leq \left|\left|\left(\vec{\boldsymbol{\omega}}^{BD}\right)_B(t)\right|\right|.$$

The proof is found in Lemma D.4 (Attitude Error Derivative Bound) in Appendix D. By inserting the system dynamics (5.13) and Lemma 5.6 (Attitude Error Derivative Bound) into (5.20), the derivative of the Lyapunov function $\dot{V}(t)$ can be bounded from above by

$$\begin{aligned}\dot{V} &\leq k_a \cdot \left(\vec{\mathbf{e}}_a^{DB}\right)_B^T \left(\vec{\boldsymbol{\omega}}^{DB}\right)_B + c \cdot \left|\left|\left(\vec{\boldsymbol{\omega}}^{DB}\right)_B\right|\right|^2 \\ &\quad + \left(\left(\vec{\boldsymbol{\omega}}^{DB}\right)_B + c \cdot \left(\vec{\mathbf{e}}_a^{DB}\right)_B\right)^T \left(-k_a \left(\vec{\mathbf{e}}_a^{DB}\right)_B - \mathbf{K}_\omega \left(\vec{\boldsymbol{\omega}}^{DB}\right)_B\right), \\ &\leq -c \cdot k_a \cdot \left|\left|\left(\vec{\mathbf{e}}_a^{DB}\right)_B\right|\right|^2 + \left(\vec{\boldsymbol{\omega}}^{DB}\right)_B^T \left(\mathbf{I} \cdot c - \mathbf{K}_\omega\right) \left(\vec{\boldsymbol{\omega}}^{DB}\right)_B - c \cdot \left(\vec{\mathbf{e}}_a^{DB}\right)_B^T \mathbf{K}_\omega \left(\vec{\boldsymbol{\omega}}^{DB}\right)_B.\end{aligned}$$

By defining the error vector $\boldsymbol{\zeta}(t) \in \mathbb{R}^6$ and the symmetric matrix $\mathbf{Q} \in \mathbb{R}^{6\times 6}$ as

$$\boldsymbol{\zeta}(t) = \begin{pmatrix} \left(\vec{\mathbf{e}}_a^{DB}\right)_B(t) \\ \left(\vec{\boldsymbol{\omega}}^{DB}\right)_B(t) \end{pmatrix}, \qquad \mathbf{Q} = \begin{bmatrix} c\cdot k_a \cdot \mathbf{I} & \frac{c}{2}\mathbf{K}_\omega \\ \frac{c}{2}\mathbf{K}_\omega & \mathbf{K}_\omega - c\cdot\mathbf{I} \end{bmatrix},$$

the derivative of the Lyapunov candidate (5.18) can be rewritten as

$$\dot{V}(t) \leq -\boldsymbol{\zeta}^T(t)\cdot\mathbf{Q}\cdot\boldsymbol{\zeta}(t) =: -W(\boldsymbol{\zeta}(t)). \tag{5.21}$$

By the correct selection of c, a positive definite matrix $\mathbf{Q}$ can always be found. This is stated in the following lemma and the corresponding proof is found in Lemma D.5 (Positive Definite $\mathbf{Q}$) in Appendix D.

Lemma 5.7 (Positive Definite Q) *If $\mathbf{K}_\omega \in \mathbb{R}^{3\times 3}$ is symmetric positive definite and $k_a > 0$, there exists a constant $c \in]0, \sqrt{k_a}[$ such that the matrix $\mathbf{Q}$*

$$\mathbf{Q} = \begin{bmatrix} c\cdot k_a \cdot \mathbf{I} & \frac{c}{2}\mathbf{K}_\omega \\ \frac{c}{2}\mathbf{K}_\omega & \mathbf{K}_\omega - c\cdot\mathbf{I} \end{bmatrix}$$

is positive definite.

Hence, from Lemmas 5.5 (Attitude Lyapunov function) and 5.7 (Positive Definite $\mathbf{Q}$), it follows that there exists a constant $c \in]0, \sqrt{k_a}[$ such that the matrix $\mathbf{Q}$ is symmetric positive definite and simultaneously the Lyapunov candidate (5.18) is positive definite. Consequently, the function $W(\boldsymbol{\zeta}(t))$ in (5.21) is positive definite with respect to $\boldsymbol{\zeta}(t)$. Furthermore, note that $\boldsymbol{\zeta}(t) = \mathbf{0}$ holds only if

$$\begin{aligned} ||\vec{\mathbf{e}}_a^{DB}(t)|| &= |\sin\left(\varphi_d\left(\left(\vec{\mathbf{z}}_d\right)_B(t)\right)\right)| = 0, \\ ||\vec{\boldsymbol{\omega}}^{DB}|| &= 0. \end{aligned}$$

Here, the magnitude of $\vec{\mathbf{e}}_a^{DB}(t)$ (5.15) has been used. $\boldsymbol{\zeta}(t) = \mathbf{0}$ corresponds to the set of equilibria $\boldsymbol{\chi}_e = \left(\left(\vec{\mathbf{z}}_{d,e}\right)_B, \vec{\boldsymbol{\omega}}_e^{DB}\right) = (\pm[0,0,1]^T, \mathbf{0})$. Therefore, the derivative $\dot{V}(t) \leq 0$ is negative semi-definite and not negative definite.

From Theorem 4.8 (Uniform Stability) in [87], it follows that the desired equilibrium $\boldsymbol{\chi}_{d,e} = ([0,0,1]^T, \mathbf{0})$ is uniformly stable and that the errors $\varphi_d\left(\left(\vec{\mathbf{z}}_d\right)_B(t)\right)$ and $\vec{\boldsymbol{\omega}}^{DB}(t)$ are bounded. Using Theorem 8.4 (Invariance-like Theorem) in [87], it further follows that $-W(\boldsymbol{\zeta}(t)) \to 0$ for $t \to \infty$. Because $\mathbf{Q}$ is symmetric positive definite, this implies that $||\boldsymbol{\zeta}(t)|| \to 0$ for $t \to \infty$. This corresponds to Condition 1 in Theorem 5.2. Conditions 2 and 3 in Theorem 5.2 are shown through linearization and summarized in the following lemma.

Lemma 5.8 (Linearization: Reduced Attitude) *The reduced attitude closed-loop system (5.13) has a stable equilibrium at $\boldsymbol{\chi}_{d,e} = \left(\left(\vec{\mathbf{z}}_{d,e}\right)_B, \vec{\boldsymbol{\omega}}_e^{DB}\right) = ([0,0,1]^T, \mathbf{0})$ and an unstable equilibrium at $\boldsymbol{\chi}_{u,e} = \left(\left(\vec{\mathbf{z}}_{d,e}\right)_B, \vec{\boldsymbol{\omega}}_e^{DB}\right) = ([0,0,-1]^T, \mathbf{0})$ if k_a is positive and $\mathbf{K}_\omega$ is a diagonal positive definite matrix. The subscript e refers to equilibrium points.*

The proof is found in Lemma D.6 (Linearization: Reduced Attitude) in Appendix D. Using Theorem 5.2 (Almost Global Asymptotic Stability) it is concluded that the desired equilibrium $\boldsymbol{\chi}_{d,e} = ([0,0,1]^T, \mathbf{0})$ is almost globally asymptotically stable and this finally proves Theorem 5.4 (Reduced Attitude AGAS). ■

From Lemma 5.8 (Linearization: Reduced Attitude) it is known that linearized system at the equilibrium $\boldsymbol{\chi}_{d,e} = ([0,0,1]^T, \mathbf{0})$ has only eigenvalues with negative real parts. Then from Theorem 4.13 (Lyapunov's Indirect Method) in [87] it follows that the equilibrium is locally exponentially stable. The set of initial states for which the exponential decay holds is refined in the following theorem and the proof is given in Lemma D.9 (Local Exponential Stability) in Appendix D.

Theorem 5.9 (Local Exponential Stability) *Given the system dynamics* (5.13)*, the desired equilibrium* $\boldsymbol{\chi}_{d,e} = ((\vec{\mathbf{z}}_{d,e})_B, \vec{\boldsymbol{\omega}}_e^{DB}) = ((\vec{\mathbf{z}}_B)_B, \mathbf{0}) = ([0,0,1]^T, \mathbf{0})$ *is locally exponentially stable and the Lyapunov function* (5.18) *decays exponentially given the initial conditions*

$$V_\varphi\left(\varphi_d(t=0)\right) \leq \overline{V}_\varphi,$$
$$||\vec{\boldsymbol{\omega}}^{DB}(t=0)|| \leq \sqrt{2k_a \cdot \left(\overline{V}_\varphi - V_\varphi\left(\varphi_d(t=0)\right)\right)},$$

for a constant $\overline{V}_\varphi \in \mathbb{R}$ *such that* $0 < \overline{V}_\varphi < 2$.

Note that given a zero velocity initial condition $||\vec{\boldsymbol{\omega}}^{DB}|| = 0$, an exponentially decaying attitude Lyapunov function can be found as long as the initial attitude error angle is $|\varphi_d(t=0)| < \pi \Rightarrow V_\varphi\left(\varphi_d(t=0)\right) < 2$. Furthermore, the smaller the initial attitude error angle $\varphi_d(t=0)$ the larger the permitted initial error velocity $\vec{\boldsymbol{\omega}}^{DB}(t=0)$ can be.

5.3.2 Yaw Control Decoupling

From the attainable control set (ACS) analysis of the two hexacopter configurations in Section 2.3, it is known that a degraded control authority case is unavoidable if the failure of any actuator is considered. Control Task 5.1 (Multirotor Control Allocation) describes a prioritizing control allocation strategy and therefore it is expected that during a failure the desired yaw moment $N_d(t)$ is not always achieved. A decoupled control of the yaw motion and the reduced attitude is desirable such that the degradation is limited to the yaw control. In this section the interaction between the subsystems is analyzed.

In a first step, the desired closed-loop system for tracking the reduced attitude (5.13) is modified assuming that the yaw control is independently designed as follows

$$\left(\dot{\vec{\mathbf{z}}}_d\right)_B^B(t) = -\left(\vec{\boldsymbol{\omega}}^{DB}\right)_B(t) \times \left(\vec{\mathbf{z}}_d\right)_B(t), \tag{5.22a}$$
$$\left(\dot{\boldsymbol{\omega}}_{xy}^{DB}\right)_B^B(t) = -k_a\left(\mathbf{e}_{a,xy}^{DB}\right)_B(t) - k_w\left(\boldsymbol{\omega}_{xy}^{DB}\right)_B(t), \tag{5.22b}$$

where $k_a \in \mathbb{R}$ is a positive spring constant and $k_w \in \mathbb{R}$ is a positive damping constant. Furthermore, $\left(\mathbf{e}_{a,xy}^{DB}\right)_B(t) \in \mathbb{R}^2$ and $\left(\boldsymbol{\omega}_{xy}^{DB}\right)_B(t) \in \mathbb{R}^2$ correspond to the first and second elements of $\left(\vec{\mathbf{e}}_a^{DB}\right)_B(t)$ and $\left(\vec{\boldsymbol{\omega}}^{DB}\right)_B(t)$ respectively. Compared to (5.13), the spring term has not changed since the third element of $\left(\vec{\mathbf{e}}_a^{DB}\right)_B(t) := \left(\vec{\mathbf{z}}_d\right)_B(t) \times \left(\vec{\mathbf{z}}_B\right)_B = \left(\vec{\mathbf{z}}_d\right)_B(t) \times [0,0,1]^T$ (5.14) is always zero

$$\left(e_{a,z}^{DB}\right)_B(t) = 0. \tag{5.23}$$

The damping term has been modified such that only roll and pitch rates are damped. Therefore, the yaw rate is not directly used by controller and the yaw rate error $(\omega_z^{DB})_B^B(t) \in \mathbb{R}$ corresponding to the third element of $\left(\vec{\boldsymbol{\omega}}^{DB}\right)_B(t)$ is considered an external signal. In the following the equilibria of (5.22) are analyzed.

Theorem 5.10 (Roll/Pitch Attitude Equilibrium Points) *Given the desired closed-loop system dynamics* (5.22) *and a time-varying signal* $\left(\omega_z^{DB}\right)_B(t)$*, the set of equilibrium points is given by* $\overline{\boldsymbol{\chi}}_e := \left(\left(\vec{\mathbf{z}}_{d,e}\right)_B, \left(\boldsymbol{\omega}_{e,xy}^{DB}\right)_B\right) = (\pm[0,0,1]^T, \mathbf{0})$*. The subscript* e *refers to the values at the equilibrium of the system and* $\left(\boldsymbol{\omega}_{xy}^{DB}\right)_B(t) \in \mathbb{R}^2$ *corresponds to the first two elements of* $\left(\vec{\boldsymbol{\omega}}^{DB}\right)_B(t)$ *and* $\left(\omega_z^{DB}\right)_B(t) \in \mathbb{R}$ *to the third element.*

Proof: For the equilibria of the pitch and roll system (5.22) the following conditions hold

$$\mathbf{0} = -\begin{pmatrix} \left(\omega_{e,x}^{DB}\right)_B \\ \left(\omega_{e,y}^{DB}\right)_B \\ \left(\omega_z^{DB}\right)_B(t) \end{pmatrix} \times \left(\vec{\mathbf{z}}_{d,e}\right)_B, \tag{5.24a}$$

$$\mathbf{0} = -k_a \begin{pmatrix} \left(z_{d,e,y}\right)_B \\ -\left(z_{d,e,x}\right)_B \end{pmatrix} - k_w \begin{pmatrix} \left(\omega_{e,x}^{DB}\right)_B \\ \left(\omega_{e,y}^{DB}\right)_B \end{pmatrix}. \tag{5.24b}$$

Here, $\left(\omega_x^{DB}\right)_B(t)$, $\left(\omega_y^{DB}\right)_B(t)$, $\left(z_{d,x}\right)_B(t)$ and $\left(z_{d,y}\right)_B(t) \in \mathbb{R}$ are the first and second elements of $\left(\vec{\boldsymbol{\omega}}^{DB}\right)_B(t)$ and $\left(\vec{\mathbf{z}}_d\right)_B(t)$ respectively. From (5.24b), the following relations are derived

$$\left(z_{d,e,y}\right)_B = -\frac{k_w}{k_a}\left(\omega_{e,x}^{DB}\right)_B, \qquad \left(z_{d,e,x}\right)_B = \frac{k_w}{k_a}\left(\omega_{e,y}^{DB}\right)_B.$$

By inserting them in the third row of (5.24a), it follows that

$$\begin{aligned} 0 &= -\left(\omega_{e,x}^{DB}\right)_B \cdot \left(z_{d,e,y}\right)_B + \left(\omega_{e,y}^{DB}\right)_B \cdot \left(z_{d,e,x}\right)_B, \\ &= \frac{k_w}{k_a}\left(\omega_{e,x}^{DB}\right)_B^2 + \frac{k_w}{k_a}\left(\omega_{e,y}^{DB}\right)_B^2. \end{aligned}$$

Since k_a and k_w are greater than zero, it holds that $\left(\omega_{e,x}^{DB}\right)_B = \left(\omega_{e,y}^{DB}\right)_B = 0$. By inserting the result back into (5.24b), it follows that $\left(z_{d,e,x}\right)_B = \left(z_{d,e,y}\right)_B = 0$ and because of the unit length constraint of $\left(\vec{\mathbf{z}}_d\right)_B$ it holds that $\left(\vec{\mathbf{z}}_{d,e}\right)_B = \pm\left(\vec{\mathbf{z}}_B\right)_B = \pm[0,0,1]^T$. The equilibrium points are valid for a nonzero yaw error rate $\left(\omega_z^{DB}\right)_B(t)$. ■

The following theorem is the main result of the section regarding the stability assessment of the closed-loop dynamics (5.22).

Theorem 5.11 (Roll/Pitch Reduced Attitude AGAS) *Given the uniformly bounded signal* $\left|\left(\omega_z^{DB}\right)_B(t)\right| < r_{max}$ *with* $r_{max} > 0 \in \mathbb{R}$ *and the system dynamics* (5.22)*, the desired equilibrium* $\overline{\boldsymbol{\chi}}_{d,e} = ([0,0,1]^T, \mathbf{0})$ *is almost globally asymptotically stable (AGAS). Furthermore, the undesired equilibrium* $\overline{\boldsymbol{\chi}}_{u,e} = ([0,0,-1]^T, \mathbf{0})$ *is unstable.*

Proof: The proof is based on checking the three conditions of Theorem 5.2 (Almost Global Asymptotic Stability) using a Lyapunov approach. Using the reduced attitude penalty function $V_\varphi\left(\varphi_d\left(\left(\vec{\mathbf{z}}_d\right)_B(t)\right)\right)$ (5.17), the following positive definite Lyapunov candidate function is defined.

Lemma 5.12 (Roll/Pitch Attitude Lyapunov function) *Let the attitude Lyapunov function be*

$$\begin{aligned} \overline{V}\left(\left(\vec{\mathbf{z}}_d\right)_B(t), \left(\boldsymbol{\omega}_{xy}^{DB}\right)_B(t)\right) = k_a \cdot V_\varphi\left(\varphi_d\left(\left(\vec{\mathbf{z}}_d\right)_B(t)\right)\right) + \frac{1}{2}\left(\boldsymbol{\omega}_{xy}^{DB}\right)_B^T(t)\left(\boldsymbol{\omega}_{xy}^{DB}\right)_B(t) \\ + c \cdot \left(\boldsymbol{\omega}_{xy}^{DB}\right)_B^T(t)\left(\mathbf{e}_{a,xy}^{DB}\right)_B(t), \end{aligned} \tag{5.25}$$

where $c \in \mathbb{R}$, $k_a > 0 \in \mathbb{R}$ *and* $\vec{\mathbf{e}}_a^{DB}(t) := \vec{\mathbf{z}}_d(t) \times \vec{\mathbf{z}}_B(t) \in \mathbb{R}^3$ *is defined as the attitude error vector. Its first two elements correspond to the vector* $\left(\mathbf{e}_{a,xy}^{DB}\right)_B(t) \in \mathbb{R}^2$*. The Lyapunov function is defined in the configuration space* $\mathbb{S}^2 \times \mathbb{R}^2$ *and is positive definite for* $|c| \leq \sqrt{k_a}$*. Positive definite means that* $\overline{V} = 0$ *if and only if* $\left(\varphi_d\left(\left(\vec{\mathbf{z}}_d\right)_B(t)\right), \left(\boldsymbol{\omega}_{xy}^{DB}\right)_B(t)\right) = (0, \mathbf{0})$*, otherwise it holds that* $\overline{V}(t) > 0$*.*

This lemma is proven exactly as Lemma D.3 (Attitude Lyapunov function) in Appendix D by noting that $\left|\left|\left(\mathbf{e}_{a,xy}^{DB}\right)_B(t)\right|\right| = ||(\mathbf{e}_a^{DB})_B(t)|| = |\sin\varphi_d(t)|$. This follows from the fact that $\left(e_{a,z}^{DB}\right)_B(t) = 0$ (5.23).

In order to analyze the stability properties of the system, the derivative of $\overline{V}(t)$ (5.25) is needed. Using the derivative of the attitude penalty function $V_\varphi(t)$ (5.19) and $\left(e_{a,z}^{DB}\right)_B(t) = 0$ (5.23), the derivative of the Lyapunov function $\dot{\overline{V}}(t)$ is then given by

$$\begin{aligned}\dot{\overline{V}} &= k_a \cdot \dot{V}_\varphi + \left(\boldsymbol{\omega}_{xy}^{DB}\right)_B^T \left(\dot{\boldsymbol{\omega}}_{xy}^{DB}\right)_B^B + c \cdot \left(\mathbf{e}_{a,xy}^{DB}\right)_B^T \left(\dot{\boldsymbol{\omega}}_{xy}^{DB}\right)_B^B + c \cdot \left(\boldsymbol{\omega}_{xy}^{DB}\right)_B^T \left(\dot{\mathbf{e}}_{a,xy}^{DB}\right)_B^B, \\ &= k_a \cdot \left(\mathbf{e}_{a,xy}^{DB}\right)_B^T \left(\boldsymbol{\omega}_{xy}^{DB}\right)_B + c \cdot \left(\boldsymbol{\omega}_{xy}^{DB}\right)_B^T \left(\dot{\mathbf{e}}_{a,xy}^{DB}\right)_B^B + \left(\left(\boldsymbol{\omega}_{xy}^{DB}\right)_B + c \cdot \left(\mathbf{e}_{a,xy}^{DB}\right)_B\right)^T \left(\dot{\boldsymbol{\omega}}_{xy}^{DB}\right)_B^B,\end{aligned}$$

where the time dependency has not been explicitly stated for the sake of readability. Since the third element of $\left(\vec{\mathbf{e}}_a^{DB}\right)_B(t)$ is always zero, the fact that $\left(\vec{\mathbf{e}}_a^{DB}\right)_B^T(t)\left(\vec{\boldsymbol{\omega}}^{DB}\right)_B(t) = \left(\mathbf{e}_{a,xy}^{DB}\right)_B^T(t)\left(\boldsymbol{\omega}_{xy}^{DB}\right)_B(t)$ has been used. The attitude error derivative is computed as follows.

Lemma 5.13 (Roll/Pitch Attitude Error Derivative) *The derivative of the first two elements of the attitude error vector* $\vec{\mathbf{e}}_a^{DB}(t) = \vec{\mathbf{z}}_d(t) \times \vec{\mathbf{z}}_B(t)$ *can be written as*

$$\left(\dot{\vec{\mathbf{e}}}_{a,xy}^{DB}\right)_B^B(t) = \mathbf{H}(t)\left(\vec{\mathbf{e}}_{a,xy}^{DB}\right)_B(t) + \left(\boldsymbol{\omega}_{xy}^{DB}\right)_B(t)\left(z_{d,z}\right)_B(t)$$

using the time varying matrix

$$\mathbf{H}(t) = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \cdot \left(\omega_z^{DB}\right)_B(t).$$

The proof of the lemma is found in Lemma D.10 (Roll/Pitch Attitude Error Derivative) in Appendix D. Using the system dynamics (5.22), Lemma 5.13 (Roll/Pitch Attitude Error Derivative) and $\left|\left(z_{d,z}\right)_B(t)\right| \leq 1$, the derivative of the Lyapunov function $\dot{\overline{V}}(t)$ can be bounded from above by

$$\begin{aligned}\dot{\overline{V}}(t) &= k_a \cdot \left(\mathbf{e}_{a,xy}^{DB}\right)_B^T(t)\left(\boldsymbol{\omega}_{xy}^{DB}\right)_B(t) + c \cdot \left(\boldsymbol{\omega}_{xy}^{DB}\right)_B^T(t)\,\mathbf{H}(t)\left(\vec{\mathbf{e}}_{a,xy}^{DB}\right)_B(t) \\ &\quad + c \cdot \left|\left|\left(\boldsymbol{\omega}_{xy}^{DB}\right)_B(t)\right|\right|^2 \left(z_{d,z}\right)_B(t) \\ &\quad + \left(\left(\boldsymbol{\omega}_{xy}^{DB}\right)_B(t) + c \cdot \left(\mathbf{e}_{a,xy}^{DB}\right)_B(t)\right)^T \left(-k_a\left(\mathbf{e}_{a,xy}^{DB}\right)_B(t) - k_\omega\left(\boldsymbol{\omega}_{xy}^{DB}\right)_B(t)\right), \\ &\leq -c \cdot k_a \cdot \left|\left|\left(\vec{\mathbf{e}}_{a,xy}^{DB}\right)_B(t)\right|\right|^2 + (c - k_w)\left(\boldsymbol{\omega}_{xy}^{DB}\right)_B^T(t)\left(\boldsymbol{\omega}_{xy}^{DB}\right)_B(t) \\ &\quad + c \cdot \left(\vec{\mathbf{e}}_{a,xy}^{DB}\right)_B^T(t)\left(\mathbf{H}^T(t) - k_w \cdot \mathbf{I}\right)\left(\boldsymbol{\omega}_{xy}^{DB}\right)_B(t).\end{aligned}$$

By defining the error vector $\overline{\boldsymbol{\zeta}}(t) \in \mathbb{R}^4$ and the symmetric matrix $\overline{\mathbf{Q}}(t) \in \mathbb{R}^{4\times4}$ as

$$\overline{\boldsymbol{\zeta}}(t) = \begin{pmatrix} \left(\mathbf{e}_{a,xy}^{DB}\right)_B(t) \\ \left(\boldsymbol{\omega}_{xy}^{DB}\right)_B(t) \end{pmatrix}, \qquad \overline{\mathbf{Q}}(t) = \begin{bmatrix} c \cdot k_a \cdot \mathbf{I} & \frac{c}{2}\left(k_w \cdot \mathbf{I} - \mathbf{H}^T(t)\right) \\ \frac{c}{2}\left(k_w \cdot \mathbf{I} - \mathbf{H}(t)\right) & (k_w - c) \cdot \mathbf{I} \end{bmatrix},$$

the derivative of the Lyapunov candidate (5.25) can be rewritten as

$$\dot{\overline{V}}(t) \leq -\overline{\boldsymbol{\zeta}}^T(t) \cdot \overline{\mathbf{Q}}(t) \cdot \overline{\boldsymbol{\zeta}}(t) \leq -\overline{q} \cdot \left|\left|\overline{\boldsymbol{\zeta}}(t)\right|\right|^2. \tag{5.26}$$

The constant $\overline{q} \in \mathbb{R}$ is positive if the matrix $\overline{\mathbf{Q}}(t)$ is positive definite for all $t \geq 0$. This is checked by the following lemma.

Lemma 5.14 (Roll/Pitch Positive Definite $\overline{\mathbf{Q}}$) *If $k_a, k_w \in \mathbb{R}$ are positive constants and $\left(\omega_z^{DB}\right)_B(t)$ is uniformly bounded by $\left|\left(\omega_z^{DB}\right)_B(t)\right| < r_{max}$, there exists a constant $c \in \left]0, \sqrt{k_a}\right[$ such that the matrix $\overline{\mathbf{Q}}$*

$$\overline{\mathbf{Q}}(t) = \begin{bmatrix} c \cdot k_a \cdot \mathbf{I} & \frac{c}{2}\left(k_w \cdot \mathbf{I} - \mathbf{H}^T(t)\right) \\ \frac{c}{2}\left(k_w \cdot \mathbf{I} - \mathbf{H}(t)\right) & (k_w - c) \cdot \mathbf{I} \end{bmatrix}$$

is positive definite.

The corresponding proof is found in Lemma D.11 (Roll/Pitch Positive Definite $\mathbf{Q}$) in Appendix D. From Lemmas 5.12 (Roll/Pitch Attitude Lyapunov function) and 5.14 (Roll/Pitch Positive Definite $\overline{\mathbf{Q}}$), it follows that there exists a constant $c \in \left]0, \sqrt{k_a}\right[$ such that the matrix $\overline{\mathbf{Q}}(t)$ is positive definite for all $t \geq 0$ and simultaneously the Lyapunov candidate (5.25) is positive definite. Consequently, the constant $\overline{q}$ in (5.26) is positive and $\dot{\overline{V}} \leq 0$. From Theorem 4.8 (Uniform Stability) in [87], it follows that the desired equilibrium $([0,0,1]^T, \mathbf{0})$ is uniformly stable and that the errors $\varphi_d\left(\left(\vec{\mathbf{z}}_d\right)_B(t)\right)$ and $\boldsymbol{\omega}_{xy}^{DB}(t)$ are bounded. Using Theorem 8.4 (Invariance-like Theorem) in [87], it further follows that $\left|\left|\overline{\boldsymbol{\zeta}}(t)\right|\right| \to 0$ for $t \to \infty$.

In order to satisfy Condition 1 in Theorem 5.2, it remains to be shown that $\overline{\boldsymbol{\zeta}}(t) = \mathbf{0}$ only holds for the equilibrium points $\overline{\boldsymbol{\chi}}_e = \left(\left(\vec{\mathbf{z}}_{d,e}\right)_B, \left(\boldsymbol{\omega}_{e,xy}^{DB}\right)_B\right) = (\pm[0,0,1]^T, \mathbf{0})$. Given that $\left(e_{a,z}^{DB}\right)_B(t) = 0$ (5.23) and from the magnitude of $\vec{\mathbf{e}}_a^{DB}(t)$ (5.15) it follows that $\overline{\boldsymbol{\zeta}}(t) = \mathbf{0}$ if and only if

$$\begin{aligned} \left|\left|\left(\mathbf{e}_{a,xy}^{DB}\right)_B(t)\right|\right| = \left|\left|\left(\vec{\mathbf{e}}_a^{DB}\right)_B(t)\right|\right| = \left|\sin\left(\varphi_d\left(\left(\vec{\mathbf{z}}_d\right)_B(t)\right)\right)\right| = 0, \\ \left(\boldsymbol{\omega}_{xy}^{DB}\right)_B = \mathbf{0}. \end{aligned}$$

This corresponds to the set of equilibria $\overline{\boldsymbol{\chi}}_e = \left(\left(\vec{\mathbf{z}}_{d,e}\right)_B, \boldsymbol{\omega}_{e,xy}^{DB}\right) = (\pm[0,0,1]^T, \mathbf{0})$. Conditions 2 and 3 in Theorem 5.2 are shown through linearization and are summarized in the following lemma.

Lemma 5.15 (Linearization: Roll/Pitch Reduced Attitude) *The reduced attitude closed-loop system (5.22) has a stable equilibrium at $\overline{\boldsymbol{\chi}}_{d,e} = \left(\left(\vec{\mathbf{z}}_{d,e}\right)_B, \left(\boldsymbol{\omega}_{e,xy}^{DB}\right)_B\right) = ([0,0,1]^T, \mathbf{0})$ and an unstable equilibrium at $\overline{\boldsymbol{\chi}}_{u,e} = \left(\left(\vec{\mathbf{z}}_{d,e}\right)_B, \left(\boldsymbol{\omega}_{e,xy}^{DB}\right)_B\right) = ([0,0,-1]^T, \mathbf{0})$ if k_a and k_w are positive constants. The subscript e refers to equilibrium points.*

The proof is found in Lemma D.12 (Linearization: Roll/Pitch Reduced Attitude) in Appendix D. Using Theorem 5.2 (Almost Global Asymptotic Stability) it is concluded that the desired equilibrium $\overline{\boldsymbol{\chi}}_{d,e} = ([0,0,1]^T, \mathbf{0})$ is almost globally asymptotically stable and this finally proves Theorem 5.11 (Roll/Pitch Reduced Attitude AGAS). ■

From Lemma 5.15 (Linearization: Roll/Pitch Reduced Attitude) it is known that linearized system at the equilibrium $\overline{\boldsymbol{\chi}}_{d,e} = ([0,0,1]^T, \mathbf{0})$ has only eigenvalues with negative real parts. Then from Theorem 4.13 (Lyapunov's Indirect Method) in [87] it follows that the equilibrium is locally exponentially stable. The set of initial states for which the exponential decay holds is refined in the following theorem.

Theorem 5.16 (Roll/Pitch Local Exponential Stability) *Given the desired closed loop dynamics* (5.22)*, the desired equilibrium* $\overline{\boldsymbol{\chi}}_{d,e} = \left(\left(\vec{\mathbf{z}}_{d,e}\right)_B, \left(\boldsymbol{\omega}_{e,xy}^{DB}\right)_B\right) = \left([0,0,1]^T, \mathbf{0}\right)$ *is locally exponentially stable and the Lyapunov function* (5.25) *decays exponentially given the initial conditions*

$$V_\varphi\left(\varphi_d(t=0)\right) \leq \overline{V}_\varphi,$$
$$\left|\left|\left(\boldsymbol{\omega}_{xy}^{DB}\right)_B(t=0)\right|\right| \leq \sqrt{2k_a \cdot \left(\overline{V}_\varphi - V_\varphi\left(\varphi_d(t=0)\right)\right)},$$

for a constant $\overline{V}_\varphi \in \mathbb{R}$ *such that* $0 < \overline{V}_\varphi < 2$*.*

The proof of the theorem is analogous to the one of Theorem 5.9 (Local Exponential Stability) and is therefore omitted here. Note that given a zero velocity initial condition $\left|\left|\left(\boldsymbol{\omega}_{xy}^{DB}\right)_B\right|\right| = 0$, an exponentially decaying attitude Lyapunov function can be found as long as the initial attitude error angle is $|\varphi_d(t=0)| < \pi \Rightarrow V_\varphi\left(\varphi_d(t=0)\right) < 2$.. Furthermore, the smaller the initial attitude error angle $\varphi_d(t=0)$ the larger the permitted initial error velocity $\left(\boldsymbol{\omega}_{xy}^{DB}\right)_B(t=0)$ can be.

In this and the previous sections, the stability properties of the desired closed-loop systems (5.13) and (5.22) have been analyzed showing that the desired equilibria are almost globally asymptotically stable (AGAS) in Theorems 5.4 and 5.11. The difference between them is that the system (5.22) considers the yaw rate $\left(\omega_z^{DB}\right)_B(t)$ as an external disturbance. In this way, the degraded control authority case has been studied. In summary, it can be concluded that the reduced attitude vector $\vec{\mathbf{z}}_B$ can be controlled even in the case that the yaw rate cannot. The main condition is that the yaw rate error remains bounded $|(\omega_z^{DB})_B(t)| \leq r_{max}$. The following sections derive the attitude reference model, the attitude baseline controller and the fault tolerant adaptive control augmentations.

5.3.3 Attitude Reference Model

The attitude reference model is a dynamic system that represents the desired system's response to the given commands. The inputs are the commanded direction of the body-fixed z-axis given in the N-frame $\left(\vec{\mathbf{z}}_C\right)_N(t) : \mathbb{R}^+ \to \mathbb{S}^2$ and the commanded body-fixed yaw rotational rate $\left(w_{c,z}\right)_B(t) : \mathbb{R}^+ \to \mathbb{R}$. Both signals are assumed to be bounded and piecewise-continuous in t. For each of the control signals, two independent reference models are designed and then combined together to generate the feed-forward terms for the tracking controller. The yaw rate reference model corresponds to a first order linear low-pass filter and the tilt reference model corresponds to a nonlinear second order stable filter. A sketch of the reference model and baseline controller is depicted in Figure 5.6.

Yaw Rate Reference Model

Given the commanded body-fixed yaw rotational rate $\left(w_{c,z}\right)_B(t)$, the yaw rate reference model is given by the following stable first order low-pass filter

$$\left(\dot{w}_{d,z}\right)_B^B(t) = k_{wz,rm}\left(\left(w_{c,z}\right)_B(t) - \left(w_{d,z}\right)_B(t)\right), \tag{5.27}$$

with a positive gain $k_{wz,rm} > 0 \in \mathbb{R}$ which corresponds to the inverse of the time constant. The desired body-fixed yaw rotational rate $\left(w_{d,z}\right)_B(t) \in \mathbb{R}$ is the state of the reference model.

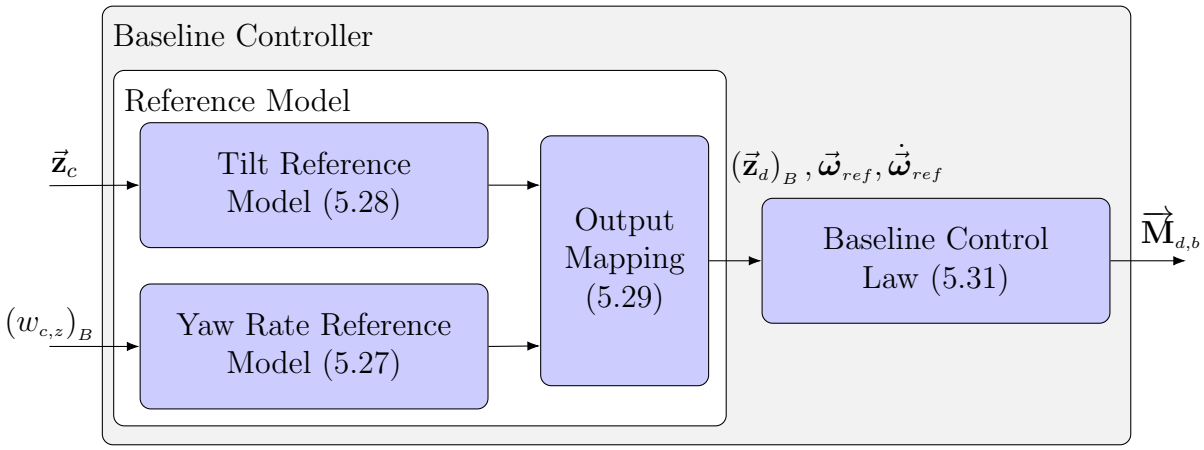


Figure 5.6: Baseline Attitude Controller Structure

Tilt Reference Model

Motivated by the previous stability analysis of the reduced attitude error dynamics (5.13), the following reference model is chosen

$$\left(\dot{\vec{\mathbf{z}}}_d\right)_N^N(t) = -\left(\vec{\boldsymbol{\omega}}^{DN}\right)_N(t) \times \left(\vec{\mathbf{z}}_d\right)_N(t), \tag{5.28a}$$

$$\left(\dot{\vec{\boldsymbol{\omega}}}^{DN}\right)_N^N(t) = -k_{a,rm}\left(\left(\vec{\mathbf{z}}_d\right)_N(t) \times \left(\vec{\mathbf{z}}_c\right)_N(t)\right) - \mathbf{K}_{\omega,rm}\left(\vec{\boldsymbol{\omega}}^{DN}\right)_N(t). \tag{5.28b}$$

The states of the reference model are given by $\left(\left(\vec{\mathbf{z}}_d\right)_N(t), \left(\vec{\boldsymbol{\omega}}^{DN}\right)_N(t)\right) \in (\mathbb{S}^2 \times \mathbb{R}^3)$. The vector $\left(\vec{\mathbf{z}}_d\right)_N(t)$ is the desired body-fixed z-axis and the rate $\left(\vec{\boldsymbol{\omega}}^{DN}\right)_N(t) = -\left(\vec{\boldsymbol{\omega}}^{ND}\right)_N(t)$ is the negative of the rotational rate corresponding to the vector $\left(\vec{\mathbf{z}}_d\right)_N$ with respect to the N-frame. Both vectors are expressed in the N-frame. The gains are chosen such that $k_{a,rm} > 0 \in \mathbb{R}$ is positive and $\mathbf{K}_{\omega,rm} \in \mathbb{R}^{3\times3}$ is diagonal positive definite. The first term in (5.28b) provides the rotation in order to align the reference model z-axis $\vec{\mathbf{z}}_d(t)$ with the commanded one $\vec{\mathbf{z}}_c(t)$. The second term is a damping term. The following corollary describes the stability properties of the reference model.

Corollary 5.17 (Reference Model Stability) *Given the system dynamics* (5.28)*, a positive proportional gain* $k_{a,rm} > 0$*, a diagonal positive definite derivative gain* $\mathbf{K}_{\omega,rm} > 0$ *and a constant command* $\vec{\mathbf{z}}_c(t) = \vec{\mathbf{z}}_{c,e}$*, it holds that:*

- *The set of equilibrium points is given by* $(\vec{\mathbf{z}}_{d,e}, \vec{\boldsymbol{\omega}}_e^{DN}) = (\pm\vec{\mathbf{z}}_{c,e}, \mathbf{0})$*. The subscript* e *refers to the values at the equilibrium of the system.*
- *The desired equilibrium* $(\vec{\mathbf{z}}_{d,e}, \vec{\boldsymbol{\omega}}_e^{DN}) = (\vec{\mathbf{z}}_{c,e}, \mathbf{0})$ *is almost globally asymptotically stable and locally exponentially stable.*
- *The undesired equilibrium* $(\vec{\mathbf{z}}_{d,e}, \vec{\boldsymbol{\omega}}_e^{DN}) = (-\vec{\mathbf{z}}_{c,e}, \mathbf{0})$ *is unstable.*

By comparing the reference model (5.28) to the closed-loop system (5.13) analyzed last section, the statements in Corollary 5.17 follow from Theorems 5.3 (Attitude Equilibrium Points), 5.4 (Reduced Attitude AGAS) and 5.9 (Local Exponential Stability).

Reference Model Output Mapping

In order to use only roll and pitch accelerations for tracking the desired reduced attitude $\vec{\mathbf{z}}_d$, the corresponding yaw rate and accelerations in (5.28b) are not used in the feed-forward path of the controller. Instead the yaw reference model (5.27) is used. This leads to the following reference signals in the body-fixed frame B which are computed from the states of the reference models.

$$\left(\vec{\mathbf{z}}_d\right)_B(t) = \mathbf{M}_{BN}(t)\left(\vec{\mathbf{z}}_d\right)_N(t) \tag{5.29a}$$

$$\left(\vec{\boldsymbol{\omega}}_{ref}\right)_B(t) = \begin{bmatrix}1&0&0\\0&1&0\\0&0&0\end{bmatrix}\mathbf{M}_{BN}(t)\left(\vec{\boldsymbol{\omega}}^{ND}\right)_N(t) + \begin{pmatrix}0\\0\\\left(w_{d,z}\right)_B(t)\end{pmatrix} \tag{5.29b}$$

$$\left(\dot{\vec{\boldsymbol{\omega}}}_{ref}\right)_B^B(t) = \begin{bmatrix}1&0&0\\0&1&0\\0&0&0\end{bmatrix}\left(\dot{\vec{\boldsymbol{\omega}}}^{ND}\right)_B^B(t) + \begin{pmatrix}0\\0\\\left(\dot{w}_{d,z}\right)_B^B(t)\end{pmatrix} \tag{5.29c}$$

The rate acceleration $\left(\dot{\vec{\boldsymbol{\omega}}}^{ND}\right)_B^B(t)$ can be related to the reference model dynamics (5.28b) via the subsequent kinematic relation. Using the derivative $\dot{\mathbf{M}}_{BN}(t)$ (5.12) and Lemma D.1 (Cross Product) it follows that

$$\begin{aligned}
\left(\vec{\boldsymbol{\omega}}^{ND}\right)_B(t) &= -\mathbf{M}_{BN}(t)\left(\vec{\boldsymbol{\omega}}^{DN}\right)_N(t),\\
\left(\dot{\vec{\boldsymbol{\omega}}}^{ND}\right)_B^B(t) &= -\dot{\mathbf{M}}_{BN}(t)\left(\vec{\boldsymbol{\omega}}^{DN}\right)_N(t) - \mathbf{M}_{BN}(t)\left(\dot{\vec{\boldsymbol{\omega}}}^{DN}\right)_N^N(t),\\
&= \mathbf{M}_{BN}(t)\,\boldsymbol{\Omega}\left(\left(\vec{\boldsymbol{\omega}}^{NB}\right)_N(t)\right)\left(\vec{\boldsymbol{\omega}}^{DN}\right)_N(t) - \mathbf{M}_{BN}(t)\left(\dot{\vec{\boldsymbol{\omega}}}^{DN}\right)_N^N(t),\\
&= \left(\vec{\boldsymbol{\omega}}^{NB}\right)_B(t) \times \left(\mathbf{M}_{BN}(t)\left(\vec{\boldsymbol{\omega}}^{DN}\right)_N(t)\right) - \mathbf{M}_{BN}(t)\left(\dot{\vec{\boldsymbol{\omega}}}^{DN}\right)_N^N(t).
\end{aligned}$$

The presented decoupling of the tilt reference model (5.28) from the yaw feed forward signals allows for an easier selection of the desired dynamics. This is mainly because the yaw authority on a multicopter is much smaller than the roll and pitch authority. The effect of the omitted terms is considered in the stability assessment of the closed-loop system in the next section.

5.3.4 Attitude Baseline Control Law

In this section, the design of the attitude baseline controller is presented. Based on the dynamics (5.10), the propulsion moments $\overrightarrow{\mathbf{M}}_P^R(t)$ are considered as the system inputs and it is assumed that no actuator faults or external disturbances are present. The goal of the baseline controller is to achieve tracking of the reference model (5.27)-(5.29) which corresponds to a combination of Control Task 5.2 (Reduced Attitude Tracking Control) and yaw rate tracking. The stability assessment is therefore based on Section 5.3.2. Tracking of the desired reduced attitude $\vec{\mathbf{z}}_d(t)$ is achieved if the closed-loop dynamics correspond to (5.22).

The first step is the definition of the rate tracking error as

$$\left(\vec{\mathbf{e}}_w\right)_B(t) := \left(\vec{\boldsymbol{\omega}}^{NB}\right)_B(t) - \left(\vec{\boldsymbol{\omega}}_{ref}\right)_B(t) = \begin{pmatrix}\left(w_x^{NB}\right)_B(t) - \left(w_x^{ND}\right)_B(t)\\\left(w_y^{NB}\right)_B(t) - \left(w_y^{ND}\right)_B(t)\\\left(w_z^{NB}\right)_B(t) - \left(w_{d,z}\right)_B(t)\end{pmatrix}, \tag{5.30}$$

where the subscripts x, y and z represent the first, second and third element of a given vector. Using the plant dynamics (5.10b) its derivative is given by

$$\begin{aligned}\left(\dot{\vec{\mathbf{e}}}_w\right)_B^B(t) &= \left(\dot{\vec{\boldsymbol{\omega}}}^{NB}\right)_B^B(t) - \left(\dot{\vec{\boldsymbol{\omega}}}_{ref}\right)_B^B(t)\,,\\ &= \left(\mathbf{I}^R\right)_{BB}^{-1}\left(\mathbf{f}_c\left(\left(\vec{\boldsymbol{\omega}}^{NB}\right)_B(t)\right) + \left(\overrightarrow{\mathbf{M}}_P^R\right)_B(t)\right) - \left(\dot{\vec{\boldsymbol{\omega}}}_{ref}\right)_B^B(t)\,.\end{aligned}$$

The feed-forward term $\left(\dot{\vec{\boldsymbol{\omega}}}_{ref}\right)_B^B(t)$ corresponds to the reference model (5.29). By comparison with the dynamics of the error system (5.22b), the following baseline control law $\left(\overrightarrow{\mathbf{M}}_{d,b}\right)_B(t)$ is selected

$$\begin{aligned}\left(\overrightarrow{\mathbf{M}}_{d,b}\right)_B(t) = &-\mathbf{f}_c\left(\left(\vec{\boldsymbol{\omega}}^{NB}\right)_B(t)\right)\\ &+\left(\mathbf{I}^R\right)_{BB}\left(-k_a\left(\vec{\mathbf{e}}_a^{DB}\right)_B(t) - \mathbf{K}_\omega\left(\vec{\mathbf{e}}_w\right)_B(t) + \left(\dot{\vec{\boldsymbol{\omega}}}_{ref}\right)_B^B(t)\right)\end{aligned} \tag{5.31}$$

Here, $k_a > 0 \in \mathbb{R}$ is the spring constant and $\mathbf{K}_\omega = diag([k_w, k_w, k_{w,z}]) \in \mathbb{R}^{3\times 3}$ is the damping constant with $k_w, k_{w,z} > 0 \in \mathbb{R}$. The derivative of the rate tracking error is then given by

$$\left(\dot{\vec{\mathbf{e}}}_w\right)_B^B(t) = -k_a\left(\vec{\mathbf{e}}_a^{DB}\right)_B(t) - \mathbf{K}_\omega\left(\vec{\mathbf{e}}_w\right)_B(t)\,. \tag{5.32}$$

The closed-loop error dynamics follow from the error kinematics (5.22a) and the error dynamics (5.32) as

$$\left(\dot{\vec{\mathbf{z}}}_d\right)_B^B(t) = -\left(\vec{\boldsymbol{\omega}}^{DB}\right)_B(t) \times \left(\vec{\mathbf{z}}_d\right)_B(t)\,, \tag{5.33a}$$

$$\left(\dot{\boldsymbol{\omega}}_{xy}^{DB}\right)_B^B(t) = -k_a\left(\mathbf{e}_{a,xy}^{DB}\right)_B(t) - k_w\left(\boldsymbol{\omega}_{xy}^{DB}\right)_B(t)\,, \tag{5.33b}$$

$$\left(\dot{e}_{w,z}\right)_B^B(t) = -k_{w,z}\left(e_{w,z}\right)_B(t)\,, \tag{5.33c}$$

where (5.33a)-(5.33b) are the reduced-attitude subsystem and (5.33c) is the independently controlled yaw rate subsystem. Here, the facts that $\left(e_{a,z}^{DB}\right)_B(t) = 0$ (5.23) and $\left(\boldsymbol{\omega}_{xy}^{DB}\right)_B(t) = \left(\mathbf{e}_{w,xy}\right)_B(t)$ (5.30) have been used. The vectors $\left(\boldsymbol{\omega}_{xy}^{DB}\right)_B(t) \in \mathbb{R}^2$ and $\left(\mathbf{e}_{w,xy}\right)_B(t) \in \mathbb{R}^2$ correspond to the first and second elements of $\left(\vec{\boldsymbol{\omega}}^{DB}\right)_B(t)$ and $\left(\vec{\mathbf{e}}_w\right)_B(t)$ respectively. The error $\left(e_{w,z}\right)_B(t) \in \mathbb{R}$ corresponds to the third element of $\left(\vec{\mathbf{e}}_w\right)_B(t)$. Note that for the reduced attitude subsystem (5.33a)-(5.33b) $\left(\omega_z^{DB}\right)_B(t)$ is regarded as an external signal and that $\left(\omega_z^{DB}\right)_B(t) \neq \left(e_{w,z}\right)_B(t)$. This is a result from the independent tilt and yaw rate reference models. The following corollary describes the stability properties of the closed-loop attitude subsystem (5.33).

Corollary 5.18 (Baseline Controller) *Given the system dynamics* (5.10)*, the reference model* (5.27)-(5.29)*, the control law* (5.31)*, positive gains* $k_{wz,rm}, k_{a,rm}, k_a, k_w, k_{w,z} > 0$*, a diagonal positive definite matrix* $\mathbf{K}_{\omega,rm}$ *and bounded commands* $\vec{\mathbf{z}}_c(t), \left(w_{c,z}\right)_B(t)$*, it holds that:*

- *The set of equilibrium points is given by* $\left(\left(\vec{\mathbf{z}}_{d,e}\right)_B, \vec{\mathbf{e}}_{w,e}\right) = \left(\pm[0,0,1]^T, \mathbf{0}\right)$*. The subscript* e *refers to the values at the equilibrium of the system.*
- *The desired equilibrium* $\left(\left(\vec{\mathbf{z}}_{d,e}\right)_B, \vec{\mathbf{e}}_{w,e}\right) = \left([0,0,1]^T, \mathbf{0}\right)$ *is almost globally asymptotically stable and locally exponentially stable.*

- *The undesired equilibrium* $\left(\left(\vec{\mathbf{z}}_{d,e}\right)_B, \vec{\mathbf{e}}_{w,e}\right) = \left([0,0,-1]^T, \mathbf{0}\right)$ *is unstable.*

Proof: From the boundedness of the commands $\vec{\mathbf{z}}_c(t), \left(w_{c,z}\right)_B(t)$ it follows that the reference model states $\left(w_{d,z}\right)_B(t)$, $\vec{\mathbf{z}}_d(t)$, $\vec{\boldsymbol{\omega}}^{DN}(t)$ and the reference model outputs $\vec{\boldsymbol{\omega}}_{ref}(t)$ are bounded. Furthermore, from the independent linear yaw error dynamics (5.33c), it directly follows that $\left(e_{w,z}\right)_B(t) = \left(w_z^{NB}\right)_B(t) - \left(w_{d,z}\right)_B(t) = 0$ is an exponentially stable equilibrium. Therefore, $\left(\omega_z^{NB}\right)_B(t)$ is uniformly bounded and $\left(\omega_z^{NB}\right)_B(t) \to \left(w_{d,z}\right)_B(t)$ for $t \to \infty$.

Given the last result, the signal $\left(\omega_z^{DB}\right)_B(t) = \left(\omega_z^{NB}\right)_B(t) - \left(\omega_z^{ND}\right)_B(t)$ is also uniformly bounded. Then, by comparing the closed-loop reduced attitude tracking subsystem (5.33a)-(5.33b) to the system (5.22) analyzed in Section 5.3.2, the statements in Corollary 5.18 follow from Theorems 5.10 (Roll/Pitch Attitude Equilibrium Points), 5.11 (Roll/Pitch Reduced Attitude AGAS) and 5.16 (Roll/Pitch Local Exponential Stability). ■

5.4 Attitude Adaptive Augmentation

In this section, the design of the attitude adaptive augmentation is presented. The final structure of the P-ACA attitude and rate tracking controller is depicted in Figure 5.7. Compared to the controller in the last section, actuator faults and external disturbances are considered as presented in the actuator model (2.16) or equivalently (5.1). For this controller the control input is the vector of propellers' squared angular velocities $\mathbf{u}(t)$ as defined in (2.7). Hence, the attitude subsystem is derived from the attitude dynamics model (5.10) and the partitioned actuator model (5.6) as

$$\dot{\mathbf{M}}_{NB}(t) = \mathbf{M}_{NB}(t)\left(\boldsymbol{\Omega}^{NB}\right)_{BB}(t) \tag{5.34a}$$

$$\left(\dot{\vec{\boldsymbol{\omega}}}^{NB}\right)_B^B(t) = \left(\mathbf{I}^R\right)_{BB}^{-1}\left(\mathbf{f}_c\left(\left(\vec{\boldsymbol{\omega}}^{NB}\right)_B(t)\right) + \left(\overrightarrow{\mathbf{M}}_P^R\right)_B(t) + \left(\vec{\mathbf{d}}_M\right)_B\right), \tag{5.34b}$$

$$\left(\overrightarrow{\mathbf{M}}_P^R\right)_B(t) = \mathbf{U}_M\mathbf{D}_M\boldsymbol{\Lambda}_M\mathbf{u}(t) \tag{5.34c}$$

where $\vec{\mathbf{d}}_M \in \mathbb{R}^3$ is a constant moment disturbance and $\mathbf{f}_c\left(\left(\vec{\boldsymbol{\omega}}^{NB}\right)_B(t)\right)$ is the abbreviation for the Coriolis term as defined in (5.2). The parameters $\boldsymbol{\Lambda}_M \in \mathbb{R}^{3\times 6}$ and $\vec{\mathbf{d}}_M \in \mathbb{R}^3$ of the plant are considered unknown in order to take actuator faults and failures into account.

A reduced order P-ACA approach is used to achieve the control task as in Section 4.2 with the difference that the attitude kinematics (5.34a) are inherently nonlinear. Therefore, the Lyapunov functions used in the stability assessment are adapted to achieve analogous results. The 4 design steps of P-ACA are listed in Remark 4.5 (P-ACA Design Steps). The first step is the design of a state predictor. The next step is the selection of the desired moment $\vec{\boldsymbol{\nu}}_{M,d}(t) \in \mathbb{R}^3$ such that the known predictor model replicates the reference model (5.27)-(5.29) up to an identification disturbance term. The third step, the control allocation, has been formulated in the Control Task 5.1 and solved in Section 5.2. The last step is the identification of the unknown plant parameters $\boldsymbol{\Lambda}_M$ and $\vec{\mathbf{d}}_M$ so that the control law $\mathbf{u}(t)$ is also valid for the plant dynamics (5.34). The P-ACA approach is directly applied to the plant and afterwards it is conceptually separated into the baseline controller from Section 5.4 and the adaptive augmentation.

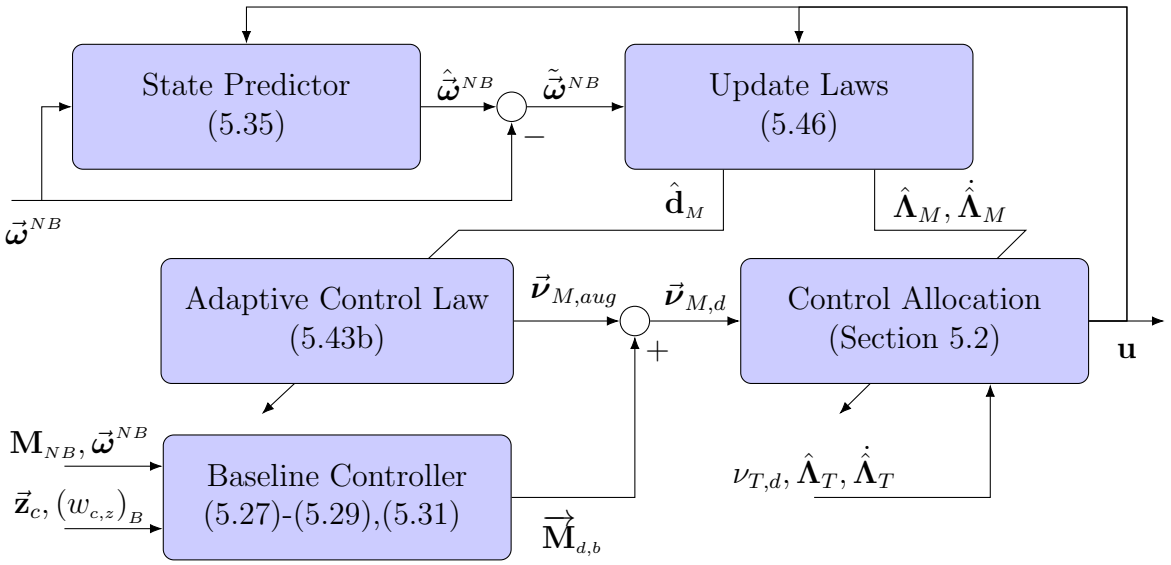


Figure 5.7: P-ACA Attitude and Rate Controller

From the equations of motion, it can be seen that the kinematics (5.34a) are known and that the uncertainties are in the rotational dynamics (5.34b) and the actuator model (5.34c). Therefore, a reduced order predictor as in (4.17) is used in order to estimate the unknown parameters. By defining $\hat{\vec{\omega}}^{NB}(t) \in \mathbb{R}^3$ as the estimate of the system angular rate $\vec{\omega}^{NB}(t)$ and $\overrightarrow{\hat{\mathbf{M}}}_P^R(t)$ as the estimate of the propulsion moments $\overrightarrow{\mathbf{M}}_P^R(t)$, the reduced-order state predictor is defined as

$$\begin{aligned}\left(\dot{\hat{\vec{\omega}}}^{NB}\right)_B(t) = &-\mathbf{L}\cdot\left(\left(\hat{\vec{\omega}}^{NB}\right)_B(t) - \left(\vec{\omega}^{NB}\right)_B(t)\right)\\ &+ \left(\mathbf{I}^R\right)^{-1}_{BB}\left(\mathbf{f}_c\left(\left(\vec{\omega}^{NB}\right)_B(t)\right) + \left(\overrightarrow{\hat{\mathbf{M}}}_P^R\right)_B(t) + \left(\hat{\vec{\mathbf{d}}}_M\right)_B(t)\right),\end{aligned} \tag{5.35a}$$

$$\left(\overrightarrow{\hat{\mathbf{M}}}_P^R\right)_B(t) = \mathbf{U}_M\mathbf{D}_M\hat{\mathbf{\Lambda}}_M(t)\cdot\mathbf{u}(t) \tag{5.35b}$$

The first term in (5.35a) is the estimation stabilizing feedback using the diagonal positive definite gain $\mathbf{L} = diag([L_{xy}, L_{xy}, L_z]) \in \mathbb{R}^{3\times3}$ and the estimation error defined as $\tilde{\vec{\omega}}^{NB}(t) := \hat{\vec{\omega}}^{NB}(t) - \vec{\omega}^{NB}(t)$. The remaining terms correspond emulate the dynamics (5.34b)-(5.34c) given the estimate $\hat{\mathbf{\Lambda}}_M(t) \in \mathbb{R}^{3\times6}$ of the control effectiveness matrix $\mathbf{\Lambda}_M$ and the estimate $\hat{\vec{\mathbf{d}}}_M(t) \in \mathbb{R}^3$ of the disturbance $\vec{\mathbf{d}}_M$. The estimation errors are defined as

$$\tilde{\mathbf{\Lambda}}_M(t) := \hat{\mathbf{\Lambda}}_M(t) - \mathbf{\Lambda}_M, \qquad \tilde{\vec{\mathbf{d}}}_M(t) := \hat{\vec{\mathbf{d}}}_M(t) - \vec{\mathbf{d}}_M.$$

In the following the control law for the known reduced order predictor (5.35) is presented

5.4.1 Adaptive Control Law

The control of the predictor model is separated into control allocation and rigid body control as presented in Sections 4.1 and 4.2. Given the control allocation as presented

in Section 5.2 with (5.7), the control input $\mathbf{u}(t)$ is computed such that

$$\vec{\boldsymbol{\nu}}_{M,d}(t) = \mathbf{U}_M \mathbf{D}_M \hat{\boldsymbol{\Lambda}}_M(t) \cdot \mathbf{u}(t) \tag{5.36}$$

for a given desired moment $\vec{\boldsymbol{\nu}}_{M,d}(t) \in \mathbb{R}^3$ and a known and time varying control effectiveness matrix $\hat{\boldsymbol{\Lambda}}_M(t)$. Therefore, in this section the design of the desired moment $\vec{\boldsymbol{\nu}}_{M,d}(t)$ as the rigid body control law is addressed. Using (5.36) and using the virtual control $\vec{\boldsymbol{\nu}}_{M,d}(t)$ as an input, the predictor (5.35) can be rewritten as

$$\left(\dot{\hat{\vec{\boldsymbol{\omega}}}}^{NB}\right)_B^B(t) = -\mathbf{L}\left(\tilde{\vec{\boldsymbol{\omega}}}^{NB}\right)_B(t) + (\mathbf{I}^R)_{BB}^{-1}\left(\mathbf{f}_c\left(\left(\vec{\boldsymbol{\omega}}^{NB}\right)_B(t)\right) + \vec{\boldsymbol{\nu}}_{M,d}(t) + \left(\hat{\vec{\mathbf{d}}}_M\right)_B(t)\right).$$

For stability analysis, the reduced-order predictor is extended by the attitude kinematics to generate a full state predictor as in (4.19). By using the kinematics (5.34a) and the estimation error definition

$$\tilde{\vec{\boldsymbol{\omega}}}^{NB}(t) := \hat{\vec{\boldsymbol{\omega}}}^{NB}(t) - \vec{\boldsymbol{\omega}}^{NB}(t), \tag{5.37}$$

it follows that

$$\dot{\mathbf{M}}_{NB}(t) = \mathbf{M}_{NB}(t)\left(\boldsymbol{\Omega}\left(\left(\hat{\vec{\boldsymbol{\omega}}}^{NB}\right)_B(t) - \left(\tilde{\vec{\boldsymbol{\omega}}}^{NB}\right)_B(t)\right)\right) \tag{5.38a}$$

$$\left(\dot{\hat{\vec{\boldsymbol{\omega}}}}^{NB}\right)_B^B(t) = (\mathbf{I}^R)_{BB}^{-1}\left(\mathbf{f}_c\left(\left(\vec{\boldsymbol{\omega}}^{NB}\right)_B(t)\right) + \vec{\boldsymbol{\nu}}_{M,d}(t) + \left(\hat{\vec{\mathbf{d}}}_M\right)_B(t)\right) - \mathbf{L}\left(\tilde{\vec{\boldsymbol{\omega}}}^{NB}\right)_B(t), \tag{5.38b}$$

The goal of the adaptive augmentation controller is to achieve asymptotic tracking of the reference model (5.27)-(5.29) . Therefore, the error dynamics are derived in a similar way as in Section 5.3.4 (Attitude Baseline Control Law). The error states are given by

$$\left(\vec{\mathbf{z}}_d\right)_B(t) = \mathbf{M}_{BN}(t) \cdot \left(\vec{\mathbf{z}}_d\right)_N(t), \tag{5.39a}$$

$$\left(\hat{\vec{\mathbf{e}}}_w\right)_B(t) = \left(\hat{\vec{\boldsymbol{\omega}}}^{NB}\right)_B(t) - \left(\vec{\boldsymbol{\omega}}_{ref}\right)_B(t). \tag{5.39b}$$

The attitude error state is given by the desired vector $\vec{\mathbf{z}}_d$ given in the B-frame. The rate error state is the difference between the estimated rotational rate of the B-frame $\hat{\vec{\boldsymbol{\omega}}}^{NB}(t)$ and the rotational rate of the reference vector $\left(\vec{\boldsymbol{\omega}}_{ref}\right)_B(t)$ given in the B-frame. This simultaneously corresponds to the estimate of $\left(\vec{\mathbf{e}}_w\right)_B(t)$. In a similar way the tracking error with respect to the tilt reference model (5.28) is defined as

$$\left(\hat{\vec{\boldsymbol{\omega}}}^{DB}\right)_B(t) = \left(\hat{\vec{\boldsymbol{\omega}}}^{NB}\right)_B(t) - \left(\vec{\boldsymbol{\omega}}^{ND}\right)_B(t).$$

Since $\left(\boldsymbol{\omega}_{xy}^{ND}\right)_B(t) = \left(\boldsymbol{\omega}_{ref,xy}\right)_B(t)$ (5.30), it follows that

$$\left(\hat{\boldsymbol{\omega}}_{xy}^{DB}\right)_B(t) = \left(\hat{\mathbf{e}}_{w,xy}\right)_B(t), \tag{5.40}$$

where $\left(\hat{\boldsymbol{\omega}}_{xy}^{DB}\right)_B(t), \left(\hat{\mathbf{e}}_{w,xy}\right)_B(t) \in \mathbb{R}^2$ correspond to the first two elements of $\left(\hat{\vec{\boldsymbol{\omega}}}^{DB}\right)_B(t)$ and $\left(\hat{\vec{\mathbf{e}}}_w\right)_B(t)$ respectively. The kinematics of the desired vector (5.33a) can be rewritten as

$$\left(\dot{\vec{\mathbf{z}}}_d\right)_B^B(t) = -\left(\hat{\vec{\boldsymbol{\omega}}}^{DB}\right)_B(t) \times \left(\vec{\mathbf{z}}_d\right)_B(t) + \left(\tilde{\vec{\boldsymbol{\omega}}}^{NB}\right)_B(t) \times \left(\vec{\mathbf{z}}_d\right)_B(t),$$

using the relation

$$\vec{\boldsymbol{\omega}}^{DB}(t) = \vec{\boldsymbol{\omega}}^{NB}(t) - \vec{\boldsymbol{\omega}}^{ND}(t) = \hat{\vec{\boldsymbol{\omega}}}^{NB}(t) - \tilde{\vec{\boldsymbol{\omega}}}^{NB}(t) - \vec{\boldsymbol{\omega}}^{ND}(t) = \hat{\vec{\boldsymbol{\omega}}}^{DB}(t) - \tilde{\vec{\boldsymbol{\omega}}}^{NB}(t).$$

Using the predictor dynamics (5.38b), the error dynamics are given by

$$\left(\dot{\vec{\mathbf{z}}}_d\right)_B^B(t) = -\left(\hat{\vec{\boldsymbol{\omega}}}^{DB}\right)_B(t) \times \left(\vec{\mathbf{z}}_d\right)_B(t) - \boldsymbol{\Omega}\left(\left(\vec{\mathbf{z}}_d\right)_B(t)\right)\left(\tilde{\vec{\boldsymbol{\omega}}}^{NB}\right)_B(t), \tag{5.41a}$$

$$\begin{aligned}\left(\dot{\hat{\vec{\mathbf{e}}}}_w\right)_B^B(t) = {}& \left(\mathbf{I}^R\right)_{BB}^{-1}\left(\mathbf{f}_c\left(\left(\vec{\boldsymbol{\omega}}^{NB}\right)_B(t)\right) + \vec{\boldsymbol{\nu}}_{M,d}(t) + \left(\hat{\vec{\mathbf{d}}}_M\right)_B(t)\right) \\ & - \mathbf{L}\left(\tilde{\vec{\boldsymbol{\omega}}}^{NB}\right)_B(t) - \left(\dot{\vec{\boldsymbol{\omega}}}_{ref}\right)_B^B(t).\end{aligned} \tag{5.41b}$$

In order to design the adaptive control law $\vec{\boldsymbol{\nu}}_{M,d}(t)$, the desired closed-loop error dynamics are selected analogously to (4.20) derived from (5.33) as

$$\left(\dot{\vec{\mathbf{z}}}_d\right)_B^B(t) = -\left(\hat{\vec{\boldsymbol{\omega}}}^{DB}\right)_B(t) \times \left(\vec{\mathbf{z}}_d\right)_B(t) - \boldsymbol{\Omega}\left(\left(\vec{\mathbf{z}}_d\right)_B(t)\right)\left(\tilde{\vec{\boldsymbol{\omega}}}^{NB}\right)_B(t), \tag{5.42a}$$

$$\left(\dot{\hat{\boldsymbol{\omega}}}_{xy}^{DB}\right)_B^B(t) = -k_a\left(\mathbf{e}_{a,xy}^{DB}\right)_B(t) - k_w\left(\hat{\boldsymbol{\omega}}_{xy}^{DB}\right)_B(t) + R_{xy}\left(\tilde{\boldsymbol{\omega}}_{xy}^{NB}\right)_B(t), \tag{5.42b}$$

$$\left(\dot{\hat{e}}_{w,z}\right)_B^B(t) = -k_{w,z}\left(\hat{e}_{w,z}\right)_B(t) + R_z\left(\tilde{\omega}_z^{NB}\right)_B(t). \tag{5.42c}$$

Here, $\left(\hat{e}_{w,z}\right)_B(t) \in \mathbb{R}$ corresponds to the third element of $\left(\hat{\vec{\mathbf{e}}}_w\right)_B(t)$. As in Section 4.2, they correspond to desired tracking error dynamics (5.33) plus estimation error terms. The first one is $-\boldsymbol{\Omega}\left(\left(\vec{\mathbf{z}}_d\right)_B(t)\right)\left(\tilde{\vec{\boldsymbol{\omega}}}^{NB}\right)_B(t)$ in (5.42a) and arises from the use of the reduced order predictor. The other two terms in (5.42b) and (5.42c) correspond to $\mathbf{R}\left(\tilde{\vec{\boldsymbol{\omega}}}^{DB}\right)_B(t)$ with the matrix $\mathbf{R} = diag([R_{xy}, R_{xy}, R_z])$. This is a design degree of freedom and its presence does not change the stability proof. By comparing the desired closed-loop error dynamics (5.42) with the predictor tracking error dynamics (5.41), the desired virtual control law is selected as

$$\begin{aligned}\vec{\boldsymbol{\nu}}_{M,d}(t) = {}& -\left(\hat{\vec{\mathbf{d}}}_M\right)_B(t) - \mathbf{f}_c\left(\left(\vec{\boldsymbol{\omega}}^{NB}\right)_B(t)\right) + \left(\mathbf{I}^R\right)_{BB}\left(\mathbf{L}\left(\tilde{\vec{\boldsymbol{\omega}}}^{NB}\right)_B(t) + \left(\dot{\vec{\boldsymbol{\omega}}}_{ref}\right)_B^B(t)\right) \\ & + \left(\mathbf{I}^R\right)_{BB}\left(-k_a\left(\vec{\mathbf{e}}_a^{DB}\right)_B(t) - \mathbf{K}_\omega\left(\hat{\vec{\mathbf{e}}}_w\right)_B(t) + \mathbf{R}\left(\tilde{\vec{\boldsymbol{\omega}}}^{NB}\right)_B(t)\right),\end{aligned}$$

with the matrix $\mathbf{K}_\omega = diag([k_w, k_w, k_{w,z}]) \in \mathbb{R}^{3\times 3}$. The complete control law has a term canceling the estimated disturbance moment, a term canceling the Coriolis term, a term canceling the estimator dynamics, a feedforward term, proportional and derivative feedback terms and finally an additional term corresponding to the rate estimation error $\left(\tilde{\vec{\boldsymbol{\omega}}}^{NB}\right)_B(t)$. There are several possible choices of the gain $\mathbf{R}$. The choice $\mathbf{R} = \mathbf{0}$ leads to closed-loop rate error dynamics (5.41b) without disturbance term. The choice $\mathbf{R} = -\mathbf{L}$ is interesting since it cancels out the estimator term $\mathbf{L}\left(\tilde{\vec{\boldsymbol{\omega}}}^{NB}\right)_B(t)$. This feature is desired since the usually fast estimator dynamics may lead to high-frequency control inputs. Here, the matrix $\mathbf{R}$ is chosen as $\mathbf{R} = \mathbf{K}_\omega - \mathbf{L}$ such that the control law is given

by

$$\vec{\boldsymbol{\nu}}_{M,d}(t) = \vec{\boldsymbol{\nu}}_{M,aug}(t) + \left(\overrightarrow{\mathbf{M}}_{d,b}\right)_B(t), \tag{5.43a}$$

$$\vec{\boldsymbol{\nu}}_{M,aug}(t) = -\left(\hat{\vec{\mathbf{d}}}_M\right)_B(t), \tag{5.43b}$$

$$\begin{aligned}\left(\overrightarrow{\mathbf{M}}_{d,b}\right)_B(t) = &-\mathbf{f}_c\left(\left(\vec{\boldsymbol{\omega}}^{NB}\right)_B(t)\right)\\ &+ \left(\mathbf{I}^R\right)_{BB}\left(-k_a\left(\vec{\mathbf{e}}_a^{DB}\right)_B(t) - \mathbf{K}_\omega\left(\vec{\mathbf{e}}_w\right)_B(t) + \left(\dot{\vec{\boldsymbol{\omega}}}_{ref}\right)_B^B(t)\right).\end{aligned} \tag{5.43c}$$

From the definitions of $\tilde{\vec{\boldsymbol{\omega}}}^{NB}(t)$ (5.37) and of $\hat{\vec{\mathbf{e}}}_w(t)$ (5.39b), the relation

$$\hat{\vec{\mathbf{e}}}_w - \tilde{\vec{\boldsymbol{\omega}}}^{NB} = \hat{\vec{\boldsymbol{\omega}}}^{NB}(t) - \vec{\boldsymbol{\omega}}_{ref}(t) - \tilde{\vec{\boldsymbol{\omega}}}^{NB} = \vec{\boldsymbol{\omega}}^{NB}(t) - \vec{\boldsymbol{\omega}}_{ref}(t) = \vec{\mathbf{e}}_w$$

has been derived and used. The control law (5.43) is decoupled from the high frequency content in the estimation error $\tilde{\vec{\boldsymbol{\omega}}}^{NB}(t)$ and in the estimate $\hat{\vec{\boldsymbol{\omega}}}^{NB}(t)$ and constitutes a simple augmentation of the baseline control law $\overrightarrow{\mathbf{M}}_{d,b}(t)$ (5.31). The adaptive augmentation term $\vec{\boldsymbol{\nu}}_{M,aug}(t) \in \mathbb{R}^3$ is in this case a cancellation of the estimated disturbance. Note that the adaptation does not only affect the control inputs $\mathbf{u}(t)$ through the virtual control law $\vec{\boldsymbol{\nu}}_{M,aug}(t)$ (5.43b), but also through the solution of the control allocation (5.36) and hence through the estimate $\hat{\boldsymbol{\Lambda}}_M(t)$. This is one of the special characteristics of the P-ACA applied to overactuated systems.

Theorem 5.19 (Full State Predictor - Boundedness) *Given the full state predictor dynamics* (5.38)*, the reference model* (5.27)-(5.29)*, the control law* (5.43)*, positive gains* $k_{wz,rm}$*,* $k_{a,rm}$*,* k_a*,* k_w*,* $k_{w,z} > 0$*, diagonal positive definite matrices* $\mathbf{K}_{\omega,rm}, \mathbf{L}$ *and bounded commands* $\vec{\mathbf{z}}_c(t), (w_{c,z})_B(t)$*, it holds that:*

- *Given a bounded input* $\tilde{\vec{\boldsymbol{\omega}}}^{NB}(t)$ *(estimation error), the error state* $\left(\left(\vec{\mathbf{z}}_d\right)_B(t), \hat{\vec{\mathbf{e}}}_w(t)\right)$ *is bounded.*
- *The trajectories approach the equilibrium set* $\left(\left(\vec{\mathbf{z}}_d\right)_B(t), \hat{\vec{\mathbf{e}}}_w(t)\right) \to (\pm[0,0,1]^T, \mathbf{0})$ *for* $\left|\left|\tilde{\vec{\boldsymbol{\omega}}}^{NB}\right|\right| \to 0$ *.*

Proof: The proof goes along the lines of the reduced-order P-ACA presented in Section 4.2 and the baseline control law from Section 5.3.4. From the boundedness of the commands $\vec{\mathbf{z}}_c(t), (w_{c,z})_B(t)$ it follows that the reference model states $(w_{d,z})_B(t)$, $\vec{\mathbf{z}}_d(t)$, $\vec{\boldsymbol{\omega}}^{DN}(t)$ and the reference model outputs $\vec{\boldsymbol{\omega}}_{ref}(t)$ are bounded.

As for the baseline controller, first the independent yaw subsystem (5.42c) is analyzed. Since it is a linear system as in, it directly follows that for a positive $k_{w,z} > 0$, the system (5.42c) is Input-to-State Stable (ISS) with respect to the input $(\tilde{\omega}_z^{NB})_B(t)$ [87, Definition 4.7]. This implies that for a bounded $(\tilde{\omega}_z^{NB})_B(t)$ the error $(\hat{e}_{w,z})_B(t)$ is bounded and that the error $(\hat{e}_{w,z})_B(t) \to 0$ for $(\tilde{\omega}_z^{NB})_B(t) \to 0$. From the boundedness of the error $(\hat{e}_{w,z})_B(t) = (\hat{\omega}_z^{NB})_B(t) - (\omega_{ref,z})_B(t)$ (5.39b) it follows that the estimate $(\hat{\omega}_z^{NB})_B(t)$ bounded. From the error definition $\tilde{\vec{\boldsymbol{\omega}}}^{NB}(t) = \hat{\vec{\boldsymbol{\omega}}}^{NB}(t) - \vec{\boldsymbol{\omega}}^{NB}(t)$ (5.37), it follows that the rate $(\omega_z^{NB})_B(t)$ is bounded too.

In the next step, the reduced attitude subsystem (5.42a)-(5.42b) is analyzed. The attitude error function (5.17) is used to formulate the attitude Lyapunov candidate function analogously to (5.25)

$$\begin{aligned}\overline{V}((\vec{\mathbf{z}}_d)_B(t), \hat{\vec{\boldsymbol{\omega}}}_{xy}^{DB}(t)) = k_a \cdot V_\varphi(\varphi_d((\vec{\mathbf{z}}_d)_B(t))) + \frac{1}{2}\left(\hat{\boldsymbol{\omega}}_{xy}^{DB}\right)_B^T(t)\left(\hat{\boldsymbol{\omega}}_{xy}^{DB}\right)_B(t) \\ + c \cdot \left(\hat{\boldsymbol{\omega}}_{xy}^{DB}\right)_B^T(t)\left(\mathbf{e}_{a,xy}^{DB}\right)_B(t),\end{aligned} \tag{5.44}$$

which is positive definite for $c \leq \sqrt{k_a}$ as described in Lemma 5.12 (Roll/Pitch Attitude Lyapunov function). For the sake of readability, the time dependency is not written in the following. Using the derivative of the attitude penalty function $V_\varphi(t)$ (5.19) and $\left(e_{a,z}^{DB}\right)_B(t) = 0$ (5.23), the derivative of the Lyapunov function $\dot{V}(t)$ can be written as

$$\begin{aligned}\dot{\overline{V}} &= k_a \cdot \dot{V}_\varphi + \left(\hat{\boldsymbol{\omega}}_{xy}^{DB}\right)_B^T\left(\dot{\hat{\boldsymbol{\omega}}}_{xy}^{DB}\right)_B^B + c \cdot \left(\mathbf{e}_{a,xy}^{DB}\right)_B^T\left(\dot{\hat{\boldsymbol{\omega}}}_{xy}^{DB}\right)_B^B + c \cdot \left(\hat{\boldsymbol{\omega}}_{xy}^{DB}\right)_B^T\left(\dot{\mathbf{e}}_{a,xy}^{DB}\right)_B^B, \\ &= k_a \cdot \left(\mathbf{e}_{a,xy}^{DB}\right)_B^T\left(\boldsymbol{\omega}_{xy}^{DB}\right)_B + c \cdot \left(\hat{\boldsymbol{\omega}}_{xy}^{DB}\right)_B^T\left(\dot{\mathbf{e}}_{a,xy}^{DB}\right)_B^B + \left(\left(\hat{\boldsymbol{\omega}}_{xy}^{DB}\right)_B + c \cdot \left(\mathbf{e}_{a,xy}^{DB}\right)_B\right)^T\left(\dot{\hat{\boldsymbol{\omega}}}_{xy}^{DB}\right)_B^B,\end{aligned}$$

Since the third element of $\left(\vec{\mathbf{e}}_a^{DB}\right)_B(t)$ is always zero, the fact that $\left(\vec{\mathbf{e}}_a^{DB}\right)_B^T(t)\left(\vec{\boldsymbol{\omega}}^{DB}\right)_B(t) = \left(\mathbf{e}_{a,xy}^{DB}\right)_B^T(t)\left(\boldsymbol{\omega}_{xy}^{DB}\right)_B(t)$ has been used. Using the closed-loop predictor dynamics (5.42a)-(5.42b), the Lemma 5.13 (Roll/Pitch Attitude Error Derivative), the relation $\vec{\boldsymbol{\omega}}^{DB} = \hat{\vec{\boldsymbol{\omega}}}^{DB} - \tilde{\vec{\boldsymbol{\omega}}}^{DB}$ (5.37), and the bound $|(z_{d,z})_B| \leq 1$, the derivative of the Lyapunov function $\dot{\overline{V}}(t)$ can be bounded from above by

$$\begin{aligned}\dot{\overline{V}} &= k_a \cdot \left(\mathbf{e}_{a,xy}^{DB}\right)_B^T\left(\hat{\boldsymbol{\omega}}_{xy}^{DB}\right)_B - k_a \cdot \left(\mathbf{e}_{a,xy}^{DB}\right)_B^T\left(\tilde{\boldsymbol{\omega}}_{xy}^{DB}\right)_B \\ &\quad + c \cdot \left(\hat{\boldsymbol{\omega}}_{xy}^{DB}\right)_B^T\left(\mathbf{H}\left(\vec{\mathbf{e}}_{a,xy}^{DB}\right)_B + \left(\left(\hat{\boldsymbol{\omega}}_{xy}^{DB}\right)_B - \left(\tilde{\boldsymbol{\omega}}_{xy}^{DB}\right)_B\right)(z_{d,z})_B\right) \\ &\quad + \left(\left(\hat{\boldsymbol{\omega}}_{xy}^{DB}\right)_B + c \cdot \left(\mathbf{e}_{a,xy}^{DB}\right)_B\right)^T\left(-k_a\left(\mathbf{e}_{a,xy}^{DB}\right)_B - k_w\left(\hat{\boldsymbol{\omega}}_{xy}^{DB}\right)_B + R_{xy}\left(\tilde{\boldsymbol{\omega}}_{xy}^{NB}\right)_B\right), \\ &\leq -c \cdot k_a \cdot \left|\left|\left(\vec{\mathbf{e}}_{a,xy}^{DB}\right)_B\right|\right|^2 - k_a \cdot \left(\mathbf{e}_{a,xy}^{DB}\right)_B^T\left(\tilde{\boldsymbol{\omega}}_{xy}^{DB}\right)_B + (c - k_w)\left(\hat{\boldsymbol{\omega}}_{xy}^{DB}\right)_B^T\left(\hat{\boldsymbol{\omega}}_{xy}^{DB}\right)_B \\ &\quad + c \cdot \left(\vec{\mathbf{e}}_{a,xy}^{DB}\right)_B^T\left(\mathbf{H}^T - k_w \cdot \mathbf{I}\right)\left(\hat{\boldsymbol{\omega}}_{xy}^{DB}\right)_B \\ &\quad - c \cdot \left(\hat{\boldsymbol{\omega}}_{xy}^{DB}\right)_B^T\left(\tilde{\boldsymbol{\omega}}_{xy}^{DB}\right)_B(z_{d,z})_B + \left(\left(\hat{\boldsymbol{\omega}}_{xy}^{DB}\right)_B + c \cdot \left(\mathbf{e}_{a,xy}^{DB}\right)_B\right)^T R_{xy}\left(\tilde{\boldsymbol{\omega}}_{xy}^{NB}\right)_B.\end{aligned}$$

By defining the error vector $\hat{\boldsymbol{\zeta}}(t) \in \mathbb{R}^4$, the symmetric matrix $\overline{\mathbf{Q}}(t) \in \mathbb{R}^{4\times4}$ and the estimation error input matrix $\mathbf{B}_e(t) \in \mathbb{R}^{4\times2}$ as

$$\hat{\boldsymbol{\zeta}} = \begin{pmatrix}\left(\mathbf{e}_{a,xy}^{DB}\right)_B(t) \\ \left(\hat{\boldsymbol{\omega}}_{xy}^{DB}\right)_B(t)\end{pmatrix}, \qquad \overline{\mathbf{Q}}(t) = \begin{bmatrix} c \cdot k_a \cdot \mathbf{I} & \frac{c}{2}\left(k_w \cdot \mathbf{I} - \mathbf{H}^T(t)\right) \\ \frac{c}{2}\left(k_w \cdot \mathbf{I} - \mathbf{H}(t)\right) & (k_w - c) \cdot \mathbf{I}\end{bmatrix},$$
$$\mathbf{B}_e(t) = \begin{bmatrix}(-k_a + c \cdot R_{xy}) \cdot \mathbf{I} \\ \left(-c \cdot (z_{d,z})_B(t) + R_{xy}\right) \cdot \mathbf{I}\end{bmatrix},$$

the derivative of the Lyapunov candidate (5.44) can be rewritten as

$$\dot{\overline{V}}(t) \leq -\hat{\boldsymbol{\zeta}}^T(t) \cdot \overline{\mathbf{Q}}(t) \cdot \hat{\boldsymbol{\zeta}}(t) + \hat{\boldsymbol{\zeta}}^T(t) \cdot \mathbf{B}_e(t) \cdot \left(\tilde{\vec{\boldsymbol{\omega}}}^{NB}\right)_B(t), \tag{5.45a}$$
$$\leq -\overline{q}\left|\left|\hat{\boldsymbol{\zeta}}(t)\right|\right|^2 + \overline{b}\left|\left|\hat{\boldsymbol{\zeta}}(t)\right|\right|\left|\left|\tilde{\boldsymbol{\omega}}_{xy}^{NB}(t)\right|\right|, \tag{5.45b}$$

The constant $\overline{q} \in \mathbb{R}$ is positive if the matrix $\overline{\mathbf{Q}}(t)$ is positive definite for all $t \geq 0$. Since from the yaw motion analysis it is known that $\left(\omega_z^{NB}\right)_B(t)$ is bounded, the constant c can

be chosen as in Lemma 5.14 (Roll/Pitch Positive Definite $\overline{\mathbf{Q}}$) such that the matrix $\overline{\mathbf{Q}}(t)$ is positive definite for all $t \geq 0$ and simultaneously the Lyapunov candidate (5.44) is positive definite. Furthermore, a finite constant $\bar{b} \geq 0 \in \mathbb{R}$ can be chosen to satisfy $\bar{b} > ||\mathbf{B}_e(t)||$ uniformly in t. This follows from the fact that $|(z_{d,z})_B(t)| \leq 1$ for all t. Therefore, it holds that

$$\dot{\overline{V}}(t) < 0, \qquad \text{for } \left|\left|\hat{\boldsymbol{\zeta}}(t)\right|\right| > \frac{\bar{b}}{\overline{q}} \left|\left|\left(\tilde{\boldsymbol{\omega}}_{xy}^{NB}\right)_B(t)\right|\right|.$$

Based on (5.45), the derivative of the Lyapunov function can be linearly bounded by the estimation error $\left(\tilde{\boldsymbol{\omega}}_{xy}^{NB}\right)_B(t)$. For linear dynamics, this leads to Input-to-State Stability (ISS) as in Section 4.2 and Theorem 5.19 (Full State Predictor - Boundedness) would have been proven. This is not directly applicable since in the case of reduced attitude control the state predictor system has two equilibria. The rest of the proof is based on the argumentation lines of Theorem 4.18 (Ultimate Bound) in [87] and is found in Theorem D.14 (Full State Predictor - Boundedness) in Appendix D. ■

5.4.2 Update Law

The next step of the reduced order P-ACA approach (Remark 4.5) is the design of the update laws for the unknown parameters. The update laws of the estimates $\hat{\boldsymbol{\Lambda}}_M(t)$ and $\hat{\vec{\mathbf{d}}}_M(t)$ follow the design methodology presented in Section 4.2 and are therefore given by

$$\left(\dot{\hat{\vec{\mathbf{d}}}}_M\right)_B^B(t) = -\boldsymbol{\Gamma}_d \cdot (\mathbf{I}^R)_{BB}^{-1} \cdot \left(\tilde{\boldsymbol{\omega}}^{NB}\right)_B(t), \tag{5.46a}$$

$$\dot{\hat{\boldsymbol{\Lambda}}}_M(t) = -\mathbf{D}_M \mathbf{U}_M^T (\mathbf{I}^R)_{BB}^{-1} \left(\tilde{\boldsymbol{\omega}}^{NB}\right)_B(t) \cdot \mathbf{u}^T(t) \cdot \boldsymbol{\Gamma}_\Lambda. \tag{5.46b}$$

Noting that the estimated parameters are constant, the parameter estimation error dynamics are given by $\left(\dot{\tilde{\vec{\mathbf{d}}}}_M\right)_B^B(t) = \left(\dot{\hat{\vec{\mathbf{d}}}}_M\right)_B^B(t)$ and $\dot{\tilde{\boldsymbol{\Lambda}}}_M(t) = \dot{\hat{\boldsymbol{\Lambda}}}_M(t)$. The state estimation error dynamics are computed by subtracting the plant dynamics (5.34b) from the predictor dynamics (5.35). Hence, the predictor error dynamics are given by

$$\left(\dot{\tilde{\boldsymbol{\omega}}}^{NB}\right)_B^B(t) = -\mathbf{L}\left(\tilde{\boldsymbol{\omega}}^{NB}\right)_B(t) + (\mathbf{I}^R)_{BB}^{-1}\mathbf{U}_M\mathbf{D}_M\tilde{\boldsymbol{\Lambda}}_M(t)\mathbf{u}(t) + (\mathbf{I}^R)_{BB}^{-1}\left(\tilde{\vec{\mathbf{d}}}_M\right)_B(t), \tag{5.47a}$$

$$\left(\dot{\tilde{\vec{\mathbf{d}}}}_M\right)_B^B(t) = -\boldsymbol{\Gamma}_d \cdot (\mathbf{I}^R)_{BB}^{-1} \cdot \left(\tilde{\boldsymbol{\omega}}^{NB}\right)_B(t), \tag{5.47b}$$

$$\dot{\tilde{\boldsymbol{\Lambda}}}_M(t) = -\mathbf{D}_M \mathbf{U}_M^T (\mathbf{I}^R)_{BB}^{-1} \left(\tilde{\boldsymbol{\omega}}^{NB}\right)_B(t) \cdot \mathbf{u}^T(t) \cdot \boldsymbol{\Gamma}_\Lambda, \tag{5.47c}$$

with symmetric positive definite gains $\boldsymbol{\Gamma}_d \in \mathbb{R}^{3\times3}$ and $\boldsymbol{\Gamma}_\Lambda \in \mathbb{R}^{6\times6}$. The underlined terms show that the estimated error dynamics are coupled between each other. These error dynamics lead to a stable equilibrium of the estimation errors and to an asymptotically vanishing tracking error between the plant (5.34) and the reference model (5.27)-(5.29). The following theorem summarizes the main result of this section.

Theorem 5.20 (Estimator Update Law) *Given the system dynamics* (5.34)*, the reduced order state predictor* (5.35)*, the reference model* (5.27)-(5.29)*, the control law* (5.43)*, the update laws* (5.46)*, positive gains* $k_{wz,rm}$*,* $k_{a,rm}$*,* k_a*,* k_w*,* $k_{w,z} > 0$*, diagonal positive definite matrices* $\mathbf{K}_{\omega,rm}, \mathbf{L}$*, symmetric positive definite adaptation gains* $\mathbf{\Gamma}_d, \mathbf{\Gamma}_\Lambda$ *and bounded commands* $\vec{\mathbf{z}}_c(t), (w_{c,z})_B(t)$*, it holds that:*

- *The equilibrium* $(\tilde{\vec{\boldsymbol{\omega}}}^{NB}, \tilde{\vec{\mathbf{d}}}_M, \widetilde{\mathbf{\Lambda}}_M) = (\mathbf{0}, \mathbf{0}, \mathbf{0})$ *is uniformly stable.*
- *The estimation error* $\left|\left|\tilde{\vec{\boldsymbol{\omega}}}^{NB}(t)\right|\right| \to 0$ *for* $t \to \infty$*.*
- *And the plant tracking errors approach the set of equilibrium points* $((\vec{\mathbf{z}}_d)_B, \vec{\mathbf{e}}_w) \to (\pm[0,0,1]^T, \mathbf{0})$ *for* $t \to \infty$*.*

Proof: In the following the time dependency is not written out for readability. Analogously to Section 4.2, in order to show the first two statements in Theorem 5.20, the Lyapunov candidate function is chosen as

$$V = \frac{1}{2}\left(\tilde{\vec{\boldsymbol{\omega}}}^{NB}\right)^T \tilde{\vec{\boldsymbol{\omega}}}^{NB} + \frac{1}{2}\left(\tilde{\vec{\mathbf{d}}}_M\right)_B^T \mathbf{\Gamma}_d^{-1}\left(\tilde{\vec{\mathbf{d}}}_M\right)_B + \frac{1}{2} tr\left(\widetilde{\mathbf{\Lambda}}_M \mathbf{\Gamma}_\Lambda^{-1} \widetilde{\mathbf{\Lambda}}_M^T\right). \tag{5.48}$$

From Lemma B.7 (Positive Definite Trace 1) it follows that the Lyapunov function is positive definite in the case of symmetric positive definite matrices $\mathbf{\Gamma}_d \in \mathbb{R}^{3\times3}$ and $\mathbf{\Gamma}_\Lambda \in \mathbb{R}^{6\times6}$, which correspond to the adaptation rates. The derivative of the Lyapunov function is calculated using the predictor estimation error dynamics (5.47) and Lemma B.2 (Trace and Scalar Product) as

$$\begin{aligned}
\dot{V} &= \left(\tilde{\vec{\boldsymbol{\omega}}}^{NB}\right)_B^T \cdot \left(\dot{\tilde{\vec{\boldsymbol{\omega}}}}^{NB}\right)_B^B + \left(\tilde{\vec{\mathbf{d}}}_M\right)_B^T \mathbf{\Gamma}_d^{-1}\left(\dot{\tilde{\vec{\mathbf{d}}}}_M\right)_B^B + tr\left(\widetilde{\mathbf{\Lambda}}_M \mathbf{\Gamma}_\Lambda^{-1} \dot{\widetilde{\mathbf{\Lambda}}}_M^T\right), \\
&= -\left(\tilde{\vec{\boldsymbol{\omega}}}^{NB}\right)_B^T \cdot \mathbf{L} \cdot \left(\tilde{\vec{\boldsymbol{\omega}}}^{NB}\right)_B + \left(\tilde{\vec{\boldsymbol{\omega}}}^{NB}\right)_B^T (\mathbf{I}^R)_{BB}^{-1}\left(\mathbf{U}_M \mathbf{D}_M \widetilde{\mathbf{\Lambda}}_M \mathbf{u} + \left(\tilde{\vec{\mathbf{d}}}_M\right)_B\right) \\
&\quad - \left(\tilde{\vec{\mathbf{d}}}_M\right)_B^T \cdot (\mathbf{I}^R)_{BB}^{-1} \cdot \left(\tilde{\vec{\boldsymbol{\omega}}}^{NB}\right)_B - tr\left(\widetilde{\mathbf{\Lambda}}_M \mathbf{u} \cdot \left(\tilde{\vec{\boldsymbol{\omega}}}^{NB}\right)_B^T (\mathbf{I}^R)_{BB}^{-1} \mathbf{U}_M \mathbf{D}_M\right), \\
&= -\left(\tilde{\vec{\boldsymbol{\omega}}}^{NB}\right)_B^T \cdot \mathbf{L} \cdot \left(\tilde{\vec{\boldsymbol{\omega}}}^{NB}\right)_B + tr\left(\widetilde{\mathbf{\Lambda}}_M \mathbf{u} \cdot \left(\tilde{\vec{\boldsymbol{\omega}}}^{NB}\right)_B^T (\mathbf{I}^R)_{BB}^{-1} \mathbf{U}_M \mathbf{D}_M\right) \\
&\quad - tr\left(\widetilde{\mathbf{\Lambda}}_M \mathbf{u} \cdot \left(\tilde{\vec{\boldsymbol{\omega}}}^{NB}\right)_B^T (\mathbf{I}^R)_{BB}^{-1} \mathbf{U}_M \mathbf{D}_M\right), \\
&= -\left(\tilde{\vec{\boldsymbol{\omega}}}^{NB}\right)_B^T \cdot \mathbf{L} \cdot \left(\tilde{\vec{\boldsymbol{\omega}}}^{NB}\right)_B.
\end{aligned}$$

Hence, the derivative of the Lyapunov function $\dot{V}(t) = -\left(\tilde{\vec{\boldsymbol{\omega}}}^{NB}\right)_B^T(t) \cdot \mathbf{L} \cdot \left(\tilde{\vec{\boldsymbol{\omega}}}^{NB}\right)_B(t)$ is negative semi-definite. From Theorem 4.8 (Uniform Stability) in [87], the equilibrium $(\tilde{\vec{\boldsymbol{\omega}}}^{NB}, \tilde{\vec{\mathbf{d}}}_M, \widetilde{\mathbf{\Lambda}}_M) = (\mathbf{0}, \mathbf{0}, \mathbf{0})$ is globally uniformly stable and $\tilde{\vec{\boldsymbol{\omega}}}^{NB}(t)$, $\tilde{\vec{\mathbf{d}}}_M(t)$, $\widetilde{\mathbf{\Lambda}}_M(t) \in \mathcal{L}_\infty$. Therefore, $\hat{\vec{\mathbf{d}}}_M(t)$ and $\hat{\mathbf{\Lambda}}_M(t)$ are also bounded.

Subsequently, the goal is to show asymptotic convergence of the estimation error $\tilde{\vec{\boldsymbol{\omega}}}^{NB}(t)$ to zero based on Barbalat's Lemma (Lemma B.9). Following the same arguments as in Section 3.2, it is known that $\lim_{t\to\infty} \int_{\tau=0}^{t} \dot{V}\left(\tilde{\vec{\boldsymbol{\omega}}}^{NB}(\tau)\right) d\tau$ exists and is finite.

It remains to be shown that $\dot{V}\left(\tilde{\vec{\omega}}^{NB}(t)\right)$ is a uniform continuous function of t. This holds if its derivative

$$\ddot{V}(t) = -2 \cdot \left(\tilde{\vec{\omega}}^{NB}\right)_B^T(t) \cdot \mathbf{L} \cdot \left(\dot{\tilde{\vec{\omega}}}^{NB}\right)_B^B(t)$$

is uniformly bounded. From the state estimation error dynamics (5.47a), it follows that $\ddot{V}(t)$ is uniformly bounded if $\tilde{\vec{\omega}}^{NB}(t)$, $\tilde{\vec{\mathbf{d}}}_M(t)$, $\tilde{\mathbf{\Lambda}}_M(t)$ and $\mathbf{u}(t)$ are uniformly bounded. Therefore, it remains to be shown that $\mathbf{u}(t)$ is uniformly bounded. Given that $\tilde{\vec{\omega}}^{NB}(t) \in \mathcal{L}_\infty$, it follows from Theorem 5.19 (Full State Predictor - Boundedness) that the error state $((\vec{\mathbf{z}}_d)_B(t), \hat{\vec{\mathbf{e}}}_w)$ is bounded. From the virtual control law (5.43) and the solution of the control allocation (5.36) it follows that the virtual control $\vec{\boldsymbol{\nu}}_{M,d}(t)$ (5.43) and the control input $\mathbf{u}(t)$ are uniformly bounded. Finally, it is concluded that $\ddot{V}(t)$ is bounded uniformly in t for all $t \geq 0$ and $\dot{V}\left(\tilde{\vec{\omega}}^{NB}(t)\right)$ is uniformly continuous. Using Lemma B.9 (Barbalat) it follows that $\dot{V}\left(\tilde{\vec{\omega}}^{NB}(t)\right) \to 0$ and hence $\tilde{\vec{\omega}}^{NB}(t) \to \mathbf{0}$ for $t \to \infty$. From the second statement in Theorem 5.19 (Full State Predictor - Boundedness) it follows that the predictor tracking error converges to the set of equilibria $((\vec{\mathbf{z}}_d)_B(t), \hat{\vec{\mathbf{e}}}_w(t)) \to (\pm[0,0,1]^T, \mathbf{0})$ and therefore the tracking error converges to the set of equilibria $((\vec{\mathbf{z}}_d)_B(t), \vec{\mathbf{e}}_w(t)) \to (\pm[0,0,1]^T, \mathbf{0})$ for $t \to \infty$. ■

An overview of the P-ACA attitude controller is depicted in Figure 5.7 and includes all the implemented equations: The reference model (5.27)-(5.29), the baseline control law (5.31), the state predictor (5.35), the adaptive control law (5.43), the update laws (5.46) and the control allocation as in Section 5.2.

5.5 Vertical Velocity Control

After having derived the control allocation in Section 5.2 and the attitude and rate controller in Sections 5.3 and 5.4, this section presents the vertical velocity controller. The main purpose of the controller is to track a given vertical velocity command $(v_{c,z})_N^N(t) \in \mathbb{R}$ using the total thrust $T(t) \in \mathbb{R}$ as a control input. The vertical velocity $(v_z)_N^N(t) \in \mathbb{R}$ corresponds to the third element of the vector $(\vec{\mathbf{v}}^R)_N^N$ and therefore the dynamics are described by the rigid body dynamics (5.1b) and the actuator model (5.6) as

$$(\dot{v}_z)_N^{NN}(t) = g - \frac{1}{m}(z_{B,z})_N(t) \cdot (T(t) + d_T), \tag{5.49a}$$

$$T(t) = U_T \cdot D_T \cdot \mathbf{\Lambda}_T \cdot \mathbf{u}(t). \tag{5.49b}$$

Here, $g \in \mathbb{R}$ is the gravitational constant and $m \in \mathbb{R}$ is the aircraft mass as in (2.3). Furthermore, $(z_{B,z})_N(t) \in \mathbb{R}$ is the third element of the vector $(z_B)_N(t)$. In order to address faults within the actuation system, the P-ACA approach presented in Section 4.1 is used.

Following the same ideas from Sections 5.3 and 5.4, a baseline controller is initially derived in order to highlight the changes due to the adaptive augmentation and to facilitate the selection of a lean, compact and unified controller structure (thesis objective 6). The baseline controller is composed of a reference model and a baseline control law presented in Sections 5.5.1 and 5.5.2 respectively. The adaptive augmentation is

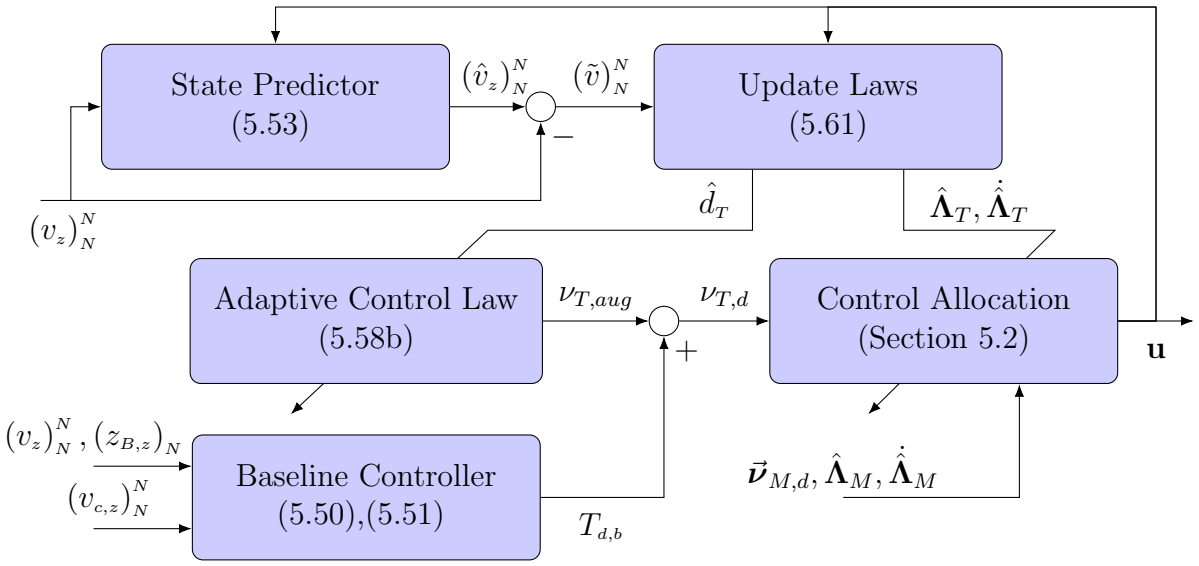


Figure 5.8: P-ACA Vertical Velocity Controller

derived in Section 5.5.3. The final structure of the P-ACA vertical velocity tracking controller is depicted in Figure 5.8. The structure is the same as for the attitude tracking controller.

5.5.1 Vertical Velocity Reference Model

The vertical velocity reference model is a dynamic system that represents the desired system's response to the given commands. The vertical velocity command $(v_{c,z})_N^N(t) \in \mathbb{R}$ comes from the pilot. It is assumed to be bounded and piecewise-continuous in t. The reference model is given by the following stable first order low-pass filter

$$(\dot{v}_{d,z})_N^{NN}(t) = k_{vz,rm} \cdot \left((v_{c,z})_N^N(t) - (v_{d,z})_N^N(t)\right), \tag{5.50}$$

with a positive gain $k_{vz,rm} > 0 \in \mathbb{R}$ corresponding to the inverse of the time constant. The desired vertical velocity $(v_{d,z})_N^N(t) \in \mathbb{R}$ is the state of the reference model. The following lemma describes the stability properties of the reference model.

Lemma 5.21 (Reference Model Stability) *Given the system dynamics* (5.50) *and a positive gain* $k_{vz,rm} > 0$*, the reference model corresponds to a stable first order low-pass filter, which is a bounded-input-bounded-output system [101].* ■

5.5.2 Vertical Velocity Baseline Control Law

Given the system dynamics (5.49a) and assuming no disturbance $d_T = 0$, the goal of the baseline control law is to track the desired vertical velocity $(v_{d,z})_N^N(t)$ from the reference model (5.50) using $T(t)$ as the control input. The control law is given by

$$T_{d,b}(t) = \frac{m}{\max\left(z_{min}, (z_{B,z})_N(t)\right)}\left(g - (\dot{v}_{d,z})_N^{NN}(t) + k_{v,z} \cdot \left((v_z)_N^N(t) - (v_{d,z})_N^N(t)\right)\right), \tag{5.51}$$

where $k_{v,z} > 0 \in \mathbb{R}$ is a positive proportional gain. The limit $z_{min} > 0 \in \mathbb{R}$ is the minimum operational value of $(z_{B,z})_N(t)$ and a protection of excessive thrust. This limit is selected using the following assumption.

Assumption 5.22 (Maximum Tilt Limit) *The operational envelope is limited by the maximum tilt angle $\varphi(t) \in \mathbb{R}$ between the body-fixed z-axis $\vec{\mathbf{z}}_B(t)$ and the N-frame z-axis $\vec{\mathbf{z}}_N$. The maximum angle is given by $\varphi_{max} < \frac{\pi}{2}$ such that*

$$(z_{B,z})_N(t) = (\vec{\mathbf{z}}_B)_N^T(t)\,(\vec{\mathbf{z}}_N)_N = \cos(\varphi(t)) \geq \cos(\varphi_{max}) = z_{min} > 0.$$

Note that $(z_{B,z})_N(t) = 0$ means that the total thrust $T(t)$ is perpendicular to the z-axis of the N coordinate system. In this case the vertical velocity cannot be influenced by the total thrust $T(t)$ regardless of the selected control law.

The tracking error is defined as $(e_{v,z})_N^N(t) := (v_z)_N^N(t) - (v_{d,z})_N^N(t)$. If Assumption 5.22 holds, closed-loop error dynamics are computed from the system dynamics (5.49a) with no disturbance ($d_T = 0$) and the baseline control law (5.51) as

$$\begin{aligned}(\dot{e}_{v,z})_N^{NN} &= (\dot{v}_z)_N^{NN} - (\dot{v}_{d,z})_N^{NN}, \\ &= -k_{v,z} \cdot (e_{v,z})_N^N(t).\end{aligned} \tag{5.52}$$

The following Lemma describes the stability properties of the closed-loop system.

Lemma 5.23 (Baseline Controller Stability) *Given the system dynamics (5.49a) with no disturbance ($d_T = 0$) and the control input $T(t)$, the reference model (5.50), the baseline control law (5.51), positive gains $k_{v,z}$, $k_{vz,rm} > 0$ and a bounded command $(v_{c,z})_N^N(t)$, the closed-loop error dynamics are linear and the zero equilibrium is globally exponentially stable if Assumption 5.22 (Maximum Tilt Limit) holds.*

The Lemma is proven from the linearity of the closed-loop error dynamics (5.52) and its negative eigenvalue $-k_{v,z}$ [101]. ■

5.5.3 Vertical Velocity Adaptive Augmentation

The vertical velocity adaptive augmentation is designed following the P-ACA approach from Section 4.1. Compared to the controller in the last section, actuator faults and external disturbances are considered as presented in (5.49). The 4 design steps of P-ACA are listed in Remark 4.5 (P-ACA Design Steps). The first step is the design of a state predictor. The next step is the selection of the desired virtual control $\nu_{T,d}(t)$ such that for $\hat{T}(t) = \nu_{T,d}(t)$ the known predictor model replicates the reference model (5.50) up to an identification disturbance term. The third step, the control allocation, has been formulated in the Control Task 5.1 and solved in Section 5.2. The last step is the identification of the unknown plant parameters $\mathbf{\Lambda}_L$ and d_T so that the control law $\mathbf{u}(t)$ is also valid for the plant dynamics (5.49).

Because the vertical velocity but not the altitude is controlled, the use of a reduced order predictor is not necessary and a predictor as in (4.7) can be used to estimate the unknown parameters. By defining $(\hat{v}_z)_N^N(t) \in \mathbb{R}$ as the estimate of the vertical velocity

$(v_z)_N^N(t)$ and $\hat{T}(t) \in \mathbb{R}$ as the estimate of the total thrust $T(t)$, the state predictor is defined as

$$\left(\dot{\hat{v}}_z\right)_N^{NN}(t) = -L_{v,z}\left((\hat{v}_z)_N^N(t) - (v_z)_N^N(t)\right) + g - \frac{1}{m}\left(z_{B,z}\right)_N(t)\left(\hat{T}(t) + \hat{d}_T(t)\right), \quad (5.53a)$$

$$\hat{T}(t) = U_T D_T \hat{\mathbf{\Lambda}}_T \cdot \mathbf{u}(t). \quad (5.53b)$$

The first term in (5.53a) is the estimation stabilizing feedback using the positive gain $L_{v,z} \in \mathbb{R}$ and the estimation error defined as

$$(\tilde{v}_z)_N^N(t) = (\hat{v}_z)_N^N(t) - (v_z)_N^N(t). \quad (5.54)$$

The remaining terms correspond to the plant dynamics (5.49) given the estimate $\hat{\mathbf{\Lambda}}_T(t) \in \mathbb{R}^{1\times 6}$ of the control effectiveness matrix $\mathbf{\Lambda}_T$ and the estimate $\hat{d}_T(t) \in \mathbb{R}$ of the disturbance d_T. The estimation errors are defined as

$$\widetilde{\mathbf{\Lambda}}_T(t) = \hat{\mathbf{\Lambda}}_T(t) - \mathbf{\Lambda}_T, \qquad \tilde{d}_T(t) = \hat{d}_T(t) - d_T.$$

Adaptive Control Law

The control of the predictor model is separated into control allocation and rigid body control as presented in Section 4.1. Given the control allocation as presented in Section 5.2 with (5.7), a control input $\mathbf{u}(t)$ is known such that

$$\nu_{T,d}(t) = U_T D_T \hat{\mathbf{\Lambda}}_T(t) \cdot \mathbf{u}(t) \quad (5.55)$$

for a given desired thrust $\nu_{T,d}(t)$ and a known and time varying control effectiveness matrix $\hat{\mathbf{\Lambda}}_T(t)$. Therefore, the design of the desired thrust $\nu_{T,d}(t)$ as the rigid body control law is addressed in this section. Using (5.55) and using the virtual control $\nu_{T,d}(t)$ as an input, the predictor (5.53) can be rewritten as

$$\left(\dot{\hat{v}}_z\right)_N^{NN}(t) = -L_{v,z}(\tilde{v}_z)_N^N(t) + g - \frac{1}{m}\left(z_{B,z}\right)_N(t)\left(\nu_{T,d}(t) + \hat{d}_T(t)\right). \quad (5.56)$$

The goal of the adaptive augmentation controller is to achieve asymptotic tracking of the reference model (5.50). The tracking error of the predictor is defined as $(\hat{e}_{v,z})_N^N(t) := (\hat{v}_z)_N^N(t) - (v_{d,z})_N^N(t)$ and the error dynamics are given by

$$\left(\dot{\hat{e}}_{v,z}\right)_N^{NN}(t) = -L_{v,z}(\tilde{v}_z)_N^N(t) + g - \frac{1}{m}\left(z_{B,z}\right)_N(t)\left(\nu_{T,d}(t) + \hat{d}_T(t)\right) - (\dot{v}_{d,z})_N^{NN}(t). \quad (5.57)$$

In a similar manner as in Section 5.4 (Attitude Adaptive Augmentation), the virtual control law is selected such that it constitutes a simple augmentation of the baseline control law $T_{d,b}(t)$ (5.51) and is given by

$$\nu_{T,d}(t) = \nu_{T,aug}(t) + T_{d,b}(t), \quad (5.58a)$$

$$\nu_{T,aug} = -\hat{d}_T(t), \quad (5.58b)$$

$$T_{d,b}(t) = \frac{m}{\max\left(z_{min}, \left(z_{B,z}\right)_N(t)\right)}\left(g - (\dot{v}_{d,z})_N^{NN}(t) + k_{v,z} \cdot (e_{v,z})_N^N(t)\right). \quad (5.58c)$$

The adaptive augmentation term $\nu_{T,aug}(t) \in \mathbb{R}$ is a cancellation of the estimated disturbance force $\hat{d}_T(t)$. The baseline control law has a term canceling the gravitational potential, a feedforward term and a proportional feedback term.

Let Assumption 5.22 (Maximum Tilt Limit) hold. Then by inserting the adaptive control law (5.58) in (5.57) the closed-loop error dynamics are given by

$$\left(\dot{\hat{e}}_{v,z}\right)_N^{NN} = -L_{v,z}\left(\tilde{v}_z\right)_N^N(t) - k_{v,z}\cdot\left(e_{v,z}\right)_N^N(t) \tag{5.59a}$$

$$= -k_{v,z}\cdot\left(\hat{e}_{v,z}\right)_N^N(t) + R_{v,z}\cdot\left(\tilde{v}_z\right)_N^N(t), \tag{5.59b}$$

where $R_{v,z} = k_{v,z} - L_{v,z}$. From $\left(v_z\right)_N^N(t) = \left(\hat{v}_z\right)_N^N(t) - \left(\tilde{v}_z\right)_N^N(t)$ (5.54), the relation

$$\begin{aligned}\left(e_{v,z}\right)_N^N(t) &= \left(v_z\right)_N^N(t) - \left(v_{d,z}\right)_N^N(t)\\ &= \left(\hat{v}_z\right)_N^N(t) - \left(\tilde{v}_z\right)_N^N(t) - \left(v_{d,z}\right)_N^N(t)\\ &= \left(\hat{e}_{v,z}\right)_N^N(t) - \left(\tilde{v}_z\right)_N^N(t)\end{aligned}$$

has been derived and used in (5.59b). As in the attitude case, the estimation disturbance term in (5.59b) arises from the choice of a control law that is decoupled from the high frequency content of the estimation error $\left(\tilde{v}_z\right)_N^N(t)$ and the estimate $\left(\hat{v}_z\right)_N^N(t)$. Since in this case no reduce-order predictor is used, the constant $R_{v,z}$ could have been alternatively selected as $R_{v,z} = 0$.

The adaptation does not only affect the control inputs $\mathbf{u}(t)$ through the virtual control law $\nu_{T,aug}(t)$ (5.58b), but also through the solution of the control allocation (5.55) and hence through the estimate $\hat{\mathbf{\Lambda}}_T(t)$. This is one of the special characteristics of P-ACA applied to overactuated systems. The following theorem summarizes the characteristics of the closed-loop system needed for the subsequent stability assessment.

Theorem 5.24 (State Predictor Boundedness) *Given are the predictor dynamics* (5.56)*, the reference model* (5.50)*, the control law* (5.58)*, positive gains* $k_{v,z}, k_{vz,rm} > 0$ *and a bounded command* $\left(v_{c,z}\right)_N^N(t)$*. Given Assumption 5.22 (Maximum Tilt Limit), it holds that:*

- *Given a bounded input* $\left(\tilde{v}_z\right)_N^N(t)$ *(estimation error), the error state* $\left(\hat{e}_z\right)_N^N(t)$ *is bounded.*
- *For* $\left|\left|\left(\tilde{v}_z\right)_N^N(t)\right|\right| \to 0$ *it holds that* $\left|\left|\left(\hat{e}_z\right)_N^N(t)\right|\right| \to 0$*.*

Proof: The error dynamics (5.59b) correspond to a linear stable first order low-pass filter with the external input $\left(\tilde{v}_z\right)_N^N(t)$. The theorem follows directly by noting that the eigenvalue of the system $-k_{v,z}$ is negative [101]. Alternatively, the error dynamics (5.59) in combination with the following Lyapunov function

$$V\left(\left(\hat{e}_{v,z}\right)_N(t)\right) = \frac{1}{2}\cdot\left(\hat{e}_{v,z}\right)_N^2(t) \tag{5.60}$$

can be used to show input-to-state stability (ISS) using Theorem 4.19 in [87]. The ISS property implies Theorem 5.24. ■

Update Law

The next step of the P-ACA approach (Remark 4.5) is the design of the update laws for the unknown parameters. The update laws of the estimates $\hat{\mathbf{\Lambda}}_T(t)$ and $\hat{d}_T(t)$ follow the design methodology presented in Section 4.1 and are given by

$$\dot{\hat{d}}_T(t) = -\gamma_{d,vz} \cdot (\tilde{v}_z)_N^N(t) \cdot \frac{-(z_{B,z})_N(t)}{m} \tag{5.61a}$$

$$\dot{\hat{\mathbf{\Lambda}}}_T(t) = -D_T U_T \cdot \frac{-(z_{B,z})_N(t)}{m} \cdot (\tilde{v}_z)_N^N(t) \cdot \mathbf{u}^T(t) \cdot \mathbf{\Gamma}_{\Lambda,vz}, \tag{5.61b}$$

Noting that the estimated parameters are constant, the parameter estimation error dynamics are given by $\dot{\tilde{d}}_T(t) = \dot{\hat{d}}_T(t)$ and $\dot{\tilde{\mathbf{\Lambda}}}_T(t) = \dot{\hat{\mathbf{\Lambda}}}_T(t)$. The state estimation error dynamics are computed by subtracting the plant dynamics (5.49) from the predictor dynamics (5.53). Hence, the predictor error dynamics are given by

$$\left(\dot{\tilde{v}}_z\right)_N^{NN}(t) = -L_{v,z}(\tilde{v}_z)_N^N(t) - \frac{1}{m}(z_{B,z})_N(t)\left(U_T D_T \tilde{\mathbf{\Lambda}}_T \cdot \mathbf{u}(t) + \tilde{d}_T(t)\right), \tag{5.62a}$$

$$\dot{\tilde{d}}_T(t) = -\gamma_{d,vz} \cdot (\tilde{v}_z)_N^N(t) \cdot \frac{-(z_{B,z})_N(t)}{m} \tag{5.62b}$$

$$\dot{\tilde{\mathbf{\Lambda}}}_T(t) = -D_T U_T \cdot \frac{-(z_{B,z})_N(t)}{m} \cdot (\tilde{v}_z)_N^N(t) \cdot \mathbf{u}^T(t) \cdot \mathbf{\Gamma}_{\Lambda,vz}, \tag{5.62c}$$

with a positive constant $\gamma_{d,vz} \in \mathbb{R}$ and a positive definite matrix $\mathbf{\Gamma}_{\Lambda,vz} \in \mathbb{R}^{6\times 6}$. The underlined terms show that the estimated error dynamics are coupled between each other. These error dynamics lead to a stable equilibrium of the estimation errors and to an asymptotically vanishing tracking error between the plant (5.49) and the reference model (5.50). The following theorem summarizes the main result of this section.

Theorem 5.25 (Estimator Update Law) *Given are the system dynamics* (5.49)*, the state predictor* (5.53)*, the reference model* (5.50)*, the control law* (5.58)*, the update laws* (5.61)*, positive gains* $k_{v,z}$*,* $k_{vz,rm}$*,* $L_{v,z} > 0$*, symmetric positive definite adaptation gains* $\gamma_{d,vz}, \mathbf{\Gamma}_{\Lambda,vz}$ *and a bounded command* $(v_{c,z})_N^N(t)$*. Given Assumption 5.22 (Maximum Tilt Limit), it holds that:*

- *The equilibrium* $((\tilde{v}_z)_N^N, \tilde{d}_T, \tilde{\mathbf{\Lambda}}_T) = (0, 0, \mathbf{0})$ *is uniformly stable.*
- *The estimation error* $\left|\left|(\tilde{v}_z)_N^N(t)\right|\right| \to 0$ *for* $t \to \infty$*.*
- *And the plant tracking error vanishes* $|e_{v,z}(t)| \to 0$ *for* $t \to \infty$*.*

Proof: Analogously to Section 4.1, in order to show the first two statements in Theorem 5.25, the Lyapunov candidate function is chosen as

$$V(t) = \frac{1}{2}\left((\tilde{v}_z)_N^N(t)\right)^2 + \frac{1}{2}\gamma_{d,vz}^{-1} \cdot \tilde{d}_T^2(t) + \frac{1}{2}\tilde{\mathbf{\Lambda}}_T(t)\mathbf{\Gamma}_{\Lambda,vz}^{-1}\tilde{\mathbf{\Lambda}}_T^T(t). \tag{5.63}$$

The Lyapunov function is positive definite since $\gamma_{d,vz} > 0 \in \mathbb{R}$ and $\mathbf{\Gamma}_{\Lambda,vz} \in \mathbb{R}^{6\times 6}$ is a symmetric positive definite matrix. Both constants correspond to the adaptation rates.

The derivative of the Lyapunov function is calculated using the predictor estimation error dynamics (5.62) as

$$\begin{aligned}\dot{V}(t) &= (\tilde{v}_z)_N^N(t)\cdot\left(\dot{\tilde{v}}_z\right)_N^{NN}(t)+\gamma_{d,vz}^{-1}\cdot\tilde{d}_T(t)\dot{\tilde{d}}_T(t)+\widetilde{\mathbf{\Lambda}}_T(t)\mathbf{\Gamma}_{\Lambda,vz}^{-1}\dot{\widetilde{\mathbf{\Lambda}}}_T^T(t)\\ &= -L_{v,z}\left((\tilde{v}_z)_N^N(t)\right)^2-(\tilde{v}_z)_N^N(t)\cdot\frac{1}{m}(z_{B,z})_N(t)\left(U_TD_T\widetilde{\mathbf{\Lambda}}_T\cdot\mathbf{u}(t)+\tilde{d}_T(t)\right)\\ &\quad+\tilde{d}_T(t)(\tilde{v}_z)_N^N(t)\cdot\frac{(z_{B,z})_N(t)}{m}+\widetilde{\mathbf{\Lambda}}_T(t)\mathbf{u}(t)\cdot(\tilde{v}_z)_N^N(t)\frac{(z_{B,z})_N(t)}{m}\cdot U_TD_T,\\ &= -L_{v,z}\left((\tilde{v}_z)_N^N(t)\right)^2.\end{aligned}$$

Hence, the derivative of the Lyapunov function $\dot{V}(t) = -L_{v,z}\left((\tilde{v}_z)_N^N(t)\right)^2$ is negative semi-definite. From Theorem 4.8 (Uniform Stability) in [87], the equilibrium $((\tilde{v}_z)_N^N, \tilde{d}_T, \widetilde{\mathbf{\Lambda}}_T) = (0, 0, \mathbf{0})$ is uniformly stable and $(\tilde{v}_z)_N^N(t)$, $\tilde{d}_T(t)$, $\widetilde{\mathbf{\Lambda}}_T(t) \in \mathcal{L}_\infty$. Therefore, $\hat{d}_T(t)$ and $\hat{\mathbf{\Lambda}}_T(t)$ are also bounded.

Subsequently, the goal is to show asymptotic convergence of the estimation error $(\tilde{v}_z)_N^N(t)$ to zero based on Barbalat's Lemma (Lemma B.9). Following the same arguments as in Section 3.2, it is known that $\lim_{t\to\infty}\int_{\tau=0}^{t}\dot{V}\left((\tilde{v}_z)_N^N(\tau)\right)d\tau$ exists and is finite. It remains to be shown that $\dot{V}\left((\tilde{v}_z)_N^N(t)\right)$ is a uniform continuous function of t. This holds if its derivative

$$\ddot{V}(t) = -2L_{v,z}(\tilde{v}_z)_N^N(t)\cdot\left(\dot{\tilde{v}}_z\right)_N^{NN}(t)$$

is uniformly bounded. From the estimation error dynamics (5.62a), it follows that $\ddot{V}(t)$ is uniformly bounded if $(\tilde{v}_z)_N^N(t)$, $\left(\tilde{d}_T\right)_N^N(t)$, $\widetilde{\mathbf{\Lambda}}_T(t)$, $(z_{B,z})_N(t)$ and $\mathbf{u}(t)$ are uniformly bounded. Since $|(z_{B,z})_N(t)| \leq 1$ for all $t \geq 0$, it remains to show that $\mathbf{u}(t)$ is uniformly bounded. Given that $(\tilde{v}_z)_N^N(t) \in \mathcal{L}_\infty$, it follows from Theorem 5.24 (State Predictor Boundedness) that the estimated tracking state $(\hat{e}_z)_N^N(t)$ and therefore the tracking error $(e_z)_N^N(t)$ are bounded. From the virtual control law (5.58) and the solution of the control allocation (5.55) it follows that the virtual control $\nu_{T,d}(t)$ (5.58) and the control input $\mathbf{u}(t)$ are uniformly bounded. Finally, it is concluded that $\ddot{V}(t)$ is bounded uniformly in t for all $t \geq 0$ and $\dot{V}\left((\tilde{v}_z)_N^N(t)\right)$ is uniformly continuous. Using Lemma B.9 (Barbalat) it follows that $\dot{V}\left((\tilde{v}_z)_N^N(t)\right) \to 0$ and hence $(\tilde{v}_z)_N^N(t) \to 0$ for $t \to \infty$. From the second statement in Theorem 5.24 (State Predictor Boundedness) it follows that the predictor tracking error converges to the equilibrium $(\hat{e}_z)_N(t) \to 0$ and therefore the tracking error converges to the equilibrium $(e_z)_N(t) = \left((v_z)_N^N(t) - (v_{d,z})_N^N(t)\right) \to 0$ for $t \to \infty$. ■

An overview of the P-ACA vertical velocity controller is depicted in Figure 5.8 and includes all the implemented equations: The reference model (5.50), the baseline control law (5.51), the state predictor (5.53), the adaptive control law (5.58b), the update laws (5.61) and the control allocation as in Section 5.2. The vertical velocity controller as in Figure 5.8 in combination with the attitude and rate controller as in Figure 5.7 complete the controller depicted in Figure 5.1. The parameters of the attitude and rate controller are given in Table 5.1 and the ones of the vertical velocity controller in Table 5.2.

Subsystem	Description	Symbol	Value
Ref. Model	Yaw rate proportional gain	$k_{wz,rm}$	8.00
Ref. Model	Attitude proportional gain	$k_{a,rm}$	71.12
Ref. Model	Rate proportional gain	$\mathbf{K}_{\omega,rm}$	$16.86 \cdot \mathbf{I}$
Baseline Ctrl.	Attitude proportional gain	k_a	18.51
Baseline Ctrl.	Roll/pitch rate proportional gain	k_ω	6.02
Baseline Ctrl.	Yaw proportional gain	$k_{\omega,z}$	3.5
State Predictor	Luenberger gain	$\mathbf{L}$	$30 \cdot \mathbf{I}$
Update Law	Disturbance adaptation rate	$\mathbf{\Gamma}_d$	$[2.25, 2.36, 7.37] \cdot 10^{-2}$
Update Law	Control effect. adaptation rate	$\mathbf{\Gamma}_\Lambda$	$2.25 \cdot 10^{-2} \cdot \mathbf{I}$

Table 5.1: Attitude Controller Gains

Subsystem	Description	Symbol	Value
Ref. Model	Vert. vel. proportional gain	$k_{vz,rm}$	2.5
Baseline Controller	Vert. vel. proportional gain	$k_{v,z}$	3.5
State Predictor	Luenberger gain	$L_{v,z}$	20
Update Law	Disturbance adaptive rate	$\gamma_{d,vz}$	100
Update Law	Control effectiveness adaptive rate	$\mathbf{\Gamma}_{\Lambda,vz}$	$2.25 \cdot 10^{-2} \cdot \mathbf{I}$

Table 5.2: Vertical Velocity Controller Gains

5.6 Bibliographical Remarks

First results using geometric attitude control laws in $SO(3)$ for multicopter control have been presented in [95]. Subsequently, the use of a reduced attitude parameterization in $\mathbb{S}^2$ has been presented in [48,55,149] using non-adaptive techniques. The $SO(3)$ attitude parameterization has been presented in combination with robust adaptive control [93] and $\mathcal{L}_1$ adaptive control [88] but faults were not addressed. In previous work [51], incremental control using the reduced attitude parameterization for addressing actuator degradation has been presented. To the best knowledge of the author, this is the first time that the reduced attitude parameterization in $\mathbb{S}^2$ has been used within an adaptive control framework. Furthermore, the stability properties are shown decoupled from the yaw motion for the first time. These properties enhance the advantages of the Prioritizing Control Allocation (PRIO CA) during degraded control situations.

The formulation of the attitude Lyapunov function (5.18) and Theorem 5.9 (Local Exponential Stability) are motivated by the analysis in $SO(3)$ in [93]. In this thesis it has been adapted to reduced attitude control in $\mathbb{S}^2$ and used for adaptive control. The structure of the almost global asymptotic stability (AGAS) proof has been inspired by [55, 113]. Finally, the selection of the attitude proportional feedback term corresponds to the reduced attitude case in [26].

Compared to the previous work in [42, 43], where PMRAC has been applied to the hexacopter's total failure scenario, the reduced attitude parameterization in $\mathbb{S}^2$ is used and the uncertainty parameterization and the control allocation strategy have been updated. The control allocation corresponds to the PRIO CA instead of a gradient-based optimization. The control effectiveness matrix $\mathbf{\Lambda}$ is not limited to a diagonal matrix and by using the PRO approach a clear separation of the update laws between

attitude (5.46b) and vertical velocity (5.61b) has been achieved. Finally, the fault model (3.5) integrates not only multiplicative but also additive uncertainties.

Chapter 6

Experimental Results

For demonstrating the performance and fault tolerance of the controller presented in Chapter 5 (Multirotor Controllers) several flight tests have been done using the hexacopter platform described in Chapter 2 (Multirotor Dynamics Model). The test cases include two hexacopter configurations with representative failure cases for demonstrating repeatable results. In order to emulate faults and total failures, two different approaches have been used. The hardware approach corresponds to the dismounting the propeller previous to a given flight. In this way is is possible to realistically analyze the performance after a total failure, but it is not possible to address the direct reaction of the system to a fault during flight and it is neither possible to consider partial faults. This is done using a software approach where the input command is multiplied by an externally set control effectiveness. In this way a total failure corresponds to an idle command to the corresponding ESC regardless of the commands of the controller. The failure scenarios are evaluated during hover flight as well as high speed cruise flight. Furthermore, the performance is analyzed post-failure during hover flight, high speed cruise flight and a 2-2-1-1 maneuver. During the experiments, the faults are unknown to the controller, all computations are done onboard, and no external sensors were used. Furthermore, a single unified controller handles the nominal and all the faulty cases.

6.1 Failure during Hover Flight - Single Case

In this section, a total failure of actuator 5 during hover flight using configuration 1 of the hexacopter system as presented in Chapter 2 is analyzed. The pilot commands the reduced attitude vector $(\vec{\mathbf{z}}_c)_N(t) \in \mathbb{S}^2$, the yaw rate $(\omega_{c,z})_B(t) \in \mathbb{R}$ and the vertical velocity $(v_{c,z})_N^N(t) \in \mathbb{R}$. In configuration 1, a total actuator failure means that yaw moment cannot be generated in one direction while hovering as depicted in Figure 2.4. In the best case the system is able to track the desired reduced attitude vector $\vec{\mathbf{z}}_d(t)$ and the desired vertical velocity $(v_{d,z})_N^N(t)$, which corresponds to a fail-graceful system. In this case it is possible to remain operational but with a degraded performance [12, p.8].

Figure 6.1 shows the general performance of the controller. The time has been shifted such that the failure time corresponds to $t = 0[s]$. The plot shows attitude, rate and vertical velocity tracking as well as horizontal velocity. For analyzing the attitude tracking performance, the Euler angles are used as attitude parameters due to ease of interpretation. The desired values of the roll $\Phi_d(t)$ and pitch angles $\Theta_d(t)$ are

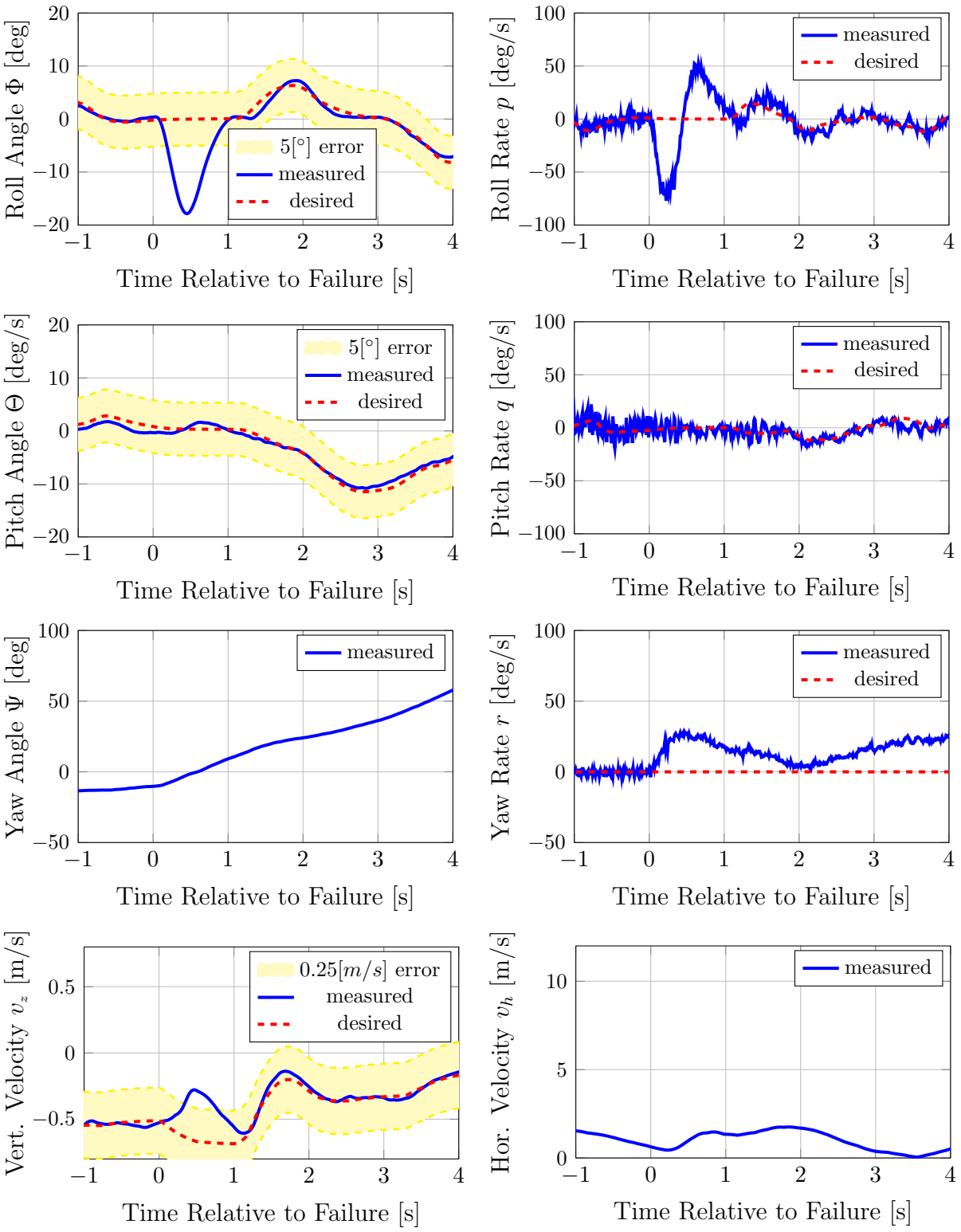


Figure 6.1: Tracking Performance - Fault Case $\lambda_5 = 0$ during Hover Flight - Config. 1

computed from the desired reduced attitude vector $(\vec{\mathbf{z}}_d)_N(t)$ by inverting the relation (2.18) derived in Chapter 2 as follows

$$\Phi_d(t) = \arcsin\left((z_{d,x})_N(t) \cdot \sin\Psi(t) - (z_{d,y})_N(t) \cdot \cos\Psi(t)\right), \tag{6.1a}$$

$$\Theta_d(t) = \arctan\left(\frac{(z_{d,x})_N(t) \cdot \cos\Psi(t) + (z_{d,y})_N \cdot \sin\Psi(t)}{(z_{d,z})_N(t)}\right). \tag{6.1b}$$

Since there is no desired heading $\Psi_d(t)$, the measured heading $\Psi(t)$ is used for the transformation. In Figure 6.1 it can be seen that the failure of actuator 5 does not affect the performance of the pitch angle tracking performance but it influences the roll angle, yaw rate and vertical velocity tracking performance. After $447\ [ms]$ the absolute roll angle tracking error $|\Phi_d(t) - \Phi(t)|$ reaches a maximum of $17.84\ [deg]$ and after $225\ [ms]$ the absolute roll rate tracking error $|(e_{w,x})_B(t)|$ reaches a maximum of $69.97\ [deg/s]$. After $792\ [ms]$ the roll angle tracking error remains under $5\ [deg]$. In Figure 6.1 this limit is marked as a yellow area. Since the actuator 5 has failed, negative yaw moments cannot be generated during hover. A moderate positive yaw rate tracking error $|(e_{w,z})_B(t)|$ under $50\ [deg/s]$ is observed and consequently the yaw angle slowly increases. The vertical velocity tracking error $|(e_{v,z})_N(t)|$ reaches a maximum of $0.39\ [m/s]$ at $486\ [ms]$. In the last plot in Figure 6.1, the horizontal velocity $v_h(t) \in \mathbb{R}$ is depicted. Since the fault has been induced during hover, the velocity remains small under $3\ m/s$. The horizontal velocity $v_h(t)$ is computed as

$$v_h(t) = \sqrt{(v_x)_N^N(t) \cdot (v_x)_N^N(t) + (v_y)_N^N(t) \cdot (v_y)_N^N(t)} \tag{6.2}$$

where $(v_x)_N^N(t) \in \mathbb{R}$ and $(v_y)_N^N(t) \in \mathbb{R}$ are the first and second elements of $(\vec{\mathbf{v}})_N^N(t)$.

In order to understand how the failure is compensated, the adaptation and the control allocation processes are analyzed in the following. The adaptation results depend on the selected predictor error dynamics (5.47) and (5.62) which combine state and parameter estimation. The adaptive parameters are the disturbances $\hat{\vec{\mathbf{d}}}_M(t)$ and $\hat{d}_T(t)$ as well as the reduced control effectiveness matrix $\hat{\mathbf{\Lambda}}_{a,r}(t)$ composed by $\hat{\mathbf{\Lambda}}_T(t)$ and $\hat{\mathbf{\Lambda}}_M(t)$. From the Lyapunov analysis it is known that adaptation is active as long as there exist non-zero state estimation errors $\left(\tilde{\vec{\boldsymbol{\omega}}}^{NB}\right)_B(t)$ and $(\tilde{v}_z)_N^N(t)$. In Figure 6.2 the rotational rates $(\vec{\boldsymbol{\omega}}^{NB})_B(t)$ and the vertical velocity $(v_z)_N^N(t)$ and their respective estimates $\left(\hat{\vec{\boldsymbol{\omega}}}^{NB}\right)_B(t)$ and $(\hat{v}_z)_N^N(t)$ are plotted. It can be seen that the estimation error increases only during a short time span of less than a second after the failure occurred. Especially the roll rate and vertical velocity estimation errors can be seen in the figure.

Figure 6.3 depicts the reduced control effectiveness matrix $\mathbf{\Lambda}_{a,r}(t)$ and its estimation $\hat{\mathbf{\Lambda}}_{a,r}(t)$. The elements are arranged by columns corresponding to each of the actuators and organized such that they geometrically match the position of the actuators. For example, actuator 1 is located to the front-right of the origin of the B-system. Each row of $\hat{\mathbf{\Lambda}}_{a,r}(t)$ represents a virtual control and has a constant color through all the plots. For facilitating the interpretation, each of the matrix elements is multiplied by the sign of its nominal value (initial value)

$$\hat{\mathbf{\Lambda}}_{a,r}(t=0) = \mathbf{V}_{a,r}^T \cdot \hat{\mathbf{\Lambda}}(t=0) = \mathbf{V}_{a,r}^T \cdot \mathbf{I}. \tag{6.3}$$

The computation is the following

$$\hat{\lambda}^*_{a,r,ij}(t) := \begin{cases} \hat{\lambda}_{a,r,ij}(t), & \text{if } \hat{\lambda}_{a,r,ij}(t=0) \geq 0 \\ -\hat{\lambda}_{a,r,ij}(t), & \text{if } \hat{\lambda}_{a,r,ij}(t=0) < 0 \end{cases}, \text{ for } i = 1...4, j = 1...6. \tag{6.4}$$

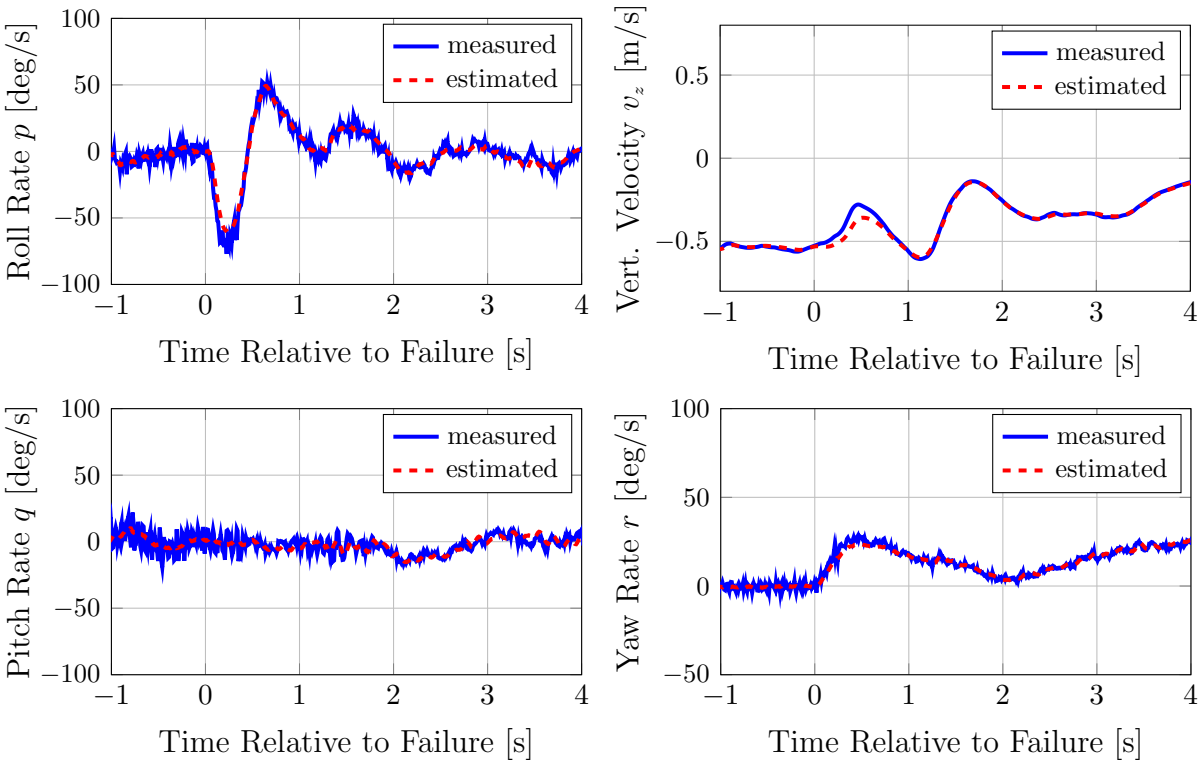


Figure 6.2: State Estimation - Fault Case $\lambda_5 = 0$ during Hover Flight - Configuration 1

In this way, an increase of $\hat{\lambda}^*(t)$ means an increase of effectiveness regardless of the nominal direction. Here, $\hat{\lambda}^*_{a,r,ij}(t), \hat{\lambda}_{a,r,ij}(t) \in \mathbb{R}$ correspond to the element in the i-th row and j-th column of $\hat{\mathbf{\Lambda}}^*_{a,r}(t) \in \mathbb{R}^{4\times6}$ and $\hat{\mathbf{\Lambda}}_{a,r}(t) \in \mathbb{R}^{4\times6}$ respectively. The theoretical ideal values $\mathbf{\Lambda}_{a,r}$ are transformed to $\mathbf{\Lambda}^*_{a,r} \in \mathbb{R}^{4\times6}$ analogously to (6.4). In Figure 6.3 it can be seen that these ideal values $\mathbf{\Lambda}^*_{a,r}$ only change for actuator 5 at the time of the failure $t = 0$. The matrices $\hat{\mathbf{\Lambda}}^*_{M,x}(t), \hat{\mathbf{\Lambda}}^*_{M,y}(t), \hat{\mathbf{\Lambda}}^*_{M,z}(t), \hat{\mathbf{\Lambda}}^*_{T}(t) \in \mathbb{R}^{1\times6}$ each represent a row of $\hat{\mathbf{\Lambda}}^*_{a,r}(t)$. In Figure 6.3 it can be seen that the elements corresponding to the roll moment $\hat{\mathbf{\Lambda}}^*_{M,x}(t)$ change the most. The ones corresponding to rotors on the left side decrease and the ones corresponding to rotors on the right side increase. The elements corresponding to the yaw moment $\hat{\mathbf{\Lambda}}^*_{M,z}(t)$ and to the rotors 2, 4 and 6 slightly increase. The ones corresponding to the rotors 1, 3 and 5 slightly decrease. All the elements of $\hat{\mathbf{\Lambda}}^*_{T}(t)$ slightly decrease. These changes match with the direction of the faulty actuator 5 as seen from the 5-th column of the nominal control input matrix $\mathbf{B}_a$ (2.9) or its normalized version $\mathbf{B}_{\nu u}(t)$ derived in Example 4.4 (Hexacopter Input Normalization).

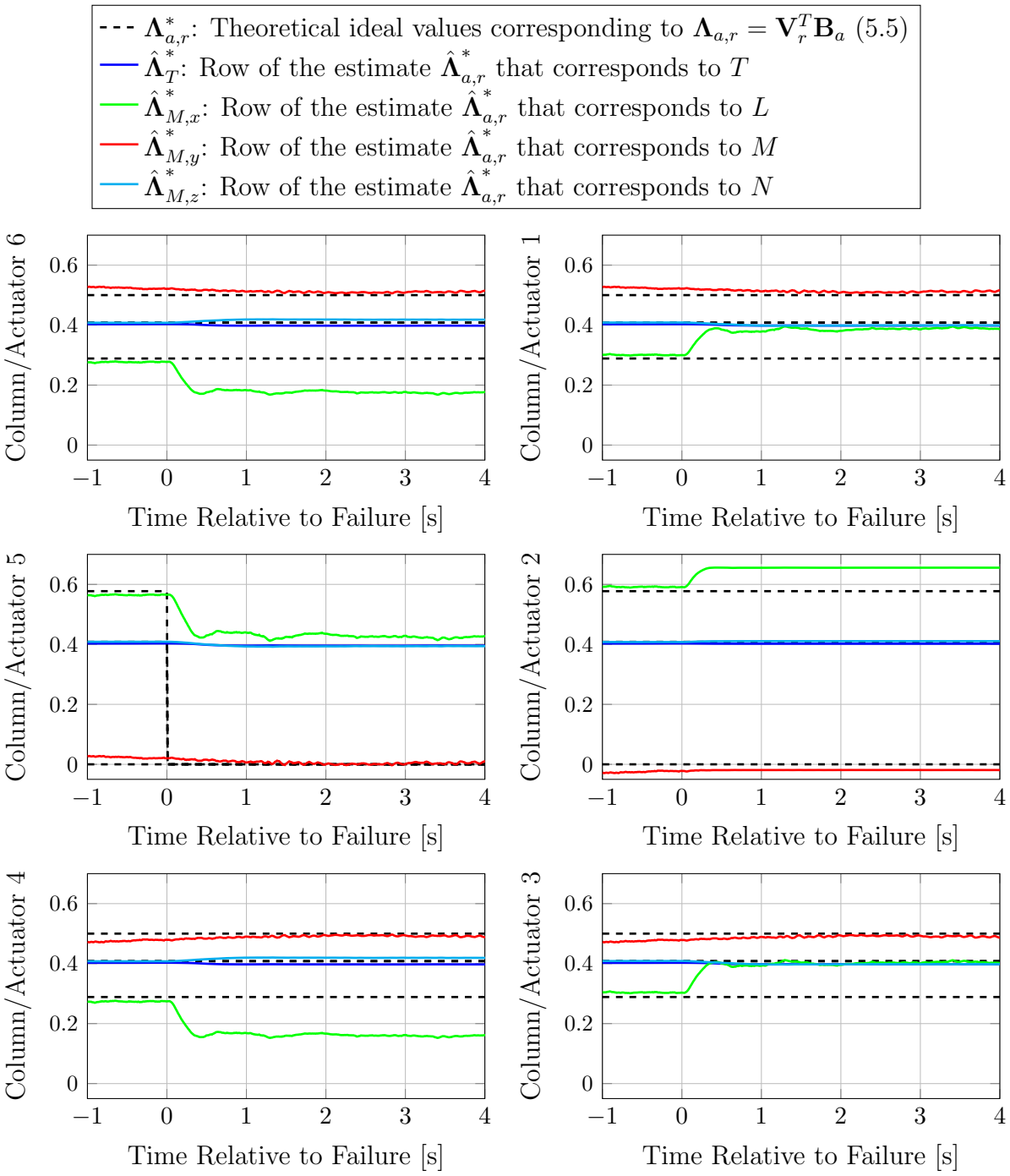


Figure 6.3: Reduced Control Effectiveness - Fault Case $\lambda_5 = 0$ during Hover Flight - Configuration 1

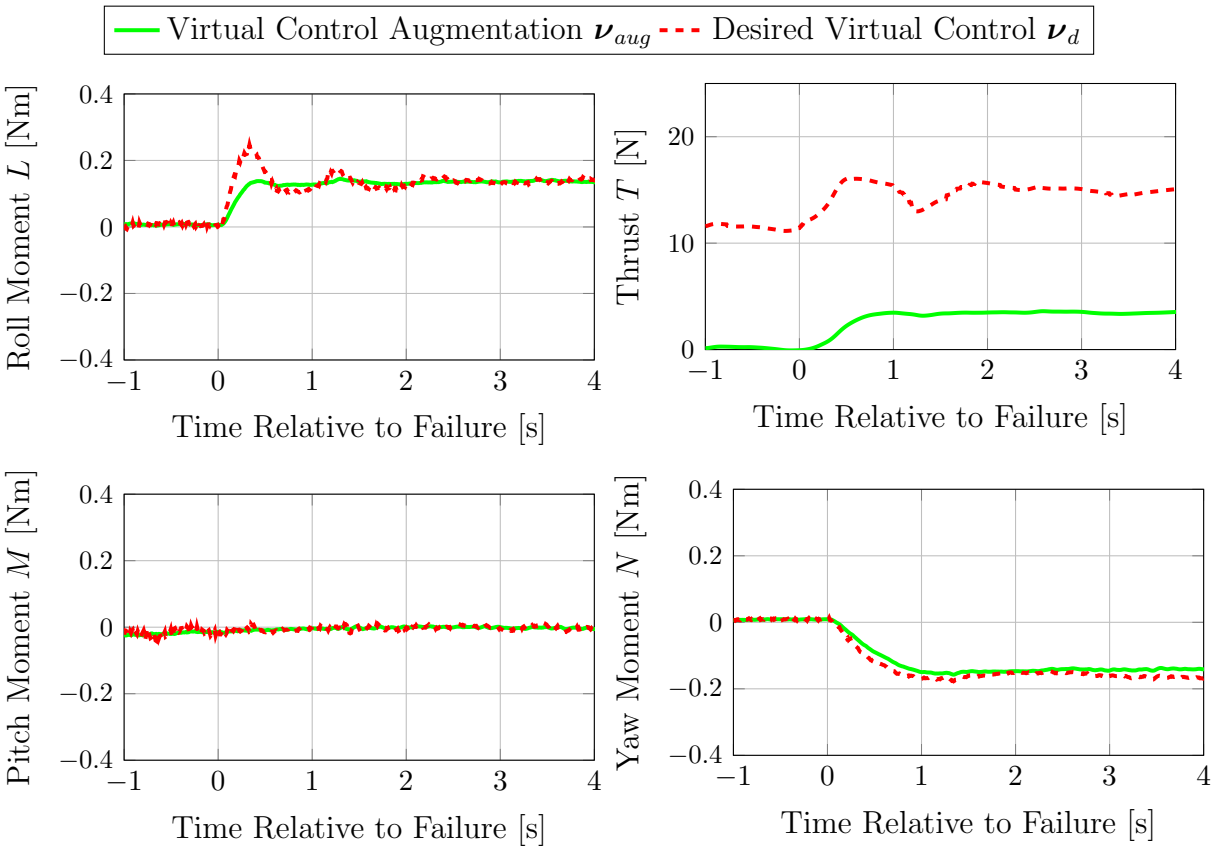


Figure 6.4: Virtual Controls - Fault Case $\lambda_5 = 0$ during Hover Flight - Configuration 1

The estimated disturbances $\hat{\vec{\mathbf{d}}}_M(t)$ and $\hat{d}_T(t)$ also compensate for the failure of actuator 5. This can be seen in a virtual control context since the adaptive augmentation term $\boldsymbol{\nu}_{aug}(t)$ corresponds to the cancellation of the estimated disturbances. In Figure 6.4, the desired virtual control $\boldsymbol{\nu}_d(t)$ (5.43) and (5.58) and the adaptive augmentation term $\boldsymbol{\nu}_{aug}(t)$ (5.43b) and (5.58b) are shown. The difference between $\boldsymbol{\nu}_d(t)$ and $\boldsymbol{\nu}_{aug}(t)$ corresponds to the virtual control commanded by the baseline controller $\boldsymbol{\nu}_b(t)$ (5.43c) and (5.58c). In the presented case, after the failure, the thrust $\nu_{T,aug}(t)$ and the roll moment demand $\nu_{M,x,aug}(t)$ increase while the yaw moment demand $\nu_{M,z,aug}(t)$ becomes negative. The adaptive augmentation term $\boldsymbol{\nu}_{aug}(t)$ matches with the direction of the faulty actuator 5 as seen from the 5-th column of the nominal control input matrix $\mathbf{B}_a$ (2.9) or its normalized version $\mathbf{B}_{\nu u}(t)$ derived in Example 4.4 (Hexacopter Input Normalization).

From Figures 6.3 and 6.4 it can be seen that shortly after the failure the unknown parameters $\hat{\boldsymbol{\Lambda}}_{r,M}(t)$, $\hat{\vec{\mathbf{d}}}_M(t)$, $\hat{\boldsymbol{\Lambda}}_{r,T}(t)$ and $\hat{d}_T(t)$ adapt but do not converge to the theoretical ideal values. From the Lyapunov analysis in Chapter 5, this is an expected result since the parameter error states have been proven to be stable but not to convergence to zero. During the analysis it additionally needs to be taken into account that the theoretical ideal values $\boldsymbol{\Lambda}_{r,M}$, $\mathbf{d}_M$, $\boldsymbol{\Lambda}_{r,T}$ and d_T may differ from the experimental ones since a dynamics model inherently contains modeling errors. In this case the most predominant uncertainties correspond to a center of gravity shift, wind disturbances like turbulence or gusts and the propeller model. Nevertheless, from the tracking per-

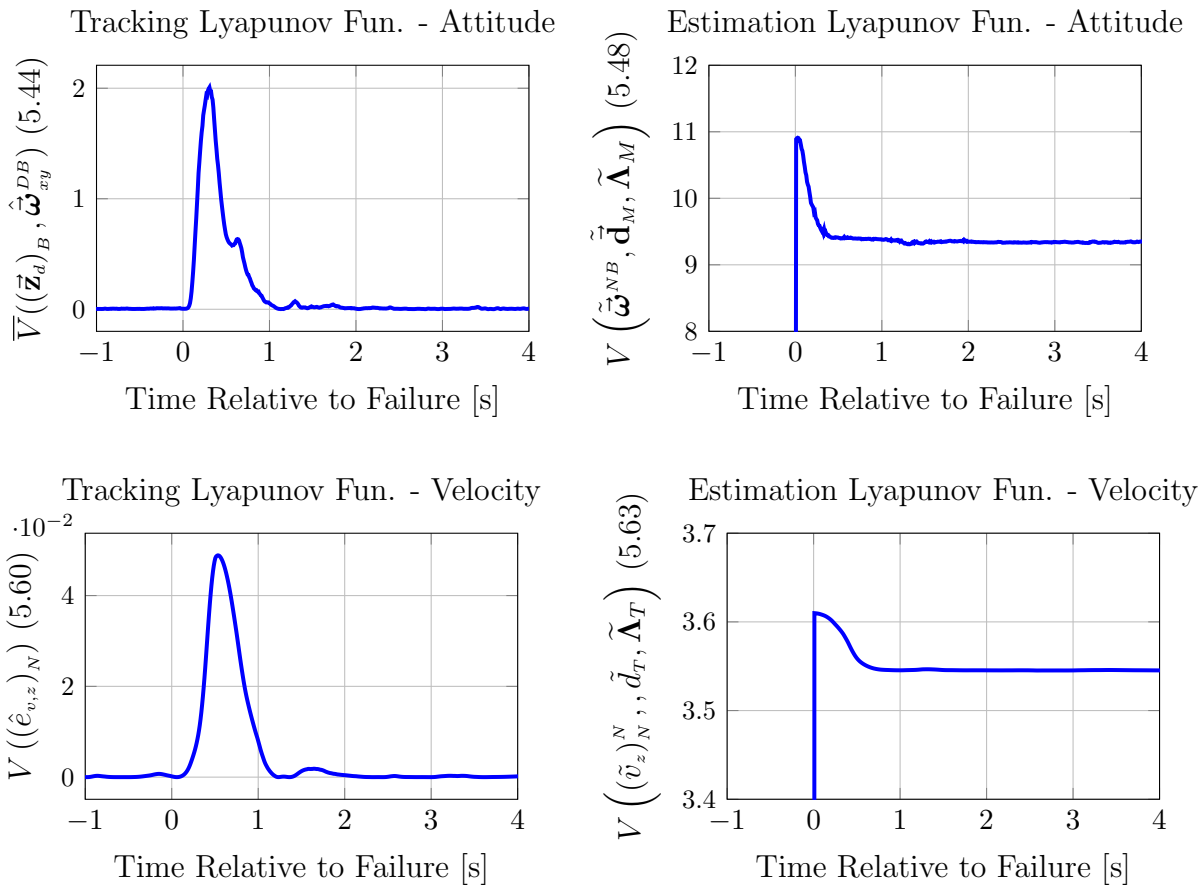


Figure 6.5: Lyapunov Analysis - Fault Case $\lambda_5 = 0$ during Hover Flight - Configuration 1

formance in Figure 6.1, it can be concluded that the considered effects in the model appropriately describe the plant dynamics for the use of the selected control strategy.

The interaction of the predictor dynamics and adaptation with the tracking performance can be analyzed using the respective Lyapunov functions presented in the last chapter. The attitude and velocity tracking Lyapunov functions (5.44) and (5.60) and the attitude and velocity estimation Lyapunov functions (5.48) and (5.63) are depicted in Figure 6.5. At the time of the failure $t = 0$, the control effectiveness $\mathbf{\Lambda}$ changes and therefore the estimation Lyapunov functions (5.48) and (5.63) increase with a discrete step. Before the failure, the estimation Lyapunov functions (5.48) and (5.63) had values lower than 0.1 and $5 \cdot 10^{-3}$ respectively. After the failure and as long as the state estimation errors $\left(\tilde{\boldsymbol{\omega}}^{NB}\right)_B(t)$ and $(\tilde{v}_z)_N^N(t)$ are large, the Lyapunov functions decrease significantly. As can be seen in Figure 6.5 both Lyapunov functions rapidly decrease to a constant positive value. This error corresponds to the adaptive parameters which have not converge to their desired values. Furthermore, from the Theorems 5.19 (Full State Predictor - Boundedness) and 5.24 (State Predictor Boundedness) it is known that the predictor tracking performance depends on the state estimation errors. This effect can be seen from the depicted tracking Lyapunov functions (5.44) and (5.60). After the failure there is an increase and once the estimation errors $\left(\tilde{\boldsymbol{\omega}}^{NB}\right)_B(t)$ and $(\tilde{v}_z)_N^N(t)$ are small enough, the Lyapunov functions start decreasing again.

The interaction between the control allocation and the adaptation can be assessed

using the virtual control space representation. An alternative presentation of the estimated control effectiveness $\hat{\mathbf{\Lambda}}_{a,r}(t)$ is the estimated attainable control set (ACS) which is defined analogously to the ACS (2.20) as introduced in Chapter 2. The estimated ACS is defined as a 4-dimensional polyhedron

$$\hat{\mathcal{V}}(t) := \{\boldsymbol{\nu} \in \mathbb{R}^4 \mid \boldsymbol{\nu} = \mathbf{B}_a \hat{\mathbf{\Lambda}}(t)\,\mathbf{u} = \mathbf{U}_{a,r}\mathbf{D}_a \hat{\mathbf{\Lambda}}_{a,r}(t)\,\mathbf{u}, \mathbf{u} \in \mathcal{U}\}, \tag{6.5}$$

where the input set is defined in (2.19). In Figures 6.6, 6.7 and 6.8, three ACS are compared at the times $t = 0[s]$, $t = 0.3[s]$ and $t = 1[s]$ respectively. The sets correspond to the nominal ACS (2.20) with $\mathbf{\Lambda} = \mathbf{I}$ and $\mathbf{d}_\nu = \mathbf{0}$, the ACS (2.20) corresponding to the fault case with

$$\mathbf{\Lambda} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

and $\mathbf{d}_\nu = \mathbf{0}$ and the estimated ACS (6.5). Each of the figures shows the roll and pitch moment L/M plane and the thrust and yaw moment T/N plane. Apart from the ACS polygons, the virtual controls $\boldsymbol{\nu}(t)$ PRIO and $\hat{\boldsymbol{\nu}}(t)$ PRIO are depicted. They are computed as

$$\begin{aligned} \boldsymbol{\nu}(t) &= \mathbf{B}_a \cdot \mathbf{\Lambda} \cdot \mathbf{u}(t), \\ \hat{\boldsymbol{\nu}}(t) &= \mathbf{B}_a \hat{\mathbf{\Lambda}}(t)\,\mathbf{u}(t) = \mathbf{U}_{a,r}\mathbf{D}_a \hat{\mathbf{\Lambda}}_{a,r}(t)\,\mathbf{u}(t), \end{aligned}$$

using the control inputs $\mathbf{u}(t)$ computed by the Prioritizing Control Allocation (PRIO) during the flight test. For plotting the L/M plane, the nominal and the fault ACS are cut at $N(t)$ and $T(t)$ and the estimated ACS (6.5) is cut at $\hat{N}(t)$ and $\hat{T}(t)$. For plotting the T/N plane, the nominal and the fault ACS (2.20) are cut at $L(t)$ and $M(t)$ and the estimated ACS (6.5) is cut at $\hat{L}(t)$ and $\hat{M}(t)$. The values $L(t), M(t), N(t), T(t)$ correspond to $\boldsymbol{\nu}(t)$ PRIO and $\hat{L}(t), \hat{M}(t), \hat{N}(t), \hat{T}(t)$ correspond to $\hat{\boldsymbol{\nu}}(t)$ PRIO as shown in the figure. For comparison, the virtual controls $\boldsymbol{\nu}(t)$ PINV and $\hat{\boldsymbol{\nu}}(t)$ PINV corresponding to the Pseudo-Inverse Control Allocation (PINV CA) are plotted. These virtual controls have been computed in post-processing since the PINV CA was not used in flight.

By comparing the nominal ACS with the fault ACS, the physical loss of control authority can be seen. The estimation error of the control effectiveness matrix $\widetilde{\mathbf{\Lambda}}_{a,r}(t)$ can be visualized in two ways. One possibility is the comparison of the fault ACS (2.20) with the estimated ACS (6.5). The second option is the comparison of $\boldsymbol{\nu}(t)$ and $\hat{\boldsymbol{\nu}}(t)$, i.e. the difference between the red and blue markers for both PRIO and PINV control allocations. In Figure 6.6, $t = 0[s]$ is depicted and therefore the difference between the ACS and the estimated ACS is the largest since the parameter adaptation has just started. This difference is rapidly reduced as can be seen in Figures 6.7 and 6.8. In Figure 6.8, it can be seen that there is a remaining control effectiveness matrix estimation error $\widetilde{\mathbf{\Lambda}}_{a,r}(t)$ although at $t = 1[s]$ the system has recovered tracking performance as seen in Figure 6.1.

An interesting effect is the interaction between the estimated constant disturbances $\hat{\vec{\mathbf{d}}}_M(t)$ and $\hat{d}_T(t)$ and the estimated control effectiveness matrices $\hat{\mathbf{\Lambda}}_M(t)$ and $\hat{\mathbf{\Lambda}}_T(t)$.

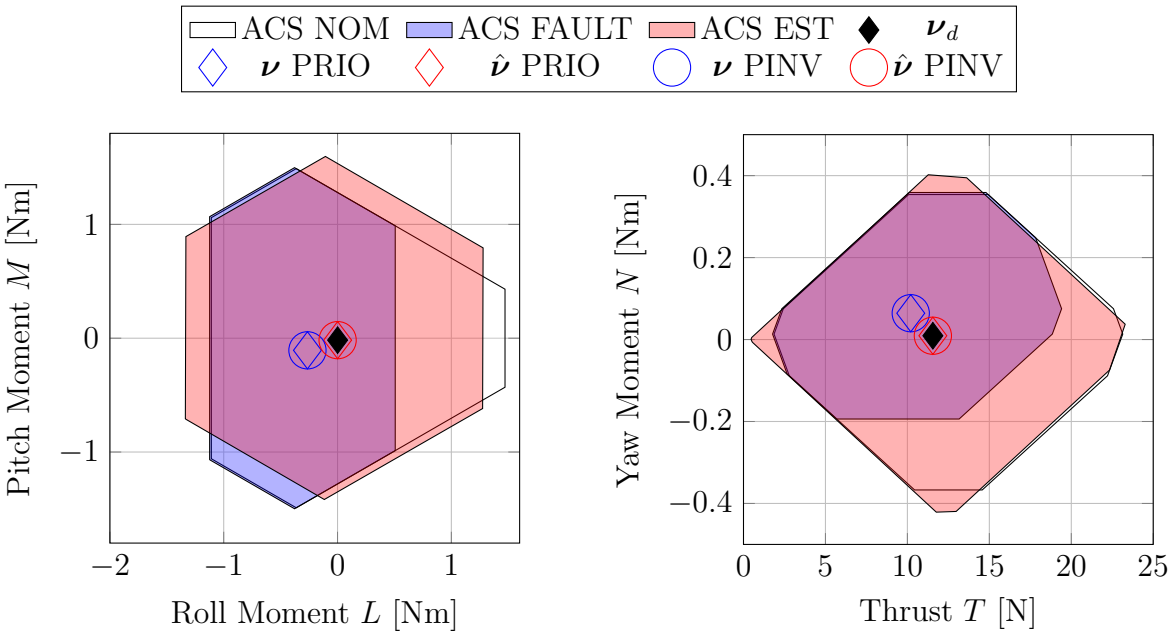


Figure 6.6: Virtual Control Space at $t = 0[s]$ - Fault Case $\lambda_5 = 0$ during Hover Flight - Configuration 1: Comparison of the Prioritizing Control Allocation (PRIO CA) and the Pseudo-Inverse Control Allocation (PINV CA)

All the adaptive parameters change while non-zero estimation errors $\left(\tilde{\vec{\omega}}^{NB}\right)_B(t)$ and $(\tilde{v}_z)_N^N(t)$ exists such that the disturbance forces and moments are counteracted. Adaptation is negligible once the estimation errors are small enough. This occurs once the combination of all parameter changes compensate for the disturbance arising from the failure. Therefore, since the estimation of the constant disturbance also compensates for the actuator failure, it inhibits a better estimation of the control effectiveness matrices. In order to improve the estimation, the system needs to be excited such that it is possible to differentiate between these two types of parameters.

In order to analyze the control allocation performance based on the available information, the estimated virtual control $\hat{\boldsymbol{\nu}}(t)$ is compared with the desired virtual control $\boldsymbol{\nu}_d(t)$. In Figures 6.6 and 6.7, the control allocation is able to match the desired virtual control $\boldsymbol{\nu}_d(t)$. Furthermore, there is no difference between the PRIO CA and the PINV CA. In Figure 6.8 it can be seen that the PRIO CA allocates the desired roll moment $L_d(t)$, pitch moment $M_d(t)$ and thrust $T_d(t)$ correctly but there is a small error in the yaw moment $N_d(t)$. This occurs due to control input saturation and is the intended behaviour due to the prioritizing strategy. In comparison it can be seen that the estimated virtual controls $\hat{\boldsymbol{\nu}}(t)$ using the PINV CA show errors in roll, yaw and thrust.

For a better visualization, the time sequence of the desired virtual control $\boldsymbol{\nu}_d(t)$ and the estimated virtual controls $\hat{\boldsymbol{\nu}}(t)$ corresponding to the PRIO and PINV control allocations are plotted in Figure 6.9. In this fault scenario, the advantage of the prioritization strategy can be clearly seen from the roll moment plot. The PRIO CA is able to satisfy the roll moment demand $L_d(t)$ while the error using PINV CA increases after $t = 0.3[s]$. This would lead to the inability of maintaining the desired position and attitude leading to a loss of control situation. In contrast, the prioritization allows for maintaining reduced attitude control. Furthermore, the PRIO CA is able to correctly allocate the

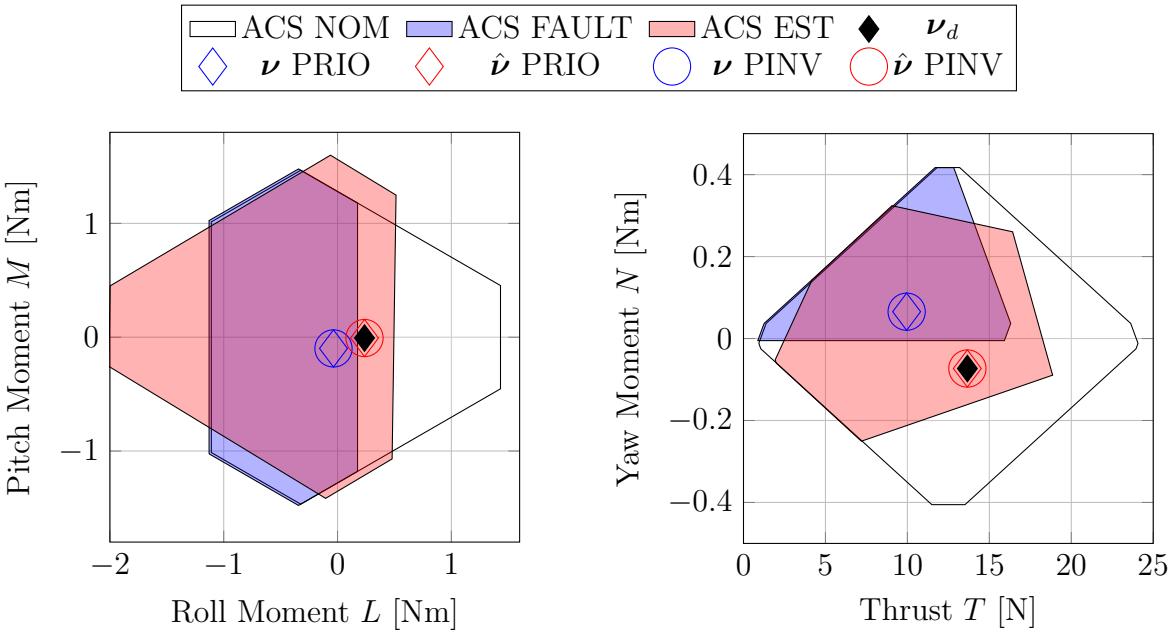


Figure 6.7: Virtual Control Space at $t = 0.3[s]$ - Fault Case $\lambda_5 = 0$ during Hover Flight - Configuration 1: Comparison of the Prioritizing Control Allocation (PRIO CA) and the Pseudo-Inverse Control Allocation (PINV CA)

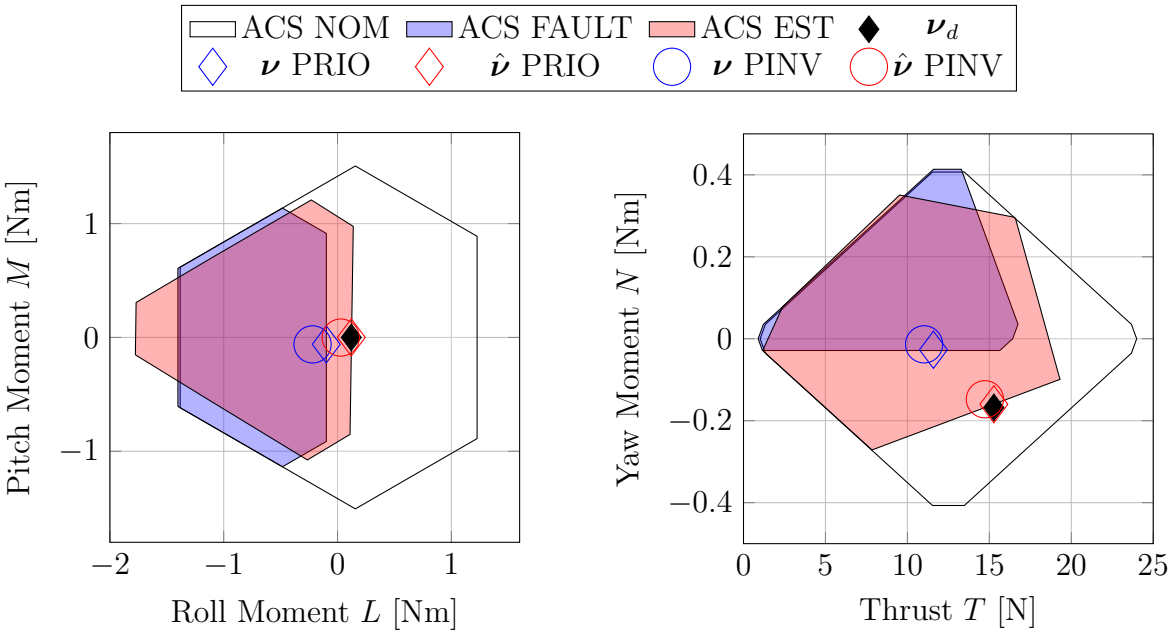


Figure 6.8: Virtual Control Space at $t = 1[s]$ - Fault Case $\lambda_5 = 0$ during Hover Flight - Configuration 1: Comparison of the Prioritizing Control Allocation (PRIO CA) and the Pseudo-Inverse Control Allocation (PINV CA)

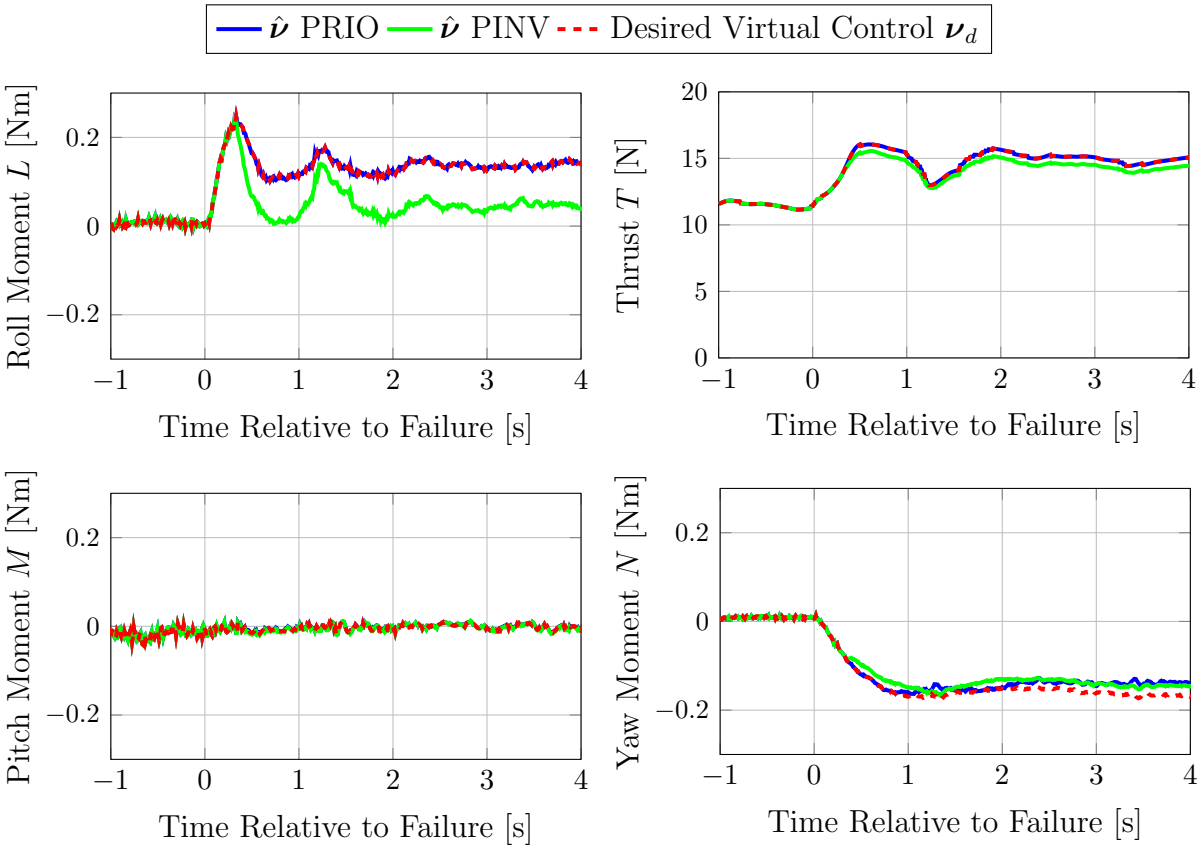


Figure 6.9: Control Allocation Performance - Estimated Virtual Controls - Fault Case $\lambda_5 = 0$ during Hover Flight - Configuration 1

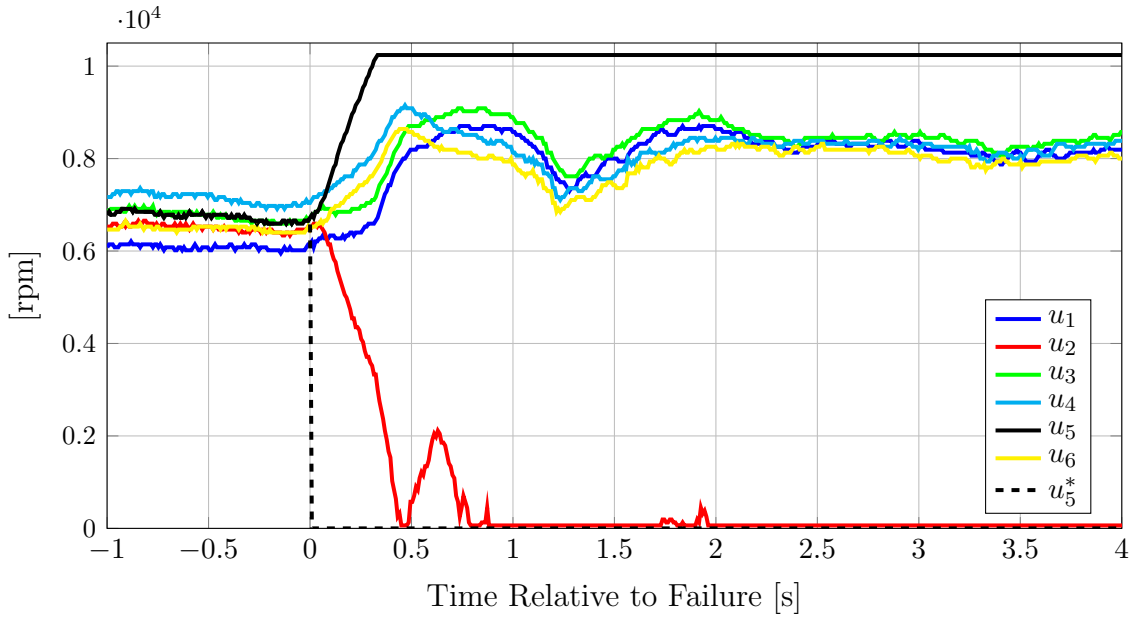


Figure 6.10: Control Inputs - Fault Case $\lambda_5 = 0$ during Hover Flight - Configuration 1

desired pitch moment $M_d(t)$ and thrust $T_d(t)$ but not the desired yaw moment $N_d(t)$. The PINV CA shows small errors in both thrust and yaw moment allocation but is able to correctly allocate the desired pitch moment. The general importance of a correct control allocation can be seen from the yaw rate tracking performance. Although the control allocation error in the yaw axis is not large using the PRIO CA, it translates to moderate errors in the yaw rate tracking as seen in Figure 6.1. By setting this result in relation with the PINV CA, it is clear that the roll moment error is unacceptable.

The adaptation together with the control allocation lead to the control input commands shown in Figure 6.10. The virtual degradation of the command 5 $u_5^*(t)$ is also depicted. From the time the failure is active, the input commands $u_2(t)$ and $u_5(t)$ rapidly change. The command $u_2(t)$ decreases and the command $u_5(t)$ increases causing the saturation of both commands. The PRIO CA starts differing from the PINV CA at the time the command $u_5(t)$ saturates. The control input commands 1, 3, 4 and 6 increase in order to compensate for the thrust deficit due to the reduction of the control input 2 and the failure of actuator 5.

In summary, the presented P-ACA control approach is able to tackle the challenges of handling fault and failure uncertainties while simultaneusly dealing with actuator redundancy and control input saturation. Due to the physical limitations of the analyzed hexacopter configuration, a fail-operational system is not possible since the four virtual controls cannot be controlled independently after a failure. The next goal is a fail-graceful system which has been achieved using a prioritization strategy implemented as the Prioritizing Control Allocation (PRIO CA).

6.2 Failure during Hover - Overview

After the in-depth controller analysis, this section aims to show the performance of the controller considering different failure cases. Figure 6.11 presents an overview of the compensation of a total failure during a hover maneuver for representative actuators of each of the two hexacopter configurations in Figure 2.1. A failure of actuators 3 and 5 are considered for both configurations and a failure of actuator 1 is considered for configuration 2. Due to the symmetry of the system, they are considered to be representative for all possible single actuator failures. Each of the propellers is represented with a different color, the normal line type represents configuration 1 and the dashed line corresponds to configuration 2. In order to emulate a total failure during flights, the failure has been induced by an idle command to the corresponding ESC regardless of the commands of the controller. For all the flights, the time has been shifted such that the time of the induced failure is zero. The pilot commands the reduced attitude vector $(\vec{\mathbf{z}}_c)_N(t) \in \mathbb{S}^2$, the yaw rate $(\omega_{c,z})_B(t) \in \mathbb{R}$ and the vertical velocity $(v_{c,z})_N^N(t) \in \mathbb{R}$.

For analyzing the attitude error, the Euler angles and the tilt angle error $\varphi_d(t) \in [0, \pi]$ between $\vec{\mathbf{z}}_d(t)$ and $\vec{\mathbf{z}}_B(t)$ are used. For the sake of readability, the desired attitude and rate values of each controller have not been plotted. They are close to a zero command but not all the same. Especially after the failure the commands differ since position corrections are made by the pilot. Nevertheless, the tilt angle error $\varphi_d(t)$ is a direct measure of the attitude error. In Figure 6.11 it can be seen that a failure of actuator 1 leads to a positive roll angle error and a negative pitch angle error, that a failure of actuator 3 leads to a positive roll angle error and a positive pitch angle error and that a failure of actuator 5 leads mainly to a negative roll angle error. This behavior is

	Configuration 1		Configuration 2		
	Act. 3	Act. 5	Act. 3	Act. 5	Act. 1
Max. tilt angle error φ_d [deg]	16.17	17.84	15.06	13.40	21.38
Time [s] for max. φ_d	0.522	0.447	0.513	0.471	0.387
Time [s] for $\varphi_d \leq 5[^\circ]$	0.918	0.795	0.933	0.858	0.768
Max. vertical vel. error $\lvert e_{v,z} \rvert$ [m/s]	0.319	0.386	0.326	0.355	0.359
Time [s] for max. $\lvert e_{v,z} \rvert$	0.516	0.486	0.444	0.621	0.408
Time [s] for $\lvert e_{v,z} \rvert \leq 0.25[m/s]$	0.750	0.780	0.750	0.855	0.765

Table 6.1: Failure during Hover Flight - Performance Comparison for different Failure Cases for the 2 Hexacopter Configurations

expected because of the location of the actuators and is the same for the two configurations. The same can be concluded for the roll and pitch rate responses. The maximum absolute error of roll and pitch rates are respectively $70[^\circ/s]$ and $79[^\circ/s]$. The maximum tilt angle error $\varphi_d(t)$ varies between $13.4 - 21.35[^\circ]$ and the time after which the error is less than $5[^\circ]$ lies between $0.768 - 0.933[s]$.

The main difference between the configurations can be seen in the yaw rate since the direction of propellers 3 and 4 is exchanged. Therefore, the failure case of actuator 3 leads to a positive yaw rate in configuration 1 and to a negative yaw rate in configuration 2. The failure case of actuator 5 leads to a positive yaw rate for both configurations. Furthermore, for both failure cases in configuration 1 the yaw rate error has a slight increase in the while in configuration 2 it is driven to zero. The failure case of actuator 1 in configuration 2 leads to a significantly larger yaw rate error than in the other cases. This is expected from the ACS analysis in Chapter 2. This follows from Figures 2.4 and 2.5. From Figure 2.4 it is known that given any actuator failure in configuration 1, yaw moment at a hover state can be excerted only in one direction. From Figure 2.5 it is known in the failure cases of actuator 3 and 5 in configuration 2, yaw moment at a hover state can be excerted both directions and therefore yaw control is not lost. A failure of actuator 1 in configuration 2 leads to a strictly positive yaw moment at a hover state with a relatively high minimum. The response to the disturbance in the vertical velocity is similar for all the test cases and the maximum vertical velocity error $\lvert e_{v,z}(t) \rvert$ lies within $0.32 - 0.39[m/s]$. Since total thrust $T(t)$ reduces due to the failure, the vertical velocity error $e_{v,z}(t)$ is positive in all cases. In order to quantify the recovery performance of the controller, various characteristic values of the tilt angle error $\varphi(t)$ and the vertical velocity error $e_{v,z}(t)$ are given in Table 6.1. In summary, it has been shown that the presented controller maintains a similar performance for all possible failure scenarios exploiting the physically achievable moments and forces for best results. The time needed for recovery after the failure occurs is less than $1[s]$.

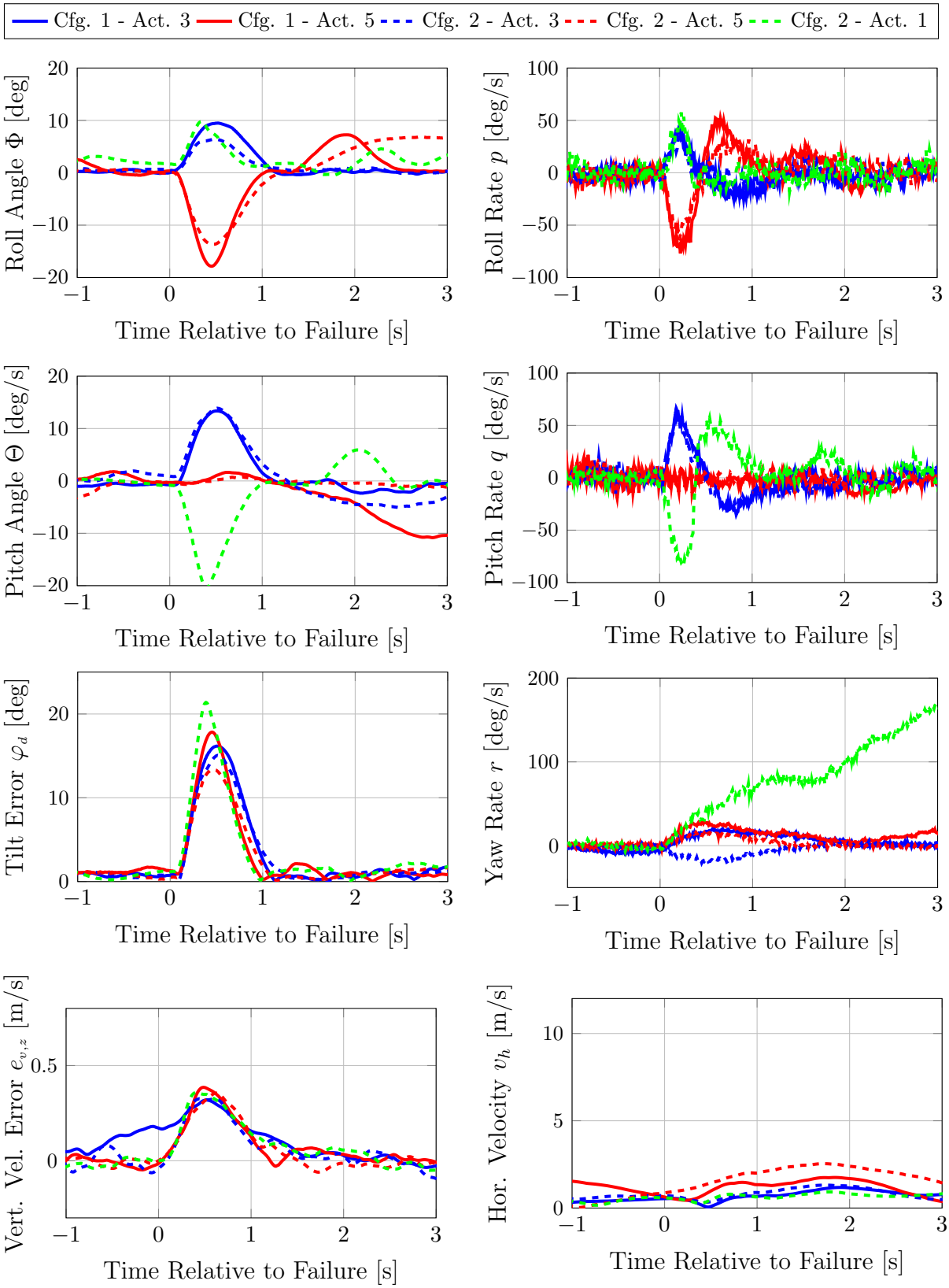


Figure 6.11: Failure during Hover Flight - Performance Comparison for different Configurations

6.3 Failures during High Speed Cruise

One of the main contributions of the thesis is the test of actuator failures during high speed maneuvers. This has never been presented before and poses several new challenges. The most noticeable changes on the dynamics compared to hover flight arise from the aerodynamic forces and the position of the center of gravity. The aerodynamic forces are proportional to the squared aerodynamic velocity and therefore significantly increase at higher speeds. Some effects that are not take into account in the control model are the aerodynamic drag forces and the effect of aerodynamic velocity on the propulsion units. Shifts of center of gravity along the body-fixed z_B-axis induce disturbance moments which are zero at hover and increase with the tilt angle. These moments therefore increase with increasing horizontal velocities since the aircraft needs to tilt for accelerating and maintaining the speed.

Figure 6.12 shows a comparison of the compensation of a total failure at different horizontal velocities for configuration 1. In order to emulate a total failure during flights, the failure is induced by an idle command to the corresponding ESC regardless of the commands of the controller. For all the flights, the time has been shifted such that the time of the induced failure is zero. Each of the different velocities is represented with a different color and correspond to $v_h = 0.5[m/s]$ (hover), $v_h = 4.9[m/s]$ and $v_h = 10.9[m/s]$. For analyzing the attitude error, the Euler angles and the tilt angle error $\varphi_d(t) \in [0, \pi]$ between $\vec{\mathbf{z}}_d(t)$ and $\vec{\mathbf{z}}_B(t)$ are used. Therefore the desired values of the roll $\Phi_d(t)$ and pitch angles $\Theta_d(t)$ are computed from (6.1). Furthermore, during the tests the attitude command $\vec{\mathbf{z}}_B(t)$ after the failure was such that the system was slowed down as can be seen in the horizontal velocity $v_h(t)$ plot.

In Figure 6.12 it can be seen that the attitude command during cruise flight is not equal zero as in the hover case. Nevertheless, a failure of actuator 1 leads to a positive roll angle error and a negative pitch angle error and a failure of actuator 3 leads to a positive roll angle error and a positive pitch angle error. This behavior is expected because of the location of the actuators. The maximum tilt angle error $\varphi_d(t)$ varies between $16.17 - 22.12[^\circ]$ and the time after which the error is less than $5[^\circ]$ lies between $0.771 - 1.044[s]$. The roll and pitch rates $p(t)$ and $q(t)$ show two large amplitude peaks. One corresponds to the failure and the other one to the correction maneuver afterwards. It can be seen that the time of the correction is similar for the three cases but the amplitude of the peaks is larger for higher speeds. This is mainly due to the change in the commanded attitude after the failure. The yaw rate $r(t)$ for all three cases increases after the failure and for $v_h = 0.5[m/s]$ and for $v_h = 4.9[m/s]$ it goes back to zero. For the case $v_h = 10.9[m/s]$ it remains positive for longer time and it reduces to zero approximately $5[s]$ after the failure. Because the total thrust $T(t)$ reduces due to the failure, in all the cases there is a positive vertical velocity error $e_{v,z}(t)$ and the maximum value lies between $0.32 - 0.45[m/s]$. In order to quantify the recovery performance of the controller, various characteristic values of the tilt angle error $\varphi_d(t)$ and the vertical velocity error $e_{v,z}(t)$ are given in Table 6.2.

The control commands corresponding to the three test cases are plotted in Figures 6.13, 6.14 and 6.15. In all three cases the control command corresponding to the failed rotor saturates at the upper limit. The command corresponding to the propulsion unit located diagonally on the other side of the hexacopter is rapidly reduced and in two cases the lower limit is reached. Furthermore, the mean value of the other four actuator commands increases in order to compensate for the total thrust $T(t)$ loss. In summary,

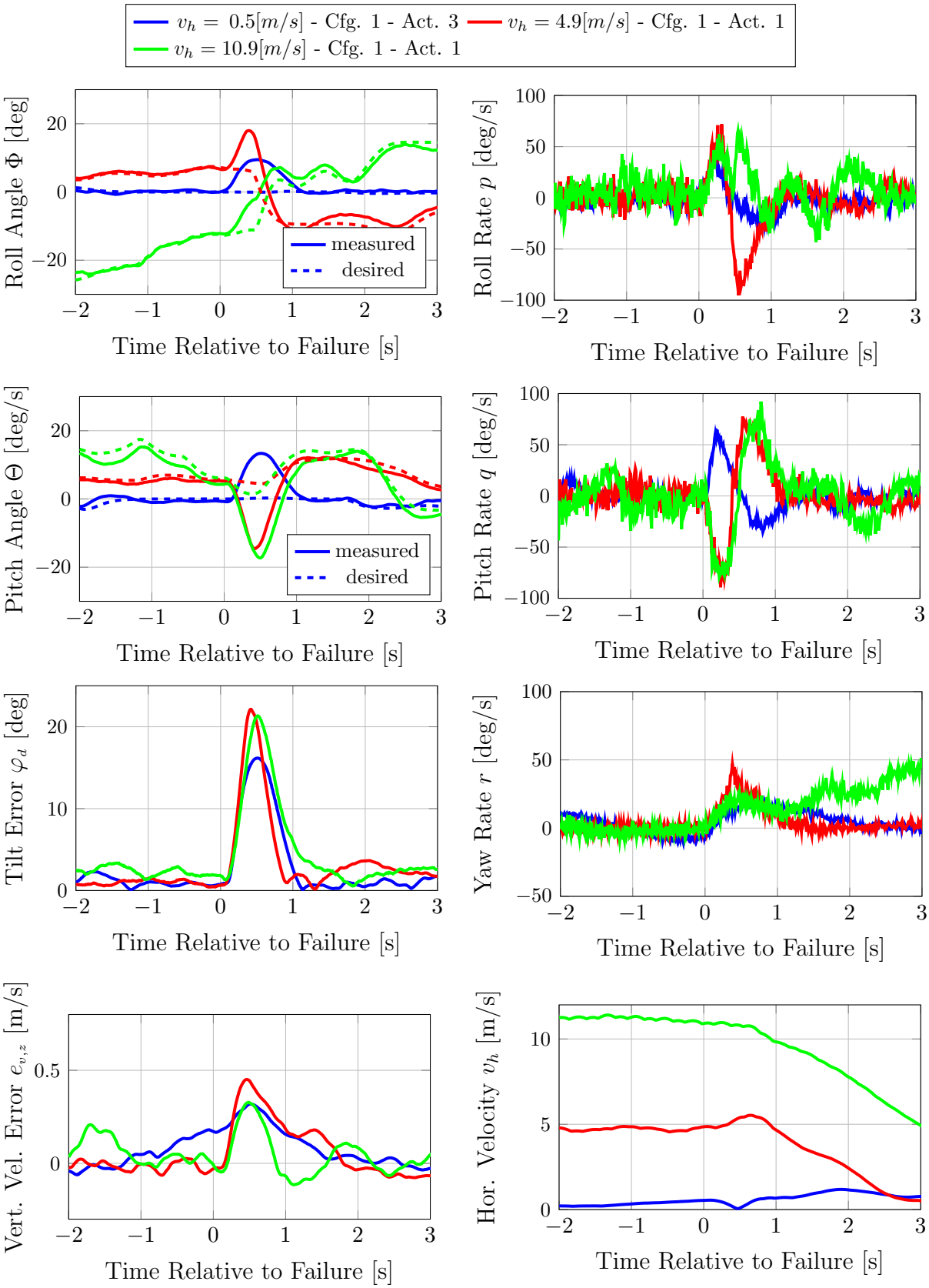


Figure 6.12: Failure during Cruise Flight - Performance Comparison for different Velocities

	Configuration 1		
	$v_h = 0.5[m/s]$ Act. 3	$v_h = 4.9[m/s]$ Act. 1	$v_h = 10.9[m/s]$ Act. 1
Max. Tilt Angle Error φ_d [deg]	16.17	22.12	21.34
Time [s] for Max. φ_d	0.522	0.423	0.507
Time [s] for $\varphi_d \leq 5[^\circ]$	0.918	0.771	1.044
Max. Vertical Vel. Error $\lvert e_{v,z} \rvert$ [m/s]	0.319	0.451	0.328
Time [s] for Max. $\lvert e_{v,z} \rvert$	0.516	0.462	0.495
Time [s] for $\lvert e_{v,z} \rvert \leq 0.25[m/s]$	0.750	0.852	0.654

Table 6.2: Failure during Cruise Flight - Performance Comparison for different Velocities

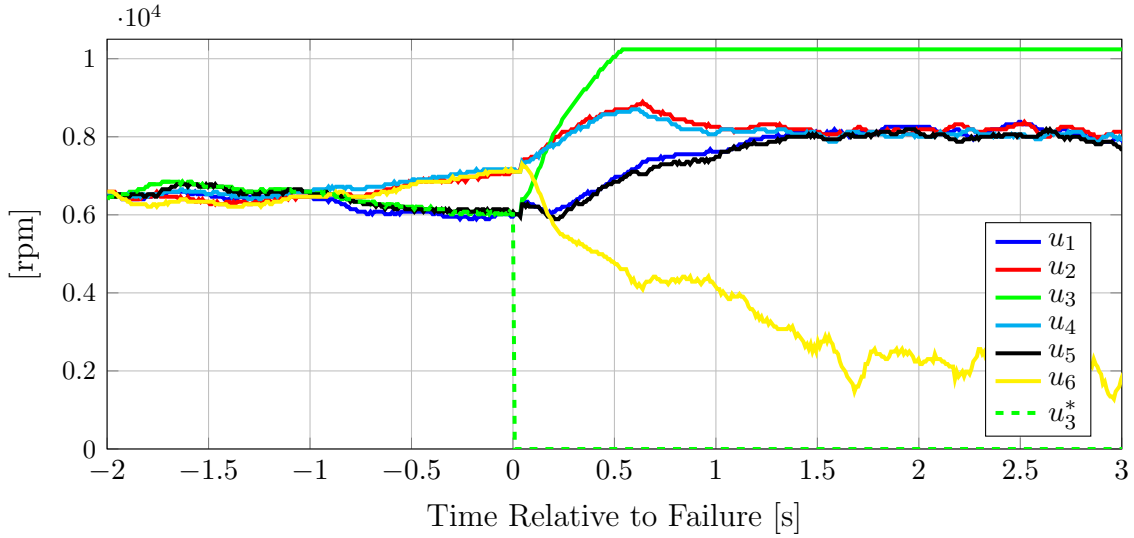


Figure 6.13: Control Inputs - Fault Case $\lambda_3 = 0$ during Cruise Flight $v_h = 0.5[m/s]$ - Configuration 1

it has been demonstrated that the P-ACA strategy is able to compensate for unknown actuator failures during high speed maneuvers. This highlights the robustness of the selected approach against unmodeled dynamics.

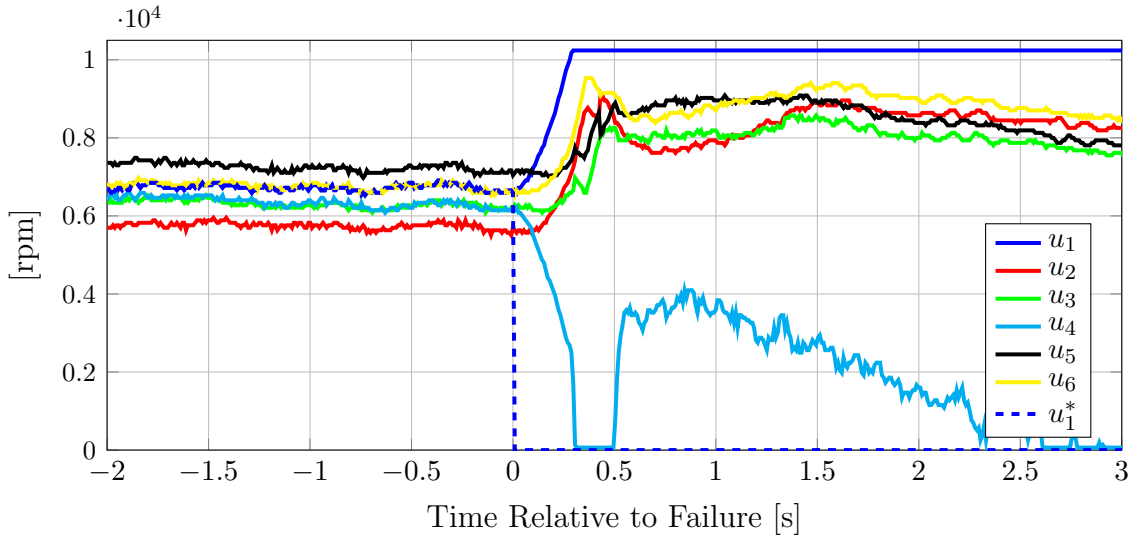


Figure 6.14: Control Inputs - Fault Case $\lambda_1 = 0$ during Cruise Flight $v_h = 4.9[m/s]$ - Configuration 1

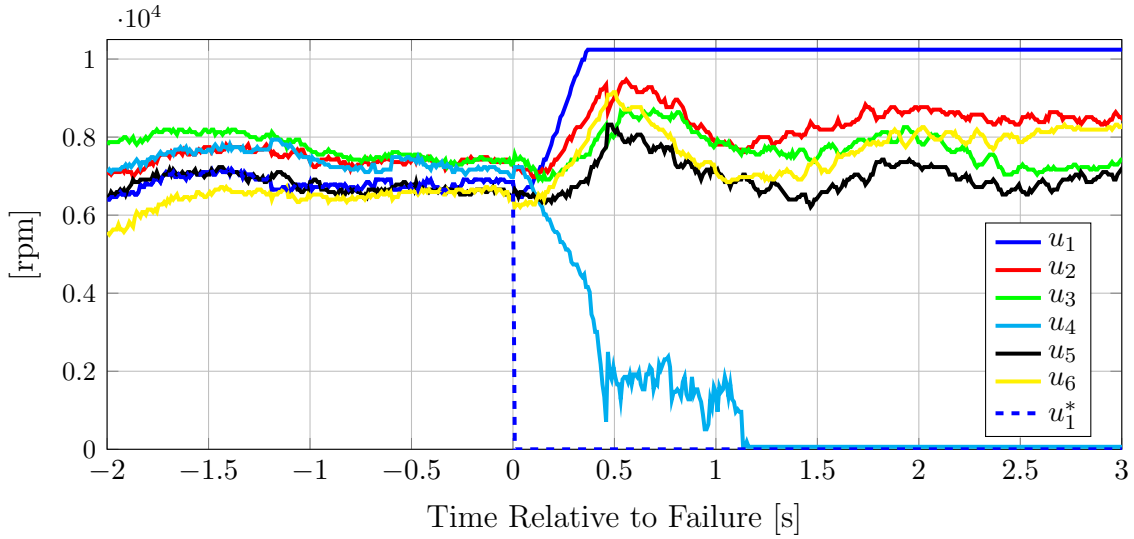


Figure 6.15: Control Inputs - Fault Case $\lambda_1 = 0$ during Cruise Flight $v_h = 10.9[m/s]$ - Configuration 1

6.4 Post-Failure Performance Analysis

As important as the immediate compensation of the actuator faults is the tracking performance during flight maneuvers after the failure occurs. Figure 6.16 shows a performance comparison of three fault scenarios during the execution the same maneuver. The three scenarios are the nominal case i.e. with no degradation, a software induced failure of actuator 5 (SW) and a hardware induced failure of actuator 1 (HW). In the figure, each of the scenarios corresponds to a color. The software induced failure corresponds to an idle command to the corresponding ESC regardless of the commands of the controller as has been done in the previous sections. The hardware induced failure corresponds to a missing propeller during the whole flight. In the figure the time has been shifted such that the start time of the maneuver is zero for all the flights. All the results shown in the figure correspond the hexacopter configuration 1.

In Figure 6.16 the maneuver corresponds to a 2-2-1-1 maneuver. In a 2-2-1-1 maneuver, the command remains constant at the amplitude α_0 for the first 2 seconds, then at $-\alpha_0$ for the next 2 seconds, then at α_0 for another second, and finally at $-\alpha_0$ for the last second. At the end the command returns to zero. The presented maneuver is commanded in the $(z_{c,y})_N(t)$ channel with the amplitude $\alpha_0 = 0.5$ which is equivalent to a tilt angle $\varphi(t)$ of $30[^\circ]$. Since the heading angle $\Psi(t)$ is not the same for the different flights, the desired Euler angles (6.1) differ and therefore the comparison plot uses the reduced attitude vector parameterization. The attitude command is $(\vec{\mathbf{z}}_c)_N(t)$, the desired attitude is $(\vec{\mathbf{z}}_d)_N(t)$ and the measured attitude corresponds to $(\vec{\mathbf{z}}_B)_N(t)$. In Figure 6.16 it can be seen that in all three cases the tracking in the commanded direction $(z_y)_N(t)$ is very good and does not couple with $(z_{B,x})_N(t)$. In the actuator 5 fault case, the maneuver could not be completed do to space constraints and therefore the desired attitude $(z_{d,y})_N(t)$ differs from the other cases during the last second. Nevertheless, the tracking performance is maintained. In order to get a measure of the tilting, the tilt angle $\varphi(t)$ and the tilt angle error $\varphi_d(t)$ are depicted. The tilt angle $\varphi(t)$ is the angle between $\vec{\mathbf{z}}_B(t)$ and $\vec{\mathbf{z}}_N(t)$ and the tilt angle error $\varphi_d(t)$ is the angle between $\vec{\mathbf{z}}_d(t)$ and $\vec{\mathbf{z}}_B(t)$. It can be seen that the changes in the tilt angle $\varphi(t)$ occur fast every time the command changes the sign. From the rate plot, it can be seen that roll $p(t)$ or pitch rates $q(t)$ larger than $100[deg/s]$ are necessary. The maximum tilt angle error $\varphi_d(t)$ varies between $8.92 - 9.74[deg]$ and the mean varies between $2.95 - 3.51[deg]$. The desired yaw rate $r_d(t)$ during the maneuver is zero but it can be seen that in the two failure cases the yaw error cannot always be reduced to zero due to the physical limitations of configuration 1. The yaw rate tracking performance differs significantly because the yaw authority largely depends on the desired roll and pitch moments. These are different for every flight since they depend on the current heading. The maximum absolute vertical velocity error $|e_{v,z}(t)|$ is $0.66[m/s]$ in the nominal case and approximately $1[m/s]$ for both failure cases. The resulting horizontal velocity $v_h(t)$ depends on the initial velocity $(\vec{\mathbf{v}})_N^N(t)$ and the combination of the reduced attitude vector $\vec{\mathbf{z}}_B(t)$ and the thrust $T(t)$.

In the presented failure cases the performance largely depends on the control allocation. This can be assessed by comparing the desired virtual controls $\boldsymbol{\nu}_d(t)$ and the estimated virtual controls $\hat{\boldsymbol{\nu}}(t)$ which are shown in Figure 6.17. In the figure, the left column corresponds to the failure of actuator 5 and the right column to the failure of actuator 1. It can be seen that in both cases the roll $L(t)$ and pitch moments $M(t)$ are correctly allocated as expected from the Prioritizing Control Allocation (PRIO CA).

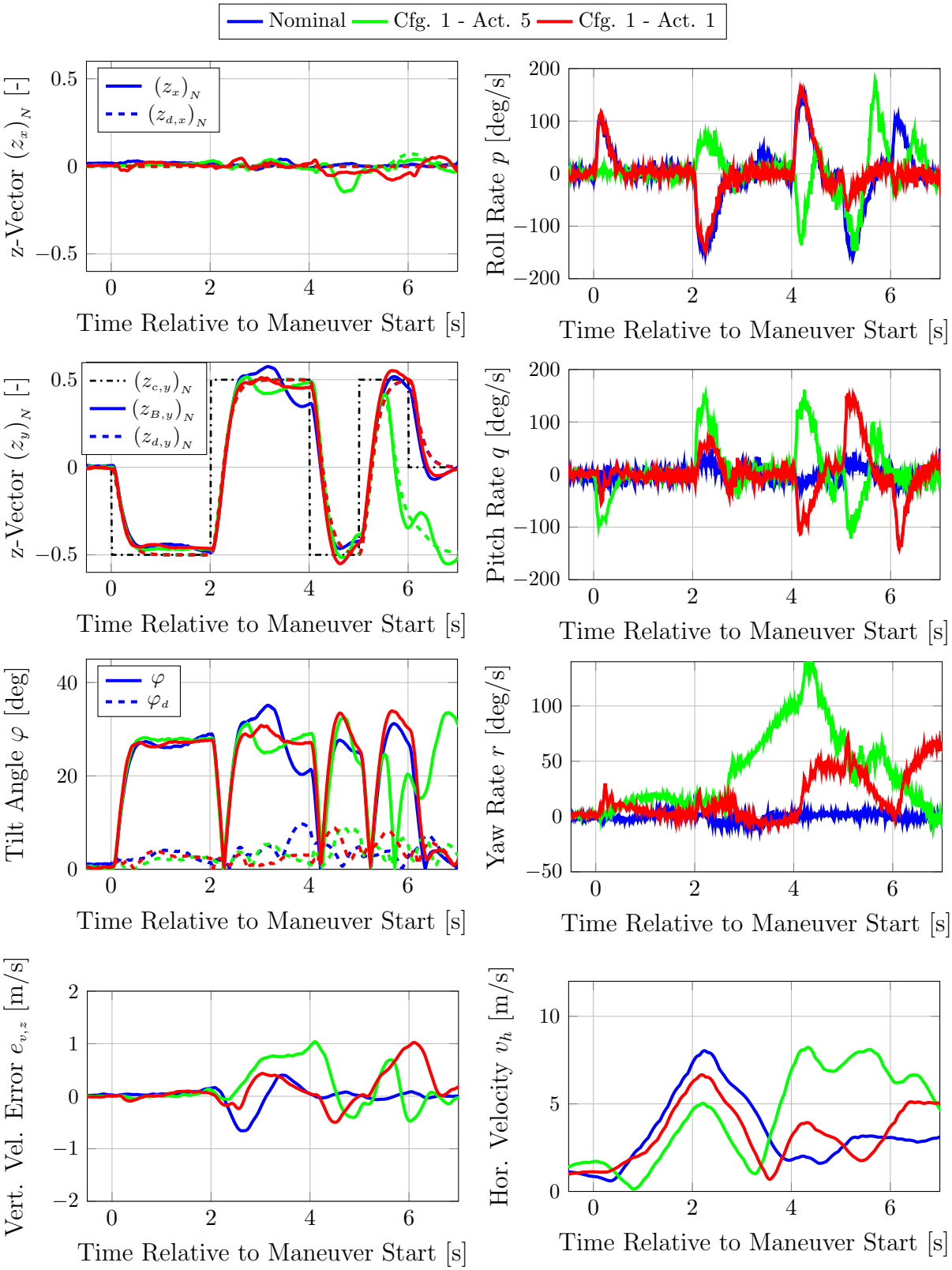


Figure 6.16: Tracking Performance Comparison for different Failure Cases - Maneuver Flight - Configuration 1

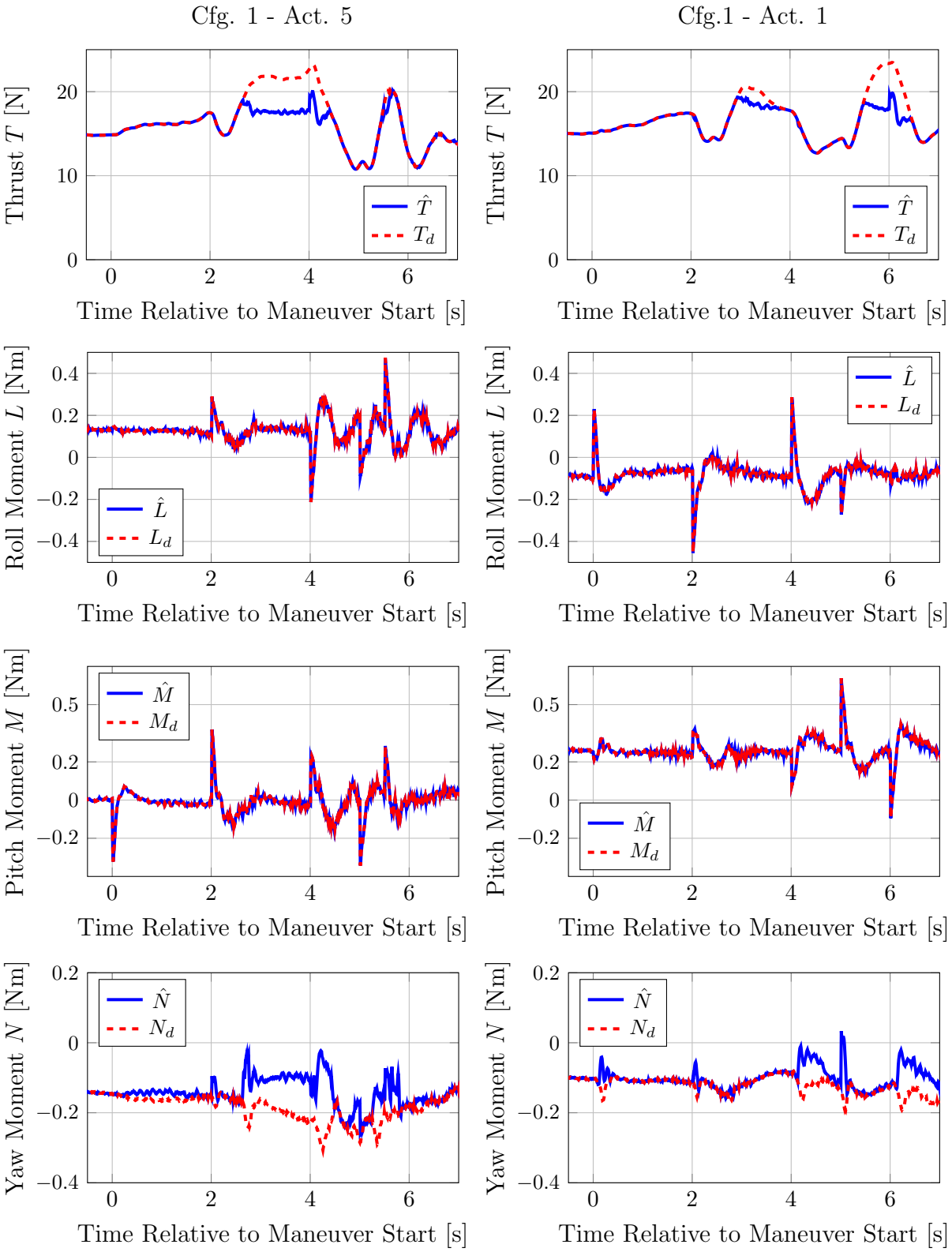
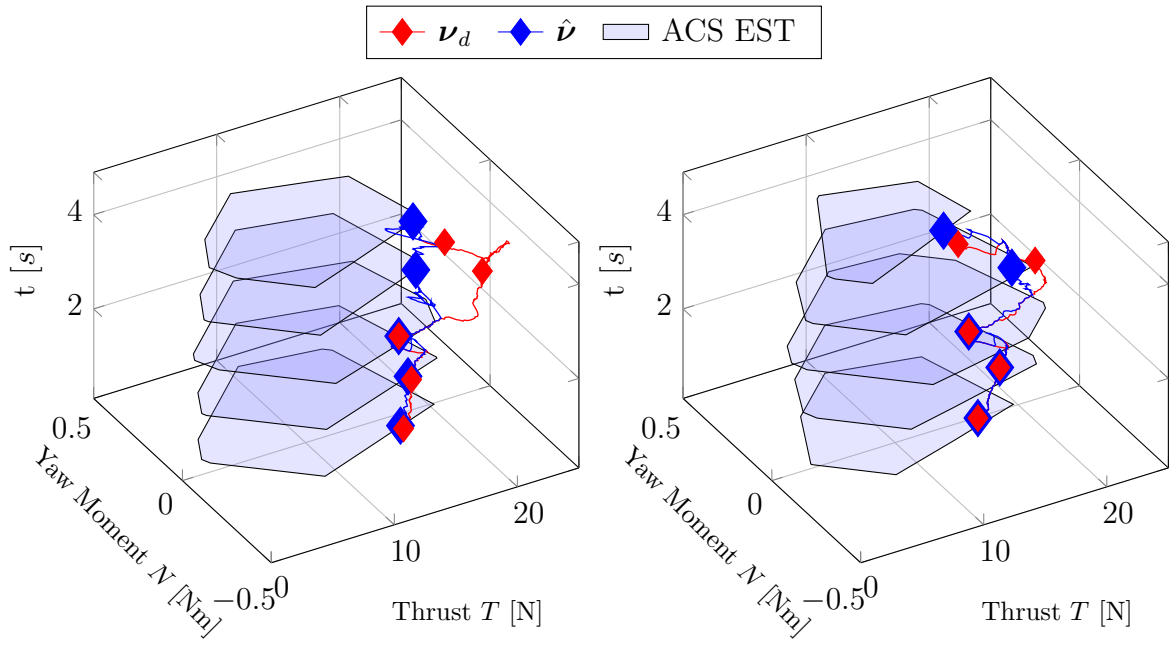


Figure 6.17: Control Allocation Performance Comparison for different Failure Cases - Maneuver Flight - Configuration 1



(a) Cfg. 1 - Act. 5

(b) Cfg. 1 - Act. 1

Figure 6.18: T/N Estimated Virtual Control Space for $t = [0.5, 1.5, 2.5, 3.5, 4.5][s]$ - Maneuver Flight - Configuration 1

Further, there are time intervals where the yaw moment $N(t)$ and the thrust $T(t)$ cannot be correctly allocated. These time intervals match with the tracking errors $e_{w,z}(t)$ and $e_{v,z}(t)$ in Figure 6.16. For example, in the actuator 5 failure case between 2.5 and $4.5[s]$, the thrust allocated is too low and the yaw moment is greater than the desired one. This translates to a increasing tracking errors in both yaw and vertical velocity.

To understand the control allocation errors, the T/N-plane of estimated attainable virtual control space (ACS EST) (6.5) is plotted for $t = [0.5, 1.5, 2.5, 3.5, 4.5][s]$ in Figure 6.18. The cut of the estimated ACS in the T/N-plane depends on the allocated roll $\hat{L}(t)$ and pitch moments $\hat{M}(t)$. Therefore, depending on the maneuver, different limits hold for the low priority virtual controls. As detailed in Section 4.3 (Prioritizing Control Allocation Strategy (PRIO CA)), the control allocation errors mainly arise from unattainable desired virtual controls $\boldsymbol{\nu}_d(t)$ as it is shown in Figure 6.18. It can be concluded that the prioritization strategy has been effective, since high priority virtual controls have been prioritized over low priority ones. Because the ACS is different for each actuator failure, the response in yaw and vertical velocity tracking will differ between them. In order to quantify the recovery performance of the controller, various characteristic values of the tilt angle error $\varphi_d(t)$, the vertical velocity error $e_{v,z}(t)$ and the yaw rate error $e_{w,z}$ are given in Table 6.3.

Figure 6.19 shows the performance of the controller for a larger time span and for a flight at higher speeds. It corresponds to the hexacopter configuration 1 and a software induced failure of actuator 1. It is the same high speed case already analyzed in Figure 6.12. In the plot the time has been shifted such that $t = 0$ corresponds to the failure time. The corresponding position trajectory $(\vec{\mathbf{r}}^R)_N(t)$ of the flight can be seen in Figure 6.20. It can be seen that in the horizontal plane, two similar patterns were flown: one before and one after the failure. In the vertical plane the altitude increased

	Configuration 1		
	Nominal	Act. 5 (SW)	Act. 1 (HW)
Max. Tilt Angle Error φ_d [deg]	9.74	9.07	8.92
Mean Tilt Angle Error φ_d [deg]	3.52	3.04	3.16
Std. Tilt Angle Error φ_d [deg]	1.95	2.15	2.04
Max. Vert. Vel. Error $\lvert e_v \rvert$ [m/s]	0.66	1.04	0.96
Mean Vert. Vel. Error e_v [m/s]	0.00	0.24	0.09
Std. Vert. Vel. Error e_v [m/s]	0.21	0.39	0.28
Max. Yaw Rate Error $\lvert e_{w,z} \rvert$ [deg/s]	13.92	141.11	73.37
Mean Yaw Rate Error $e_{w,z}$ [deg/s]	2.87	46.05	15.13
Std. Yaw Rate Error $e_{w,z}$ [deg/s]	2.25	36.78	16.15

Table 6.3: Maneuver Flight - Performance Comparison for different Failure Cases

during the flight. In Figure 6.19, the reduced attitude vector $\vec{\mathbf{z}}_B(t)$ is used as attitude parameterization and the tilt angle $\varphi(t)$ and the tilt angle error $\varphi_d(t)$ are depicted. The maximum tilt angle $\varphi(t)$ is $27.15\ [deg]$ before the failure and $17.42\ [deg]$ after the failure. The tilt angle error $\varphi_d(t)$ is most of the time very small. The mean error is $1.22\ [deg]$ before the failure and $1.72\ [deg]$ after the failure. There are three prominent peaks after the failure at $t = 0.52[s]$, $t = 16.86[s]$ and $t = 40.57[s]$. The first peak directly arises from the unknown failure condition. The other two peaks correspond to the times where the yaw tracking error $e_{w,z}(t)$ increases. The large errors are all positive since the failed actuator is number 1. The maximum yaw tracking error $e_{w,z}(t)$ is $74.14\ [deg/s]$. Further, it can be seen that roll $p(t)$ and pitch rates $q(t)$ smaller than $100\ [deg/s]$ are necessary to follow the commands. The vertical velocity tracking can also be seen in Figure 6.19. The vertical velocity tracking error $e_{v,z}(t)$ has a maximum absolute value of $0.31\ [m/s]$ before the failure and $0.46\ [m/s]$ after the failure. The resulting horizontal velocity $v_h(t)$ shows an acceleration phase up to $11.87\ [m/s]$ before the failure and a fast deceleration after it. This is followed by a subsequent acceleration up to $13.54\ [m/s]$, a cruise flight phase at approximately $8.5\ [m/s]$ and finally another deceleration phase. In order to quantify the performance of the controller before and after the failure, various characteristic values of the tilt angle error $\varphi_d(t)$, the vertical velocity error $e_{v,z}(t)$ and the yaw rate error $e_{w,z}$ are given in Table 6.4.

Figure 6.21 shows the desired virtual controls $\boldsymbol{\nu}_d(t)$ and the estimated virtual controls $\hat{\boldsymbol{\nu}}(t)$. Before the failure, all the virtual controls are correctly allocated. After the failure, the priorization strategy is used whenever input saturation is encountered. The maximum absolute value of the control allocation error after the failure is $[0.02, 0.00, 0.00, 0.12]^T$ and the mean is $[2, 0, 0, -6]^T \cdot 10^{-3}$ (with the order (T,L,M,N)). The largest performance degradation corresponds to the yaw rate tracking. It can be seen that whenever the yaw moment is allocated incorrectly, the yaw tracking error $e_{w,z}(t)$ increases. Finally, the control input commands $\mathbf{u}(t)$ and the degraded value of actuator 1 $u_1^*(t)$ are depicted in Figure 6.22. It can be seen that the main limitation for correctly allocating the yaw moment $N(t)$ is the lower saturation of the control input $u_4(t)$.

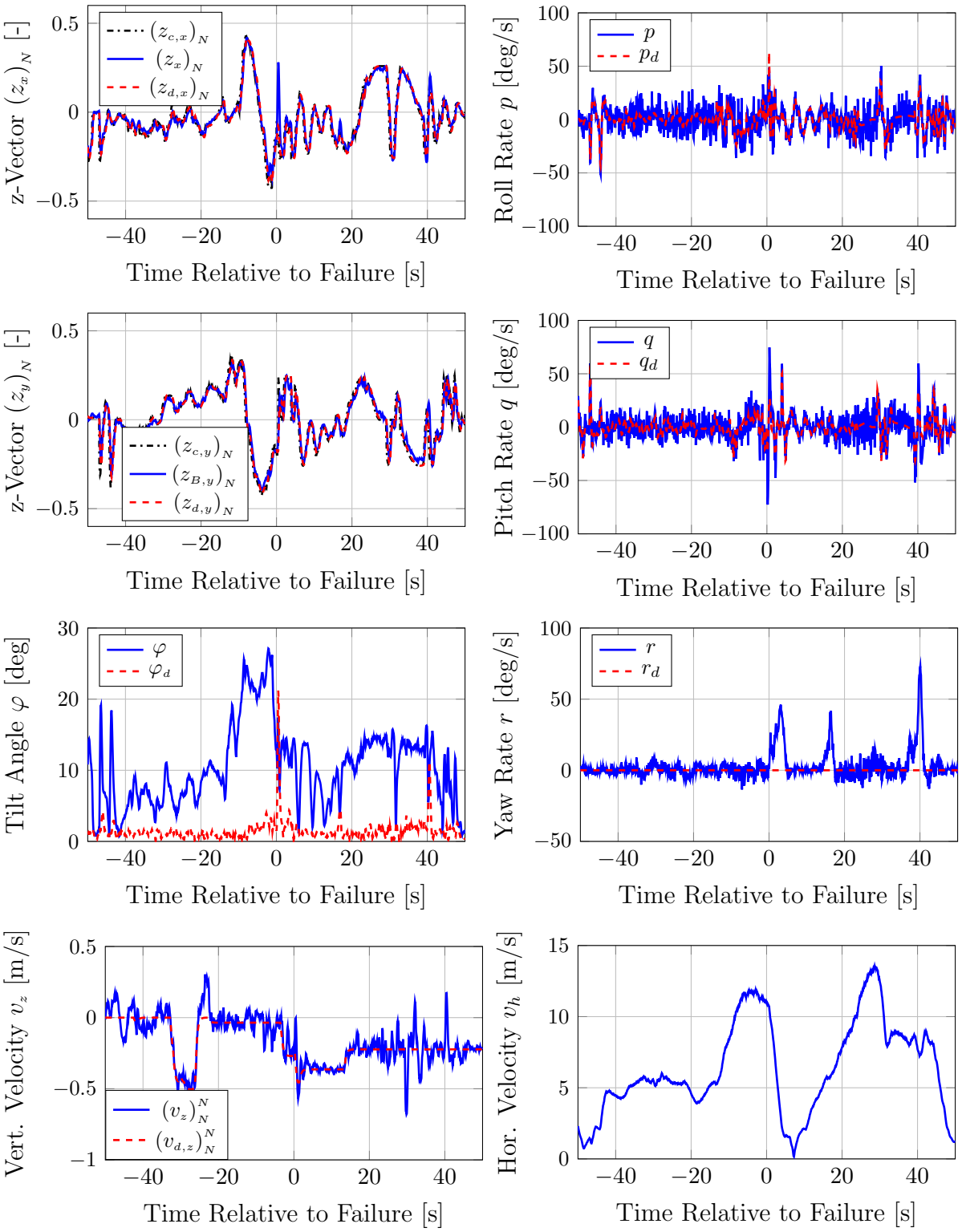


Figure 6.19: Tracking Performance - Fault Case $\lambda_1 = 0$ during High-Speed Flight - Configuration 1

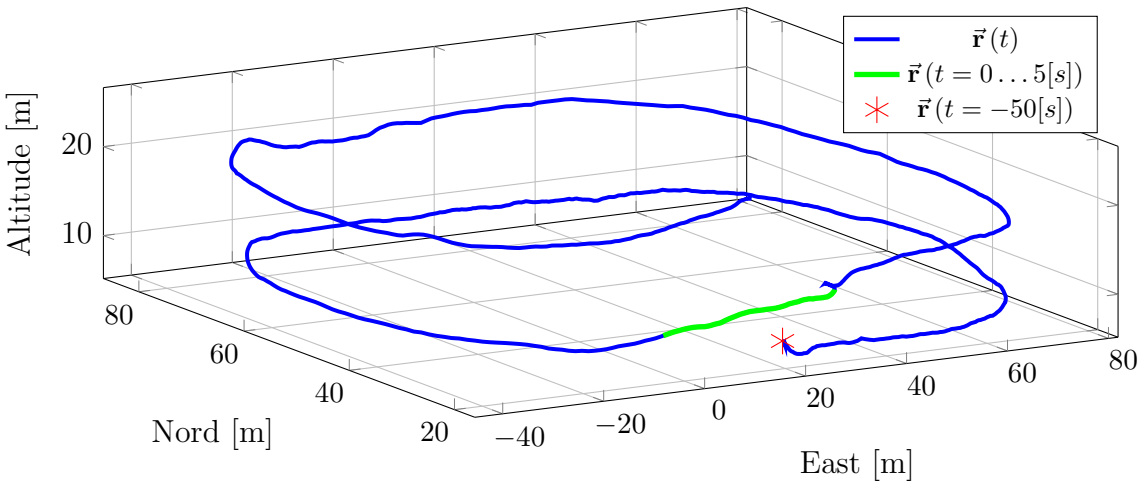


Figure 6.20: Position Trajectory - Fault Case $\lambda_1 = 0$ during High-Speed Flight - Configuration 1

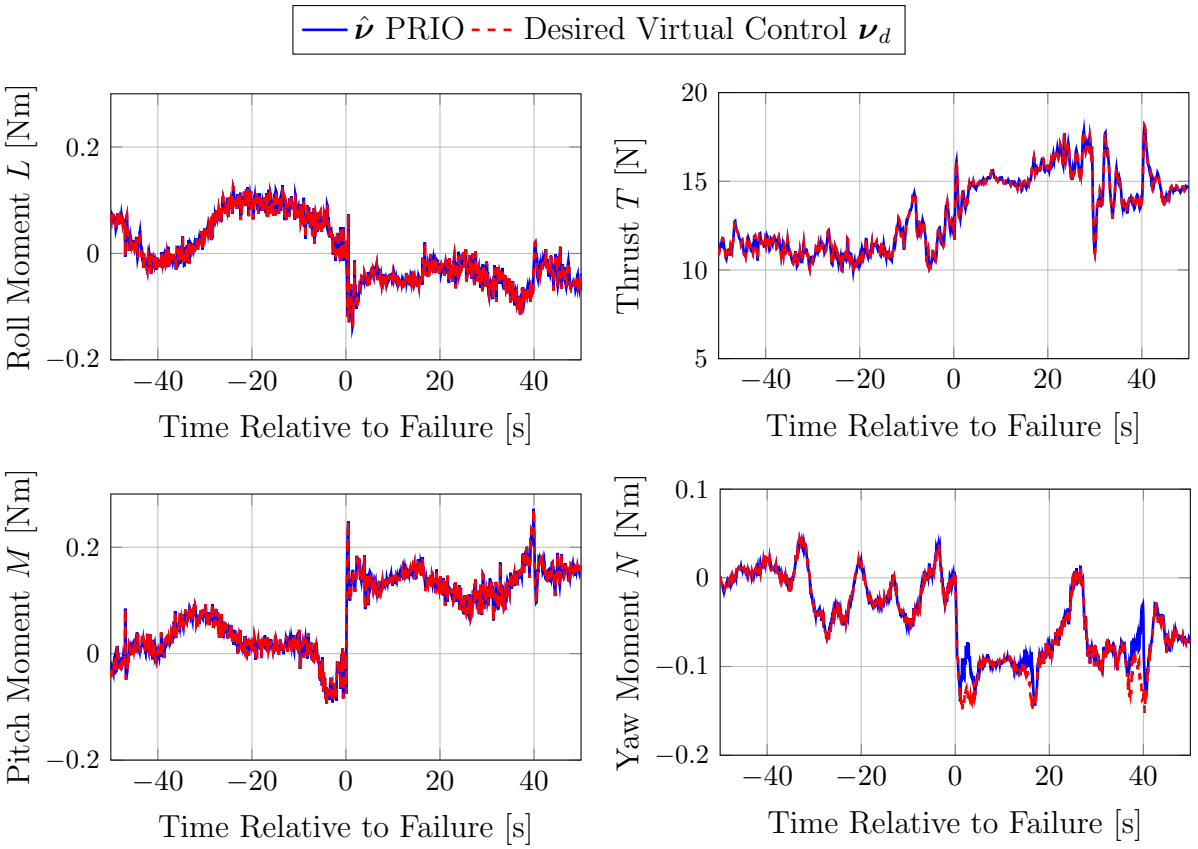


Figure 6.21: Control Allocation Performance - Fault Case $\lambda_1 = 0$ during High-Speed Flight - Configuration 1

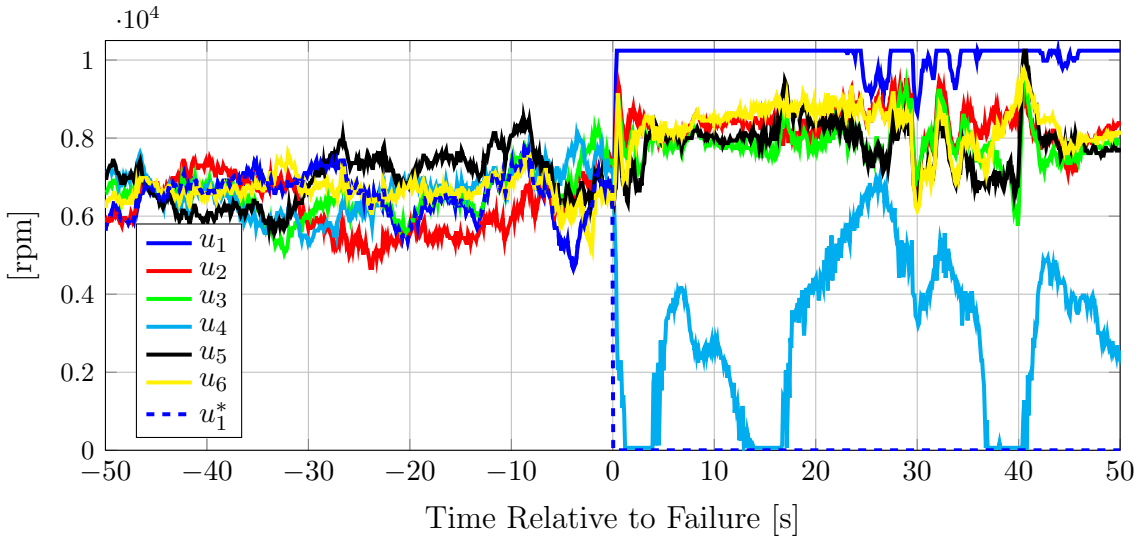


Figure 6.22: Control Inputs - Fault Case $\lambda_1 = 0$ during High Speed Flight - Config. 1

	Configuration 1	
	Nominal	Act. 1 (SW)
Max. Tilt Angle Error φ_d [deg]	4.28	21.34
Mean Tilt Angle Error φ_d [deg]	1.22	1.72
Std. Tilt Angle Error φ_d [deg]	0.71	2.08
Max. Vert. Vel. Error $\lvert e_v \rvert$ [m/s]	0.31	0.46
Mean Vert. Vel. Error e_v [m/s]	0.02	0.00
Std. Vert. Vel. Error e_v [m/s]	0.08	0.09
Max. Yaw Rate Error $\lvert e_{w,z} \rvert$ [deg/s]	15.57	74.14
Mean Yaw Rate Error $e_{w,z}$ [deg/s]	2.70	8.11
Std. Yaw Rate Error $e_{w,z}$ [deg/s]	2.23	12.05

Table 6.4: High-Speed Flight - Performance Comparison before and after the Total Failure of Actuator 1

6.5 Bibliographical Remarks

For analyzing the flight test results with respect to the state of the art, the scenario of a full failure during hover flight is chosen. The maximum tilt angle error $\varphi_d(t)$ and the corresponding time of occurrence are chosen as a comparison measure and depicted in Table 6.5. The results are inferred from the data in the cited works and are therefore approximated values. Furthermore, since the systems have different inertia and actuator characteristics, the comparison has rather an informative character. Table 6.5 considers the experimental results including the full actuator failure cases from Tables 1.2 and 1.3 with the exception of [1, 60, 103, 114]. In [1] a discrete change of control allocation with knowledge of the failure is presented and therefore no transient error occurs. In [103] there is no data showing the performance of the controller. In [60, 114] the failure does not occur in flight but before taking off. In Table 6.5 the test cases are classified into full and degraded authority. Furthermore, it is specified if a fault detection and isolation algorithm (FDI) is needed and existing (✓) or if it is only assumed (x). Otherwise, the FDI is not explicitly needed as is the case for the controller presented in this thesis.

In Table 6.5 it can be seen that the time of the maximum tilt error is in the range $0.35 - 0.63\ [s]$. Furthermore, only [106] and [154] achieve clearly smaller maximum errors $|\varphi_d(t)|$. This can be explained as follows. In [106] the failure of 2 rotors which are symmetrically and simultaneously shut off is shown. Due to the symmetry, the attitude transient error is expected to be smaller. Furthermore, no FDI was implemented. In [154] the FDI module is based on the actuator and not the rigid body model and therefore the failure can be detected faster. Hence, it is concluded that the controller analyzed in this chapter achieves a very good performance which could be improved by using fault information coming directly from the actuator subsystem. Furthermore, for the degraded authority case the author has slightly improved the performance compared to previous work [42, 43], where the hexacopter configuration 1 has been presented. Compared to the other degraded authority case result in [34] the performance is clearly better.

The results presented in this chapter include a failure case during high speed flight at $10.9\ [m/s]$ that shows the robustness of the algorithms. To the best of the author's knowledge, this is the first time that this achievement has been presented. Finally, the post-failure performance analysis also includes fast maneuvers that have not been yet reported during failure scenarios. On the one hand, fast step attitude commands were tracked with rates surpassing well over $100\ [deg/s]$. On the other hand, an acceleration in cruise flight from 0 up to $13.54\ [m/s]$ was shown.

	[34]	[42]	[147]	[154]	[43]	[106]
Max. tilt angle error φ_d [deg]	41.5	18	20.8	9	18.12	9.9
Time [s] for max. φ_d	0.55	0.5	0.35	0.43	0.36	0.63
Failure (full authority)			✓	✓		✓
Failure (degraded authority)	✓	✓			✓	
FDI	x		✓	✓		x

(a) Literature Review

	Configuration 1		Configuration 2		
	Act. 3	Act. 5	Act. 3	Act. 5	Act. 1
Max. tilt angle error φ_d [deg]	16.17	17.84	15.06	13.40	21.38
Time [s] for max. φ_d	0.522	0.447	0.513	0.471	0.387
Failure (full authority)			✓	✓	
Failure (degraded authority)	✓	✓			✓
FDI					

(b) Thesis Results from Table 6.1

Table 6.5: Failure during Hover Flight - Performance Comparison with existing Literature

Chapter 7

Conclusion

This thesis addresses the challenge of developing an adaptive fault tolerant controller for a VTOL multicopter system that considers actuator redundancy and actuator limits and implements a strategy for degraded control authority. In Chapter 2 (Multirotor Dynamics Model) the hexacopter system architecture was presented and the control dynamics model was derived. From the analysis of the attainable control set (ACS) it became clear that a strategy for degraded control authority is necessary. Subsequently in Chapter 3 (Adaptive Fault Tolerant Control), adaptive control methods were systematically analyzed for systems with redundant actuators. Direct and indirect MRAC approaches were examined with respect to the conditions that need to be met in order to solve the actuator failure problem. The Parameter Reduction due to Overactuation (PRO) approach was proposed in order to reduce the number of adaptive parameters and simplify the matching conditions. This allowed for the definition of a reduced control effectiveness matrix which formed the basis for developing the control authority conditions for redundant systems.

In Chapter 4 (Adaptive Control Allocation), predictor-based MRAC was used as a basis to formulate a framework that can integrate control allocation algorithms: Predictor Based Adaptive Control Allocation (P-ACA). In this way, actuator redundancy and limits are directly addressed by the control allocation problem which is solved separately from the rigid body control and the parameter adaptation. The strategy for degraded control authority is implemented by the proposed Prioritizing Control Allocation (PRIO CA). It is able to exploit the overactuation in order to allocate the virtual controls in the selected priority order. In order to maintain the real-time capabilities it is combined with the presented SVD update algorithm.

Based on these methods, the problems of attitude and vertical velocity control of the hexacopter system have been addressed in Chapter 5 (Multirotor Controllers). In order to maximize the utilizable flight envelope and to facilitate the prioritization strategy in the case of degraded control, the reduced attitude vector is selected as an attitude parameter. The controllers have been developed such that they can be separated into a baseline controller and an adaptive augmentation. The corresponding stability assessment has been presented for the adaptive fault tolerant controller. During degraded control authority the control allocation prioritizes the virtual controls in the following order: roll and pitch moments, thrust and yaw moment.

Finally, the results of flight tests were presented in Chapter 6 (Experimental Results). During the experiments, the faults are unknown to the controller, all computations are done onboard, and no external sensors were used. Furthermore, a single

unified controller handled the nominal and all the faulty cases. The first tests correspond to 5 different total failure cases during hover flight covering the two hexacopter configurations. Hence, they included full and degraded authority control cases. It was demonstrated that the results in roll, pitch and vertical velocity were repeatable regardless of the configuration and failure case. The yaw rate tracking authority depended on the physical limitations of the specific failure case. Further experiments showed good post-failure performance during maneuver flight with rotational rates surpassing well over $100\ [deg/s]$. The robustness of the approach was validated during high speed cruise flight with a failure induced at a velocity of $10.9[m/s]$. After the failure the system was able to recover performance and to show controlled high speed cruise flight.

7.1 Contributions

In the following the contributions of this thesis are revised.

Parameter Reduction due to Overactuation (PRO)

It has been shown that for overactuated systems it is possible to reduce the set of adaptive parameters without any information loss. This is possible if the parameter update law has the form in (3.49), where the transposed input matrix $\mathbf{B}_p^T$ is its left most element. In the MRAC approaches, this is true for estimates of the control effectiveness matrix $\hat{\mathbf{\Lambda}}\,(t)$. The PRO approach improves the usage of computational resources without affecting the controller performance. The initial work of this approach has been presented in [46, 47].

Control Authority Conditions for Overactuated Systems

Based on the new uncertainty parameterization (PRO), control authority conditions have been formulated for systems with redundant actuators such that actuator failure cases can be considered within the stability proofs of model reference adaptive control (MRAC). The conditions are derived for direct and indirect structures as presented in Theorem 3.22 (Fault Tolerant Control Authority Conditions). The result shows that the original assumptions regarding the control effectiveness matrix $\mathbf{\Lambda}$ can be replaced by using a reduced control effectiveness matrix $\mathbf{\Lambda}_{rr}$. Interesting is the result for a positive semi-definite control effectiveness matrix $\mathbf{\Lambda}$ which covers degradation of control effectiveness, actuator failures and stuck actuators as listed in Table 3.1. In this case, the conditions in Theorem 3.22 (Fault Tolerant Control Authority Conditions) are equivalent to Assumption 3.20 (Control Authority - Fault Case) and therefore the controller exploits the maximum physical capability of the plant. Initial work has been presented in [46, 47].

Predictor-Based Adaptive Control Allocation (P-ACA)

Based on Predictor-Based MRAC and the Parameter Reduction due to Overactuation (PRO), a framework for integrating control allocation algorithms has been presented.

In this Predictor-Based Adaptive Control Allocation (P-ACA) formulation, the rigid body control law is not overparameterized and the actuator redundancy is fully handled within the control allocation. The control allocation permits the consideration of actuator limits and a better use of the redundancy degrees of freedom. Furthermore, a reduced-order version of the approach is presented.

Prioritizing Control Allocation (PRIO CA)

In order to take actuator limits into account and to implement a strategy for degraded control authority for VTOL aircraft, the Prioritizing Control Allocation (PRIO CA) is developed. In the case of unattainable or suboptimal solutions, the algorithm prioritizes virtual controls in a given order such that control degradation is reduced to a minimum. This approach has the advantage of having a fixed maximum number of iterations to return a solution.

Extension of the SVD Udpate Algorithm to Non-Square Matrices

The Prioritizing Control Allocation (PRIO CA) needs the knowledge of the Singular Value Decomposition (SVD) of the input matrix. This has been achieved by using the SVD update algorithm presented in [69]. In this work, it has been extended to non-square matrices and a numerical drift correction term has been added.

Attitude Parameterization and Yaw Motion Decoupling

In order to maximize the utilizable flight envelope, the reduced attitude vector in $\mathbb{S}^2$ is used. The advantage is that non-uniqueness and singularities are avoided. Furthermore, the stability properties are shown decoupled from the yaw motion. This enhances the advantages of the Prioritizing Control Allocation (PRIO CA) during degraded control situations. This is the first time that this type of parameterization is analyzed using an adaptive controller. Non-adaptive controllers have been presented in [48, 51, 55, 149] and previous adaptive geometric results use a full attitude parameterization in $SO(3)$ [88, 93].

Experimental Validation

The presented controller is validated using the hexacopter testbed. During the experiments, the faults are unknown to the controller, all computations are done onboard, and no external sensors were used. Furthermore, a single unified controller handles the nominal and all the faulty cases. The test cases include two hexacopter configurations with representative failure cases for demonstrating repeatable results. The experiments include post-failure performance analysis and include software-induced and hardware failure scenarios.

Experimental Validation - High Speed

The test cases include hover flight as well as high speed cruise flight. These are the first results showing the failure scenario occurring during high speed cruise flight at

$10.9[m/s]$. This shows the robustness of the controller against unmodeled aerodynamic effects which increase overproportionally with respect to the airspeed.

7.2 Future work

Based on the results in this thesis, several new questions have been raised and further research topics are sketch in the following.

Direct MRAC - Consideration of Actuator Limits for Overactuated Systems

In the direct MRAC framework, one of the first ideas to counteract control magnitude saturation was the consideration of the control deficit within the reference model like in [85], where a single input system was considered. The positive μ-modification extends these results by guaranteeing that the control input will never saturate [90]. In [134] and [91], extensions for a multi-input plant were presented. A natural continuation for overactuated systems is the use of the new parameterization of the PRO approach to specifically address actuator redundancy and try to reduce conservatism of the enforced limits.

Output Feedback and Unmatched Uncertainties

The Control Task 3.1 (Model Reference Adaptive Control) has been restricted to state feedback. An extension to output feedback is a logical next step. Furthermore, since the focus of the thesis was actuator faults, only matched uncertainties were considered in the predictor-based approaches (PMRAC PRO, P-ACA). The case of unmatched uncertainties can be further studied.

Robust and Approximation-Based Adaptive Control

Robust adaptive control aims at increasing robustness against small disturbances and unmodeled dynamics. Some examples are the dead zone modification [124], the σ-modification [77, 78], the e-modification [116] or the projection operator [125]. Approximation based adaptive control approximates unknown regressor functions with special basis functions. One of the main approaches is the use of neural networks [92]. The combination of the PRO approach with robust and approximation-based adaptive control is an important next step since it links well-known approaches to systems with redundant actuators in a straightforward manner.

Control Allocation and Parameter Estimation

In order to address the different fault cases presented in Table 3.1, additive and multiplicative faults have been integrated in the actuator model (3.5). From the stability assessment it is known that the parameter errors are stable but do not necessarily converge to zero. Further, the inability to distinguish actuator faults and failures from other disturbances or model uncertainty prevents unbiased estimates [22]. Therefore, it is interesting to analyze the effects of these biased estimates regarding the control

allocation performance in view of input saturation. Specifically, the study of Assumption 4.3 (Control Allocation) under biased estimates can be addressed. The final goal is the understanding of how to match uncertainty parameterization, estimation strategy and control allocation approaches for a general mechanical system. In this work, the combination of the Predictor Based Adaptive Control Allocation (P-ACA) and the Prioritizing Control Allocation (PRIO CA) has proven robustness during the presented VTOL flight tests.

Propeller Rotation Direction Inversion

An interesting next step is the integration of the change of the propeller's rotation direction during flight within the adaptive control approach. In this way, the attainable control set (ACS) can be increased and the degraded authority scenario can be avoided. The main challenge is the consideration of the actuator behavior during the rotation direction changes.

Integration of External Fault Information

Because the actuator dynamics are faster than the rigid body dynamics, the detection of the fault directly within the actuator subsystem may lead to a faster detection and isolation of the fault as shown in [154]. Therefore, the integration of this external fault information within the current approach has the potential of improving the fault reaction performance.

Appendix A

Coordinate Systems

In this section, the coordinate frames needed for the flight dynamics description in Chapter 2 (Multirotor Dynamics Model). The descriptions corresponds to the one used at the Institute of Flight System Dynamics (TUM) as presented in [72].

Earth Centered Inertial Frame (ECI)	
Index	I
Role	Notation frame for Newtonian inertial physics
Origin	Center of the Earth
Translation	Around the sun with the solar system
Rotation	None
x-axis	In equatorial plane, points towards the vernal equinox
y-axis	In equatorial plane, builds a right-hand system
z-axis	Rotation axis of the Earth

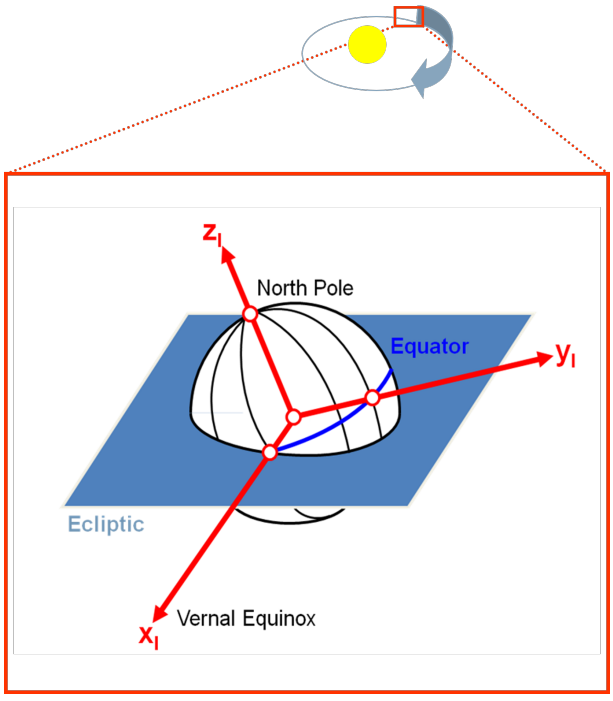


Figure A.1: Earth Centered Inertial Frame (ECI) (taken from [72])

Earth-Centered Earth-Fixed Frame (ECEF)	
Index	E
Role	Notation frame for positioning and navigation
Origin	Center of the Earth
Translation	Moves with the ECI-Frame
Rotation	Rotates with the Earth about the z-axis
x-axis	In the equatorial plane, points towards the Greenwich meridian
y-axis	In the equatorial plane, forms a right-hand system
z-axis	Rotation axis of the Earth

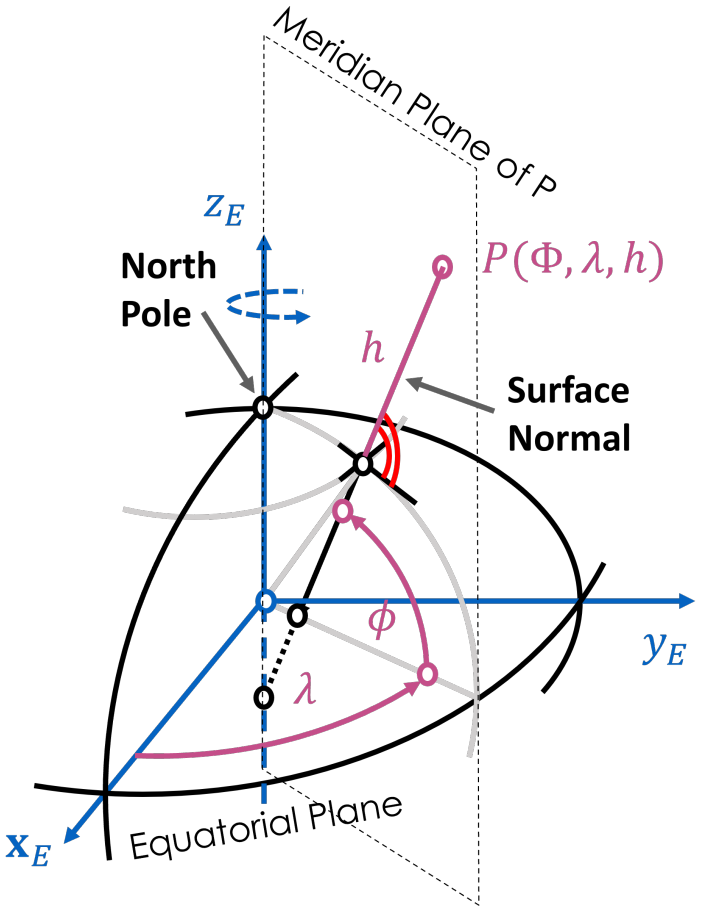


Figure A.2: Earth-Centered Earth-Fixed Frame (ECEF) (adapted from [72])

North-East-Down Frame (NED)	
Index	O
Role	Orientation reference frame
Origin	Aircraft reference point
Translation	Moves with the aircraft reference point
Rotation	Rotates with transport rate to comply with the NED alignment
x-axis	Parallel to the local geoid surface, points to the geographic north pole
y-axis	Parallel to the local geoid surface, points east to form a right-hand system
z-axis	Points downwards, perpendicular to the local geoid surface

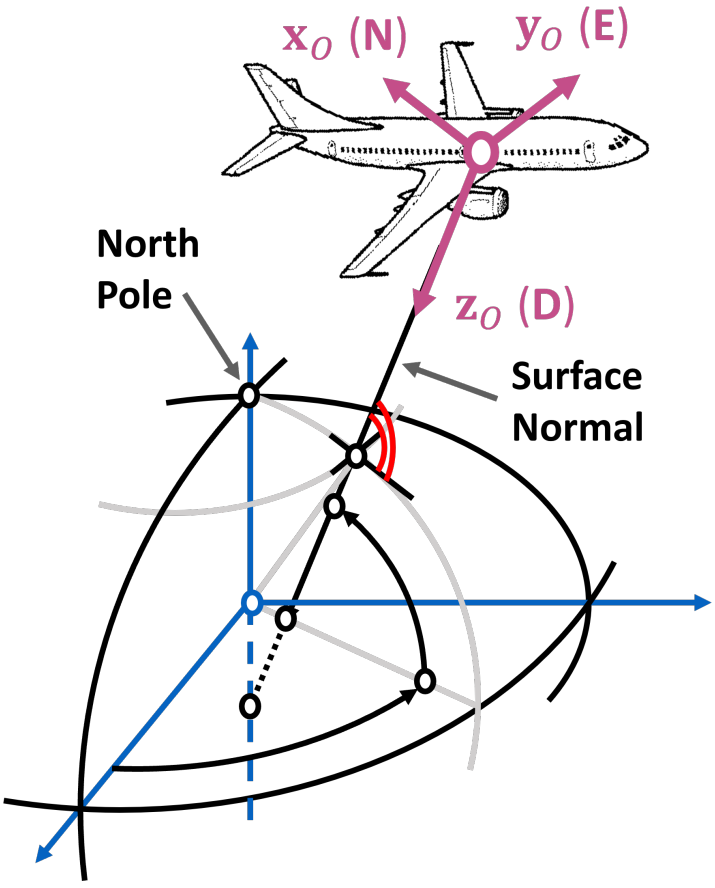


Figure A.3: North-East-Down Frame (NED) (adapted from [72])

Body-Fixed Frame (B)	
Index	B
Role	Notation frame, orientation frame
Origin	Aircraft reference point
Translation	Moves with the aircraft reference point
Rotation	Rotates with the aircraft
x-axis	Points to the front of the aircraft in symmetry plane
y-axis	Points to starboard side to form a right-hand system
z-axis	Points downwards in symmetry plane

Figure A.4: Body-Fixed Frame (B)

Local Navigation System (N)	
Index	N
Role	Local navigation frame derived from the NED frame
Origin	Point on the Earth's surface
Translation	Moves with the Earth's rotation
Rotation	Rotates with ECEF frame
x-axis	Parallel to the local geoid surface, corresponds to the NED x-axis rotated about the angle χ_N and about the NED z-axis
y-axis	Parallel to the local geoid surface, forms a right-hand system
z-axis	Points downwards, perpendicular to the local geoid surface

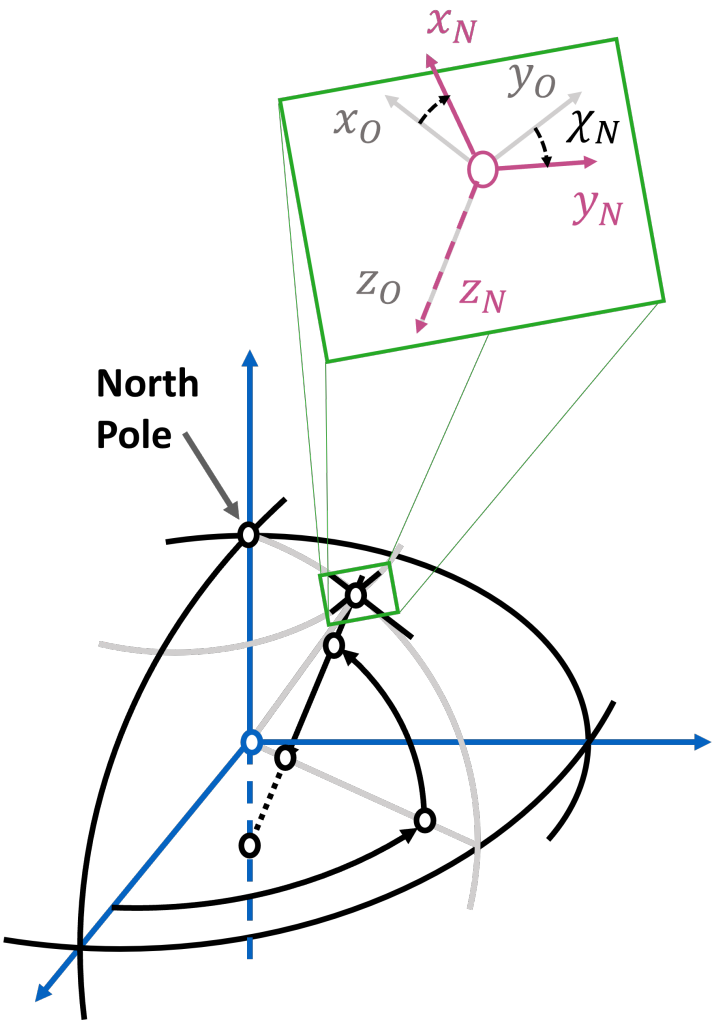


Figure A.5: Local Navigation System (N) (adapted from [72])

Appendix B

Mathematical Background

In this appendix the main mathematical tools used for the stability proofs are recompiled.

Lemma B.1 (Matrix Product) *For any two matrices* $\mathbf{A} \in \mathbb{R}^{n\times m}$, $\mathbf{B} \in \mathbb{R}^{p\times m}$, *the following product can be written as the sum of dyadic product*

$$\mathbf{A} \cdot \mathbf{B}^T = \sum_{k=1}^{m} \mathbf{a}_{c,k}\mathbf{b}_{c,k}^T,$$

where $\mathbf{a}_{c,k}$ *and* $\mathbf{b}_{c,k}$ *are the* k*-th columns of* $\mathbf{A}$ *and* $\mathbf{B}$ *respectively.*

Proof: Let the matrix $\mathbf{C}$ be defined as $\mathbf{C} = \mathbf{A}\cdot\mathbf{B}^T$. Each of its elements can be computed as

$$C_{ij} = \sum_{k=1}^{m} A_{ik}B_{jk}.$$

This corresponds exactly to sum of dyadic products. ■

Lemma B.2 (Trace and Scalar Product) *For any two vectors* $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$, *it holds that*

$$\mathbf{a}^T\mathbf{b} = tr\left(\mathbf{b}\mathbf{a}^T\right).$$

This property is found in Section 3.2.8.3 in [59].

Lemma B.3 (Trace Cyclic Property) *For any three matrices* $\mathbf{A} \in \mathbb{R}^{n\times m}$, $\mathbf{B} \in \mathbb{R}^{m\times p}$ *and* $\mathbf{C} \in \mathbb{R}^{p\times n}$, *it holds that*

$$tr\left(\mathbf{ABC}\right) = tr\left(\mathbf{CAB}\right) = tr\left(\mathbf{BCA}\right).$$

This property is found in Section 3.2.4 in [59].

Lemma B.4 (Rank) *Given a real matrix* $\mathbf{A} \in \mathbb{R}^{n\times m}$, *the following statements hold.*

1. $\operatorname{rank}\left(\mathbf{A}\right) = \operatorname{rank}\left(\mathbf{A}^T\right)$.
2. $\operatorname{rank}\left(\mathbf{A}^T\mathbf{A}\right) = \operatorname{rank}\left(\mathbf{A}\right)$.

3. $\text{rank}\,(\mathbf{A}) \leq \min(n, m)$.

4. *For a matrix* $\mathbf{B} \in \mathbb{R}^{m\times k}$*, it holds that* $\text{rank}\,(\mathbf{A}\cdot\mathbf{B}) \leq \min\left(\text{rank}\,(\mathbf{A}), \text{rank}\,(\mathbf{B})\right)$.

5. *For non-singular matrices* $\mathbf{Q}_L \in \mathbb{R}^{n\times n}$ *and* $\mathbf{Q}_R \in \mathbb{R}^{m\times m}$ *it holds that* $\text{rank}\,(\mathbf{Q}_L\cdot\mathbf{A}) = \text{rank}\,(\mathbf{A})$ *and* $\text{rank}\,(\mathbf{A}\cdot\mathbf{Q}_R) = \text{rank}\,(\mathbf{A})$.

6. *Full rank factorization: The matrix* $\mathbf{A}$ *has* $\text{rank}\,(\mathbf{A}) = k$ *if and only if there exist two full column rank matrices* $\mathbf{X} \in \mathbb{R}^{n\times k}$ *and* $\mathbf{Y} \in \mathbb{R}^{m\times k}$ *such that* $\mathbf{A} = \mathbf{X}\cdot\mathbf{Y}^T$*. Further,* $\text{rank}\,(\mathbf{A}) = \text{rank}\,(\mathbf{X}) = \text{rank}\,(\mathbf{Y}) = k$.

7. *If* $\mathbf{C} \in \mathbb{R}^{k\times n}$ *is a full column rank matrix,* $\text{rank}\,(\mathbf{C}) = n$*, then* $\text{rank}\,(\mathbf{C}\cdot\mathbf{A}) = \text{rank}\,(\mathbf{A})$.

8. *If* $\mathbf{R} \in \mathbb{R}^{m\times k}$ *is a full row rank matrix,* $\text{rank}\,(\mathbf{R}) = m$*, then* $\text{rank}\,(\mathbf{A}\cdot\mathbf{R}) = \text{rank}\,(\mathbf{A})$.

The last two item of the lemma are found in Section 3.3.9.1 in [59]. All the previous items are in Section 0.4 in [74].

Lemma B.5 (Frobenius Norm Invariance) *The Frobenius norm of a matrix* $\boldsymbol{\Theta} \in \mathbb{R}^{n\times m}$ *does not change if multiplied from the left and right by two orthogonal matrices* $\mathbf{U} \in \mathbb{R}^{m\times m}$, $\mathbf{V} \in \mathbb{R}^{n\times n}$*, that is*

$$||\boldsymbol{\Theta}||_F = ||\mathbf{V}\boldsymbol{\Theta}\mathbf{U}||_F\,.$$

Proof: Using $\mathbf{U}\mathbf{U}^T = \mathbf{I}$ and $\mathbf{V}\mathbf{V}^T = \mathbf{I}$, the definition of the Frobenius norm and Lemma B.3 (Trace Cyclic Property) it follows that

$$||\mathbf{V}\boldsymbol{\Theta}\mathbf{U}||_F = tr\left(\mathbf{V}\boldsymbol{\Theta}\mathbf{U}\cdot\mathbf{U}^T\boldsymbol{\Theta}^T\mathbf{V}^T\right) = tr\left(\mathbf{V}\boldsymbol{\Theta}\cdot\boldsymbol{\Theta}^T\mathbf{V}^T\right) = tr\left(\mathbf{V}^T\mathbf{V}\boldsymbol{\Theta}\boldsymbol{\Theta}^T\right) = tr\left(\boldsymbol{\Theta}\boldsymbol{\Theta}^T\right),$$

which corresponds the definition of $||\boldsymbol{\Theta}||_F$. ■

Lemma B.6 (Orthogonal Transformation Invariance) *Given an orthogonal matrix* $\mathbf{V} \in \mathbb{R}^{m\times m}$*, the product* $\mathbf{V}^T\boldsymbol{\Lambda}\mathbf{V}$ *has the same eigenvalues as the matrix* $\boldsymbol{\Lambda} \in \mathbb{R}^{m\times m}$*. Therefore* $\mathbf{V}^T\boldsymbol{\Lambda}\mathbf{V}$ *is positive definite / positive semi-definite / non-singular if and only if the matrix* $\boldsymbol{\Lambda}$ *is positive definite / positive semi-definite / non-singular.*

The lemma follows from Theorem 1.3.3 in [74].

Lemma B.7 (Positive Definite Trace 1) *Given two symmetric positive definite matrices* $\boldsymbol{\Gamma} \in \mathbb{R}^{n\times n}$, $\boldsymbol{\Lambda} \in \mathbb{R}^{m\times m}$*, the term* $tr\left(\boldsymbol{\Theta}\boldsymbol{\Gamma}\boldsymbol{\Theta}^T\boldsymbol{\Lambda}\right)$ *is positive definite in* $\boldsymbol{\Theta} \in \mathbb{R}^{m\times n}$ *and is bounded by*

$$\underline{\lambda}\left(\boldsymbol{\Gamma}\right)\underline{\lambda}\left(\boldsymbol{\Lambda}\right)||\boldsymbol{\Theta}||_F^2 \leq tr\left(\boldsymbol{\Theta}\boldsymbol{\Gamma}\boldsymbol{\Theta}^T\boldsymbol{\Lambda}\right) \leq \overline{\lambda}\left(\boldsymbol{\Gamma}\right)\overline{\lambda}\left(\boldsymbol{\Lambda}\right)||\boldsymbol{\Theta}||_F^2\,,$$

where $\underline{\lambda}\left(\cdot\right), \overline{\lambda}\left(\cdot\right) \in \mathbb{R}$ *are the smallest and largest eigenvalues of the respective argument.*

Proof: Because $\boldsymbol{\Gamma}$ and $\boldsymbol{\Lambda}$ are symmetric positive definite matrices, there exist an eigenvalue decomposition such that

$$\boldsymbol{\Gamma} = \mathbf{Q}_\Gamma\mathbf{D}_\Gamma\mathbf{Q}_\Gamma^T, \qquad \boldsymbol{\Lambda} = \mathbf{Q}_\Lambda\mathbf{D}_\Lambda\mathbf{Q}_\Lambda^T,$$

with two diagonal matrices $\mathbf{D}_\Gamma \in \mathbb{R}^{n\times n}$ and $\mathbf{D}_\Lambda \in \mathbb{R}^{m\times m}$ with positive entries and two orthogonal matrices $\mathbf{Q}_\Gamma \in \mathbb{R}^{n\times n}$ and $\mathbf{Q}_\Lambda \in \mathbb{R}^{m\times m}$ [74, Th. 4.1.5]. Then, using

the parameter transformation $\overline{\boldsymbol{\Theta}} = \mathbf{Q}_{\Lambda}^{T}\boldsymbol{\Theta}\mathbf{Q}_{\Gamma}$ and Lemma B.3 (Trace Cyclic Property) it follows that

$$\begin{aligned} tr\left(\boldsymbol{\Theta}\boldsymbol{\Gamma}\boldsymbol{\Theta}^{T}\boldsymbol{\Lambda}\right) &= tr\left(\boldsymbol{\Theta}\mathbf{Q}_{\Gamma}\mathbf{D}_{\Gamma}\mathbf{Q}_{\Gamma}^{T}\boldsymbol{\Theta}^{T}\mathbf{Q}_{\Lambda}\mathbf{D}_{\Lambda}\mathbf{Q}_{\Lambda}^{T}\right), \\ &= tr\left(\mathbf{Q}_{\Lambda}^{T}\boldsymbol{\Theta}\mathbf{Q}_{\Gamma}\mathbf{D}_{\Gamma}\mathbf{Q}_{\Gamma}^{T}\boldsymbol{\Theta}^{T}\mathbf{Q}_{\Lambda}\mathbf{D}_{\Lambda}\right), \\ &= tr\left(\overline{\boldsymbol{\Theta}}\mathbf{D}_{\Gamma}\overline{\boldsymbol{\Theta}}^{T}\mathbf{D}_{\Lambda}\right), \\ &= tr\left(\left[\overline{\boldsymbol{\theta}}_{c1}d_{\Gamma 1},\ldots,\overline{\boldsymbol{\theta}}_{cn}d_{\Gamma 1}\right]\left[\overline{\boldsymbol{\theta}}_{r1}^{T}d_{\Lambda 1},\ldots,\overline{\boldsymbol{\theta}}_{rm}^{T}d_{\Lambda m}\right]\right), \\ &= \sum_{j=1}^{m}\sum_{i=1}^{n}\overline{\boldsymbol{\theta}}_{ji}^{2}d_{\Gamma i}d_{\Lambda j}. \end{aligned}$$

Here, $\overline{\boldsymbol{\theta}}_{ci} \in \mathbb{R}^{m}$ and $\overline{\boldsymbol{\theta}}_{ri} \in \mathbb{R}^{1\times n}$ correspond to the i-th column and row of $\overline{\boldsymbol{\Theta}}$ respectively. $d_{\Gamma i}, d_{\Lambda i} \in \mathbb{R}$ are the i-th diagonal entry of $\mathbf{D}_{\Gamma}$ and $\mathbf{D}_{\Lambda}$ respectively, i.e. they are the eigenvalues of $\boldsymbol{\Gamma}$ and $\boldsymbol{\Lambda}$. Subsequently, from the definition of the Frobenius norm and Lemma B.5 (Frobenius Norm Invariance) it follows that

$$\begin{aligned} \underline{\lambda}\left(\boldsymbol{\Gamma}\right)\underline{\lambda}\left(\boldsymbol{\Lambda}\right)\left(\sum_{j=1}^{m}\sum_{i=1}^{n}\overline{\boldsymbol{\theta}}_{ji}^{2}\right) &\leq tr\left(\boldsymbol{\Theta}\boldsymbol{\Gamma}\boldsymbol{\Theta}^{T}\boldsymbol{\Lambda}\right) \leq \overline{\lambda}\left(\boldsymbol{\Gamma}\right)\overline{\lambda}\left(\boldsymbol{\Lambda}\right)\left(\sum_{j=1}^{m}\sum_{i=1}^{n}\overline{\boldsymbol{\theta}}_{ji}^{2}\right), \\ \underline{\lambda}\left(\boldsymbol{\Gamma}\right)\underline{\lambda}\left(\boldsymbol{\Lambda}\right)\left|\left|\overline{\boldsymbol{\Theta}}\right|\right|_{F}^{2} &\leq tr\left(\boldsymbol{\Theta}\boldsymbol{\Gamma}\boldsymbol{\Theta}^{T}\boldsymbol{\Lambda}\right) \leq \overline{\lambda}\left(\boldsymbol{\Gamma}\right)\overline{\lambda}\left(\boldsymbol{\Lambda}\right)\left|\left|\overline{\boldsymbol{\Theta}}\right|\right|_{F}^{2}, \\ \underline{\lambda}\left(\boldsymbol{\Gamma}\right)\underline{\lambda}\left(\boldsymbol{\Lambda}\right)\left|\left|\boldsymbol{\Theta}\right|\right|_{F}^{2} &\leq tr\left(\boldsymbol{\Theta}\boldsymbol{\Gamma}\boldsymbol{\Theta}^{T}\boldsymbol{\Lambda}\right) \leq \overline{\lambda}\left(\boldsymbol{\Gamma}\right)\overline{\lambda}\left(\boldsymbol{\Lambda}\right)\left|\left|\boldsymbol{\Theta}\right|\right|_{F}^{2}. \end{aligned}$$

■

Lemma B.8 (Positive Definite Trace) *Given a symmetric positive definite matrix $\boldsymbol{\Gamma} \in \mathbb{R}^{n\times n}$ and a positive definite matrix $\boldsymbol{\Lambda} \in \mathbb{R}^{m\times m}$, the term $tr\left(\boldsymbol{\Theta}\boldsymbol{\Gamma}\boldsymbol{\Theta}^{T}\boldsymbol{\Lambda}\right)$ is positive definite in $\boldsymbol{\Theta} \in \mathbb{R}^{m\times n}$ and is bounded by*

$$\underline{\lambda}\left(\boldsymbol{\Gamma}\right)\underline{\lambda}\left(\boldsymbol{\Lambda}_{s}\right)\left|\left|\boldsymbol{\Theta}\right|\right|_{F}^{2} \leq tr\left(\boldsymbol{\Theta}\boldsymbol{\Gamma}\boldsymbol{\Theta}^{T}\boldsymbol{\Lambda}\right) \leq \overline{\lambda}\left(\boldsymbol{\Gamma}\right)\overline{\lambda}\left(\boldsymbol{\Lambda}_{s}\right)\left|\left|\boldsymbol{\Theta}\right|\right|_{F}^{2},$$

where $\underline{\lambda}\left(\cdot\right), \overline{\lambda}\left(\cdot\right) \in \mathbb{R}$ are the smallest and largest eigenvalues of the respective argument and $\boldsymbol{\Lambda}_{s} = \frac{1}{2}\left(\boldsymbol{\Lambda} + \boldsymbol{\Lambda}^{T}\right)$ is the symmetric part of $\boldsymbol{\Lambda}$.

Proof: First separate the matrix $\boldsymbol{\Lambda}$ into its symmetric $\boldsymbol{\Lambda}_{s} = \frac{1}{2}\left(\boldsymbol{\Lambda} + \boldsymbol{\Lambda}^{T}\right)$ and skew-symmetric $\boldsymbol{\Lambda}_{a} = \frac{1}{2}\left(\boldsymbol{\Lambda} - \boldsymbol{\Lambda}^{T}\right)$ parts. For the skew-symmetric part $\boldsymbol{\Lambda}_{a}$, it holds that $\boldsymbol{\Lambda}_{a}^{T} = -\boldsymbol{\Lambda}_{a}$ and hence using Lemma B.3 (Trace Cyclic Property) it follows that

$$\begin{aligned} tr\left(\boldsymbol{\Theta}\boldsymbol{\Gamma}\boldsymbol{\Theta}^{T}\boldsymbol{\Lambda}_{a}\right) &= tr\left(\left(\boldsymbol{\Theta}\boldsymbol{\Gamma}\boldsymbol{\Theta}^{T}\boldsymbol{\Lambda}_{a}\right)^{T}\right) \\ &= tr\left(\boldsymbol{\Lambda}_{a}^{T}\boldsymbol{\Theta}\boldsymbol{\Gamma}^{T}\boldsymbol{\Theta}^{T}\right), \\ &= -tr\left(\boldsymbol{\Lambda}_{a}\boldsymbol{\Theta}\boldsymbol{\Gamma}\boldsymbol{\Theta}^{T}\right), \\ &= -tr\left(\boldsymbol{\Theta}\boldsymbol{\Gamma}\boldsymbol{\Theta}^{T}\boldsymbol{\Lambda}_{a}\right), \\ &= 0. \end{aligned}$$

Furthermore, due to the linearity of the trace it can be concluded that

$$\begin{aligned} tr\left(\boldsymbol{\Theta}\boldsymbol{\Gamma}\boldsymbol{\Theta}^{T}\boldsymbol{\Lambda}\right) &= tr\left(\boldsymbol{\Theta}\boldsymbol{\Gamma}\boldsymbol{\Theta}^{T}\left(\boldsymbol{\Lambda}_{s} + \boldsymbol{\Lambda}_{a}\right)\right) \\ &= tr\left(\boldsymbol{\Theta}\boldsymbol{\Gamma}\boldsymbol{\Theta}^{T}\boldsymbol{\Lambda}_{s}\right) + tr\left(\boldsymbol{\Theta}\boldsymbol{\Gamma}\boldsymbol{\Theta}^{T}\boldsymbol{\Lambda}_{a}\right) \\ &= tr\left(\boldsymbol{\Theta}\boldsymbol{\Gamma}\boldsymbol{\Theta}^{T}\boldsymbol{\Lambda}_{s}\right) \end{aligned}$$

Using the fact that $\mathbf{\Lambda}_s$ is symmetric positive definite if and only if $\mathbf{\Lambda}$ is positive definite, Lemma B.7 (Positive Definite Trace 1) can be applied for $tr\left(\mathbf{\Theta\Gamma\Theta}^T\mathbf{\Lambda}_s\right)$ given that $\mathbf{\Gamma}$ and $\mathbf{\Lambda}_s$ are symmetric positive definite. ■

Lemma B.9 (Barbalat) *Let $\phi(t) : \mathbb{R} \to \mathbb{R}$ be a uniformly continuous function on $[0, \infty)$. Suppose that $\lim_{t\to\infty} \int_{\tau=0}^{t} \phi(\tau)d\tau$ exists and is finite. Then,*

$$\phi(t) \to 0 \quad as \quad t \to \infty.$$

This corresponds to Lemma 8.2 in [87], where the proof can be found.

Lemma B.10 (Orthogonal Matrix) *Let $\mathbf{V} \in \mathbb{R}^{m\times m}$ be an orthogonal matrix, i.e. $\mathbf{V}^T\mathbf{V} = \mathbf{V}\mathbf{V}^T = \mathbf{I}$. Let $\mathbf{V}_r \in \mathbb{R}^{m\times r}, \mathbf{V}_n \in \mathbb{R}^{m\times(m-r)}$ be partitions of $\mathbf{V}$ such that $\mathbf{V} = [\mathbf{V}_r, \mathbf{V}_n]$. Then, it holds that*

$$1)\ \mathbf{V}_r^T\mathbf{V}_r = \mathbf{I}, \quad 2)\ \mathbf{V}_n^T\mathbf{V}_n = \mathbf{I}, \quad 3)\ \mathbf{V}_r^T\mathbf{V}_n = \mathbf{0}, \quad 4)\ \mathbf{I} = \mathbf{V}_r\mathbf{V}_r^T + \mathbf{V}_n\mathbf{V}_n^T.$$

Proof: The lemma follows directly from the orthogonality conditions

$$\mathbf{V}^T\mathbf{V} = \begin{bmatrix}\mathbf{V}_r^T \\ \mathbf{V}_n^T\end{bmatrix}\begin{bmatrix}\mathbf{V}_r & \mathbf{V}_n\end{bmatrix} = \begin{bmatrix}\mathbf{V}_r^T\mathbf{V}_r & \mathbf{V}_r^T\mathbf{V}_n \\ \mathbf{V}_n^T\mathbf{V}_r & \mathbf{V}_n^T\mathbf{V}_n\end{bmatrix} = \begin{bmatrix}\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}\end{bmatrix},$$
$$\mathbf{V}\mathbf{V}^T = \begin{bmatrix}\mathbf{V}_r & \mathbf{V}_n\end{bmatrix}\begin{bmatrix}\mathbf{V}_r^T \\ \mathbf{V}_n^T\end{bmatrix} = \mathbf{V}_r\mathbf{V}_r^T + \mathbf{V}_n\mathbf{V}_n^T = \mathbf{I}.$$

■

Lemma B.11 (Transformation Matrix Inverse) *Given is the transformation matrix as defined in Section 4.3.4*

$$\mathbf{T} := \begin{bmatrix}\mathbf{U}_b \cdot \mathbf{D}_b \cdot \mathbf{V}_{b,r}^T \\ \mathbf{V}_{b,n}^T\end{bmatrix} \in \mathbb{R}^{m\times m},$$

with an orthogonal matrix $\mathbf{U}_b \in \mathbb{R}^{n\times n}$, a positive definite diagonal matrix $\mathbf{D}_b \in \mathbb{R}^{n\times n}$ and the partitions $\mathbf{V}_{b,r} \in \mathbb{R}^{m\times r}, \mathbf{V}_{b,n} \in \mathbb{R}^{m\times(m-r)}$ of the orthogonal matrix $\mathbf{V}_b$ such that $\mathbf{V}_b = [\mathbf{V}_{b,r}, \mathbf{V}_{b,n}]$. Its inverse is then given by

$$\mathbf{T}^{-1} = \begin{bmatrix}\mathbf{V}_{b,r} \cdot \mathbf{D}_b^{-1} \cdot \mathbf{U}_b^T, & \mathbf{V}_{b,n}\end{bmatrix} \in \mathbb{R}^{m\times m}.$$

Proof: The lemma is proved by computing the product of the matrices and using Lemma B.10 (Orthogonal Matrix). For the first product it holds that

$$\begin{aligned}\mathbf{T}\cdot\mathbf{T}^{-1} &= \begin{bmatrix}\mathbf{U}_b \cdot \mathbf{D}_b \cdot \mathbf{V}_{b,r}^T \\ \mathbf{V}_{b,n}^T\end{bmatrix} \cdot \begin{bmatrix}\mathbf{V}_{b,r} \cdot \mathbf{D}_b^{-1} \cdot \mathbf{U}_b^T, & \mathbf{V}_{b,n}\end{bmatrix}, \\ &= \begin{bmatrix}\mathbf{U}_b \cdot \mathbf{D}_b \cdot \mathbf{V}_{b,r}^T \cdot \mathbf{V}_{b,r} \cdot \mathbf{D}_b^{-1} \cdot \mathbf{U}_b^T & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_{b,n}^T \cdot \mathbf{V}_{b,n}\end{bmatrix}, \\ &= \begin{bmatrix}\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}\end{bmatrix}.\end{aligned}$$

For the second product it holds that

$$\begin{aligned}\mathbf{T}^{-1}\cdot\mathbf{T} &= \begin{bmatrix}\mathbf{V}_{b,r}\cdot\mathbf{D}_b^{-1}\cdot\mathbf{U}_b^T, & \mathbf{V}_{b,n}\end{bmatrix}\cdot\begin{bmatrix}\mathbf{U}_b\cdot\mathbf{D}_b\cdot\mathbf{V}_{b,r}^T\\ \mathbf{V}_{b,n}^T\end{bmatrix},\\ &= \mathbf{V}_{b,r}\cdot\mathbf{D}_b^{-1}\cdot\mathbf{U}_b^T\cdot\mathbf{U}_b\cdot\mathbf{D}_b\cdot\mathbf{V}_{b,r}^T+\mathbf{V}_{b,n}\cdot\mathbf{V}_{b,n}^T,\\ &= \mathbf{V}_{b,r}\cdot\mathbf{V}_{b,r}^T+\mathbf{V}_{b,n}\cdot\mathbf{V}_{b,n}^T,\\ &= \mathbf{I}\end{aligned}$$

which proves the lemma. ■

Lemma B.12 (Schur condition for positive definiteness) *Given is a symmetric block matrix* $\mathbf{P}$ *partitioned as*

$$\mathbf{P}=\begin{pmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{B}^T & \mathbf{C}\end{pmatrix},$$

with $\mathbf{A}\in\mathbb{R}^{n\times n}$, $\mathbf{C}\in\mathbb{R}^{m\times m}$ *and* $\mathbf{B}\in\mathbb{R}^{n\times m}$. *The matrix* $\mathbf{P}$ *is positive definite*

- *if and only if* $\mathbf{A}$ *and the Schur complement* $\mathbf{C}-\mathbf{B}^T\mathbf{A}^{-1}\mathbf{B}$ *are both positive definite,*
- *if and only if* $\mathbf{C}$ *and the Schur complement* $\mathbf{A}-\mathbf{B}\mathbf{C}^{-1}\mathbf{B}^T$ *are both positive definite.*

Furthermore, the matrix $\mathbf{P}$ *is positive semi-definite*

- *if and only if* $\mathbf{A}$ *is positive definite and the Schur complement* $\mathbf{C}-\mathbf{B}^T\mathbf{A}^{-1}\mathbf{B}$ *is positive semi-definite.*

This is a part of Theorem 1.12 in [155].

Appendix C

Control Allocation

In this appendix, a special case of the prioritizing control allocation (PRIO CA) is treated. Specifically, the solution of the linear inequality (4.54)

$$\mathbf{A}\mathbf{x} \leq \mathbf{b} \tag{C.1}$$

for $\mathbf{A} \in \mathbb{R}^{2\times 2}$, $\mathbf{x} \in \mathbb{R}^2$, $\mathbf{b} \in \mathbb{R}^2$ and linear dependent rows of $\mathbf{A}$ is analyzed based on the solution proposed in Section 4.3.5 (Update Direction - Singular Cases).

Qualitatively, each condition $\mathbf{A}_{ri} \cdot \mathbf{x} = b_i$ for $i = [1, 2]$ defines a line in the two dimensional plane and $\mathbf{A}_{ri} \cdot \mathbf{x} \leq b_i$ defines a half-plane. Here, $\mathbf{A}_{ri} \in \mathbb{R}^{1\times 2}$ and $b_i \in \mathbb{R}$ are the i-th row of $\mathbf{A}$ and $\mathbf{b}$ respectively. The problem is separated in two possible cases depending on the lines defining the limit of the inequalities:

- Case A: the two lines defined by the equality $\mathbf{A}\mathbf{x} = \mathbf{b}$ are linear dependent but not the same.
- Case B: the two lines defined by the equality $\mathbf{A}\mathbf{x} = \mathbf{b}$ are linear dependent and the same.

Each of these cases can be separated into qualitatively different sub-cases depending on the direction of the half-plane that each inequality defines. For each line, there are two possible half-planes that correspond to a solution. The sub-cases of case A are depicted in Figure C.1 and the ones of case B in Figure C.2. For cases 1 to 4 (case A), the matrices correspond to

$$\begin{aligned}
\text{Case 1:} \quad & \mathbf{A} = \begin{bmatrix} -2 & 1 \\ 2 & -1 \end{bmatrix}, \quad & \mathbf{b} = \begin{bmatrix} 0.2 \\ 1 \end{bmatrix}, \\
\text{Case 2:} \quad & \mathbf{A} = \begin{bmatrix} 2 & -1 \\ -2 & 1 \end{bmatrix}, \quad & \mathbf{b} = \begin{bmatrix} -0.2 \\ -1 \end{bmatrix}, \\
\text{Case 3:} \quad & \mathbf{A} = \begin{bmatrix} 2 & -1 \\ 2 & -1 \end{bmatrix}, \quad & \mathbf{b} = \begin{bmatrix} -0.2 \\ 1 \end{bmatrix}, \\
\text{Case 4:} \quad & \mathbf{A} = \begin{bmatrix} -2 & 1 \\ 2 & -1 \end{bmatrix}, \quad & \mathbf{b} = \begin{bmatrix} 0.2 \\ 1 \end{bmatrix}.
\end{aligned}$$

Hence, the rows of $\mathbf{A}$ are linear dependent but the limit of each linear inequality in

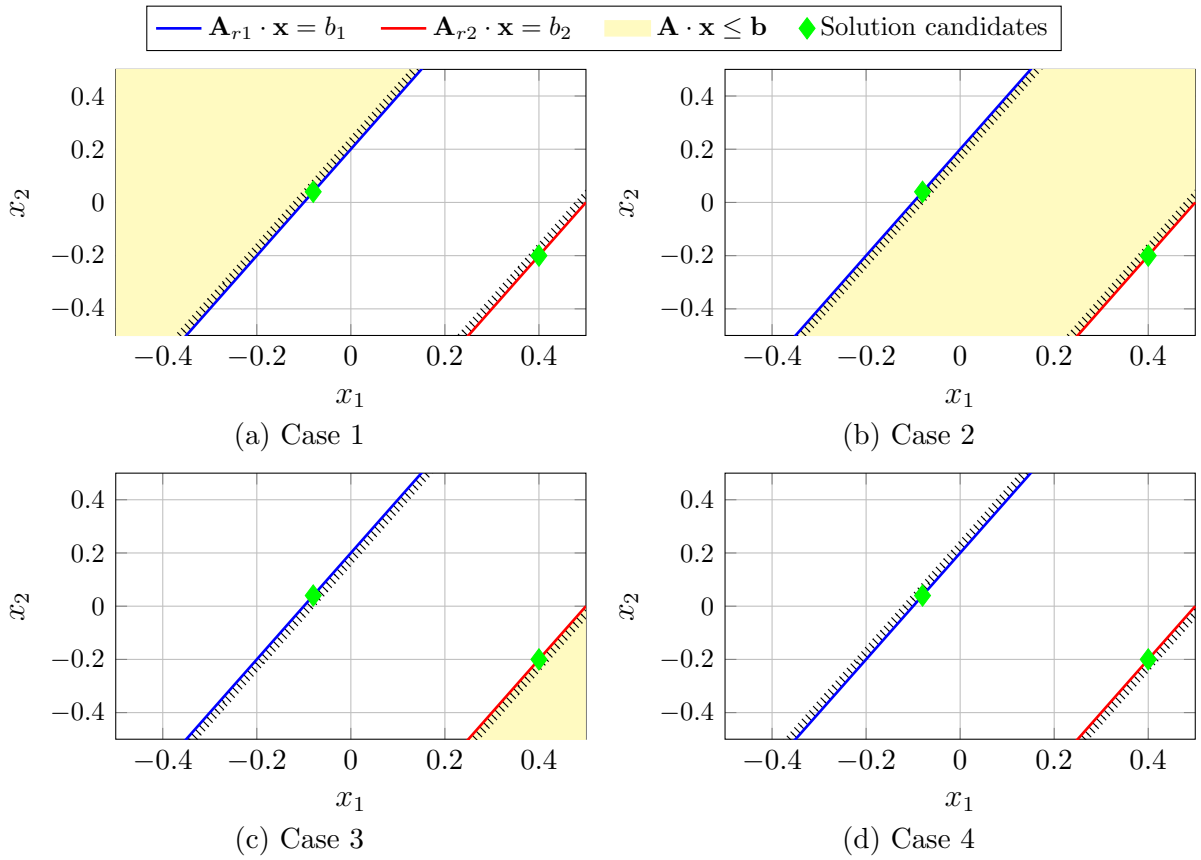


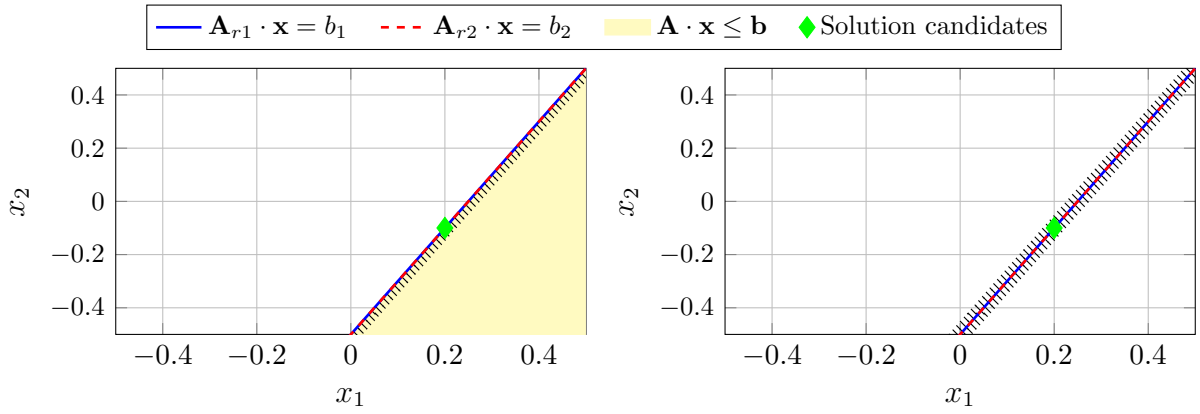
Figure C.1: Control Allocation - Update Direction: 2D Singular Case - Case A

(C.1) defines a different line. For the cases 5 and 6 (case B), the matrices correspond to

$$\text{Case 5:} \quad \mathbf{A} = \begin{bmatrix} 2 & -1 \\ 2 & -1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix},$$
$$\text{Case 6:} \quad \mathbf{A} = \begin{bmatrix} -2 & 1 \\ 2 & -1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} -0.5 \\ 0.5 \end{bmatrix}.$$

Hence, the rows of $\mathbf{A}$ are linear dependent and the limit of the two linear inequalities in (C.1) define the same line.

In Figures C.1 and C.2 the equality constraints, the solution set and the solution candidates proposed in Section 4.3.5 (Update Direction - Singular Cases) are plotted. In Section 4.3.5 (Update Direction - Singular Cases), the proposed solution candidates are the pseudo-inverse solution of $\mathbf{A}_{r1} \cdot \mathbf{x} = b_1$ and $\mathbf{A}_{r2} \cdot \mathbf{x} = b_2$. For all possible variants presented in Figures C.1 and C.2, it can be seen that if a solution set exists, at least one of the solution candidates is always part of it.



(a) Case 5

(b) Case 6

Figure C.2: Control Allocation - Update Direction: 2D Singular Case - Case B

Appendix D

Multirotor Control

D.1 Nonlinear Attitude Control

In this appendix the main mathematical tools used for the stability proofs of Chapter 5 (Multirotor Controllers) are recompiled.

Lemma D.1 (Cross Product) *Given are the vectors $\vec{\mathbf{a}}, \vec{\mathbf{b}} \in \mathbb{R}^3$ and the rotation matrix $\mathbf{M} \in SO(3)$. The skew symmetric matrix $\boldsymbol{\Omega}(\cdot)$ (2.14) has the following properties*

$$\begin{aligned}
\boldsymbol{\Omega}(\vec{\mathbf{a}})^T &= -\boldsymbol{\Omega}(\vec{\mathbf{a}}), \\
\boldsymbol{\Omega}(\vec{\mathbf{a}})\vec{\mathbf{b}} &= -\boldsymbol{\Omega}\left(\vec{\mathbf{b}}\right)\vec{\mathbf{a}}, \\
\boldsymbol{\Omega}(\vec{\mathbf{a}})\vec{\mathbf{a}} &= \mathbf{0}, \\
\mathbf{M}\boldsymbol{\Omega}(\vec{\mathbf{a}}) &= \boldsymbol{\Omega}(\mathbf{M}\vec{\mathbf{a}})\mathbf{M}, \\
\boldsymbol{\Omega}\left(\vec{\mathbf{a}} + \vec{\mathbf{b}}\right) &= \boldsymbol{\Omega}(\vec{\mathbf{a}}) + \boldsymbol{\Omega}\left(\vec{\mathbf{b}}\right),
\end{aligned}$$

The properties can be found in Section 2.1.1 in [55] or in [136].

Lemma D.2 (Scalar Triple Product) *Given the vectors $\vec{\mathbf{a}}, \vec{\mathbf{b}}, \vec{\mathbf{c}} \in \mathbb{R}^3$, it holds that*

$$\vec{\mathbf{a}}^T\left(\vec{\mathbf{b}} \times \vec{\mathbf{c}}\right) = \vec{\mathbf{b}}^T(\vec{\mathbf{c}} \times \vec{\mathbf{a}}) = \vec{\mathbf{c}}^T\left(\vec{\mathbf{a}} \times \vec{\mathbf{b}}\right)$$

In the following the proof of Lemma 5.5 (Attitude Lyapunov function) in Chapter 5 is given. For readability the lemma is rewritten.

Lemma D.3 (Attitude Lyapunov function) *Let the attitude Lyapunov function be*

$$\begin{aligned}
V\left(\left(\vec{\mathbf{z}}_d\right)_B(t), \vec{\boldsymbol{\omega}}^{DB}(t)\right) = k_a \cdot V_\varphi\left(\varphi_d\left(\left(\vec{\mathbf{z}}_d\right)_B(t)\right)\right) + \frac{1}{2}\left(\vec{\boldsymbol{\omega}}^{DB}\right)_B^T(t)\left(\vec{\boldsymbol{\omega}}^{DB}\right)_B(t) \\
+ c \cdot \left(\vec{\boldsymbol{\omega}}^{DB}\right)_B^T(t)\left(\vec{\mathbf{e}}_a^{DB}\right)_B(t),
\end{aligned}$$

where $c \in \mathbb{R}$, $k_a > 0 \in \mathbb{R}$ and $\vec{\mathbf{e}}_a^{DB}(t) := \vec{\mathbf{z}}_d(t) \times \vec{\mathbf{z}}_B(t) \in \mathbb{R}^3$ is defined as the attitude error vector. The Lyapunov function is defined in the configuration space $\mathbb{S}^2 \times \mathbb{R}^3$ and is positive definite for $|c| \leq \sqrt{k_a}$. Positive definite means that $V = 0$ if and only if $\left(\varphi_d\left(\left(\vec{\mathbf{z}}_d\right)_B(t)\right), \vec{\boldsymbol{\omega}}^{DB}(t)\right) = (0, \mathbf{0})$, otherwise it holds that $V(t) > 0$.

Proof: For the sake of readability the time dependency is dropped in the following. Using the fact that $||(\vec{\mathbf{e}}_a^{DB})_B|| = |\sin\varphi_d|$ and completing the square polynomial leads to the following lower bound

$$\begin{aligned} V(\varphi_d, \vec{\boldsymbol{\omega}}^{DB}) &\geq k_a\left(1-\cos\varphi_d\right) + \frac{1}{2}\left|\left|\left(\vec{\boldsymbol{\omega}}^{DB}\right)_B\right|\right|^2 - |c| \cdot \left|\left|\left(\vec{\boldsymbol{\omega}}^{DB}\right)_B\right|\right| |\sin\varphi_d|, \\ &= \underbrace{k_a\left(1-\cos\varphi_d\right) - \frac{c^2}{2}\sin^2\varphi_d}_{:=G_1(\varphi_d)} + \underbrace{\left(\frac{1}{\sqrt{2}}\left|\left|\left(\vec{\boldsymbol{\omega}}^{DB}\right)_B\right|\right| - \frac{|c|}{\sqrt{2}}|\sin\varphi_d|\right)^2}_{:=G_2\left(\varphi_d, \vec{\boldsymbol{\omega}}^{DB}\right)}. \end{aligned}$$

The first two terms can be studied using the attitude dependent function

$$G_1\left(\varphi_d\right) := k_a\left(1-\cos\varphi_d\right) - \frac{c^2}{2}\sin^2\varphi_d,$$

which has the following derivative

$$\frac{\partial G_1\left(\varphi_d\right)}{\partial\varphi_d} = k_a\sin\varphi_d - c^2\sin\varphi_d\cos\varphi_d = \sin\varphi_d\left(k_a - c^2\cos\varphi_d\right).$$

For $\varphi_d \in]0,\pi[$ the derivative $\frac{\partial G_1(\varphi_d)}{\partial\varphi_d}$ is positive if $(k_a - c^2\cos\varphi_d) > 0$. This is true for $|c| \leq \sqrt{k_a}$ since $\cos\varphi_d < 1$. For $\varphi_d \in]-\pi,0[$ the derivative $\frac{\partial G_1(\varphi_d)}{\partial\varphi_d}$ is negative for $|c| \leq \sqrt{k_a}$. Hence, because $G_1\left(\varphi_d\right)$ is a continuous function, $G_1\left(0\right) = 0$, $\frac{\partial G_1(0)}{\partial\varphi_d} = \frac{\partial G_1(\pi)}{\partial\varphi_d} = \frac{\partial G_1(-\pi)}{\partial\varphi_d} = 0$, $\frac{\partial G_1(\varphi_d)}{\partial\varphi_d} > 0$ for $\varphi_d \in]0,\pi[$, and $\frac{\partial G_1(\varphi_d)}{\partial\varphi_d} < 0$ for $\varphi_d \in]-\pi,0[$, the function $G_1\left(\varphi_d\right)$ is positive definite for $c \leq \sqrt{k_a}$. Therefore, $G_1\left(\varphi_d\right) = 0$ only for $\varphi_d = 0$. Since the Lyapunov function $V(\varphi_d, \vec{\boldsymbol{\omega}}^{DB}) = 0$ if and only if simultaneously $G_1\left(\varphi_d\right) = 0$ and $G_2\left(\varphi_d, \vec{\boldsymbol{\omega}}^{DB}\right) = 0$ hold, it is concluded that $V(\varphi_d, \vec{\boldsymbol{\omega}}^{DB}) = 0$ for $(\varphi_d, \vec{\boldsymbol{\omega}}^{DB}) = (0, \mathbf{0})$ and $V(\varphi_d, \vec{\boldsymbol{\omega}}^{DB}) > 0$ otherwise. ■

In the following the proof of Lemma 5.6 (Attitude Error Derivative Bound) in Chapter 5 is given. For readability the lemma is rewritten.

Lemma D.4 (Attitude Error Derivative Bound) *The norm of the derivative of the attitude error vector $\vec{\mathbf{e}}_a^{DB}\left(t\right) = \vec{\mathbf{z}}_d\left(t\right) \times \vec{\mathbf{z}}_B\left(t\right)$ can be bounded by the norm of the error rotational rates*

$$\left|\left|\left(\dot{\vec{\mathbf{e}}}_a^{DB}\right)_B^B\left(t\right)\right|\right| \leq \left|\left|\left(\vec{\boldsymbol{\omega}}^{BD}\right)_B\left(t\right)\right|\right|.$$

Proof: The upper bound can be computed from $||\vec{\mathbf{u}} \times \vec{\mathbf{v}}|| \leq ||\vec{\mathbf{u}}||\,||\vec{\mathbf{v}}||$ [59], the fact that $||\vec{\mathbf{z}}_d\left(t\right)|| = ||\vec{\mathbf{z}}_B\left(t\right)|| = 1$ and the kinematics of the desired vector $\vec{\mathbf{z}}_d\left(t\right)$ (5.13a) as follows

$$\begin{aligned} \left|\left|\left(\dot{\vec{\mathbf{e}}}_a^{DB}\right)_B^B\left(t\right)\right|\right| &= \left|\left|\left(\dot{\vec{\mathbf{z}}}_d\right)_B^B\left(t\right) \times \left(\vec{\mathbf{z}}_B\right)_B\right|\right|, \\ &= \left|\left|\left(-\left(\vec{\boldsymbol{\omega}}^{DB}\right)_B\left(t\right) \times \left(\vec{\mathbf{z}}_d\right)_B\left(t\right)\right) \times \left(\vec{\mathbf{z}}_B\right)_B\right|\right|, \\ &\leq \left|\left|\left(\vec{\boldsymbol{\omega}}^{DB}\right)_B\left(t\right) \times \left(\vec{\mathbf{z}}_d\right)_B\left(t\right)\right|\right| \cdot \left|\left|\left(\vec{\mathbf{z}}_B\right)_B\right|\right|, \\ &\leq \left|\left|\left(\vec{\boldsymbol{\omega}}^{DB}\right)_B\left(t\right)\right|\right| \cdot \left|\left|\left(\vec{\mathbf{z}}_d\right)_B\left(t\right)\right|\right| \cdot \left|\left|\left(\vec{\mathbf{z}}_B\right)_B\right|\right|, \\ &\leq \left|\left|\left(\vec{\boldsymbol{\omega}}^{DB}\right)_B\left(t\right)\right|\right|. \end{aligned}$$

■

In the following the proof of Lemma 5.7 (Positive Definite $\mathbf{Q}$) in Chapter 5 is given. For readability the lemma is rewritten.

Lemma D.5 (Positive Definite Q) *If $\mathbf{K}_\omega \in \mathbb{R}^{3\times3}$ is symmetric positive definite and $k_a > 0$, there exists a constant $c \in]0, \sqrt{k_a}[$ such that the matrix $\mathbf{Q}$*

$$\mathbf{Q} = \begin{bmatrix} c \cdot k_a \cdot \mathbf{I} & \frac{c}{2}\mathbf{K}_\omega \\ \frac{c}{2}\mathbf{K}_\omega & \mathbf{K}_\omega - c \cdot \mathbf{I} \end{bmatrix}$$

is positive definite.

Proof: Because the matrix $\mathbf{Q}$ is a symmetric block matrix, Lemma B.12 (Schur condition for positive definiteness) is used to prove positive definiteness. Therefore it needs to be verified that the matrix $c \cdot k_a \cdot \mathbf{I}$ and the Schur complement of $\mathbf{Q}$ given by

$$\begin{aligned} \mathbf{S}_q &= \mathbf{K}_\omega - c \cdot \mathbf{I} - \frac{c}{2} \cdot \mathbf{K}_\omega \cdot (c \cdot k_a \cdot \mathbf{I})^{-1} \cdot \frac{c}{2} \cdot \mathbf{K}_\omega, \\ &= \mathbf{K}_\omega - c \cdot \mathbf{I} - \frac{c}{4 \cdot k_a}\mathbf{K}_\omega^2. \end{aligned}$$

are positive definite. The matrix $c \cdot k_a \cdot \mathbf{I}$ is positive definite for $c > 0$ and $k_a > 0$. Furthermore, since $\mathbf{K}_\omega$ is symmetric positive definite, it admits the eigenvalue decomposition $\mathbf{K}_\omega = \mathbf{Q}_\omega \mathbf{\Lambda}_\omega \mathbf{Q}_\omega^T$ [59], where $\mathbf{Q}_\omega \in \mathbb{R}^{3\times3}$ is an orthogonal matrix and $\mathbf{\Lambda}_\omega \in \mathbb{R}^{3\times3}$ is a diagonal matrix with positive entries. The Schur complement of $\mathbf{Q}$ can then be rewritten as

$$\begin{aligned} \mathbf{S}_q &= \mathbf{Q}_\omega \left(\mathbf{\Lambda}_\omega - c \cdot \mathbf{I} - \frac{c}{4 \cdot k_a}\mathbf{\Lambda}_\omega^2 \right) \mathbf{Q}_\omega^T, \\ &= \mathbf{Q}_\omega \left(\mathbf{\Lambda}_\omega - c \left(\mathbf{I} + \frac{1}{4 \cdot k_a}\mathbf{\Lambda}_\omega^2 \right) \right) \mathbf{Q}_\omega^T. \end{aligned}$$

Since the matrix $\mathbf{\Lambda}_\omega - c \left(\mathbf{I} + \frac{1}{4 \cdot k_a}\mathbf{\Lambda}_\omega^2 \right)$ is diagonal, its entries correspond to the eigenvalues of $\mathbf{S}_q$. Hence, $\mathbf{S}_q$ is positive definite if for every eigenvalue $\lambda_i(\mathbf{K}_\omega)$ of $\mathbf{K}_\omega$ it holds that

$$\begin{aligned} 0 &< \lambda_i(\mathbf{K}_\omega) - c \cdot \left(1 + \frac{\lambda_i(\mathbf{K}_\omega)^2}{4 \cdot k_a} \right), \\ c &< \frac{\lambda_i(\mathbf{K}_\omega)}{1 + \frac{\lambda_i(\mathbf{K}_\omega)^2}{4 \cdot k_a}} =: c_{ub}\left(\lambda_i(\mathbf{K}_\omega), k_a\right). \end{aligned}$$

Therefore, the condition that the matrix $\mathbf{S}_q$ is positive definite translates to an upper bound function $c_{ub}\left(\lambda_i(\mathbf{K}_\omega), k_a\right)$ of c which can be seen in Figure D.1 for a fixed k_a. It has its maximum $\sqrt{k_a}$ at $\lambda_i(\mathbf{K}_\omega) = 2\sqrt{k_a}$ and for $\lambda_i(\mathbf{K}_\omega) \to \infty$ it tends to zero. Given that the upper bound is positive $c_{ub}\left(\lambda_i(\mathbf{K}_\omega), k_a\right) > 0$ for $\lambda_i(\mathbf{K}_\omega) > 0$, there exists a constant $c \in]0, \sqrt{k_a}[$ such that $\mathbf{Q}$ is positive definite. ■

In the following the proof of Lemma 5.8 (Linearization: Reduced Attitude) in Chapter 5 is given. For readability the lemma is rewritten.

Lemma D.6 (Linearization: Reduced Attitude) *The reduced attitude closed-loop system*

$$\left(\dot{\vec{\mathbf{z}}}_d\right)_B^B(t) = -\left(\vec{\boldsymbol{\omega}}^{DB}\right)_B(t) \times \left(\vec{\mathbf{z}}_d\right)_B(t), \tag{D.1}$$

$$\left(\dot{\vec{\boldsymbol{\omega}}}^{DB}\right)_B^B(t) = -k_a\left(\left(\vec{\mathbf{z}}_d\right)_B(t) \times \left(\vec{\mathbf{z}}_B\right)_B(t)\right) - \mathbf{K}_\omega\left(\vec{\boldsymbol{\omega}}^{DB}\right)_B(t), \tag{D.2}$$

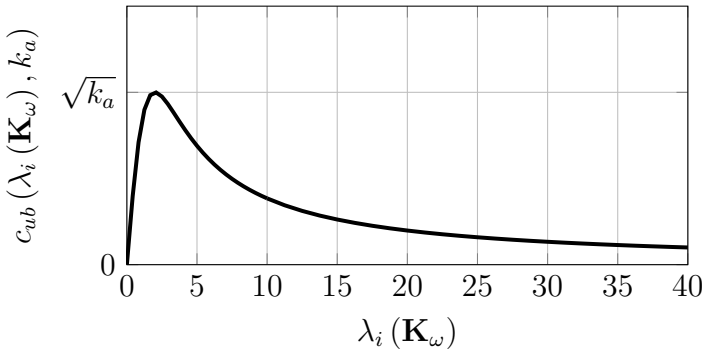


Figure D.1: Upper Bound Function

has a stable equilibrium at $\boldsymbol{\chi}_{d,e} = ((\vec{\mathbf{z}}_{d,e})_B, \vec{\boldsymbol{\omega}}_e^{DB}) = ([0,0,1]^T, \mathbf{0})$ *and an unstable equilibrium at* $\boldsymbol{\chi}_{u,e} = ((\vec{\mathbf{z}}_{d,e})_B, \vec{\boldsymbol{\omega}}_e^{DB}) = ([0,0,-1]^T, \mathbf{0})$ *if* k_a *is positive and* $\mathbf{K}_\omega$ *is a diagonal positive definite matrix. The subscript* e *refers to equilibrium points.*

Proof: From Theorem 5.3 (Attitude Equilibrium Points) it is known that the presented dynamic system has two equilibria $\boldsymbol{\chi}_{d,e}$ and $\boldsymbol{\chi}_{u,e}$. In order to asses their stability properties, the linear approximation of the dynamics at the two equilibria is analyzed. Using the definition of the skew symmetric matrix $\boldsymbol{\Omega}(\cdot)$ (2.14), the first derivative of the kinematics equation (D.1) with respect to the states is given by

$$\left.\frac{\partial\left(\dot{\vec{\mathbf{z}}}_d\right)_B^B}{\partial\left(\vec{\mathbf{z}}_d\right)_B}\right|_{\boldsymbol{\chi}_e} = -\boldsymbol{\Omega}\left(\left(\vec{\boldsymbol{\omega}}^{DB}\right)_B\right)\big|_{\boldsymbol{\chi}_e} = \mathbf{0}_{3\times3},$$

$$\left.\frac{\partial\left(\dot{\vec{\mathbf{z}}}_d\right)_B^B}{\partial\left(\vec{\boldsymbol{\omega}}^{DB}\right)_B}\right|_{\boldsymbol{\chi}_e} = \boldsymbol{\Omega}\left(\left(\vec{\mathbf{z}}_d\right)_B\right)\big|_{\boldsymbol{\chi}_e} = \boldsymbol{\Omega}\left(\left(\vec{\mathbf{z}}_{d,e}\right)_B\right).$$

The first derivative of the dynamics equation (D.2) with respect to the states is given by

$$\left.\frac{\partial\left(\dot{\vec{\boldsymbol{\omega}}}^{DB}\right)_B^B}{\partial\left(\vec{\mathbf{z}}_d\right)_B}\right|_{\boldsymbol{\chi}_e} = k_a \cdot \boldsymbol{\Omega}\left(\left(\vec{\mathbf{z}}_B\right)_B\right),$$

$$\left.\frac{\partial\left(\dot{\vec{\boldsymbol{\omega}}}^{DB}\right)_B^B}{\partial\left(\vec{\boldsymbol{\omega}}^{DB}\right)_B}\right|_{\boldsymbol{\chi}_e} = -\mathbf{K}_\omega.$$

By defining the linearized states as $\delta\vec{\mathbf{z}}(t) := (\vec{\mathbf{z}}_d)_B(t) - (\vec{\mathbf{z}}_{d,e})_B(t)$ and $\delta\vec{\boldsymbol{\omega}}(t) := (\vec{\boldsymbol{\omega}}^{DB})_B(t) - (\vec{\boldsymbol{\omega}}_e^{DB})_B(t)$, the linear approximation is given by

$$\begin{pmatrix}\delta\dot{\vec{\mathbf{z}}}(t)\\ \delta\dot{\vec{\boldsymbol{\omega}}}(t)\end{pmatrix} = \begin{bmatrix}\mathbf{0}_{3\times3} & \boldsymbol{\Omega}\left(\left(\vec{\mathbf{z}}_{d,e}\right)_B\right)\\ k_a \cdot \boldsymbol{\Omega}\left(\left(\vec{\mathbf{z}}_B\right)_B\right) & -\mathbf{K}_\omega\end{bmatrix}\begin{pmatrix}\delta\vec{\mathbf{z}}(t)\\ \delta\vec{\boldsymbol{\omega}}(t)\end{pmatrix}.$$

Given that $(\vec{\mathbf{z}}_B)_B = [0,0,1]^T$ is constant in the body-fixed frame, the desired equilibrium $\boldsymbol{\chi}_{d,e}$ corresponds to $(\vec{\mathbf{z}}_{d,e})_B = (\vec{\mathbf{z}}_B)_B = [0,0,1]^T$ and the undesired equilibrium $\boldsymbol{\chi}_{u,e}$ to $(\vec{\mathbf{z}}_{d,e})_B = -(\vec{\mathbf{z}}_B)_B = [0,0,-1]^T$ Note that the state $[\delta\vec{\mathbf{z}}^T(t), \delta\vec{\boldsymbol{\omega}}^T(t)]^T \in \mathbb{R}^6$ is not a minimal representation since $dim(\mathbb{S}^2 \times \mathbb{R}^3) = 5$. From the structure of (2.14)

$$\boldsymbol{\Omega}\left((\vec{\mathbf{z}}_B)_B\right) = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \text{ and } \boldsymbol{\Omega}\left(-(\vec{\mathbf{z}}_B)_B\right) = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

it is clear that $\delta\vec{\mathbf{z}}_z(t)$ does not influences either $\delta\dot{\vec{\mathbf{z}}}(t)$ nor $\delta\dot{\vec{\boldsymbol{\omega}}}(t)$ and therefore it is only an integrator state. Furthermore, at the equilibrium points $(\vec{\mathbf{z}}_{d,e})_B = \pm[0,0,1]^T$ it holds that its derivative $\delta\dot{\vec{\mathbf{z}}}_z(t) = 0$. In fact, since $\dot{\vec{\mathbf{z}}}_d(t)$ always lies in a plane perpendicular to $\vec{\mathbf{z}}_d(t)$, two degrees of freedom can always be chosen for describing its motion linearly. At the equilibrium points $(\vec{\mathbf{z}}_{d,e})_B = \pm[0,0,1]^T$, this plane is the body-fixed xy-plane.

Using the diagonal structure of $\mathbf{K}_\omega = diag([k_{wx}, k_{wy}, k_{wz}]) > 0$, the reduced attitude linear dynamics for the desired equilibrium $\boldsymbol{\chi}_{d,e} = ((\vec{\mathbf{z}}_{d,e})_B, \vec{\boldsymbol{\omega}}_e^{DB}) = ([0,0,1]^T, \mathbf{0})$ are given by

$$\begin{pmatrix} \delta\dot{\vec{\mathbf{z}}}_x \\ \delta\dot{\vec{\mathbf{z}}}_y \\ \delta\dot{\vec{\boldsymbol{\omega}}}_x \\ \delta\dot{\vec{\boldsymbol{\omega}}}_y \\ \delta\dot{\vec{\boldsymbol{\omega}}}_z \end{pmatrix} = \begin{bmatrix} 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -k_a & -k_{wx} & 0 & 0 \\ k_a & 0 & 0 & -k_{wy} & 0 \\ 0 & 0 & 0 & 0 & -k_{wz} \end{bmatrix} \begin{pmatrix} \delta\vec{\mathbf{z}}_x \\ \delta\vec{\mathbf{z}}_y \\ \delta\vec{\boldsymbol{\omega}}_x \\ \delta\vec{\boldsymbol{\omega}}_y \\ \delta\vec{\boldsymbol{\omega}}_z \end{pmatrix}$$

By a simple rearrangement of the states as

$$\begin{pmatrix} \delta\dot{\vec{\mathbf{z}}}_x \\ \delta\dot{\vec{\boldsymbol{\omega}}}_y \\ \delta\dot{\vec{\mathbf{z}}}_y \\ \delta\dot{\vec{\boldsymbol{\omega}}}_x \\ \delta\dot{\vec{\boldsymbol{\omega}}}_z \end{pmatrix} = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 \\ k_a & -k_{wy} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -k_a & -k_{wx} & 0 \\ 0 & 0 & 0 & 0 & -k_{wz} \end{bmatrix} \begin{pmatrix} \delta\vec{\mathbf{z}}_x \\ \delta\vec{\boldsymbol{\omega}}_y \\ \delta\vec{\mathbf{z}}_y \\ \delta\vec{\boldsymbol{\omega}}_x \\ \delta\vec{\boldsymbol{\omega}}_z \end{pmatrix},$$

it can be seen that roll, pitch and yaw axes are independent from each other. Furthermore, the roll subsystem composed by $(\delta\vec{\mathbf{z}}_y(t), \delta\vec{\boldsymbol{\omega}}_x(t))$ and the pitch subsystem composed by $(\delta\vec{\mathbf{z}}_x(t), \delta\vec{\boldsymbol{\omega}}_y(t))$ are second order systems while the yaw subsystem composed by $\delta\vec{\boldsymbol{\omega}}_z(t)$ is of first order.

Because there is no coupling between the 3 subsystems, the eigenvalues can be computed for each of the submatrices. The eigenvalues of the roll subsystem are the solution of

$$\det(s \cdot \mathbf{I} - \mathbf{A}_{roll}) = s \cdot (s + k_{\omega x}) + k_a = s^2 + k_{\omega x}s + k_a \stackrel{!}{=} 0$$

and this corresponds to a stable second order filter with

$$\text{the eigenfrequency } \omega_r = \sqrt{k_a},$$
$$\text{the damping } \zeta_r = \frac{k_{\omega x}}{2\sqrt{k_a}},$$
$$\text{and the stable poles } \lambda_{r,1/2} = -\frac{1}{2}k_{\omega x} \pm \frac{1}{2}\sqrt{k_{\omega x}^2 - 4k_a} < 0.$$

The eigenvalues of the pitch subsystem are the solution of

$$\det(s \cdot \mathbf{I} - \mathbf{A}_{pitch}) = s \cdot (s + k_{\omega y}) + k_a = s^2 + k_{\omega y} s + k_a \stackrel{!}{=} 0$$

and this corresponds to a stable second order filter with

$$\begin{aligned} \text{the eigenfrequency } \omega_p &= \sqrt{k_a}, \\ \text{the damping } \zeta_p &= \frac{k_{\omega y}}{2\sqrt{k_a}}, \\ \text{and the stable poles } \lambda_{p,1/2} &= -\frac{1}{2} k_{\omega y} \pm \frac{1}{2}\sqrt{k_{\omega y}^2 - 4k_a} < 0. \end{aligned}$$

The eigenvalue of the yaw subsystem is the solution of

$$\det(s - A_{yaw}) = s + k_{\omega z} \stackrel{!}{=} 0$$

and is given by $\lambda_y = -k_{\omega z} < 0$. It is concluded that all the eigenvalues of the linearized subsystem at the equilibrium $\boldsymbol{\chi}_{d,e} = \left(\left(\vec{\mathbf{z}}_{d,e}\right)_B, \vec{\boldsymbol{\omega}}_e^{DB}\right) = \left([0,0,1]^T, \mathbf{0}\right)$ have negative real parts and therefore the equilibrium is stable.

For the undesired equilibrium $\boldsymbol{\chi}_{u,e} = \left(\left(\vec{\mathbf{z}}_{d,e}\right)_B, \vec{\boldsymbol{\omega}}_e^{DB}\right) = \left([0,0,-1]^T, \mathbf{0}\right)$, the reduced attitude linear dynamics are given by

$$\begin{pmatrix} \delta\dot{\vec{\mathbf{z}}}_x \\ \delta\dot{\vec{\mathbf{z}}}_y \\ \delta\dot{\vec{\boldsymbol{\omega}}}_x \\ \delta\dot{\vec{\boldsymbol{\omega}}}_y \\ \delta\dot{\vec{\boldsymbol{\omega}}}_z \end{pmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & -k_a & -k_{wx} & 0 & 0 \\ k_a & 0 & 0 & -k_{wy} & 0 \\ 0 & 0 & 0 & 0 & -k_{wz} \end{bmatrix} \begin{pmatrix} \delta\vec{\mathbf{z}}_x \\ \delta\vec{\mathbf{z}}_y \\ \delta\vec{\boldsymbol{\omega}}_x \\ \delta\vec{\boldsymbol{\omega}}_y \\ \delta\vec{\boldsymbol{\omega}}_z \end{pmatrix}.$$

Repeating the same state rearrangement as before

$$\begin{pmatrix} \delta\dot{\vec{\mathbf{z}}}_x \\ \delta\dot{\vec{\boldsymbol{\omega}}}_y \\ \delta\dot{\vec{\mathbf{z}}}_y \\ \delta\dot{\vec{\boldsymbol{\omega}}}_x \\ \delta\dot{\vec{\boldsymbol{\omega}}}_z \end{pmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ k_a & -k_{wy} & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & -k_a & -k_{wx} & 0 \\ 0 & 0 & 0 & 0 & -k_{wz} \end{bmatrix} \begin{pmatrix} \delta\vec{\mathbf{z}}_x \\ \delta\vec{\boldsymbol{\omega}}_y \\ \delta\vec{\mathbf{z}}_y \\ \delta\vec{\boldsymbol{\omega}}_x \\ \delta\vec{\boldsymbol{\omega}}_z \end{pmatrix}$$

it can be seen that roll, pitch and yaw axes are independent from each other.

Because there is no coupling between the thee subsystems, the eigenvalues can be computed from the submatrices as in the previous analysis. The eigenvalues of the roll subsystem are the solution of

$$\det(s \cdot \mathbf{I} - \mathbf{A}_{roll}) = s \cdot (s + k_{\omega x}) - k_a = s^2 + k_{\omega x} s - k_a \stackrel{!}{=} 0$$

and this corresponds to an unstable second order filter with

$$\begin{aligned} \text{one stable pole } \lambda_{r,1} &= -\frac{1}{2} k_{\omega x} - \frac{1}{2}\sqrt{k_{\omega x}^2 + 4k_a} < 0 \\ \text{and one unstable pole } \lambda_{r,2} &= -\frac{1}{2} k_{\omega x} + \frac{1}{2}\sqrt{k_{\omega x}^2 + 4k_a} > 0. \end{aligned}$$

The eigenvalues of the pitch subsystem are the solution of

$$\det(s \cdot \mathbf{I} - \mathbf{A}_{pitch}) = s \cdot (s + k_{\omega y}) - k_a = s^2 + k_{\omega y} s - k_a \stackrel{!}{=} 0$$

and this corresponds to an unstable second order filter with

$$\text{one stable pole } \lambda_{p,1} = -\frac{1}{2} k_{\omega y} - \frac{1}{2}\sqrt{k_{\omega y}^2 + 4k_a} < 0$$
$$\text{and one unstable pole } \lambda_{p,2} = -\frac{1}{2} k_{\omega y} + \frac{1}{2}\sqrt{k_{\omega y}^2 + 4k_a} > 0.$$

The eigenvalue of the yaw subsystem is the solution of

$$\det(s - A_{yaw}) = s + k_{\omega z} \stackrel{!}{=} 0$$

and is given by $\lambda_y = -k_{\omega z} < 0$ It is concluded that the linearized subsystem at the equilibrium $\boldsymbol{\chi}_{u,e} = ((\vec{\mathbf{z}}_{d,e})_B, \vec{\boldsymbol{\omega}}_e^{DB}) = ([0,0,-1]^T, \mathbf{0})$ has three eigenvalues with negative real parts and two eigenvalues with positive real parts. Hence, the equilibrium is an unstable saddle point. Following the argumentation of Theorem 3.1 in [55], this result can be extended to a symmetric positive definite matrix $\mathbf{K}_\omega$. ■

In order to proof of Theorem 5.9 (Local Exponential Stability) in Chapter 5, the following two lemmas are needed.

Lemma D.7 (Attitude Penalty Function Limits) *The attitude penalty function* (5.17)

$$V_\varphi\left(\varphi_d\left((\vec{\mathbf{z}}_d)_B(t)\right)\right) = 1 - \cos\varphi_d(t)$$

is bounded from below and above by

$$\frac{1}{2} \cdot ||\vec{\mathbf{e}}_a^{DB}(t)||^2 \leq V_\varphi\left(\varphi_d\left((\vec{\mathbf{z}}_d)_B(t)\right)\right) \leq r \cdot ||\vec{\mathbf{e}}_a^{DB}(t)||^2$$

within the set $V_\varphi(\varphi_d) \leq \overline{V}_\varphi < 2$ with a constant $\overline{V}_\varphi > 0 \in \mathbb{R}$. Here, $r \in \mathbb{R}$ is a finite constant that satisfies $\frac{1}{2} < 1/(2 - \overline{V}_\varphi) \leq r$. Note that this set corresponds to an upper limit on the error angle $\varphi_d(t)$ and that $V_\varphi(\pi) = 2$. Furthermore, the attitude error vector is defined as $\vec{\mathbf{e}}_a^{DB}(t) = \vec{\mathbf{z}}_d(t) \times \vec{\mathbf{z}}_B(t)$.

Proof: The norm of the attitude error vector $\vec{\mathbf{e}}_a^{DB}(t)$ can be computed using the fact that $\vec{\mathbf{z}}_d(t)$ and $\vec{\mathbf{z}}_B(t)$ are unit vectors as

$$\begin{aligned} ||\vec{\mathbf{e}}_a^{DB}(t)|| &= ||\vec{\mathbf{z}}_d(t)|| \cdot ||\vec{\mathbf{z}}_B(t)|| \cdot |\sin\varphi_d(t)|, \\ &= |\sin\varphi_d(t)|. \end{aligned}$$

For the sake of readability, the time dependency is not explicitly written in the following. The lower bound of the theorem translates to

$$\frac{1}{2}\sin^2\varphi_d \leq 1 - \cos\varphi_d$$

and follows from

$$\begin{aligned} \frac{1}{2}\sin^2\varphi_d &= \frac{1}{2}\left(1 - \cos^2\varphi_d\right), \\ &= \frac{1}{2}\left(1 + \cos\varphi_d\right)\left(1 - \cos\varphi_d\right), \\ &\leq 1 - \cos\varphi_d. \end{aligned}$$

The upper bound of the theorem can be rewritten in a similar manner as

$$\begin{aligned}(1-\cos\varphi_d) &\leq r\sin^2\varphi_d,\\ &\leq r\,(1+\cos\varphi_d)\,(1-\cos\varphi_d)\,.\end{aligned}$$

For $\varphi_d = 0$, the equality holds regardless of the value of r. For $\varphi_d \neq 0$, it holds that $V_\varphi(\varphi_d) = (1-\cos\varphi_d) > 0$. Hence, by dividing the last inequality by $(1-\cos\varphi_d)$ it suffices to show that

$$\begin{aligned}r\,(1+\cos\varphi_d) &\geq 1,\\ r &\geq \frac{1}{1+\cos\varphi_d} = \frac{1}{2-V_\varphi(\varphi_d)}.\end{aligned}$$

Taking the limit $V_\varphi(\varphi_d) \leq \overline{V}_\varphi < 2$ into account, a finite r exists for every constant $\overline{V}_\varphi$ such that

$$\frac{1}{2-V_\varphi(\varphi_d)} \leq \frac{1}{2-\overline{V}_\varphi} \leq r < \infty$$

holds. In the limit $\overline{V}_\varphi \to 2$ the value of r tends to infinity. The attitude penalty function $V_\varphi(\varphi_d)$, its lower bound and two upper bounds for different values of r can be seen in Figure D.2. ■

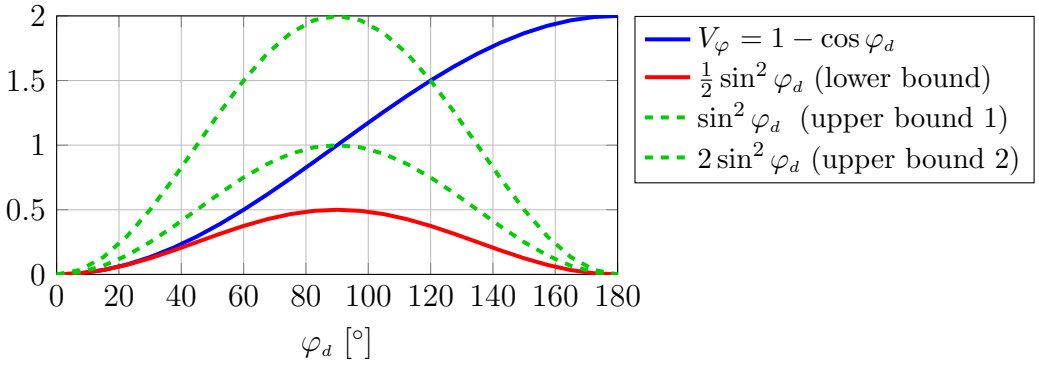


Figure D.2: Attitude Penalty Function and its Upper and Lower Bounds

Lemma D.8 (Attitude Lyapunov Function Limits) *For $V_\varphi(\varphi_d) \leq \overline{V}_\varphi < 2$, the attitude Lyapunov function* (5.18)

$$\begin{aligned}V((\vec{\mathbf{z}}_d)_B(t), \vec{\boldsymbol{\omega}}^{DB}(t)) = k_a \cdot V_\varphi((\varphi_d((\vec{\mathbf{z}}_d)_B(t))) + \frac{1}{2}(\vec{\boldsymbol{\omega}}^{DB})_B^T(t)(\vec{\boldsymbol{\omega}}^{DB})_B(t)\\ + c\cdot(\vec{\boldsymbol{\omega}}^{DB})_B^T(t)(\vec{\mathbf{e}}_a^{DB})_B(t),\end{aligned}$$

is bounded by

$$\boldsymbol{\zeta}^T(t)\cdot\mathbf{P}_1\cdot\boldsymbol{\zeta}(t) \leq V((\vec{\mathbf{z}}_d)_B(t), \vec{\boldsymbol{\omega}}^{DB}(t)) \leq \boldsymbol{\zeta}^T(t)\cdot\mathbf{P}_2\cdot\boldsymbol{\zeta}(t),$$

with the following definitions

$$\boldsymbol{\zeta}(t) = \begin{pmatrix}(\vec{\mathbf{e}}_a^{DB})_B(t)\\ (\vec{\boldsymbol{\omega}}^{DB})_B(t)\end{pmatrix}, \qquad \mathbf{P}_1 = \begin{bmatrix}\frac{1}{2}k_a\cdot\mathbf{I} & \frac{1}{2}c\cdot\mathbf{I}\\ \frac{1}{2}c\cdot\mathbf{I} & \frac{1}{2}\cdot\mathbf{I}\end{bmatrix}, \qquad \mathbf{P}_2 = \begin{bmatrix}r\cdot k_a\cdot\mathbf{I} & \frac{1}{2}c\cdot\mathbf{I}\\ \frac{1}{2}c\cdot\mathbf{I} & \frac{1}{2}\cdot\mathbf{I}\end{bmatrix}.$$

$\mathbf{P}_1, \mathbf{P}_2 \in \mathbb{R}^{6\times6}$ are symmetric positive definite if $k_a > 0$, if $|c| \leq \sqrt{k_a}$ and if $r = 1/(2-\overline{V}_\varphi)$ such that Lemma D.7 (Attitude Penalty Function Limits) holds.

Proof: Lemma D.7 (Attitude Penalty Function Limits) leads to the lower bound in Lemma D.8 as follows

$$\begin{aligned} V((\vec{\mathbf{z}}_d)_B(t), \vec{\boldsymbol{\omega}}^{DB}(t)) &\geq \frac{k_a}{2} \cdot (\vec{\mathbf{e}}_a^{DB})_B^T(t) (\vec{\mathbf{e}}_a^{DB})_B(t) + \frac{1}{2} (\vec{\boldsymbol{\omega}}^{DB})_B^T(t) (\vec{\boldsymbol{\omega}}^{DB})_B(t) \\ &\quad + c \cdot (\vec{\boldsymbol{\omega}}^{DB})_B^T(t) (\vec{\mathbf{e}}_a^{DB})_B(t), \\ &\geq \boldsymbol{\zeta}^T(t) \cdot \mathbf{P}_1 \cdot \boldsymbol{\zeta}(t). \end{aligned}$$

The upper bound can be computed analogously.

Given that $\mathbf{P}_1$ and $\mathbf{P}_2$ are symmetric block matrices, the Schur complement condition for positive definiteness can be used (Lemma B.12). For the matrix $\mathbf{P}_1$, it is known that the submatrix $\frac{1}{2} \cdot \mathbf{I}$ is positive definite and that the Schur complement of $\mathbf{P}_1$ is given by

$$\begin{aligned} \mathbf{S}_{p1} &= \left(\frac{1}{2} k_a \cdot \mathbf{I}\right) - \left(\frac{1}{2} c \cdot \mathbf{I}\right) \cdot \left(\frac{1}{2} \cdot \mathbf{I}\right)^{-1} \cdot \left(\frac{1}{2} c \cdot \mathbf{I}\right), \\ &= \frac{1}{2} k_a \cdot \mathbf{I} - \frac{1}{2} c^2 \cdot \mathbf{I}, \\ &= \left(\frac{1}{2} k_a - \frac{1}{2} c^2\right) \cdot \mathbf{I}. \end{aligned}$$

Hence, the matrix $\mathbf{S}_{p1}$ and therefore the matrix $\mathbf{P}_1$ are positive definite if $|c| < \sqrt{k_a}$. Computing the Schur complement condition for positive definiteness for the matrix $\mathbf{P}_2$ leads to the condition $|c| < \sqrt{2r \cdot k_a}$. If r is chosen as in Lemma D.7 (Attitude Penalty Function Limits), this is a redundant condition since $1 \leq 2r$ and $\sqrt{k_a} \leq \sqrt{2r \cdot k_a}$. Therefore, the matrix $\mathbf{P}_2$ is positive definite if the matrix $\mathbf{P}_1$ is positive definite. ■

In the following the proof of Theorem 5.9 (Local Exponential Stability) in Chapter 5 is given. For readability the lemma is rewritten.

Theorem D.9 (Local Exponential Stability) *Given the system dynamics* (5.13)*, the desired equilibrium* $\boldsymbol{\chi}_{d,e} = ((\vec{\mathbf{z}}_{d,e})_B, \vec{\boldsymbol{\omega}}_e^{DB}) = ([0,0,1]^T, \mathbf{0})$ *is locally exponentially stable and the Lyapunov function* (5.18) *decays exponentially given the initial conditions*

$$\begin{aligned} &V_\varphi(\varphi_d(t=0)) \leq \overline{V}_\varphi, \\ &||\vec{\boldsymbol{\omega}}^{DB}(t=0)|| \leq \sqrt{2k_a \cdot \left(\overline{V}_\varphi - V_\varphi(\varphi_d(t=0))\right)}, \end{aligned}$$

for a constant $\overline{V}_\varphi \in \mathbb{R}$ *such that* $0 < \overline{V}_\varphi < 2$.

The proof of this theorem goes along the argumentation lines of [95] and Theorem 4.10 (Uniform Exponential Stability) in [87]. Given the system dynamics (5.13), the derivative of the Lyapunov function (5.18) is limited from above as in (5.21) by

$$\dot{V}(t) \leq -\boldsymbol{\zeta}^T(t) \cdot \mathbf{Q} \cdot \boldsymbol{\zeta}(t) \leq -\underline{\lambda}(\mathbf{Q}) \cdot ||\boldsymbol{\zeta}(t)||^2, \tag{D.3}$$

where $\underline{\lambda}(\mathbf{Q})$ is the smallest eigenvalue of $\mathbf{Q}$. By choosing a positive definite matrix $\mathbf{Q}$ as in Lemma D.5 (Positive Definite $\mathbf{Q}$), it holds that $\underline{\lambda}(\mathbf{Q}) > 0$.

By choosing $r \in \mathbb{R}$ such that $\frac{1}{2} < 1/(2 - \overline{V}_\varphi) = r$, Lemma D.8 (Attitude Lyapunov Function Limits) holds for the set $V_\varphi(\varphi_d(t)) \leq \overline{V}_\varphi < 2$. The upper limit can be rewritten as

$$V(t) \leq \boldsymbol{\zeta}^T(t) \cdot \mathbf{P}_2 \cdot \boldsymbol{\zeta}(t) \leq \overline{\lambda}(\mathbf{P}_2) \cdot ||\boldsymbol{\zeta}(t)||^2, \tag{D.4}$$

where $\overline{\lambda}(\mathbf{P}_2) > 0 \in \mathbb{R}$ is the maximum eigenvalue of $\mathbf{P}_2$. By multiplying (D.4) with $-\frac{1}{\overline{\lambda}(\mathbf{P}_2)}$ it holds that

$$-\frac{V(t)}{\overline{\lambda}(\mathbf{P}_2)} \geq -||\boldsymbol{\zeta}(t)||^2. \tag{D.5}$$

Finally, inserting (D.5) in (D.3) leads to

$$\dot{V}(t) \leq -\frac{\underline{\lambda}(\mathbf{Q})}{\overline{\lambda}(\mathbf{P}_2)} \cdot V(t).$$

Hence, it holds that the attitude Lyapunov function (5.18) decays exponentially

$$V(t) \leq V(t=0) \cdot e^{-\frac{\underline{\lambda}(\mathbf{Q})}{\overline{\lambda}(\mathbf{P}_2)} \cdot t}$$

if $V_\varphi(\varphi_d(t)) \leq \overline{V}_\varphi < 2$ for all $t \geq 0$. The initial conditions in this theorem guarantee this constraint. First note that by choosing $c = 0$, the attitude Lyapunov function (5.18) is

$$V(t) = k_a \cdot V_\varphi(t) + \frac{1}{2} (\vec{\boldsymbol{\omega}}^{DB})_B^T(t) (\vec{\boldsymbol{\omega}}^{DB})_B(t).$$

From Lemma (5.5) it is known that it is positive definite. Furthermore, by inserting $c = 0$ into the derivative $\dot{V}(t)$ (5.21) it follows that it is never positive

$$\dot{V}(t) \leq -(\vec{\boldsymbol{\omega}}^{DB})_B^T(t) \cdot \mathbf{K}_\omega \cdot (\vec{\boldsymbol{\omega}}^{DB})_B(t) \leq 0.$$

Therefore every level set of the Lyapunov function with $c = 0$ is a positively invariant set. This means that the condition $V_\varphi(\varphi_d(t)) \leq \overline{V}_\varphi$ is satisfied for all $t \geq 0$ if $V(0) \leq k_a \cdot \overline{V}_\varphi$. This is the case if $V_\varphi(\varphi_d(t=0)) \leq \overline{V}_\varphi$ and $||\vec{\boldsymbol{\omega}}^{DB}(t=0)|| \leq \sqrt{2k_a \cdot \left(\overline{V}_\varphi - V_\varphi(\varphi_d(t=0))\right)}$. ■

D.2 Yaw Control Decoupling

In the following the proof of Lemma 5.13 (Roll/Pitch Attitude Error Derivative) in Chapter 5 is given. For readability the lemma is rewritten.

Lemma D.10 (Roll/Pitch Attitude Error Derivative) *The derivative of the first two elements of the attitude error vector $\vec{\mathbf{e}}_a^{DB}(t) = \vec{\mathbf{z}}_d(t) \times \vec{\mathbf{z}}_B(t)$ can be written as*

$$\left(\dot{\vec{\mathbf{e}}}_{a,xy}^{DB}\right)_B(t) = \mathbf{H}(t) \left(\vec{\mathbf{e}}_{a,xy}^{DB}\right)_B(t) + \left(\boldsymbol{\omega}_{xy}^{DB}\right)_B(t) (z_{d,z})_B(t) \tag{D.6}$$

using the time varying matrix

$$\mathbf{H}(t) = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \cdot (\omega_z^{DB})_B(t). \tag{D.7}$$

Proof: Using the kinematics of the desired vector $\vec{\mathbf{z}}_d(t)$ (5.13a), the derivative of the attitude error vector can be expanded to

$$\begin{aligned}
\left(\dot{\vec{\mathbf{e}}}_a^{DB}\right)_B^B(t) &= \left(\dot{\vec{\mathbf{z}}}_d\right)_B^B(t) \times \left(\vec{\mathbf{z}}_B\right)_B, \\
&= \left(-\left(\vec{\boldsymbol{\omega}}^{DB}\right)_B(t) \times \left(\vec{\mathbf{z}}_d\right)_B(t)\right) \times \left(\vec{\mathbf{z}}_B\right)_B, \\
&= -\begin{pmatrix} \left(\omega_y^{DB}\right)_B(t)\left(z_{d,z}\right)_B(t) - \left(\omega_z^{DB}\right)_B(t)\left(z_{d,y}\right)_B(t) \\ \left(\omega_z^{DB}\right)_B(t)\left(z_{d,x}\right)_B(t) - \left(\omega_x^{DB}\right)_B(t)\left(z_{d,z}\right)_B(t) \\ \left(\omega_x^{DB}\right)_B(t)\left(z_{d,y}\right)_B(t) - \left(\omega_y^{DB}\right)_B(t)\left(z_{d,x}\right)_B(t) \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \\
&= -\begin{pmatrix} \left(\omega_z^{DB}\right)_B(t)\left(z_{d,x}\right)_B(t) - \left(\omega_x^{DB}\right)_B(t)\left(z_{d,z}\right)_B(t) \\ -\left(\omega_y^{DB}\right)_B(t)\left(z_{d,z}\right)_B(t) + \left(\omega_z^{DB}\right)_B(t)\left(z_{d,y}\right)_B(t) \\ 0 \end{pmatrix}.
\end{aligned}$$

By taking the first two elements of the equation and noting that per definition

$$\left(\vec{\mathbf{e}}_{a,xy}^{DB}\right)_B(t) = \begin{pmatrix} \left(z_{d,y}\right)_B \\ -\left(z_{d,x}\right)_B \end{pmatrix},$$

the lemma is proven. ■

In the following the proof of Lemma 5.14 (Roll/Pitch Positive Definite $\overline{\mathbf{Q}}$) in Chapter 5 is given. For readability the lemma is rewritten.

Lemma D.11 (Roll/Pitch Positive Definite Q) *If $k_a, k_w \in \mathbb{R}$ are positive constants and $\left(\omega_z^{DB}\right)_B(t)$ is uniformly bounded by $\left|\left(\omega_z^{DB}\right)_B(t)\right| < r_{max}$, there exists a constant $c \in \left]0, \sqrt{k_a}\right[$ such that the matrix $\overline{\mathbf{Q}}$*

$$\overline{\mathbf{Q}}(t) = \begin{bmatrix} c \cdot k_a \cdot \mathbf{I} & \frac{c}{2}\left(k_w \cdot \mathbf{I} - \mathbf{H}^T(t)\right) \\ \frac{c}{2}\left(k_w \cdot \mathbf{I} - \mathbf{H}(t)\right) & (k_w - c) \cdot \mathbf{I} \end{bmatrix}$$

is positive definite.

Proof: Because the matrix $\overline{\mathbf{Q}}$ is a symmetric block matrix, Lemma B.12 (Schur condition for positive definiteness) is used to prove positive definiteness. Therefore it needs to be verified that the matrix $c \cdot k_a \cdot \mathbf{I}$ and the Schur complement of $\overline{\mathbf{Q}}$ given by

$$\begin{aligned}
\mathbf{S}_{\overline{q}} &= (k_w - c) \cdot \mathbf{I} - \frac{c}{2}\left(k_w \cdot \mathbf{I} - \mathbf{H}(t)\right) \cdot (c \cdot k_a \cdot \mathbf{I})^{-1} \cdot \frac{c}{2}\left(k_w \cdot \mathbf{I} - \mathbf{H}^T(t)\right), \\
&= (k_w - c) \cdot \mathbf{I} - \frac{c}{4 \cdot k_a}\left(k_w^2 \cdot \mathbf{I} - k_w \mathbf{H}^T(t) - \mathbf{H}(t)\, k_w + \mathbf{H}(t)\, \mathbf{H}^T(t)\right), \\
&= (k_w - c) \cdot \mathbf{I} - \frac{c}{4 \cdot k_a}\left(k_w^2 + \left(\omega_z^{DB}\right)_B^2(t)\right) \cdot \mathbf{I},
\end{aligned}$$

are positive definite. The following relations derived from the definition of $\mathbf{H}(t)$ (D.7) in Lemma D.10 (Roll/Pitch Attitude Error Derivative) have been used

$$\mathbf{H}^T(t) + \mathbf{H}(t) = \mathbf{0}, \qquad \mathbf{H}(t)\, \mathbf{H}^T(t) = \left(\omega_z^{DB}\right)_B^2(t) \cdot \mathbf{I}.$$

The matrix $c \cdot k_a \cdot \mathbf{I}$ is positive definite for $c > 0$ and $k_a > 0$. Since the matrix $\mathbf{S}_{\overline{q}}$ is diagonal, its entries correspond to its eigenvalues. Hence, for $\left(\omega_z^{DB}\right)_B(t) \leq r_{max}$ the matrix $\mathbf{S}_{\overline{q}}$ is positive definite if

$$\begin{aligned}
0 &< k_\omega - c \cdot \left(1 + \frac{1}{4 \cdot k_a}\left(k_\omega^2 + \left(\omega_z^{DB}\right)_B^2(t)\right)\right), \\
c &< \frac{k_\omega}{1 + \frac{1}{4 \cdot k_a}\left(k_\omega^2 + r_{max}^2\right)} =: c_{ub,2}\left(k_\omega, k_a, r_{max}\right).
\end{aligned}$$

Therefore, the condition that $\mathbf{S}_{\overline{q}}$ is positive definite translates to an upper bound. Note that the bound $c_{ub,2}\left(k_\omega, k_a, r_{max}\right)$ is always less or equal than $c_{ub}\left(k_\omega, k_a\right)$ in Lemma D.5 (Positive Definite **Q**)

$$c_{ub,2}\left(k_\omega, k_a, r_{max}\right) \leq \frac{k_\omega}{1+\frac{1}{4\cdot k_a}\left(k_\omega^2+\left(\omega_z^{DB}\right)_B^2(t)\right)} \leq \frac{k_\omega}{1+\frac{1}{4\cdot k_a}k_\omega^2} = c_{ub}\left(k_\omega, k_a\right)$$

and therefore $c_{ub,2}\left(k_\omega, k_a, r_{max}\right) \leq \sqrt{k_a}$. For $k_\omega > 0$ and a finite $r_{max} \geq 0$, the upper bound is positive $c_{ub,2}\left(k_\omega, k_a, r_{max}\right) > 0$ and therefore there exists a constant $c \in]0, \sqrt{k_a}[$ such that $\overline{\mathbf{Q}}$ is positive definite. ■

In the following the proof of Lemma 5.15 (Linearization: Roll/Pitch Reduced Attitude) in Chapter 5 is given. For readability the lemma is rewritten.

Lemma D.12 (Linearization: Roll/Pitch Reduced Attitude) *The reduced attitude closed-loop system* (5.22) *has a stable equilibrium at* $\overline{\boldsymbol{\chi}}_{d,e} = \left(\left(\vec{\mathbf{z}}_{d,e}\right)_B, \left(\boldsymbol{\omega}_{e,xy}^{DB}\right)_B\right) = \left([0,0,1]^T, \mathbf{0}\right)$ *and an unstable equilibrium at* $\overline{\boldsymbol{\chi}}_{u,e} = \left(\left(\vec{\mathbf{z}}_{d,e}\right)_B, \left(\boldsymbol{\omega}_{e,xy}^{DB}\right)_B\right) = \left([0,0,-1]^T, \mathbf{0}\right)$ *if* k_a *and* k_w *are positive constants. The subscript* e *refers to equilibrium points.*

Proof: From Theorem 5.10 (Roll/Pitch Attitude Equilibrium Points) it is known that the presented dynamic system has two equilibria $\overline{\boldsymbol{\chi}}_{d,e}$ and $\overline{\boldsymbol{\chi}}_{u,e}$. In order to asses their stability properties, the linear approximation of the dynamics at the two equilibria is analyzed. By noting that the reduced attitude closed-loop system (5.22) corresponds to a subsystem of (5.13) (or equivalently $(\mathrm{D}.1)$-$(\mathrm{D}.2)$) the results of Lemma D.6 (Linearization: Reduced Attitude) can be used. The linearized system in Lemma D.6 can be separated into two independent systems corresponding to the states $(\delta\vec{\mathbf{z}}_x, \delta\vec{\mathbf{z}}_y, \delta\vec{\boldsymbol{\omega}}_x, \delta\vec{\boldsymbol{\omega}}_y)$ and $\delta\vec{\boldsymbol{\omega}}_z$. The system (5.22) corresponds to $(\delta\vec{\mathbf{z}}_x, \delta\vec{\mathbf{z}}_y, \delta\vec{\boldsymbol{\omega}}_x, \delta\vec{\boldsymbol{\omega}}_y)$ and therefore the result follows directly from Lemma D.6 (Linearization: Reduced Attitude) . ■

D.3 Adaptive Attitude Control

In order to proof of Theorem 5.19 (Full State Predictor - Boundedness) in Chapter 5, the following lemma is needed.

Lemma D.13 (Attitude Lyapunov Function Limits 2) *The attitude Lyapunov function is defined by*

$$\begin{aligned}\overline{V}\left(\left(\vec{\mathbf{z}}_d\right)_B(t), \boldsymbol{\omega}_{xy}^{DB}(t)\right) = k_a \cdot V_\varphi\left(\varphi_d\left(\left(\vec{\mathbf{z}}_d\right)_B(t)\right)\right) + \frac{1}{2}\left(\boldsymbol{\omega}_{xy}^{DB}\right)_B^T(t)\left(\boldsymbol{\omega}_{xy}^{DB}\right)_B(t) \\ + c\cdot\left(\boldsymbol{\omega}_{xy}^{DB}\right)_B^T(t)\left(\mathbf{e}_{a,xy}^{DB}\right)_B(t)\end{aligned}$$

and is bounded by

$$\overline{V}\left(\left(\vec{\mathbf{z}}_d\right)_B(t), \boldsymbol{\omega}_{xy}^{DB}(t)\right) \leq 2\cdot k_a + \boldsymbol{\zeta}^T(t)\,\mathbf{P}_3\boldsymbol{\zeta}(t),$$

with the following definitions

$$\boldsymbol{\zeta} = \begin{pmatrix}\left(\mathbf{e}_{a,xy}^{DB}\right)_B(t) \\ \left(\boldsymbol{\omega}_{xy}^{DB}\right)_B(t)\end{pmatrix}, \qquad \mathbf{P}_3 = \begin{bmatrix}k_a\cdot\mathbf{I} & \frac{1}{2}c\cdot\mathbf{I} \\ \frac{1}{2}c\cdot\mathbf{I} & \frac{1}{2}\cdot\mathbf{I}\end{bmatrix}.$$

Furthermore, $\mathbf{P}_3 \in \mathbb{R}^{4\times4}$ *is symmetric positive definite if* $k_a > 0$ *and if* $|c| \leq \sqrt{2\cdot k_a}$.

Proof: First note that from $V_\varphi\left(\varphi_d\left(\left(\vec{\mathbf{z}}_d\right)_B(t)\right)\right) = 1 - \cos\varphi_d \leq 2$ it follows that

$$V_\varphi\left(\varphi_d\left(\left(\vec{\mathbf{z}}_d\right)_B(t)\right)\right) \leq 2 + \left|\left|\left(\mathbf{e}_{a,xy}^{DB}\right)_B(t)\right|\right|^2.$$

Then, it follows that

$$\begin{aligned}\overline{V}\left(\left(\vec{\mathbf{z}}_d\right)_B(t), \boldsymbol{\omega}_{xy}^{DB}(t)\right) &\leq k_a \cdot \left(2 + \left(\mathbf{e}_{a,xy}^{DB}\right)_B^T(t)\left(\mathbf{e}_{a,xy}^{DB}\right)_B(t)\right) + \frac{1}{2}\left(\boldsymbol{\omega}_{xy}^{DB}\right)_B^T(t)\left(\boldsymbol{\omega}_{xy}^{DB}\right)_B(t) \\ &\quad + c \cdot \left(\boldsymbol{\omega}_{xy}^{DB}\right)_B^T(t)\left(\mathbf{e}_{a,xy}^{DB}\right)_B(t), \\ &\leq 2 \cdot k_a + \boldsymbol{\zeta}^T(t) \cdot \mathbf{P}_3 \cdot \boldsymbol{\zeta}(t).\end{aligned}$$

Analogously to Lemma D.8 (Attitude Lyapunov Function Limits), it can be shown that the matrix $\mathbf{P}_3$ is positive definite if $|c| < \sqrt{2 \cdot k_a}$ using Lemma B.12 (Schur condition for positive definiteness). ■

In the following the proof of Theorem 5.19 (Full State Predictor - Boundedness) in Chapter 5 is given. For readability the lemma is rewritten.

Theorem D.14 (Full State Predictor - Boundedness) *Given the full state predictor dynamics* (5.38)*, the reference model* (5.27)-(5.29)*, the control law* (5.43)*, positive gains* $k_{wz,rm}$*,* $k_{a,rm}$*,* k_a*,* k_w*,* $k_{w,z} > 0$*, diagonal positive definite matrices* $\mathbf{K}_{\omega,rm}, \mathbf{L}$ *and bounded commands* $\vec{\mathbf{z}}_c(t), \left(w_{c,z}\right)_B(t)$*, it holds that:*

- *Given a bounded input* $\tilde{\tilde{\boldsymbol{\omega}}}^{NB}(t)$ *(estimation error), the error state* $\left(\left(\vec{\mathbf{z}}_d\right)_B(t), \hat{\vec{\mathbf{e}}}_w(t)\right)$ *is bounded.*
- *The trajectories approach the equilibrium set* $\left(\left(\vec{\mathbf{z}}_d\right)_B(t), \hat{\vec{\mathbf{e}}}_w(t)\right) \rightarrow \left(\pm[0,0,1]^T, \mathbf{0}\right)$ *for* $\left|\left|\tilde{\tilde{\boldsymbol{\omega}}}^{NB}\right|\right| \rightarrow 0$.

In Section 5.4, the candidate Lyapunov function (5.44) with the derivative (5.45) has been proposed in order to prove the theorem based on the argumentation lines of Theorem 4.18 in [87] (Ultimate Bound). In the following the first statement of the theorem is checked. Given an upper bound on the estimation error input $\left(\tilde{\boldsymbol{\omega}}_{xy}^{NB}\right)_B(t)$ as

$$\frac{\overline{b}}{\overline{q}}\left|\left|\left(\tilde{\boldsymbol{\omega}}_{xy}^{NB}\right)_B(t)\right|\right| \leq c_\zeta \tag{D.8}$$

with a positive constant $c_\zeta > 0 \in \mathbb{R}$, the following two sets can be defined. One is the set of error states B_ζ

$$B_\zeta := \left\{\hat{\boldsymbol{\zeta}}(t) \;\middle|\; \left|\left|\hat{\boldsymbol{\zeta}}(t)\right|\right| \leq c_\zeta\right\}$$

for which a negative derivative $\dot{\overline{V}}(t)$ (5.45) could not be shown. Outside this set it holds that

$$\left|\left|\hat{\boldsymbol{\zeta}}(t)\right|\right| > c_\zeta \geq \frac{\overline{b}}{\overline{q}}\left|\left|\left(\tilde{\boldsymbol{\omega}}_{xy}^{NB}\right)_B(t)\right|\right| \Rightarrow \dot{\overline{V}}(t) < 0. \tag{D.9}$$

The second set Ω_b is limited by a level set of the Lyapunov function $\overline{V}(t)$ (5.44) and is defined as

$$\begin{aligned}&\Omega_b := \left\{\left(\varphi_d(\left(\vec{\mathbf{z}}_d\right)_B(t)), \left(\hat{\boldsymbol{\omega}}_{xy}^{NB}\right)_B(t)\right) \;\middle|\; \overline{V}\left(\left(\vec{\mathbf{z}}_d\right)_B(t), \left(\hat{\boldsymbol{\omega}}_{xy}^{NB}\right)_B(t)\right) \leq c_v\right\} \\ &c_v = 2 \cdot k_a + \overline{\lambda}(\mathbf{P}_3) \cdot c_\zeta^2 > 0.\end{aligned}$$

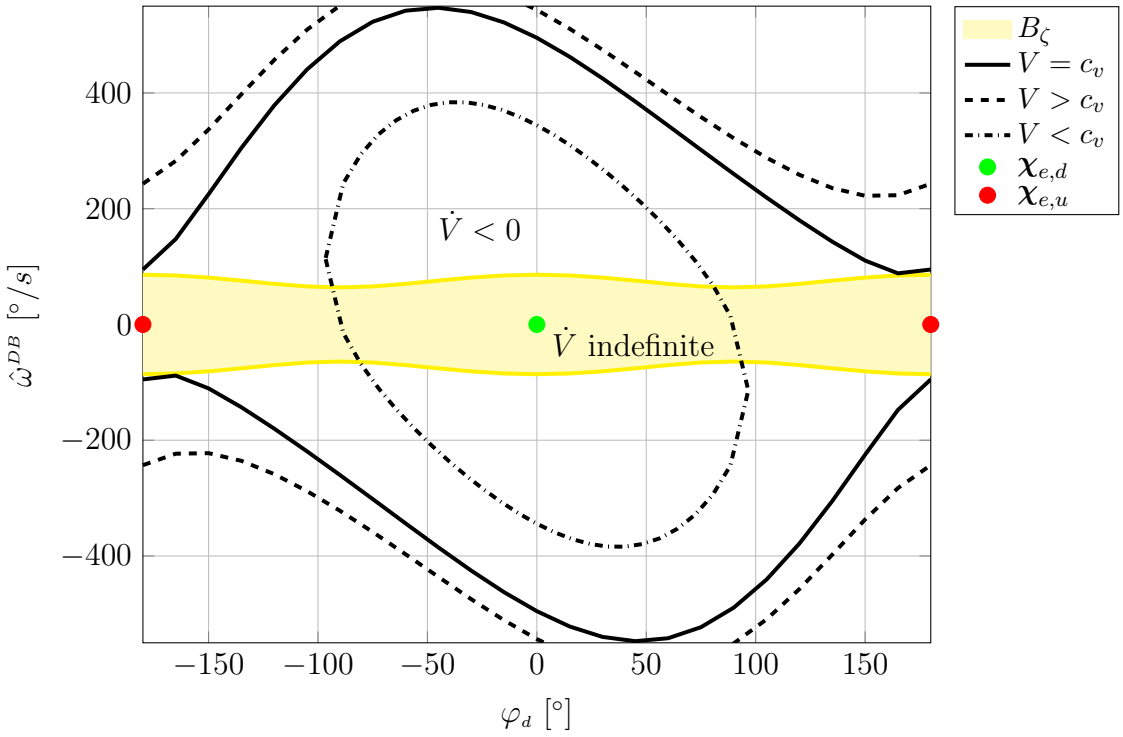


Figure D.3: Theorem D.14 - Lyapunov Analysis for $c_\zeta = 1.5$

Here, the constant $c_v > 0 \in \mathbb{R}$ defines the maximum value of the Lyapunov function $\overline{V}(t)$ and $\overline{\lambda}(\mathbf{P}_3) > 0 \in \mathbb{R}$ is the maximum eigenvalue of the symmetric positive definite matrix $\mathbf{P}_3$ defined in Lemma D.13 (Attitude Lyapunov Function Limits 2). In Figure D.3, the set B_ζ for $c_\zeta = 1.5$, different level sets of $\overline{V}(t)$ and the desired and undesired equilibria are depicted for a one dimensional version of the attitude Lyapunov function. Note that although two undesired equilibria are plotted in the figure, both represent the same equilibrium since $\varphi_d(t) = \pi$ and $\varphi_d(t) = -\pi$ are physically the same point.

The set Ω_b is defined such that $B_\zeta \subseteq \Omega_b$. This can be seen using Lemma D.13 (Attitude Lyapunov Function Limits 2) and checking that outside the set Ω_b for $\overline{V}(t) > c_v$ it holds that

$$c_v = 2 \cdot k_a + \overline{\lambda}(\mathbf{P}_3) \cdot c_\zeta^2 < \overline{V}(t) \leq 2 \cdot k_a + \overline{\lambda}(\mathbf{P}_3) \left|\left|\hat{\boldsymbol{\zeta}}(t)\right|\right|^2,$$
$$\Rightarrow c_\zeta < \left|\left|\hat{\boldsymbol{\zeta}}(t)\right|\right|.$$

Hence, the derivative of the Lyapunov function is negative $\dot{\overline{V}}(t) < 0$ outside the set Ω_b. Using the same argumentation, on the boundary of Ω_b, $\overline{V}(t) > c_v$, it holds that $\dot{\overline{V}}(t) \leq 0$. The set Ω_b is positively invariant since it is compact and $\dot{\overline{V}}(t) \leq 0$ holds on its boundaries [87]. Therefore, every trajectory starting within Ω_b stays in the set and remains bounded for $t \geq 0$. Furthermore, for every initial condition $(\varphi_d((\vec{\mathbf{z}}_d)_B(0), \hat{\vec{\boldsymbol{\omega}}}^{DB}(0))$ starting outside of the set Ω_b, $\dot{\overline{V}}(t) < 0$ guarantees boundedness of the error states and that the solutions converge to the set Ω_b for $t \to \infty$. This proves the first statement in Theorem D.14 (Full State Predictor - Boundedness).

Note that the boundary of the set Ω_b is given by the level set at c_v and that $c_v \to 2 \cdot k_a > 0$ for $c_\zeta \to 0$. Hence, this result cannot be directly used to prove the second statement in Theorem D.14 (Full State Predictor - Boundedness). In the following, the assessment is done for qualitatively different scenarios depending on the value of c_ζ. For $c_\zeta \leq 1$, the set B_ζ can be separated in two sets centered at the desired and undesired attitude errors $\varphi_d = \{0, \pi\}$. They are defined as

$$B_{\zeta,d} := \left\{ \hat{\boldsymbol{\zeta}}(t) \;\middle|\; \left|\left| \hat{\boldsymbol{\zeta}}(t) \right|\right| \leq c_\zeta \leq 1, |\varphi_d| \leq \frac{\pi}{2} \right\},$$
$$B_{\zeta,u} := \left\{ \hat{\boldsymbol{\zeta}}(t) \;\middle|\; \left|\left| \hat{\boldsymbol{\zeta}}(t) \right|\right| \leq c_\zeta \leq 1, |\varphi_d| \geq \frac{\pi}{2} \right\}.$$

The set $B_{\zeta,d}$ centered at origin is contained within a positively invariant set $\Omega_{d,in}$ and the set $B_{\zeta,u}$ lies outside another positively invariant set $\Omega_{d,out}$ (the border may contain points of $B_{\zeta,u}$). These positively invariant sets are defined as

$$\Omega_{d,in} := \left\{ (\varphi_d((\vec{\mathbf{z}}_d)_B(t)), \hat{\boldsymbol{\omega}}_{xy}^{DB}(t)) \;\middle|\; \overline{V}((\vec{\mathbf{z}}_d)_B(t), \hat{\boldsymbol{\omega}}_{xy}^{DB}(t)) \leq c_{v,in} \right\},$$
$$c_{v,in} = \overline{\lambda}(\mathbf{P}_2(r)) \cdot c_\zeta^2 \geq 0, \quad r = \frac{1}{1 + \sqrt{1 - c_\zeta^2}},$$
$$\Omega_{d,out} := \left\{ (\varphi_d((\vec{\mathbf{z}}_d)_B(t)), \hat{\boldsymbol{\omega}}_{xy}^{DB}(t)) \;\middle|\; \overline{V}((\vec{\mathbf{z}}_d)_B(t), \hat{\boldsymbol{\omega}}_{xy}^{DB}(t)) \leq c_{v,out} \right\},$$
$$c_{v,out} = \overline{\lambda}(\mathbf{P}_2(r)) \cdot c_\zeta^2 \geq 0, \quad r = \frac{1}{1 - \sqrt{1 - c_\zeta^2}}.$$

Here the positive symmetric matrix $\mathbf{P}_2(r)$ is defined in Lemma D.8 (Attitude Lyapunov Function Limits). In the boundaries of the sets $\Omega_{d,in}$ and $\Omega_{d,out}$ it holds that $\dot{\overline{V}}(t) \leq 0$. This can be seen using Lemma D.8 (Attitude Lyapunov Function Limits)

$$c_{v,in} = \overline{\lambda}(\mathbf{P}_2(r)) \cdot c_\zeta^2 = \overline{V}(t) \leq \overline{\lambda}(\mathbf{P}_2(r)) \cdot ||\boldsymbol{\zeta}(t)||^2,$$
$$\Rightarrow c_\zeta \leq \left|\left| \hat{\boldsymbol{\zeta}}(t) \right|\right| \Rightarrow \dot{\overline{V}}(t) \leq 0.$$

The same argumentation can be done for $\Omega_{d,out}$.

Both of these sets are positively invariant since they are compact and $\dot{\overline{V}}(t) \leq 0$ holds on their boundaries [87]. Furthermore, the sets only exist for $c_\zeta \leq 1$ and it holds that $c_{v,out} > c_{v,in}$ for $c_\zeta < 1$ and $c_{v,out} = c_{v,in}$ for $c_\zeta = 1$. In Figure D.4, the sets $B_{\zeta,d}$ and $B_{\zeta,u}$, the level sets $\overline{V} = c_v$, $\overline{V} = c_{v,in}$ and $\overline{V} = c_{v,out}$, and the desired and undesired equilibria are depicted for a one dimensional version of the attitude Lyapunov function for $c_\zeta = 1$. Figure D.5 shows the case $c_\zeta = 0.5$.

Given that $\dot{\overline{V}}(t) < 0$ holds for the set $\Omega_{d,out} \setminus \Omega_{d,in}$, i.e. for $c_{v,in} < \overline{V}((\vec{\mathbf{z}}_d)_B(t), \hat{\boldsymbol{\omega}}_{xy}^{DB}) < c_{v,out}$, the solutions of the dynamic system can be classified depending on the value of c_ζ as follows:

- Case $c_\zeta > 1$ (Figure D.3): for $t \to \infty$ the solutions converge to the set Ω_b.
- Case $c_\zeta = 1$ (Figure D.4): for $t \to \infty$ the solutions
  - converge to the set Ω_b and never enter $\Omega_{d,in}$,
  - or converge to the set $\Omega_{d,in}$.

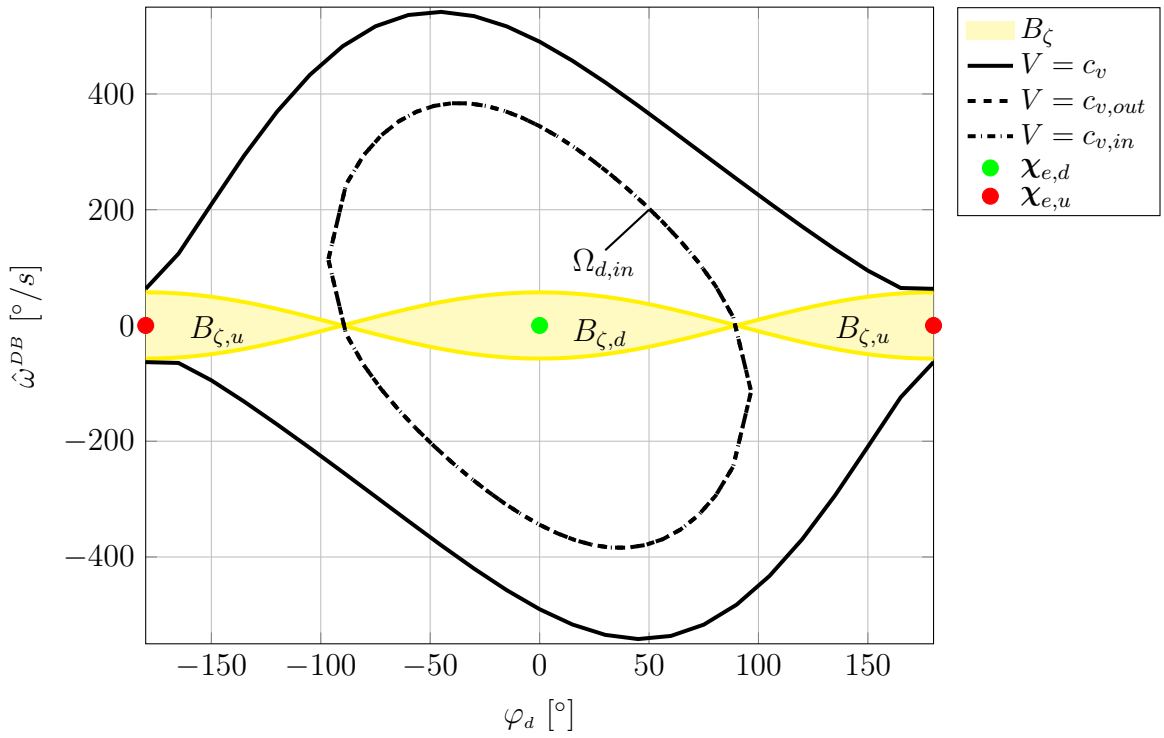


Figure D.4: Theorem D.14 - Lyapunov Analysis for $c_\zeta = 1$

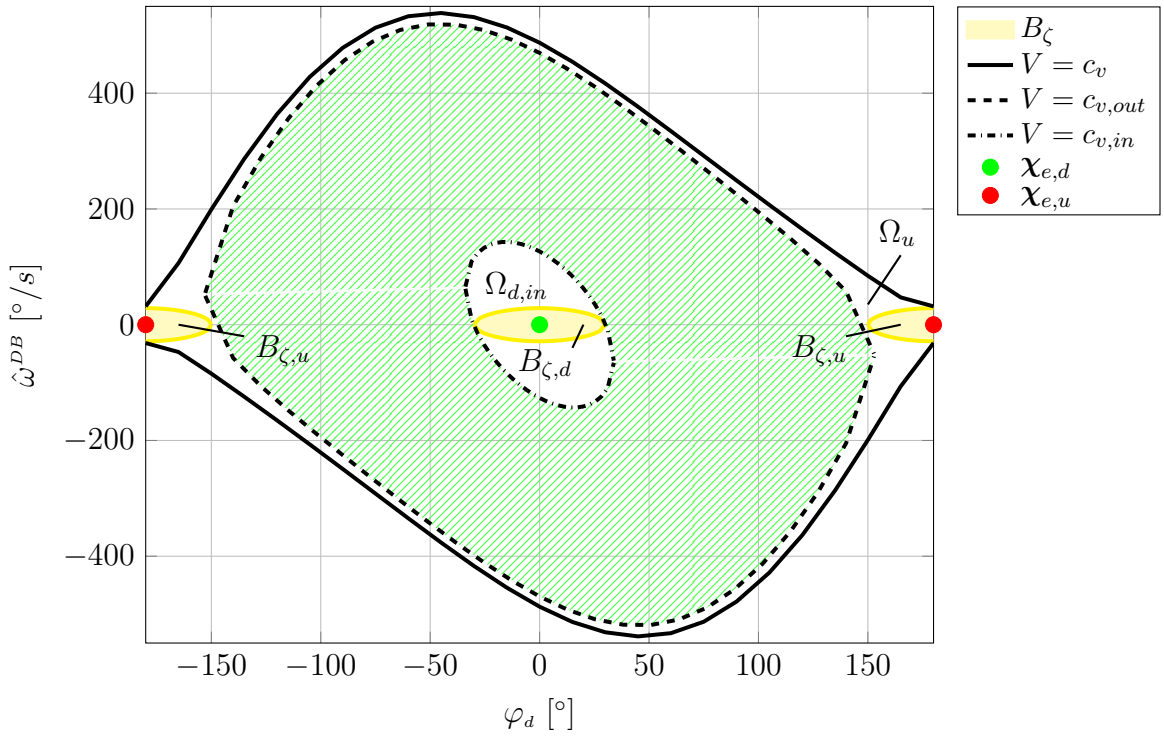


Figure D.5: Theorem D.14 - Lyapunov Analysis for $c_\zeta = 0.5$

- Case $0 < c_\zeta < 1$ (Figure D.5): for $t \to \infty$ the solutions
  - converge to the set
    $\Omega_u = \left\{(\varphi_d((\vec{\mathbf{z}}_d)_B(t)), \hat{\boldsymbol{\omega}}_{xy}^{DB}(t)) \mid c_{v,out} \leq \overline{V}((\vec{\mathbf{z}}_d)_B(t), \hat{\boldsymbol{\omega}}_{xy}^{DB}(t)) \leq c_v\right\}$,
  - or converge to the set $\Omega_{d,in}$.
- Case $c_\zeta = 0$: for $t \to \infty$ the solutions
  - converge to the undesired equilibrium $((\vec{\mathbf{z}}_d)_B(t), \hat{\boldsymbol{\omega}}_{xy}^{DB}(t)) \to ([0,0,-1]^T, \mathbf{0})$,
  - or converge to the desired equilibrium $((\vec{\mathbf{z}}_d)_B(t), \hat{\boldsymbol{\omega}}_{xy}^{DB}(t)) \to ([0,0,1]^T, \mathbf{0})$.

The case $c_\zeta = 0$ corresponds to the case without disturbance $\left(\tilde{\boldsymbol{\omega}}_{xy}^{NB}\right)_B(t) = \mathbf{0}$ (D.8) and therefore the result is equivalent to Theorem 5.11 presented in Section 5.3.2 (Yaw Control Decoupling). Furthermore, For $c_\zeta \to 0$, i.e. $\left(\tilde{\boldsymbol{\omega}}_{xy}^{NB}\right)_B(t) \to \mathbf{0}$, the sets Ω_b, $\Omega_{d,in}$ and $\Omega_{d,out}$ converge to

$$\begin{aligned}
\Omega_b &\to \left\{(\varphi_d((\vec{\mathbf{z}}_d)_B(t)), \hat{\boldsymbol{\omega}}_{xy}^{DB}(t)) \mid \overline{V}((\vec{\mathbf{z}}_d)_B(t), \hat{\boldsymbol{\omega}}_{xy}^{DB}(t)) \leq 2 \cdot k_a\right\} \\
\Omega_{d,in} &\to \left\{(\varphi_d((\vec{\mathbf{z}}_d)_B(t)), \hat{\boldsymbol{\omega}}_{xy}^{DB}(t)) \mid \overline{V}((\vec{\mathbf{z}}_d)_B(t), \hat{\boldsymbol{\omega}}_{xy}^{DB}(t)) \leq 0\right\} \\
\Omega_{d,out} &\to \left\{(\varphi_d((\vec{\mathbf{z}}_d)_B(t)), \hat{\boldsymbol{\omega}}_{xy}^{DB}(t)) \mid \overline{V}((\vec{\mathbf{z}}_d)_B(t), \hat{\boldsymbol{\omega}}_{xy}^{DB}(t)) \leq 2 \cdot k_a\right\}
\end{aligned}$$

.

Therefore statement 2 of Theorem D.14 (Full State Predictor - Boundedness) has been proved. ■

Acronyms

ACS	attainable control set
AGAS	almost globally asymptotically stable
AscTec	Ascending Technologies
CA	control allocation
ECEF	Earth-centered Earth-Fixed frame
ECI	Earth centered inertial frame
FCC	flight control computer
FDI	fault detection and isolation
FTC	fault tolerant control
GPS	global positioning system
IFCS	intelligent flight control system
IMU	inertial measurement unit
ISS	input-to-state stabiliy
LOC-I	loss of control in-flight
MRAC	model reference adaptive control
NDI	nonlinear dynamic inversion
NED	north-east-down frame
PCPH	position-command-position-hold
PRIO CA	prioritizing control allocation
PRO	parameter reduction due to overactuation
RCAH	Rate-command-attitude-hold
RESTORE	reconfigurable control for a tailless fighter aircraft
SVD	singular value decomposition
UAV	unmanned aerial vehicles
UCS	unattainable control set
VCPH	velocity-command-position-hold
VTOL	vertical take-off and landing

Symbols and Indices

Sets

$\mathcal{F}$	set of free/unsaturated inputs (control allocation)
$\mathbb{N}$	natural numbers
$\mathbb{R}$	real numbers
$\mathcal{S}$	set of saturated inputs (control allocation)
$\mathcal{S}_d$	set of directionally saturated inputs (control allocation)
$SO(3)$	special orthogonal group $\{\mathbf{M}\mathbf{M}^T = \mathbf{I}, \det(\mathbf{M}) = 1\}$
$\mathbb{S}^2$	unit sphere in $\mathbb{R}^3$ $\{\vec{\mathbf{v}} \in \mathbb{R}^3, \|\vec{\mathbf{v}}\| = 1\}$
$\mathcal{U}$	control input set
$\mathcal{V}$	attainable control set

Indices

B	body-fixed frame as defined in Appendix A
E	Earth-centered Earth-fixed frame as defined in Appendix A
I	Earth centered inertial Frame as defined in Appendix A
N	local navigation frame as defined in Appendix A
O	North-East-Down frame as defined in Appendix A

Symbols

$\mathbf{A}_m$	dynamic matrix of the reference model
$\mathbf{A}_p$	dynamic matrix of the plant
$\mathbf{B}_m$	input matrix of the reference model
$\mathbf{B}_p$	input matrix of the plant
$\mathbf{B}_a$	matrix mapping control inputs to virtual controls
$\mathbf{B}_{\nu u}$	normalized matrix mapping control inputs to virtual controls (control allocation)
$\mathbf{c}_k$	update direction in the input space at the k-th iteration (control allocation)
$\mathbf{D}, \mathbf{D}_a$	diagonal matrix containing singular values (SVD)
$\mathbf{d}$	constant disturbance vector
$\vec{\mathbf{e}}_a^{DB}$	attitude error vector
$\vec{\mathbf{e}}_w$	rotational rate tracking error w.r.t. reference model
$e_{v,z}$	vertical velocity tracking error w.r.t. reference model
$\vec{\mathbf{F}}$	force vector

g	constant gravitational acceleration
$\mathbf{G}_j$	directions of the low priority virtual controls and nullspace variables at the k-th iteration (control allocation)
$\mathbf{h}_k$	vector of help variables: scaled low priority virtual controls and nullspace variables at the k-th iteration (control allocation)
$(\mathbf{I}^R)_{BB}$	moment of inertia
L	roll moment
$\mathbf{L}$	Luenberger gain
M	pitch moment
$\vec{\mathbf{M}}$	moment vector
m	mass
$\mathbf{M}_{NB}$	rotation matrix, rotates a vector from the B into the N-frame
M	yaw moment
p	roll rate
$\mathbf{p}_j$	j-th column of the pseudo inverse $\mathbf{B}_{\nu u}^{+}$ (control allocation)
$\mathbf{P}, \overline{\mathbf{P}}$	solution matrix of the Lyapunov equation
p	pitch rate
$\mathbf{Q}, \overline{\mathbf{Q}}$	positive definite matrix
$\vec{\mathbf{r}}$	position vector
r	yaw rate
T	total thrust
$\mathbf{U}$	left singular vectors (SVD)
$\mathbf{U}_n$	left singular vectors that are the orthogonal complement of $\mathbf{U}_r$(SVD)
$\mathbf{U}_r$	left singular vectors that span the range of a matrix (SVD)
$\mathbf{u}$	control input vector
$V, \overline{V}$	Lyapunov function
$\mathbf{V}$	right singular vectors (SVD)
$\mathbf{V}_n$	right singular vectors that span the nullspace of a matrix (SVD)
$\mathbf{V}_r$	right singular vectors that are the orthogonal complement of $\mathbf{V}_n$ (SVD)
$\vec{\mathbf{v}}$	velocity vector
$\mathbf{x}_m$	state vector of the reference model
$\mathbf{x}_p$	state vector of the plant
$\vec{\mathbf{z}}_B$	body-fixed unit z vector
$\vec{\mathbf{z}}_d$	desired body-fixed unit z vector, tilt reference model
Δ_k	update length at the k-th iteration (control allocation)
$\gamma, \mathbf{\Gamma}$	aptation rates
$\mathbf{\Lambda}$	control effectiveness matrix
$\mathbf{\Lambda}_{rr}$	reduced control effectiveness matrix
$\boldsymbol{\nu}$	virtual controls vector
$\vec{\boldsymbol{\omega}}^{NB}$	rotational rate of the B-frame w.r.t. the N-frame
$\vec{\boldsymbol{\omega}}^{ND}$	rotational rate of the tilt reference model w.r.t. the N-frame
$\vec{\boldsymbol{\omega}}^{DB}$	rotational rate tracking error w.r.t. the tilt reference model

$\boldsymbol{\Omega}(\cdot)$	skew-symmetric matrix operator
Θ	pitch angle (Euler angles)
$\boldsymbol{\Theta}$	unknown constant parameter
$\boldsymbol{\Theta}_r$	reduced unknown constant parameter
$\boldsymbol{\Sigma}$	rectangular diagonal matrix containing singular values (SVD)
Φ	bank angle (Euler angles)
φ	tilt angle between $\vec{\mathbf{z}}_B$ and $\vec{\mathbf{z}}_N$
φ_d	tilt angle error $\vec{\mathbf{z}}_B$ and $\vec{\mathbf{z}}_d$
Ψ	azimuth angle (Euler angles)

Bibliography

[1] M. Achtelik, K.-M. Doth, D. Gurdan, and J. Stumpf. Design of a Multi Rotor MAV with regard to Efficiency, Dynamics and Redundancy. In *AIAA Guidance, Navigation, and Control Conference*. AIAA, 2012.

[2] Airbus. CityAirbus. `https://www.airbus.com/innovation/zero-emission/urban-air-mobility/cityairbus.html`, 13-Nov-20. Accessed: 15-Nov-20.

[3] V. S. Akkinapalli, G. P. Falconí, and F. Holzapfel. Attitude control of a multicopter using L1 augmented quaternion based backstepping. In *IEEE International Conference on Aerospace Electronics and Remote Sensing Technology (ICARES)*, pages 170–178. IEEE, 2014.

[4] V. S. Akkinapalli, G. P. Falconí, and F. Holzapfel. Fault tolerant incremental attitude control using online parameter estimation for a multicopter system. In *25th Mediterranean Conference on Control and Automation (MED)*, pages 454–460. IEEE, 2017.

[5] H. Alwi and C. Edwards. LPV sliding mode fault tolerant control of an octorotor using fixed control allocation. In *Conference on Control and Fault-Tolerant Systems (SysTol)*, 2013.

[6] M. H. Amoozgar, A. Chamseddine, and Y. Zhang. Fault-Tolerant Fuzzy Gain-Scheduled PID for a Quadrotor Helicopter Testbed in the Presence of Actuator Faults. *IFAC Proceedings Volumes*, 45(3):282–287, 2012.

[7] K. J. Åström and B. Wittenmark. *Adaptative control*. Addison-Wesley, Reading, 1989.

[8] Aurora Flight Sciences. PAV – eVTOL Passenger Air Vehicle. `https://www.aurora.aero/pav-evtol-passenger-air-vehicle/`, 14-Nov-20. Accessed: 15-Nov-20.

[9] Autel Robotics. EVO II specification. `https://auteldrones.com/pages/evo-ii-specification`, 15-Nov-20. Accessed: 15-Nov-20.

[10] S. P. Bhat and D. S. Bernstein. A topological obstruction to continuous global stabilization of rotational motion and the unwinding phenomenon. *Systems & Control Letters*, 39(1):63–70, 2000.

[11] T. Bierling. *Comparative Analysis of Adaptive Control Techniques for Improved Robust Performance*. Doctoral Dissertation, Technische Universität München, Munich, 2014.

[12] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki. *Diagnosis and fault-tolerant control*. Springer, Heidelberg and New York, 3rd edition, 2016.

[13] M. Bodson. Evaluation of Optimization Methods for Control Allocation. *Journal of Guidance, Control, and Dynamics*, 25(4):703–711, 2002.

[14] K. Bordignon and W. Durham. Null-space augmented solutions to constrained control allocation problems. In *Guidance, Navigation, and Control Conference*. AIAA, 1995.

[15] S. Bouabdallah and R. Siegwart. Backstepping and Sliding-mode Techniques Applied to an Indoor Micro Quadrotor. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2247–2252. IEEE, 2005.

[16] S. Bouabdallah and R. Siegwart. Full control of a quadrotor. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 153–158, 2007.

[17] D. Bozorgnia and A. K. Sedigh. A new control allocation methodology based on the pseudo inverse along the null space. In *21st Iranian Conference on Electrical Engineering (ICEE)*, pages 1–4. IEEE, 2013.

[18] J. S. Brinker and K. A. Wise. Flight Testing of Reconfigurable Control Law on the X-36 Tailless Aircraft. *Journal of Guidance, Control, and Dynamics*, 24(5):903–909, 2001.

[19] R. Brockhaus, W. Alles, and R. Luckner. *Flugregelung*. Springer Berlin, Berlin, 3rd edition, 2010.

[20] M. Buhl, O. Fritsch, and B. Lohmann. Exact Input-Output Linearization for the Translational Dynamics of a Quadrotor Helicopter. *at - Automatisierungstechnik*, 59(6):374–381, 2011.

[21] J. Burken, C. Hanson, J. Lee, and J. Kaneshige. Flight Test Comparison of Different Adaptive Augmentations of Fault Tolerant Control Laws for a Modified F-15 Aircraft. In *AIAA Infotech@Aerospace Conference*. AIAA, 2009.

[22] A. Casavola and E. Garone. Fault-tolerant adaptive control allocation schemes for overactuated systems. *International Journal of Robust and Nonlinear Control*, 20(17):1958–1980, 2010.

[23] P. Castillo, R. Lozano, and A. Dzul. Stabilization of a mini rotorcraft with four rotors. *IEEE Control Systems Magazine*, 25(6):45–55, 2005.

[24] A. Chamseddine, Y. Zhang, C.-A. Rabbath, J. Apkarian, and C. Fulford. Model Reference Adaptive Fault Tolerant Control of a Quadrotor UAV. In *Infotech@Aerospace*. AIAA, 2011.

[25] N. Chaturvedi, N. McClamroch, and D. Bernstein. Asymptotic Smooth Stabilization of the Inverted 3-D Pendulum. *IEEE Transactions on Automatic Control*, 54(6):1204–1215, 2009.

[26] N. Chaturvedi, A. Sanyal, and N. McClamroch. Rigid-Body Attitude Control. *IEEE Control Systems*, 31(3):30–51, 2011.

[27] G. Chowdhary, E. Johnson, M. S. Kimbrell, R. Chandramohan, and A. Calise. Flight Test Results of Adaptive Controllers in Presence of Severe Structural Damage. In *AIAA Guidance, Navigation, and Control Conference*. AIAA, 2010.

[28] G. Cimini and A. Bemporad. Complexity and convergence certification of a block principal pivoting method for box-constrained quadratic programs. *Automatica*, 100:29–37, 2019.

[29] R. Cohen, G. Davy, E. Feron, and P.-L. Garoche. Formal Verification for Embedded Implementation of Convex Optimization Algorithms. *IFAC-PapersOnLine*, 50(1):5867–5874, 2017.

[30] P. de Monte and B. Lohmann. Position trajectory tracking of a quadrotor helicopter based on L1 adaptive control. In *European Control Conference (ECC)*, 2013.

[31] DJI. Mavic Air 2 - Technische Daten. `https://www.dji.com/de/mavic-air-2/specs`, 15-Nov-20. Accessed: 15-Nov-20.

[32] G. R. Drozeski. *A Fault-Tolerant Control Architecture for Unmanned Aerial Vehicles*. Doctoral Dissertation, Georgia Institute of Technology, Atlanta, USA, 2005.

[33] G. Du, Q. Quan, and K.-Y. Cai. Additive-state-decomposition-based dynamic inversion stabilized control of a hexacopter subject to unknown propeller damages. In *32nd Chinese Control Conference (CCC)*, 2013.

[34] G.-X. Du, Q. Quan, and K.-Y. Cai. Controllability Analysis and Degraded Control for a Class of Hexacopters Subject to Rotor Failures. *Journal of Intelligent & Robotic Systems*, 78(1):143–157, 2015.

[35] W. Durham, K. A. Bordignon, and R. Beck. *Aircraft control allocation*. Aerospace series. John Wiley & Sons Ltd, Chichester, West Sussex, United Kingdom, 2017.

[36] W. C. Durham. Constrained control allocation. *Journal of Guidance, Control, and Dynamics*, 16(4):717–725, 1993.

[37] Z. Dydek, A. Annaswamy, and E. Lavretsky. Adaptive Control of Quadrotor UAVs in the Presence of Actuator Uncertainties. In *AIAA Infotech@Aerospace Conference*. AIAA, Atlanta, USA, 2010.

[38] Z. T. Dydek, A. M. Annaswamy, and E. Lavretsky. Adaptive Control and the NASA X-15-3 Flight Revisited. *IEEE Control Systems*, 30(3):32–48, 2010.

[39] Z. T. Dydek, A. M. Annaswamy, and E. Lavretsky. Adaptive Control of Quadrotor UAVs: A Design Trade Study With Flight Evaluations. *IEEE Transactions on Control Systems Technology*, 21(4):1400–1406, 2013.

[40] C. Edwards, T. Lombaerts, and H. Smaili, editors. *Fault Tolerant Flight Control*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[41] J. S. Eterno, J. L. Weiss, D. P. Looze, and A. Willsky. Design issues for fault tolerant-restructurable aircraft control. In *24th IEEE Conference on Decision and Control*, pages 900–905. IEEE, 1985.

[42] G. P. Falconí, J. Angelov, and F. Holzapfel. Hexacopter Outdoor Flight Test Results Using Adaptive Control Allocation Subject to an Unknown Complete Loss of One Propeller. In *3rd Conference on Control and Fault-Tolerant Systems (SysTol)*, pages 373–380, 2016.

[43] G. P. Falconí, J. Angelov, and F. Holzapfel. Adaptive Fault–Tolerant Position Control of a Hexacopter Subject to an Unknown Motor Failure. *International Journal of Applied Mathematics and Computer Science*, 28(2):309–321, 2018.

[44] G. P. Falconí, O. Fritsch, B. Lohmann, and F. Holzapfel. Admissible Thrust Control Laws for Quadrotor Position Tracking. In *Proceedings of the American Control Conference*, pages 4844–4849, 2013.

[45] G. P. Falconí, C. Heise, and F. Holzapfel. Fault-Tolerant Position Tracking of a Hexacopter using an Extended State Observer. In *6th International Conference on Automation, Robotics and Applications (ICARA)*, pages 550–556, 2015.

[46] G. P. Falconí, C. Heise, and F. Holzapfel. Novel Control Law for Predictor-Based MRAC for Overactuated Systems. In *14th International Conference on Control, Automation, Robotics & Vision (ICARCV)*, pages 1–6, 2016.

[47] G. P. Falconí, C. Heise, and F. Holzapfel. Novel Stability Analysis of Direct MRAC with Redundant Inputs. In *24th Mediterranean Conference on Control and Automation (MED)*, pages 176–181, 2016.

[48] G. P. Falconí and F. Holzapfel. Position Tracking of a Multicopter using a Geometric Backstepping Control Law. In *CEAS EuroGNC*, 2013.

[49] G. P. Falconí and F. Holzapfel. Position tracking of a hexacopter using a geometric backstepping control law - Experimental results. In *IEEE International Conference on Aerospace Electronics and Remote Sensing Technology (ICARES)*, pages 20–25. IEEE, 2014.

[50] G. P. Falconí and F. Holzapfel. Adaptive Fault Tolerant Control Allocation for a Hexacopter System. In *American Control Conference (ACC)*, pages 6760–6766, 2016.

[51] G. P. Falconí, V. A. Marvakov, and F. Holzapfel. Fault Tolerant Control for a Hexarotor System Using Incremental Backstepping. In *IEEE Multi-Conference on Systems and Control (CCA)*, pages 237–242, 2016.

[52] G. P. Falconí, S. P. Schatz, and F. Holzapfel. Fault Tolerant Control of a Hexarotor using a Command Governor Augmentation. In *24th Mediterranean Conference on Control and Automation (MED)*, pages 182–187, 2016.

[53] M. Fässler, D. Falanga, and D. Scaramuzza. Thrust Mixing, Saturation, and Body-Rate Control for Accurate Aggressive Quadrotor Flight. *IEEE Robotics and Automation Letters*, 2(2):476–482, 2017.

[54] A. Fekih. Fault diagnosis and Fault Tolerant Control design for aerospace systems: A bibliographical review. In *2014 American Control Conference*, pages 1286–1291. IEEE, 2014.

[55] O. Fritsch. *Energiebasierte Lage- und Positionsfolgeregelung für einen Quadrocopter*. Doctoral Dissertation, Technische Universität München, Munich, 2014.

[56] O. Fritsch, P. de Monte, M. Buhl, and B. Lohmann. Quasi-static feedback linearization for the translational dynamics of a quadrotor helicopter. In *American Control Conference (ACC)*, pages 125–130, 2012.

[57] Z. Gao, C. Cecati, and S. X. Ding. A Survey of Fault Diagnosis and Fault-Tolerant Techniques—Part I: Fault Diagnosis With Model-Based and Signal-Based Approaches. *IEEE Transactions on Industrial Electronics*, 62(6):3757–3767, 2015.

[58] Z. Gao, C. Cecati, and S. X. Ding. A Survey of Fault Diagnosis and Fault-Tolerant Techniques—Part II: Fault Diagnosis with Knowledge-Based and Hybrid/Active Approaches. *IEEE Transactions on Industrial Electronics*, 62(6):3768–3774, 2015.

[59] J. E. Gentle. *Matrix Algebra: Theory, Computations and Applications in Statistics*. Springer Texts in Statistics. Springer, Cham, second edition, 2017.

[60] J. I. Giribet, C. D. Pose, A. S. Ghersin, and I. Mas. Experimental Validation of a Fault Tolerant Hexacopter with Tilted Rotors. *International Journal of Electrical and Electronic Engineering & Telecommunications*, pages 58–65, 2018.

[61] F. Goßmann, V. S. Akkinapalli, G. P. Falconí, and F. Svaricek. Performance Oriented Fault Tolerant Attitude Control of a Hexacopter Using LPV-Control with Partly-Measurable Parameters. In *2018 IEEE Conference on Control Technology and Applications (CCTA)*, pages 722–728. IEEE, 2018.

[62] F. Goßmann, G. P. Falconí, and F. Svaricek. Fault tolerant control approach for passive handling of actuator degradation of a hexacopter using an LPV state-feedback controller with partly-measured parameters. In *25th Mediterranean Conference on Control and Automation (MED)*, pages 467–472. IEEE, 2017.

[63] I. Gregory, E. Xargay, C. Cao, and N. Hovakimyan. Flight Test of an L1 Adaptive Controller on the NASA AirSTAR Flight Test Vehicle. In *AIAA Guidance, Navigation, and Control Conference*. AIAA, 2010.

[64] D. Gurdan, J. Stumpf, M. Achtelik, K.-M. Doth, G. Hirzinger, and D. Rus. Energy-efficient Autonomous Four-rotor Flying Robot Controlled at 1 kHz. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 361–366. IEEE, 2007.

[65] O. Harkegard. Efficient active set algorithms for solving constrained least squares problems in aircraft control allocation. In *Proceedings of the 41st IEEE Conference on Decision and Control*, pages 1295–1300. IEEE, 2002.

[66] C. Heise, G. P. Falconí, and F. Holzapfel. Hexacopter outdoor flight test results of an Extended State Observer based controller. In *IEEE International Conference on Aerospace Electronics and Remote Sensing Technology (ICARES)*, pages 26–33. IEEE, 2014.

[67] C. D. Heise. *Survivable Flight Control with Guaranteed Stability and Performance Characteristics*. Doctoral Dissertation, Technische Universität München, Munich, 2017.

[68] M. Herceg, M. Kvasnica, C. Jones, and M. Morari. Multi-Parametric Toolbox 3.0. In *Proc. of the European Control Conference*, pages 502–510, Zürich, Switzerland, 2013.

[69] L. Höcht. *Advances in Stability Analysis for Model Reference Adaptive Control Systems and Application to Unmanned Aerial Systems*. Doctoral Dissertation, Technische Universität München, Garching, 2014.

[70] G. M. Hoffmann, Haomiao Huang, S. L. Wasl, and Er Claire J. Tomlin. Quadrotor helicopter flight dynamics and control: Theory and experiment. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, 2007.

[71] G. M. Hoffmann, H. Huang, S. L. Waslander, and C. J. Tomlin. Precision flight control for a multi-vehicle quadrotor helicopter testbed. *Control Engineering Practice*, 19(9):1023–1036, 2011.

[72] F. Holzapfel. *Flight System Dynamics 1 & 2 - Lecture Notes*. 2016.

[73] F. Holzapfel. *Nichtlineare adaptive Regelung eines unbemannten Fluggerätes*. Doctoral Dissertation, Technische Universität München, Munich, 2004.

[74] R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge University Press, New York, NY, second edition, 2013.

[75] N. Hovakimyan and C. Cao. *L1 adaptive control theory: Guaranteed robustness with fast adaptation*. Society for Industrial and Applied Mathematics, Philadelphia, 2010.

[76] International Air Transport Association. *Safety report 2017*. Montréal, Québec, Canada, 54th edition, 2018.

[77] P. A. Ioannou and P. V. Kokotović. *Adaptive systems with reduced models*, volume 47 of *Lecture Notes in Control and Information Sciences*. Springer, Berlin and Heidelberg, 2. pr edition, 1983.

[78] P. A. Ioannou and J. Sun. *Robust adaptive control*. PTR Prentice-Hall, Upper Saddle River, NJ, 1996.

[79] R. Isermann. *Fault diagnosis systems: An introduction from fault detection to fault tolerance*. Springer, Berlin and Heidelberg and New York, 2006.

[80] T. A. Johansen. Toward Dependable Embedded Model Predictive Control. *IEEE Systems Journal*, 11(2):1208–1219, 2017.

[81] T. A. Johansen and T. I. Fossen. Control allocation—A survey. *Automatica*, 49(5):1087–1103, 2013.

[82] E. N. Johnson and A. J. Calise. Limited Authority Adaptive Flight Control for Reusable Launch Vehicles. *Journal of Guidance, Control, and Dynamics*, 26(6):906–913, 2003.

[83] N. E. Kahveci and P. A. Ioannou. An indirect adaptive control design with anti-windup compensation: Stability analysis. In *2007 46th IEEE Conference on Decision and Control*, pages 1294–1299. IEEE, 2007.

[84] N. E. Kahveci and P. A. Ioannou. Indirect adaptive control for systems with input rate saturation. In *2008 American Control Conference*, pages 3396–3401. IEEE, 2008.

[85] S. P. Karason and A. M. Annaswamy. Adaptive control in the presence of input constraints. *IEEE Transactions on Automatic Control*, 39(11):2325–2330, 1994.

[86] F. Kendoul, Y. Zhenyu, and K. Nonami. Embedded autopilot for accurate waypoint navigation and trajectory tracking: Application to miniature rotorcraft UAVs. In *2009 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2884–2890, 2009.

[87] H. K. Khalil. *Nonlinear systems*. Prentice Hall, Upper Saddle River and N.J, 3rd edition, 2002.

[88] P. Kotaru, R. Edmonson, and K. Sreenath. Geometric L1 Adaptive Attitude Control for a Quadrotor Unmanned Aerial Vehicle. *Journal of Dynamic Systems, Measurement, and Control*, 142(3), 2020.

[89] M. Krstić, I. Kanellakopoulos, and P. V. Kokotović. *Nonlinear and adaptive control design*. Wiley, New York, 1995.

[90] E. Lavretsky and N. Hovakimyan. Positive μ-modification for stable adaptation in the presence of input constraints. In *Proceedings of the 2004 American Control Conference*, pages 2545–2550 vol.3. IEEE, 2004.

[91] E. Lavretsky and N. Hovakimyan. Stable Adaptation in the Presence of Multiple Actuator Constraints. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*. AIAA, 2006.

[92] E. Lavretsky and K. A. Wise. *Robust and adaptive control: With aerospace applications*. Advanced textbooks in control and signal processing. Springer, London and New York, 2013.

[93] T. Lee. Robust Adaptive Attitude Tracking on $\mathrm{SO}(3)$ With an Application to a Quadrotor UAV. *IEEE Transactions on Control Systems Technology*, 21(5):1924–1930, 2013.

[94] T. Lee. Global Exponential Attitude Tracking Controls on SO(3). *IEEE Transactions on Automatic Control*, 60(10):2837–2842, 2015.

[95] T. Lee, M. Leok, and N. H. McClamroch. Geometric Tracking Control of a Quadrotor UAV on SE(3). In *Proceedings of the 49th IEEE Conference on Decision and Control*, 2010.

[96] A. Leonessa, W. M. Haddad, T. Hayakawa, and Y. Morel. Adaptive control for nonlinear uncertain systems with actuator amplitude and rate saturation constraints. *International Journal of Adaptive Control and Signal Processing*, 23(1):73–96, 2009.

[97] D. Li, N. Hovakimyan, and C. Cao. L1 Adaptive Controller in the Presence of Input Saturation. In *AIAA Guidance, Navigation, and Control Conference*. AIAA, 2009.

[98] H. Liu, D. Derawi, J. Kim, and Y. Zhong. Robust optimal attitude control of hexarotor robotic vehicles. *Nonlinear Dynamics*, 74(4):1155–1168, 2013.

[99] T. J. J. Lombaerts. *Fault Tolerant Flight Control: A Physical Model Approach*. Doctoral Dissertation, Technische Universiteit Delft, Delft, 2010.

[100] J. Lunze. *Regelungstechnik 2*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2020.

[101] P. J. Lunze. *Regelungstechnik 1*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2020.

[102] S. Lupashin, A. Schöllig, M. Sherback, and R. D'Andrea. A simple learning strategy for high-speed quadrocopter multi-flips. In *2010 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1642–1648, 2010.

[103] S. Mallikarjunan, B. Nesbitt, E. Kharisov, E. Xargay, N. Hovakimyan, and C. Cao. L1 Adaptive Controller for Attitude Control of Multirotors. In *AIAA Guidance, Navigation, and Control Conference*. AIAA, 2012.

[104] A. Marks, J. F. Whidborne, and I. Yamamoto. Control allocation for fault tolerant control of a VTOL octorotor. In *Proceedings of 2012 UKACC International Conference on Control*, pages 357–362. IEEE, 2012.

[105] R. Matsuda, T. Ibuki, and M. Sampei. A Hoverability Analysis Method for Multirotor UAVs with a Case Study on Fault Tolerance. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 4264–4269. IEEE, 2018.

[106] H. Mazeh, M. Saied, H. Shraim, and C. Francis. Fault-Tolerant Control of an Hexarotor Unmanned Aerial Vehicle Applying Outdoor Tests and Experiments. *IFAC-PapersOnLine*, 51(22):312–317, 2018.

[107] D. Mellinger and V. Kumar. Minimum Snap Trajectory Generation and Control for Quadrotors. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[108] D. Mellinger, N. Michael, and V. Kumar. Trajectory Generation and Control for Precise Aggressive Maneuvers with Quadrotors. In *Proceedings of the International Symposium on Experimental Robotics*, 2010.

[109] A.-R. Merheb, H. Noura, and F. Bateman. Emergency Control of AR Drone Quadrotor UAV Suffering a Total Loss of One Rotor. *Mechatronics, IEEE/ASME Transactions on*, 22(2):961–971, 2017.

[110] K. Meyberg and P. Vachenauer. *Höhere Mathematik 1*. Springer Berlin Heidelberg, 2001.

[111] G. Michieletto, M. Ryll, and A. Franchi. Control of Statically Hoverable Multi-Rotor Aerial Vehicles and Application to Rotor-Failure Robustness for Hexarotors. In *2017 IEEE International Conference on Robotics and Automation*, Singapore, Singapore, 2017.

[112] B. Michini and J. How. L1 Adaptive Control for Indoor Autonomous Vehicles: Design Process and Flight Testing. In *AIAA Guidance, Navigation, and Control Conference*. AIAA, 2009.

[113] P. Monzón. *Almost Global Stability of Dynamical Systems*. Doctoral Dissertation, Universidad de la República, Montevideo, 2006.

[114] M. W. Mueller and R. D'Andrea. Stability and control of a quadrocopter despite the complete loss of one, two, or three propellers. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[115] M. Mühlegg, P. Niermeyer, G. P. Falconí, and F. Holzapfel. L1 Fault Tolerant Adaptive Control of a Hexacopter with Control Degradation. In *IEEE Conference on Control Applications (CCA)*, pages 750–755. IEEE, 2015.

[116] K. Narendra and A. Annaswamy. A new adaptive law for robust adaptation without persistent excitation. *IEEE Transactions on Automatic Control*, 32(2):134–145, 1987.

[117] K. S. Narendra and A. M. Annaswamy. *Stable adaptive systems*. Prentice Hall, Englewood Cliffs and N.J, 1989.

[118] D.-T. Nguyen, D. Saussie, and L. Saydy. Fault-Tolerant Control of a Hexacopter UAV based on Self-Scheduled Control Allocation. In *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 385–393. IEEE, 2018.

[119] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer New York, 2006.

[120] M. W. Oppenheimer, D. B. Doman, and M. A. Bolender. Control Allocation. In W. S. Levine, editor, *The control handbook*, The electrical engineering handbook series. CRC Press, Boca Raton, Fla., 2011.

[121] R. J. Patton. Fault-Tolerant Control: The 1997 Situation. *IFAC Proceedings Volumes*, 30(18):1029–1051, 1997.

[122] R. J. Patton. Fault-Tolerant Control. In J. Baillieul and T. Samad, editors, *Encyclopedia of Systems and Control*, pages 422–428. Springer London, London, 2015.

[123] J. A. M. Petersen and M. Bodson. Interior-Point Algorithms for Control Allocation. *Journal of Guidance, Control, and Dynamics*, 28(3):471–480, 2005.

[124] B. Peterson and K. Narendra. Bounded error adaptive control. *IEEE Transactions on Automatic Control*, 27(6):1161–1168, 1982.

[125] J.-B. Pomet and L. Praly. Adaptive nonlinear regulation: estimation from the Lyapunov equation. *IEEE Transactions on Automatic Control*, 37(6):729–740, 1992.

[126] X. Qi, D. Theilliol, J. Qi, Y. Zhang, J. Han, D. Song, L. Wang, and Y. Xia. Fault diagnosis and fault tolerant control methods for manned and unmanned helicopters: A literature review. In *Conference on Control and Fault-Tolerant Systems (SysTol)*, pages 132–139, 2013.

[127] Quantum-Systems. Trinity F90. `https://www.quantum-systems.com/project/trinity-f90/`, 29-Oct-20. Accessed: 15-Nov-20.

[128] T. Raffler, J. Wang, and F. Holzapfel. Path Generation and Control for Unmanned Multirotor Vehicles Using Nonlinear Dynamic Inversion and Pseudo Control Hedging. In *Automatic Control in Aerospace*, IFAC proceedings volumes, pages 194–199. Elsevier, IFAC, 2013.

[129] S. Richter, C. N. Jones, and M. Morari. Certification aspects of the fast gradient method for solving the dual of parametric convex programs. *Mathematical Methods of Operations Research*, 77(3):305–321, 2013.

[130] D. Rotondo, F. Nejjari, and V. Puig. Robust quasi-LPV model reference FTC of a quadrotor UAV subject to actuator faults. *International Journal of Applied Mathematics and Computer Science*, 25(1), 2015.

[131] I. Sadeghzadeh, A. Mehta, Y. Zhang, and C.-A. Rabbath. Fault-Tolerant Trajectory Tracking Control of a Quadrotor Helicopter Using Gain-Scheduled PID and Model Reference Adaptive Control. In *Annual Conference of the Prognostics and Health Management Society*. PHM Society, Montréal, 2011.

[132] S. Sastry and M. Bodson. *Adaptive control: Stability, convergence, and robustness*. s.n, Englewood Cliffs, NJ, 1989.

[133] T. Schneider, G. Ducard, K. Rudin, and P. Strupler. Fault-tolerant Control Allocation for Multirotor Helicopters using Parametric Programming. In *International Micro Air Vehicle Conference (IMAV)*, 2012.

[134] M. Schwager and A. M. Annaswamy. Direct Adaptive Control of Multi-Input Plants with Magnitude Saturation Constraints. In *Proceedings of the 44th IEEE Conference on Decision and Control*, 2005.

[135] Y. Sheng, G. Tao, and P. Beling. An Adaptive Actuator Failure Compensation Scheme for A Hexarotor System with Parameter Uncertainties. In *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 721–730. IEEE, 2018.

[136] M. Shuster. A Survey of Attitude Representations. *The Journal of the Astronautical Sciences*, 41(4):439–517, 1993.

[137] M. Steinberg. Historical Overview of Research in Reconfigurable Flight Control. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 219(4):263–275, 2005.

[138] G. Tao. *Adaptive control of systems with actuator failures*. Springer, London, 2004.

[139] F. M. Thiel. *Adaptive control of plants with input saturation*. Doctoral Dissertation, Universität Rostock, Rostock, 2019.

[140] J. Tjønnås and T. A. Johansen. Optimizing Nonlinear Adaptive Control Allocation. *IFAC Proceedings Volumes*, 38(1):1160–1165, 2005.

[141] J. Tjønnås and T. A. Johansen. Adaptive control allocation. *Automatica*, 44(11):2754–2765, 2008.

[142] S. S. Tohidi, A. Khaki Sedigh, and D. Buzorgnia. Fault tolerant control design using adaptive control allocation based on the pseudo inverse along the null space. *International Journal of Robust and Nonlinear Control*, 26(16):3541–3557, 2016.

[143] S. S. Tohidi, Y. Yildiz, and I. Kolmanovsky. Adaptive Control Allocation for Over-Actuated Systems with Actuator Saturation. *IFAC-PapersOnLine*, 50(1):5492–5497, 2017.

[144] S. S. Tohidi, Y. Yildiz, and I. Kolmanovsky. Model Reference Adaptive Control Allocation for Constrained Systems with Guaranteed Closed Loop Stability. *ArXiv*, 2019.

[145] M. Valenti, B. Bethke, G. Fiore, J. How, and E. Feron. Indoor Multi-Vehicle Flight Testbed for Fault Detection, Isolation, and Recovery. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*. AIAA, 2006.

[146] D. Vey and J. Lunze. Structural reconfigurability analysis of multirotor UAVs after actuator failures. In *54th IEEE Conference on Decision and Control (CDC)*. IEEE, 2015.

[147] D. Vey and J. Lunze. Experimental evaluation of an active fault-tolerant control scheme for multirotor UAVs. In *3rd Conference on Control and Fault-Tolerant Systems (SysTol)*, pages 125–132, 2016.

[148] Volocopter. VoloCity. `https://www.volocopter.com/en/product/`, 06-Nov-20. Accessed: 15-Nov-20.

[149] J. Wang, T. Raffler, and F. Holzapfel. Nonlinear Position Control Approaches for Quadrocopters Using a Novel State Representation. In *Proceedings of the AIAA Guidance, Navigation and Control Conference*. AIAA, 2012.

[150] T. Wang, R. Jobredeaux, M. Pantel, P.-L. Garoche, E. Feron, and D. Henrion. Credible autocoding of convex optimization algorithms. *Optimization and Engineering*, 17(4):781–812, 2016.

[151] W. Wang and C. Wen. Adaptive actuator failure compensation control of uncertain nonlinear systems with guaranteed transient performance. *Automatica*, 46(12):2082–2091, 2010.

[152] Wingcopter GmbH. Technology. `https://wingcopter.com/technology`, 15-Nov-20. Accessed: 15-Nov-20.

[153] Wisk. Meet Cora. Discover the future. `https://wisk.aero/cora/`, 01-Jun-20. Accessed: 15-Nov-20.

[154] Y. Yang, D. Iwakura, A. Namiki, K. Nonami, and W. Wang. Autonomous Flight of Hexacopter Under Propulsion System Failure. *Journal of Robotics and Mechatronics*, 28(6):899–910, 2016.

[155] F. Zhang. *The Schur complement and its applications*, volume 4 of *Numerical Methods and Algorithms*. Springer, New York, 2010.

[156] Y. Zhang and J. Jiang. Bibliographical review on reconfigurable fault-tolerant control systems. *Annual Reviews in Control*, 32(2):229–252, 2008.

[157] Z. Zuo and S. Mallikarjunan. $\mathcal{L}_1$ adaptive backstepping for robust trajectory tracking of uavs. *IEEE Transactions on Industrial Electronics*, 64(4):2944–2954, 2017.