TECHNISCHE UNIVERSITÄT MÜNCHEN

Lehrstuhl für Flugsystemdynamik

A Total Capability Approach for the Development of Safety-Critical Functions

Dipl.-Ing. Univ. David Löbl

Vollständiger Abdruck der von der Fakultät für Maschinenwesen der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender:	Prof. Dr. Markus Zimmermann
Prüfer der Dissertation:	1. Prof. Dr.-Ing. Florian Holzapfel
	2. Prof. Dr.-Ing. Manfred Hajek

Die Dissertation wurde am 16.01.2018 bei der Technischen Universität München eingereicht und durch die Fakultät für Maschinenwesen am 22.05.2018 angenommen.

Abstract

The development and certification of safety-critical functions for aircraft with novel topologies or operations pose a serious challenge when it is required to follow common, experience-based certification specifications, which prescribe specific design solutions to ensure safety. A lack in past experience for novel applications makes this conventional approach unfeasible.
The research leading to this thesis aims at filling this specification gap by a model-based development approach, the so-called *Total Capability Approach* (TCA). The idea is to consider the total capabilities of a system and its components in the presence of uncertainties, disturbances and failures in an integrated manner throughout the whole development process. A physically motivated break-down of safety-driven top-level requirements results in design specifications that are specifically derived for the intended application. The resulting specifications are not as conservative as prescriptive design requirements for common applications, since they do not have to ensure safety for a broad range of possible designs.
The shift in paradigm towards the proposed top-down requirements derivation process comes along with the need for more sophisticated verification methods, but also enables additional powerful means for validation, design and implementation of safety-critical functions. This dissertation presents the concept of this novel and promising development approach, it discusses challenges and objectives and proposes new methods for the different development phases, which include requirements derivation, validation, design, implementation and verification. A flexible framework for efficient evaluation of specified and developed functions during the individual phases is established, to ensure easy utilization of the introduced methods and procedures for many different applications. This smoothes the way towards a quick transition to real industry applications, which is highly intended to fill the upcoming specification gap. The implementation of the TCA is boosted by latest findings on efficient stochastic methods, which are required to evaluate very small failure probabilities usually inherent to the safety-critical functions in aviation. These methods are extensively used for the implementation of the TCA, which is why they are explained in detail in this thesis.
Beyond the (offline) development process, runtime verification is considered as additional powerful means to further exploit the potential of novel aircraft operations and support their certification. Therefore, this topic is thoroughly discussed and a novel online monitoring algorithm for dynamic systems with safety-critical functions is introduced and evaluated.
Eventually, all ideas, methods and developed algorithms described in this thesis are applied to real-life examples, to demonstrate proof of principle.

Zusammenfassung

Die Entwicklung und Zulassung sicherheitskritischer Funktionen für Flugzeuge mit neuartigen Topologien oder operationellen Konzepten stellen eine große Herausforderung dar, wenn es erforderlich ist, erfahrungsbasierten Zertifizierungsanforderungen zu folgen. Diese schreiben oftmals spezifische Entwurfslösungen vor um das erforderliche Maß an Sicherheit zu gewährleisten. Der Mangel an Erfahrungen mit solch neuartigen Anwendungen führt dazu, dass der konventionelle Ansatz nicht mehr praktikabel ist.

Die dieser Arbeit zugrundeliegende Forschung zielt darauf ab, die entstehende Spezifikationsdiskrepanz durch einen modellbasierten Ansatz aufzulösen, den sogenannten *Total Capability Approach* (TCA) ("*Gesamtleistungsansatz*"). Die Idee ist, während des gesamten Entwicklungsprozesses die Gesamtleistung eines Systems und seiner Komponenten mit all ihren Unsicherheiten, Störungen und möglichen Fehlern in einer ganzheitlichen Weise zu betrachten. Ein physikalisch getriebener Ableitungsprozess für Sicherheitsanforderungen von höchster Ebene führt zu dedizierten Entwurfsanforderungen, die speziell auf ein Fluggerät und seinen Betrieb zugeschnitten sind. Daher sind diese Anforderungen auch weniger konservativ im Gegensatz zu erfahrungsbasierten Anforderungen aus Spezifikationskatalogen, die üblicherweise ein breites Spektrum an möglichen Flugzeugkonfigurationen abdecken müssen.

Der Paradigmenwechsel hin zu einem Top-Down Ableitungsprozess für sicherheitsgetriebene Anforderungen erfordert komplexere Verifikationsmethoden, eröffnet jedoch auch den Weg für zusätzliche, zum Teil mächtige Werkzeuge für die Validierung, den Entwurf und die Implementierung von sicherheitskritischen Funktionen. Diese Dissertation stellt den vielversprechenden TCA vor und diskutiert die daraus resultierenden Herausforderungen und Ziele. Des Weiteren werden die erforderlichen neuen Methoden und Vorgehensweisen für die unterschiedlichen Entwicklungsschritte eingeführt, von Anforderungsableitung bis zur Verifikation. Es wird ein flexibler Rahmen für die effiziente Auswertung von spezifizierten, entwickelten und implementierten Funktionen begründet, der eine einfache Anwendung der vorgestellten Methoden auf eine Vielzahl von unterschiedlichen Entwicklungsaufgaben ermöglicht. Dies ebnet den Weg hin zur industriellen Anwendung, was höchst wünschenswert ist, um der aufkommenden Spezifikationsdiskrepanz frühzeitig zu begegnen. Die Anwendung des TCA wird auch ermöglichst durch neueste Erkenntnisse im Bereich effizienter stochastischer Methoden, welche für die Auswertung üblicherweise sehr kleiner zulässiger Fehlerwahrscheinlichkeiten von sicherheitskritischen Funktionen in der Luftfahrt erforderlich sind. Diese Methoden werden ausgiebig bei der Umsetzung des TCA eingesetzt, weshalb sie auch ausführlicher beschrieben werden.

Neben dem (Offline-)Entwicklungsprozess ist Verifikation während des Betriebs ("runtime verification") ein zusätzliches wirkungsvolles Werkzeug um das volle Potential neuartiger operationeller Konzepte auszunutzen und deren Zertifizierung zu unterstützen. Deshalb wird dieses Thema auch ausgiebig diskutiert und ein neuartiger Echtzeit-Überwachungsalgorithmus für dynamische Systeme mit sicherheitskritischen Funktionen vorgestellt.
Schlussendlich werden die in dieser Dissertation beschriebenen Ideen, Methoden und entwickelten Algorithmen auf beispielhafte, realistische Entwicklungsaufgaben angewendet, um das Prinzip des TCA und seine Anwendung zu demonstrieren.

Danksagung

Diese Arbeit entstand während meiner Zeit als wissenschaftlicher Mitarbeiter am Lehrstuhl für Flugsystemdynamik der TU München, der von Prof. *Florian Holzapfel* geleitet wird. Ihm gilt zu allererst mein Dank für die Ermöglichung dieser Arbeit, seine weise Voraussicht, die er bei der Themenfindung hatte und vor allem auch für das in mich gesetzte Vertrauen.

Weiterer Dank, insbesondere im Rahmen des Promotionsverfahrens, gilt Prof. *Manfred Hajek* für seinen Einsatz als Zweitgutachter, was ich vor allem ob seines fachlichen Hintergrundes und Wissens in für diese Arbeit relevanten Themenbereichen sehr schätze. Auch Prof. *Markus Zimmermann* sei Dank, dass er den Vorsitz der Prüfungskommission übernommen hat, darüber hinaus aber auch starkes Interesse an dem Thema dieser Arbeit gezeigt hat.

Des Weiteren möchte ich mich bei den vielen Kollegen am Lehrstuhl für Flugsystemdynamik bedanken. Hier gilt mein erster Dank *Nils Mumm* für die vielen fachlichen Diskussionen und dass er den vorgestellten Ansatz mit seiner Forschung noch näher an die Realität bringt. *Matthias Bittner* und *Chong Wang* gilt mein Dank für die vielen kritischen Diskussionen, die überaus konstruktive Zusammenarbeit und die wertvollen Hinweise aus anderen Perspektiven. Auch gilt mein Dank all jenen Kollegen, die ich bei der Anwendung neuer stochastischer Methoden unterstützen und damit wertvolle Erfahrung in diesem Bereich sammeln durfte.

Bei all den Projektpartnern, mit denen ich über die letzten Jahre zusammenarbeiten durfte, möchte ich mich für die vielen fruchtbaren Diskussionen bedanken.

Mein größter Dank gilt meiner Mutter *Gabriele*, die mich auf dem weiten Weg bis zur Promotion stets bestärkt und nach besten Möglichkeiten unterstützt hat.

Garching, im Januar 2018 David Löbl

Table of Contents

List of Figures

List of Tables

Acronyms

AFCS	Automatic Flight Control System
AMC	Acceptable Means of Compliance
ARP	Aerospace Recommended Practice
CCDF	Complementary Cumulative Distribution Function
CDF	Cumulative Distribution Function
DX	Diagnosis
EASA	European Aviation Safety Agency
FAA	Federal Aviation Administration
FCS	Flight Control System
FDI	Fault Detection and Isolation
FEM	Finite Element Method
FOQA	Flight Operational Quality Assurance
FSD	Institute of Flight System Dynamics
FTA	Fault Tree Analysis
FTE	Flight Technical Error
GPS	Global Positioning System
HIL	Hardware-in-the-Loop
HUMS	Health and Usage Monitoring System
i.i.d.	Independent and identically distributed
ICAO	International Civil Aviation Organization
IS	Importance Sampling
MC	Monte Carlo
MCMC	Markov Chain Monte Carlo
NSE	Navigation System Error
PBN	Performance Based Navigation
PDE	Path Definition Error
PDF	Probability Density Function
QAR	Quick Access Recorder
RV	Response variable
SESAR	Single European Sky ATM Research Joint Undertaking
SIL	Software-in-the-Loop
TCA	Total Capability Approach

TSE	Total System Error
TUM	Technical University of Munich
UAS	Unmanned Aerial System
UAV	Unmanned Aerial Vehicle
UP	Uncertain Parameter

Symbols and Indices

Symbols

CoV	Coefficient of variation
D	Desired probability distribution
E	Expectation
F	Failure event
I	General indicator function for set given in subscript
I_F	Indicator function for a failure
L_w	Wind turbulence scale length
$\mathcal{N}$	Gaussian distribution
N	Sample size
N_C	Number of samples per chain
N_S	Number of samples per subset
P	Probability
$\mathbf{P}$	Covariance matrix
$\hat{P}$	Estimated probability
P^*	Candidate vector of uncertain parameters
P_e	Equilibrium distribution of Markov chain
P_F	Failure probability
P_0	Desired conditional failure probability
Q	Process noise
Q	Joint probability density function
R	Autocorrelation
S	Importance sampling density
$\mathcal{U}$	Uniform distribution
$\mathbf{V}$	Gradient matrix of system derivatives with respect to design parameters
$\mathbf{W}$	Gradient matrix of system states with respect to design parameters
cov	Covariance
g	Gradient
h	Response function
h	Altitude
p	Design parameter for monitoring

q	Pitch rate
r	Response variable
r	Acceptance ratio
$\mathbf{u}$	Control vector
$\mathbf{x}$	State vector
y	System output
y^*	Extrapolated system output
$z_{1-\alpha}$	$1-\alpha$ quantile
$\mathbf{\Gamma}$	Discrete input transition matrix
$\mathbf{\Phi}$	Discrete state transition matrix
Σ	Covariance matrix
Θ	Pitch angle
$\mathbf{\Upsilon}$	Discrete disturbance transition matrix
α	Probability of exceedance of $z_{1-\alpha}$
α	Angle of attack
γ_k	Autocovariance with lag k
λ	Design parameter
ω	Natural frequency
σ	Standard deviation
σ_w	Wind turbulence intensity
$\boldsymbol{\theta}$	Vector of uncertain parameters
$\boldsymbol{\theta}^*$	Candidate vector of uncertain parameters
ζ	Damping

Indices

K	Kinematic frame
O	Earth-fixed frame describing local tangent plain of geoid
R	Receiver
T	Tanker
des	Desired quantity
rot	Rotational motion
$trans$	Translational motion
$wake$	Wake interaction

1

Introduction

1.1 Background

Beginning with the early days of aviation, improving and maintaining safety standards is of outstanding importance for aviation industry [Fed08, chapt. 1]. In the late 1940s, the *International Civil Aviation Organization* (ICAO) first published international standards on airworthiness of aircraft, known as Annex 8 to the Convention on International Civil Aviation [Int10]. This document gives broad standards which define the minimum basis for the recognition of certificates of airworthiness by other states. More detailed design standards are specified by individual states and supranational institutions, e.g. by the *Federal Aviation Administration* (FAA) in the USA and the *European Aviation Safety Agency* (EASA) in the EU. Although the so called design code has been continuously extended since their original release in the 1960s (e.g. airworthiness standards for transport category airplanes 14 CFR Part 25 [Fed64]), the given certification specifications mainly focus on conventional aircraft topologies and operations, which have not changed much during the past 50 years. However, in recent years aircraft design and operation undergo a radical change. While this has not been an issue at initiation of this research, clear changes can be identified today on various fronts, especially driven by technological advances in the field of electric propulsion. With increasing complexity of aircraft, futuristic aircraft concepts and especially also fast growing markets of unmanned and autonomous aircraft [SES16], new areas of conflict in the field of aircraft design and certification emerged.

First, aircraft functions become increasingly complex, often requiring a high number of components. However, especially for the market of small to mid-size *Unmanned Aerial Vehicle* (UAV), weight, size, cost and energy consumption limitations are in conflict with conventional design standards requiring components with high quality and quantity to obtain a desired level of safety and survivability. Furthermore, development teams for novel aerospace applications are often small and fast development and certification is aspired, which is in contrast to lots of artifacts, proofs, and documents which must be generated to demonstrate safety and reliability. The result is usually a long and expensive certification process, to prove safety. This is mainly, because "*the process of certifying a civil aircraft, whilst technically about proving that the air-*

craft is safe, in practice is about proving that the aircraft/ engine/ system meets the relevant certification standard" [Gra15, p. 4]. Additional conflicts emerge from the non-existence of specific design and certification standards for novel aircraft concepts and operations.
This work aims at resolving the challenges of developing and certifying novel aircraft applications by introducing a model based development approach. It takes the total capabilities of developed functions and resulting systems into account throughout the whole development process and operation to achieve a certain desired level of safety. In the context of aircraft and its systems, development means to "*[...] establish the process and methods to be used to provide the framework for the aircraft/system architecture development, integration and implementation*" [SAE10, p. 19]. This includes requirements management, validation, implementation, verification and certification.
Non-compliance of safety-critical functions with today's design standards is interpreted as non-admissible safety risk. This introduction shall highlight the drawbacks of today's approaches for development and certification of safety-critical aircraft functions and outline how this research contributes to tackle future challenges of implementing novel, pioneering aircraft functions and systems.

1.2 State of the Art and Motivation

Safety is of utmost importance for aviation. A reasonable measure of safety is risk, which is the probability that an event occurs that results in an unsafe condition. Unfortunately, today's safety measures for development of safety-critical functions mainly rely on past experience. The underlying assumption is that if certain specifications resulted in safe systems for past years and decades, these specifications will also lead to safe implementations for new similar designs. Examples for such historically grown specifications can be found in airworthiness standards for aircraft, which are continuously updated and extended since their initial release in the 1960s [Fed64, Fed67], mainly when incidents or accidents revealed flaws in the specifications. Such detailed airworthiness standards give requirements on how specific functions must look like. If safety-critical functions are designed, implemented and verified according to these specifications, they are considered to be safe since they fulfill the underlying airworthiness standards. The main disadvantages of this experience-based specification and certification approach are:

- It is non-probabilistic: Although a certain level of safety is achieved, no statement on the actual risk is given, i.e. the resulting overall failure probability and hence the level of safety is unknown. While for one system, today's prescriptive design requirements could lead to a very safe implementation, for others safety could be lower than expected.
- It is conservative: Usually, the prescriptive design requirements are conservative to allow their application for a certain range of aircraft designs, which however results in overly safe systems. To comply with these requirements, functions often need to be complex,

requiring components with high quality and quantity. By that, developments become complex and expensive. Using today's approach, the actual level of safety cannot be quantified, preventing alternative, simpler implementations to be used, which would still result in a desired level of safety although violating today's conservative prescriptive design requirements.

- It is only applicable to specific topologies: Aircraft topologies – i.e. the layouts of aircraft – have not changed much during past decades. With only a few exceptions, aircraft designs were very similar. Hence, airworthiness standards are mainly formulated for these common designs, also because the prescriptive design requirements are based on experience. This is a show-stopper for novel concepts, which do not follow conventional aircraft categories. Furthermore, this also slows down advance through technology, since novel ideas without any historic example lack in specific design requirements and so called "*special conditions*" must be negotiated with certification authorities, which can take a long time [Eur13, p. 11]. Special conditions are determined by a panel of experts [Eur07, Section 2, Article 3] and are likely to follow a conservative line to qualitatively ensure safety.

Despite the drawbacks, the currently established procedures and development approaches are applied due to the lack of alternatives. Only during the recent months, a shift in paradigm started to take place towards a more probabilistic development approach, caused by aircraft highly deviating from conventional designs for which a safe design according to conventional standards is no longer possible. The latest revision of the *European Certification Specifications for Normal-Category Aeroplanes CS-23* [Eur17d] released in March 2017 is completely reorganized compared to previous releases and now follows a more proportionate and risk-based approach to aeroplane standards by "*[...] removing the arbitrary weight-driven separation of technical standards that are no longer appropriate for modern technology*" [Eur17a, p. 2]. Similar actions are carried out in the USA [Dep16, p. 1]. The objective of this new rulemaking is "*[...] to provide clear safety objectives without prescribing design solutions*" [Dep16, p. 326]. This results in replacement of many requirements on how specific functions must look like by considerably less requirements on how safe those functions must be, where safety is now defined in a quantitative probabilistic and hence risk-based manner.
Similar certification and operation standards are envisaged for unmanned systems. Current draft documents for a regulatory framework for the operation of drones mainly require that "*UAS of all class shall be designed and manufactured to fly safely*" [Eur17c, p. 49]. Again, this is a risk-based approach, where a specific safety goal must be met for systems and operation. Although a risk-based approach is independent from technological changes and theoretically eliminates the disadvantages of today's deterministic experience-based approach, the unavailability of clear design requirements now leads to a huge challenge for the overall development process: How can the risk of a specific function be quantified and hence compliance with novel probabilistic certification standards be shown?

The shift towards a risk-based certification approach must not be confused with one probabilistic aspect of the aircraft and systems development process that is already established today: There is one paragraph in current certification specifications that deals with safety impacts of installed equipment (e.g. CS23.1309 and CS25.1309 for small and large aircraft respectively [Eur17d, Eur17b]). It specifies that equipment, systems and installations must not cause critical events with a probability higher than specified by admissible failure probabilities. However, this paragraph only results in probabilistic requirements for availability and integrity of the implementation of safety-critical functions, where the function itself is still defined in a conservative and prescriptive manner as described above [SAE10].

1.3 Objectives

Several years ago, the limitations of conventional airworthiness standards for development and certification of upcoming novel aircraft concepts and operations were recognized, also at the *Institute of Flight System Dynamics* (FSD) of the *Technical University of Munich* (TUM). The idea of a probabilistic development approach was born, where the major design objective to obtain a safe system is achieved by a top-down approach, which allows a quantitative evaluation of risk instead of using the prescriptive design specifications from airworthiness standards. As discussed in the previous section, this change in paradigm also started to become of interest for certification authorities.
To close the gap between risk-based top level requirements and the lack of specific design requirements, the so called *Total Capability Approach* (TCA) for development of safety-critical functions originated, where the idea is to use a physically driven system development process that takes all components and uncertainties contributing to the overall system performance into account during requirements derivation, design and verification. This approach promises to enable quantification of the actual risk emerging from a developed function and hence its quantitative contribution to safety of the overall system.
The TCA affects the overall development process. Very briefly, this especially includes the following steps:

1. **Requirements formalization:** In contrast to the conventional bottom-up process for specification of safety-critical functions, the TCA is a top-down process, where probabilistic top-level requirements are broken down to a level that allows design and implementation of the intended functionality.

2. **Design and verification:** Taking the whole knowledge about uncertainties, disturbances and failures into account during system design, allows design solutions tailored to the actual application and mission. However, this comes with additional efforts required for verification to proof compliance with probabilistic requirements.

3. **Runtime verification:** During development following the TCA, models and uncertainties are not exactly known and hence assumptions must be made. This necessitates

runtime verification to ensure that assumptions are met in reality and safe operation results.

The research presented in this PhD thesis condenses the first efforts on establishing the Total Capability Approach. Further research conducted by researches at the Institute of Flight System Dynamics already focuses on specific aspects of such a probabilistic development approach as well as a proof in practice.

1.4 Contributions

This thesis aims at the introduction of the Total Capability Approach, working out the influences of this novel development approach on the individual steps of the development process, highlighting challenges related to it and introducing possible solutions for some of the challenges. The following points condense the contributions of this thesis beyond the state of the art, which correlate well with the main steps of the TCA presented above:

- **Establishing the Total Capability Approach:** The central contribution is the introduction of the overall concept of the Total Capability Approach. This also includes in-depth discussions of the implications for the individual steps of today's well established aircraft and system development process. For that, profound evaluation of today's standards is conducted and the arising challenges are elaborated. To the best knowledge of the author, no comparable top-down, model-based approach for development and certification of safety-critical functions has been introduced before. To highlight the advantages of the novel concept, the individual steps are demonstrated using selected examples from the development of an autopilot for close formation flight of large transport aircraft.

- **Model-based derivation of requirements:** One major challenge of the TCA is the breakdown of probabilistic top-level requirements to specific lower-level requirements that can be used for system design. Besides the introduction of the overall concept, special attention is paid to the fact that the admissible failure probabilities are usually very small and hence conventional stochastic methods cannot be applied for risk assessment. Main contribution for this development task is the specification of the simulation environment needed for model-based requirements breakdown. Based thereon, a novel method for quantification of lower-level requirements is developed, with the focus on break-down of top-level safety requirements with very low admissible violation probabilities. Furthermore, the model-based derivation process supports requirements validation during early development phases – this is also introduced in this thesis.

- **Stochastic verification:** Verification in the context of the TCA must be conducted to ensure that the designed and implemented function meets the probabilistic top-level requirements. Again, the usually very small admissible failure probabilities pose additional

challenges, even more since the models, which are suitable for model-based verification, are usually very sophisticated and hence require a lot of computational resources. The major contribution here is the definition of a framework for model-based verification of probabilistic requirements, with special attention to efficiency and ease of operation. For that, state of the art stochastic algorithms are applied to the underlying engineering problem.

- **Runtime verification**: A major tool for runtime verification is online monitoring. To facilitate this, an online monitoring algorithm for one common, important type of requirements is developed that ensures that the assumptions made during development are also valid in reality and hence the offline verification results are true. Furthermore, the developed algorithm can be used to monitor the current safety level and hence operation-dependent safety goals can be implemented. This can highly increase availability of novel aviation applications. Note that only a certain aspect of runtime verification is considered, while there is currently a lof of research conducted in the field of online monitoring [Rus08, Leu08, Gro+17].

- **Enhanced stochastic analysis** is applied for different steps of the TCA. Therefore, a detailed evaluation of applicable stochastic algorithms is provided in chapter 2. Much emphasis is put on descriptive explanations and derivations supporting the understanding of the methods. The goal is clearly not to deliver another stochastic textbook including well-known proofs, but to provide the necessary relations between theory and application for risk assessment. Despite no novel stochastic methods are developed, this chapter contributes a very condensed guide to enhanced stochastic evaluation for risk assessment.

A detailed literature review, concerning the contributions and particular aspects of the development process investigated in this work, can be found in the introductions of the topics in the respective chapters.

1.5 Outline

This thesis is structured as follows. In chapter 2 detailed descriptions of enhanced stochastic methods applied throughout the overall development process are given. To ensure a common understanding, first types of uncertainties and principles of failure probability estimation are discussed in sections 2.1 and 2.2. Afterwards, stochastic algorithms for risk assessment are discussed, starting with conventional Monte Carlo simulation in section 2.3, followed by rare event sampling methods in the subsequent sections. The explanations do not only include the algorithms themselves, but also the underlying mechanisms as well as a detailed evaluation of confidence intervals, which are essential for the interpretation of the results of risk analysis.
In chapter 3, the concept of the Total Capability Approach is presented. After a general introduction, the influence of this novel development approach on today's well established

aircraft and system development processes are discussed and arising challenges elaborated. Furthermore, the relations between the different steps of the TCA are discussed, while detailed considerations of the individual steps follow in subsequent chapters. Requirements derivation in the scope of the TCA is discussed in chapter 4. Initially, the underlying idea of model-based derivation of requirements is given in section 4.1. Afterwards, a new method for quantification of lower-level requirements based on probabilistic top-level requirements is described in 4.2 and 4.3. In section 4.4 it is shown how this model-based approach can support requirements validation. Open challenges in this field are discussed in 4.5.
The first section in chapter 5 is dedicated to the design process of functions in a probabilistic framework. It is discussed how design methods for controlled systems can be utilized to optimize the compliance with probabilistic requirements. However, no dedicated research has been conducted in this field, since it is currently subject of independent research by associates at the Institute of Flight System Dynamics. Universal verification methods are described in section 5.2 that can be used to check compliance of implemented functions with probabilistic requirements. Here, especially the application of enhanced stochastic methods for the verification process is discussed. Finally, chapter 6 describes the necessity for runtime verification, the state-of-the-art of online monitoring algorithms is evaluated and a novel method for monitoring of compliance of an implemented function with a certain type of probabilistic requirements is presented in section 6.4.
In chapter 7, the different steps of the TCA are demonstrated using selected examples from development of an autopilot for close formation flight of two civil transport aircraft. An introduction of the formation flight concept, problem formulation and specification of top-level requirements are given in section 7.1. Models for flight dynamics and uncertainties used in this example are given in 7.2. In section 7.3, it is shown how the methods developed in this research support requirements derivation, verification and online monitoring. The examples given in chapter 7 need to be seen as illustrative examples for the future potential of the TCA. Finally, chapter 8 concludes this thesis. There, special attention is given to the outlook, highlighting the way towards a proof in practice.

2

Fundamentals of Stochastic Analysis

The Total Capability Approach relies on stochastic analysis throughout the whole development process. The fields of application for the analysis of effects of uncertainties in this context are:

- **Risk analysis**: Risk analysis is the quantification of the probability of occurrence of critical events. Particularly here, risk analysis is about the evaluation of usually very unlikely events, quantified by the tails of probability distributions of relevant system responses.
- **Failure analysis**: Failure analysis deals with the identification of most probable scenarios that lead to failure events. While risk analysis is only interested in the pure knowledge about criticality, failure analysis enables in-depth analysis of critical events and their causes.

Risk and failure analysis usually go hand in hand, where in the majority of cases, analysis is only possible using numerical methods due to the complexity of models usually required for a quantitative risk analysis. In the context of developing safety-critical functions, the analysis process must have a high degree of automation to facilitate ease of operation and usability. In the following, methods for risk and failure analysis are presented that are of importance for this work. Beforehand, uncertainties and their modeling are discussed.

2.1 Uncertainties

Models are used for risk and failure analysis. Usually models cannot exactly reproduce the behavior of a real system due to uncertainties. This section does not discuss possible sources of uncertainties – for that, see section 4.1.4. Instead, it is assumed that a model is available that includes effects of relevant uncertainties.

2.1.1 Types of Uncertainties

On an abstract level, uncertainties can be described by *Uncertain Parameters* (UPs) and their known probabilistic properties, e.g. their *Probability Density Function* (PDF). The UPs can

be distinguished into two groups:

- **Influential uncertain parameters**: Change of the value of a single UP can have a high influence on the system response. For an aircraft dynamics model, this is for example the case for location of center of gravity, mass and moment of inertia.
- **Non-influential uncertain parameters**: Change of the value of a single UP has only infinitesimal influence on system response, i.e. the response is insensitive to a single parameter. Instead, the combination of uncertain parameters is crucial, which can be interpreted as an "integral" effect of these parameters. This is the case for stochastic uncertainties, which are discretized for simulation and are described by single uncertain parameters per time step. While the change of the value of an uncertain parameter for a single time step does only infinitesimally change the system response, it is the combination of the values at many subsequent time steps that leads to a noticeable system response. The number of time steps is proportional to simulation time and sample frequency, hence the total number of non-influential UPs is usually much higher than the number of influential UPs.

2.1.2 Modeling and Realization of Uncertain Parameters

Uncertain parameters are described by their joint PDF. In most cases, the probability distribution of individual parameters is known or at least possible ranges of an uncertain parameter limited by known bounds are available. In the latter case, there is no knowledge about the actual parameter distribution. Often uniform distributions are used to model such uncertain parameters. This assumption is legitimate if the objective is to identify the whole solution space, i.e. to evaluate all possible effects of bounded uncertainties. However, this could lead to biased failure probability estimations in the context of risk analysis since a more critical parameter value could occur with a higher or lower probability than modeled by the uniform distribution. Figure 2.1 shows an example where the uncertain parameter θ is limited by $\theta_{low} < \theta < \theta_{up}$. For the exemplary case that $\theta = a$, the effect of this specific realization of the uncertain parameter is overrated if a uniform distribution is assumed, while for $\theta = b$ the effect is underrated. Note that this does not allow inference on the criticality of a certain parameter realization – underestimation for the case $\theta = b$ means that if this uncertain parameter value would lead to a critical event, the probability for this event would be estimated too low since the unknown actual probability of occurrence of this parameter is higher. To be on the safe side, the most critical parameter value, i.e. the parameter value that leads to the most critical system response, must be assumed to occur with a probability of one. This assumption gives an valid, but conservative upper bound for the estimated failure probability in case that no probability distribution for some bounded influential uncertain parameters is available. Theoretically, this procedure would also be applicable for non-influential uncertain parameters, which could be for example an arbitrary but bounded input signal. However, the practical implementation is virtually impossible since the individual effect of each of many non-

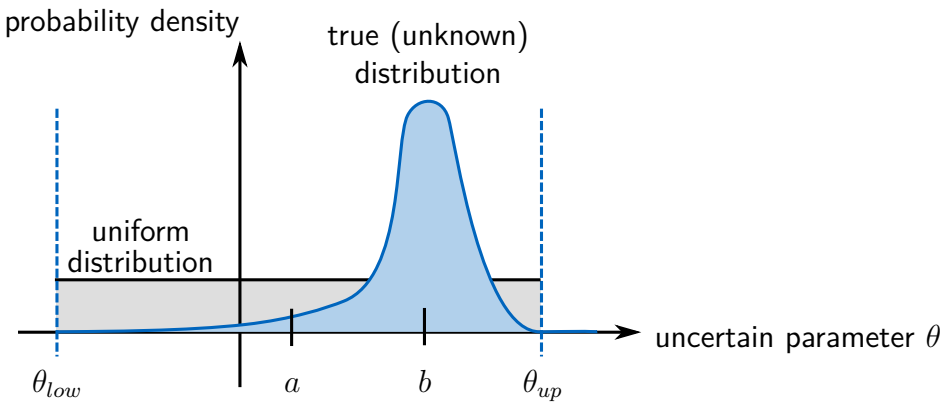


Figure 2.1: *Modeling of uncertain parameters by intervals – effect of assuming a uniform distribution*

influential parameters on the system response is usually infinitesimally small. In combination with the arbitrariness of the values, this theoretically leads to infinite most critical cases.
For simulation, samples of the uncertain parameters according to their probability density function must be generated. For each parameter, a high number of *Independent and identically distributed* (i.i.d.) random values (i.e. realizations for each uncertain parameter with the according PDF) must be generated. For simulation of random numbers from a uniform distribution, quasirandom number generators are used. Efficient methods are available for generation of samples from common distributions. For arbitrary distributions, methods exist that transform uniformly or Gaussian distributed samples to any desired distribution. Well known representatives are Inverse CDF Method, Acceptance/Rejection Methods and use of Markov chains. Generation of random numbers is a well researched and documented process with verified and efficient tools available for that task. Hence, this aspect is not discussed in this thesis. More details on generation of random numbers can be found for example in [Dev86, BDF98, Gen03].

2.2 Risk Analysis

2.2.1 Definition of Failure Events

Estimation of the failure probability is similar to the question of reliability of a system. Therefore it is often also referred to as reliability analysis. The objective of reliability analysis is the estimation of the probability of occurrence of predefined events, taking all uncertainties, represented by uncertain parameters, into account. The failure probability is defined as

$$P(F) = P_F = P(r < r_{limit}) \tag{2.1}$$

where F denotes the failure event and r the scalar positive *Response variable* (RV), defined as continuous response metric

$$r = h\left(\mathbf{x}\left(\boldsymbol{\theta}\right), \boldsymbol{\theta}\right), \quad h : \mathbb{R}^n \mapsto \mathbb{R} \tag{2.2}$$

with $\mathbf{x} \in \mathbb{R}^n$ being the state vector of the system and $\boldsymbol{\theta} \in \mathbb{R}^k$ a vector containing the k uncertain parameters. The value of the response variable is calculated by the function h usually based on the results of a simulation run that takes the uncertain parameter vector $\boldsymbol{\theta}$ as input. In most cases, the critical system response can be adequately represented by a scalar response variable. A simultaneous consideration of multiple response variables is usually not possible since in this case the solution would lie on a Pareto frontier on which the probability for one elementary event depends on the probability of another event. For failure probability analysis, two elementary cases can be identified: The individual failure events are either linked by AND- or OR conjunctions. In the first case, failure only occurs if all response variables exceed their limits (logical conjunction – intersection of failure events). In the second case, the failure occurs when at least one out of many response variables exceeds its limit (logical disjunction – union of failure events). For these cases a scalar response variable can be defined by:

$$r = \begin{cases} \min\limits_{i=1\ldots k} r_i / r_{i,limit} & \text{for} \quad r_1 \cap r_2 \cap \cdots \cap r_k \text{ (conjunction)} \\ \max\limits_{i=1\ldots k} r_i / r_{i,limit} & \text{for} \quad r_1 \cup r_2 \cup \cdots \cup r_k \text{ (disjunction)} \end{cases} \tag{2.3}$$

where the limit value of the joint response variable r is $r_{limit} = 1$ due to normalization of the individual r_i. For the definition given in equation (2.3), the individual limit values $r_{i,limit}$ must not be zero, which however would be a meaningless selection since r is defined as positive measure. This is no limitation since any arbitrary response can be transformed to a positive measure with a non-zero limit value $r_{i,limit}$ by shifting and scaling. The resulting joint response variable for both cases is shown in figures 2.2 and 2.3 exemplarily for the case of two independent response variables. Combinations of logical con- and disjunctions can be formulated using $\min$ and $\max$ appearing in the same order. For example, when a failure occurs if $(r_1 \cap r_2) \cup (r_3 \cap r_4)$, the resulting response variable is calculated by

$$r = \max\left(\min\left(\frac{r_1}{r_{1,limit}}, \frac{r_2}{r_{2,limit}}\right), \min\left(\frac{r_3}{r_{3,limit}}, \frac{r_4}{r_{4,limit}}\right)\right) \tag{2.4}$$

2.2.2 Estimation of the Failure Probability

The failure probability can be calculated by solving the following integral over the uncertain parameter space:

$$P_F = \int_F Q\left(\boldsymbol{\theta}\right) \mathrm{d}\boldsymbol{\theta} = \int Q\left(\boldsymbol{\theta}\right) I_F\left(\mathbf{x}\left(\boldsymbol{\theta}\right), \boldsymbol{\theta}\right) \mathrm{d}\boldsymbol{\theta} \tag{2.5}$$

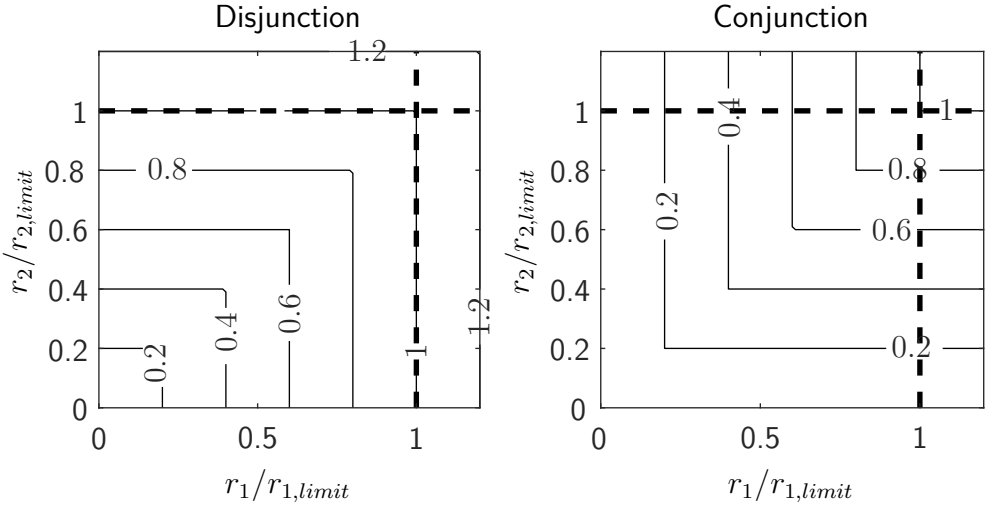


Figure 2.2: *Contour map for the response variable r of disjunct and conjunct failure events.*

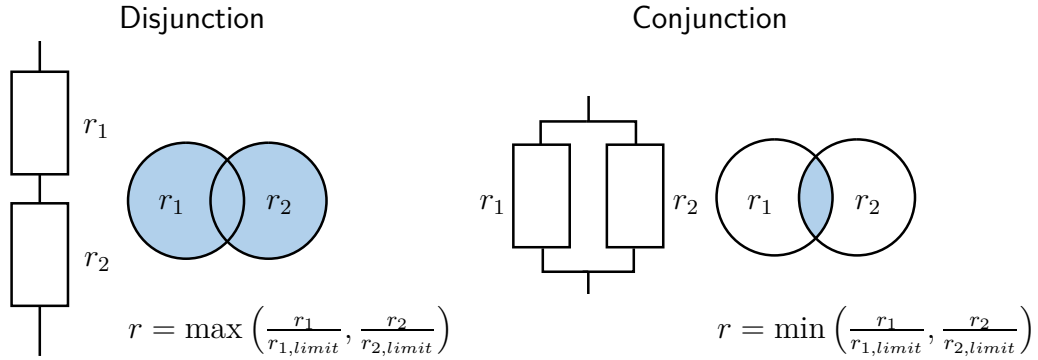


Figure 2.3: *Common relations of elementary failure events.*

where $Q(\boldsymbol{\theta})$ denotes the k-dimensional probability density function of $\boldsymbol{\theta}$. Since the failure region F is usually not a-priori known, the integral is solved over the whole uncertain parameter space and at each point the indicator function I_F is evaluated, which is one if a sample lies in the failure region and zero otherwise:

$$I_F(\mathbf{x}(\boldsymbol{\theta}), \boldsymbol{\theta}) = \begin{cases} 1 & if \quad r = h(\mathbf{x}(\boldsymbol{\theta}), \boldsymbol{\theta}) > r_{limit} \\ 0 & else \end{cases} \tag{2.6}$$

For the sake of readability, subsequently the indirect dependency of the indicator function on the uncertainties via the state vector $\mathbf{x}$ is written as a general function of $\boldsymbol{\theta}$, i.e. $I_F(\boldsymbol{\theta}) = I_F(\mathbf{x}(\boldsymbol{\theta}), \boldsymbol{\theta})$.

Mostly, the integral (2.5) cannot be solved analytically. A numeric solution of the integral by discretization of $\boldsymbol{\theta}$ is not possible due to the exponential growth of required evaluations of the indicator function I_F with increasing dimension of the uncertain parameter space k. The evaluation of I_F is usually the most computationally expensive and hence limiting task, since it is often linked to the evaluation of exhaustive simulation models. Therefore the number of evaluations of I_F should be kept as small as possible.

2.3 Monte Carlo Simulation

2.3.1 Algorithm

The idea of *Monte Carlo* (MC) simulation is to solve the integral in equation (2.5) by treating it as estimation, i.e.

$$P_F = \int Q(\boldsymbol{\theta})\, I_F(\boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta} = E_q\left[I_F(\boldsymbol{\theta})\right] \tag{2.7}$$

where the samples used to obtain the expectation E_q are distributed according to $Q(\boldsymbol{\theta})$. The estimated failure probability $\hat{P}_F$ for a sample size N is calculated by

$$\hat{P}_F = \frac{1}{N} \cdot \sum_{i=1}^{N} I_F(\boldsymbol{\theta}_i) = \frac{\text{number of failure samples}}{\text{total number of samples}} \tag{2.8}$$

with independent realizations of the uncertain parameter vector $\boldsymbol{\theta}_i$. In general, a hat above a symbol denotes an estimated quantity. The Monte Carlo algorithm for failure probability estimation can be described as follows:

Conventional Monte Carlo algorithm

1. Generate N independent uncertain parameter vectors $\boldsymbol{\theta}_i, i = 1 \ldots N$ distributed according to $Q(\boldsymbol{\theta})$.
2. For each sample, run a simulation to obtain the system response $\mathbf{x}(\boldsymbol{\theta}_i)$ and evaluate the indicator function $I_F(\boldsymbol{\theta}_i)$.
3. Calculate the estimated probability by averaging the indicator function according to equation (2.8).

With increasing number of samples N, the estimated failure probability $\hat{P}_F$ lies closer to P_F and converges almost surely to P_F for $N \to \infty$. This is known as strong law of large numbers [GT13, p. 13ff.]. The unbiasedness of the estimator can be proven by taking the expectation of equation (2.8):

$$\begin{aligned} E\left[\hat{P}_F\right] = E\left[\frac{1}{N} \cdot \sum_{i=1}^{N} I_F(\boldsymbol{\theta}_i)\right] &= \frac{1}{N} \cdot \sum_{i=1}^{N} E\left[I_F(\boldsymbol{\theta}_i)\right] \\ &= 1/N \cdot N P_F = P_F \end{aligned} \tag{2.9}$$

where the interchange of expectation and sum is possible because of the independence of the individual samples $\boldsymbol{\theta}_i$.

2.3.2 Variance Estimation

Equation (2.8) provides a good estimation of the failure probability if the sample size is large enough. In the following, an expression for reasonable sample sizes is derived. The variance of a single sample $\mathrm{var}\left[I_F\left(\boldsymbol{\theta}_i\right)\right]$ is the same for every sample since the uncertain parameter vectors $\boldsymbol{\theta}_i$ are independent and identically distributed. $I_F\left(\boldsymbol{\theta}_i\right)$ follows a Bernoulli distribution, i.e. it takes the value 1 with failure probability P_F and 0 with probability $(1 - P_F)$. The variance of a Bernoulli-distributed random variable is

$$\begin{aligned}\mathrm{var}\left[I_F\left(\boldsymbol{\theta}_1\right)\right] = E\left[I_F\left(\boldsymbol{\theta}_1\right)^2\right] - E\left[I_F\left(\boldsymbol{\theta}_1\right)\right]^2 &= P_F - P_F^2 \\ &= P_F\left(1 - P_F\right)\end{aligned} \tag{2.10}$$

where the algebraic formula for the variance is used [CH13b, p. 50]:

$$\mathrm{var}\left[X\right] = E\left[\left(X - E(X)\right)^2\right] = E\left[X^2\right] - E\left[X\right]^2 \tag{2.11}$$

Using the variance of a single sample, the variance of the estimated failure probability can be derived. First note that

$$\begin{aligned}\mathrm{var}\left[\hat{P}_F\right] = E\left[\left(\hat{P}_F - P_F\right)^2\right] &= E\left[\left(\frac{1}{N}\sum_{i=1}^{N}\left(I_F\left(\boldsymbol{\theta}_i\right) - P_F\right)\right)^2\right] \\ &= \frac{1}{N^2}E\left[\left(\sum_{i=1}^{N}\left(I_F\left(\boldsymbol{\theta}_i\right) - P_F\right)\right)^2\right] \\ &= \frac{1}{N^2}\sum_{i=1}^{N}\mathrm{var}\left[I_F\left(\boldsymbol{\theta}\right)\right]\end{aligned} \tag{2.12}$$

Again, the interchange of sum and variance in the last step is possible since the individual samples are not correlated, i.e. the sum of the variance of independent random numbers is the variance of the sum of those numbers [DS12, p. 35]. Substituting equation (2.10) into (2.12) yields

$$\begin{aligned}\mathrm{var}\left[\hat{P}_F\right] = \frac{1}{N^2}\cdot\sum_{i=1}^{N}\mathrm{var}\left[I_F\left(\boldsymbol{\theta}_i\right)\right] &= 1/N^2 \cdot N\mathrm{var}\left[I_F\left(\boldsymbol{\theta}_1\right)\right] \\ &= \frac{\mathrm{var}\left[I_F\left(\boldsymbol{\theta}_1\right)\right]}{N} \\ &= \frac{P_F\left(1 - P_F\right)}{N}\end{aligned} \tag{2.13}$$

with $I_F\left(\boldsymbol{\theta}_1\right) = I_F\left(\boldsymbol{\theta}_2\right) = I_F\left(\boldsymbol{\theta}_i\right)$ since the realizations of $\boldsymbol{\theta}$ are independent and identically distributed. Although theoretically correct, this equation cannot be used for estimation of the variance in practice since the exact failure probability and hence the variance of a single sample is usually not known. Instead, the sample variance can be used, which is an unbiased

estimator of the sample distribution variance [Beh13, p. 292]:

$$\begin{aligned} \operatorname{var}\left[I_F\left(\boldsymbol{\theta}_1\right)\right] &\approx \frac{1}{N-1} \sum_{i=1}^{N} \left(I_F\left(\boldsymbol{\theta}_1\right) - \hat{P}_F\right)^2 \\ &= \frac{1}{N-1} \left(\sum_{i=1}^{N} I_F\left(\boldsymbol{\theta}_1\right)^2 - N \cdot \hat{P}_F^2\right) \end{aligned} \tag{2.14}$$

where the second line results from equation (2.11). Using $\sum_{i=1}^{N} I_F\left(\boldsymbol{\theta}_i\right)^2 = \sum_{i=1}^{N} I_F\left(\boldsymbol{\theta}_1\right) = N \cdot \hat{P}_F$, since $I_F\left(\boldsymbol{\theta}_i\right)$ is either 0 or 1, leads to

$$\operatorname{var}\left[I_F\left(\boldsymbol{\theta}_1\right)\right] \approx \frac{N}{N-1} \left(\hat{P}_F\left(1 - \hat{P}_F\right)\right) \tag{2.15}$$

and hence for the variance of the estimated failure probability

$$\operatorname{var}\left[\hat{P}_F\right] \approx \frac{\hat{P}_F\left(1 - \hat{P}_F\right)}{N-1} \tag{2.16}$$

The factor $N-1$ compared to N in equation (2.13) arises from the fact that the failure probability used in equation (2.14) is an estimated quantity instead of its expectation $E\left[\hat{P}_F\right] = P_F$, which would be the formal definition of the variance. This can be interpreted as loss of information: For only one sample, the estimated probability is similar to the one and only sample – i.e. 0 or 1 – and hence the first sample does not provide any information for the variance. The factor $N-1$ compensates this effect and leads to an unbiased estimator of the variance. The proof can be found in [Beh13, p. 293].

For risk analysis, the variance is not informative without relation to the estimated probability. Therefore, the coefficient of variation CoV is introduced, which is a standardized measure for the variability of the estimation. It relates the square root of the variance (known as standard deviation σ) to the mean:

$$CoV = \frac{\sqrt{\operatorname{var}\left[\hat{P}_F\right]}}{E\left[\hat{P}_F\right]} \tag{2.17}$$

The estimation error of the failure probability $\left(\hat{P}_F - P_F\right)$ usually follows a Gaussian distribution. Hence, the coefficient of variation gives the standard deviation of the relative estimation error $\left(\hat{P}_F - P_F\right)/P_F$. This means that the estimated probability $\hat{P}_F$ lies between $P_F\left(1 - CoV\right)$ and $P_F\left(1 + CoV\right)$ with a probability of approximately 68.3%, which is the probability that a value lies within $\pm 1\sigma$ for a standard Gaussian distribution. By inserting equations (2.9) and (2.10) into (2.17), the CoV for Monte Carlo simulation is obtained:

$$CoV = \sqrt{\frac{1 - P_F}{N P_F}} \approx \sqrt{\frac{1 - \hat{P}_F}{N_F}} \tag{2.18}$$

where N_F is the number of failure samples. According to a rule of thumb for Monte Carlo

simulation, the number of samples should be approximately ten times the inverse of the probability $N \approx 10/P_F$. In average, this results in ten failure samples. Inserting this into equation (2.18) leads to a $CoV \approx 0.3$ for small failure probabilities where $1 - \hat{P}_F \to 1$. Given a specific probability, (2.18) can be used to calculate the required number of samples to obtain a desired accuracy CoV_{des}:

$$N_{req} = \frac{1 - P_F}{P_F CoV_{des}^2} \tag{2.19}$$

For small probabilities, the required number of samples is inversely proportional to the probability, i.e. it increases hyperbolically with decreasing probability. Writing the probability in exponential form $P_F = m \cdot 10^{-n}$, the statement is equivalent to an exponential growth of the number of samples with increasing (negative) order of the probability n.

Note that according to (2.18), CoV can never be larger than one if there is at least one failure sample. However, the assumption that the estimation error follows a Gaussian distribution only holds if there are sufficient samples in the failure domain, which will be shown in the next section. That this assumption cannot be true for high CoV can be deduced from the following consideration: If $CoV = 1$, the lower single standard deviation bound would be $P_F(1 - CoV) = 0$. Since the estimated failure probability cannot be smaller than zero, it is obvious that the resulting estimation error cannot be Gaussian distributed, which would mean that with a probability of $(1 - 68.3\%)/2$ the estimated failure probability would be smaller than zero.

2.3.3 Confidence Intervals

In this section, the assumption that the estimation error is Gaussian distributed will be justified and an expression for the confidence interval of $\hat{P}_F$ will be derived. For that, an interval around the estimation $\hat{P}_F$ is looked for that overlaps with the actual value of P_F with a high probability $(1 - \alpha)$:

$$P\left(\hat{P}_{low} < P_F < \hat{P}_{up}\right) = 1 - \alpha \tag{2.20}$$

The interval $\left[\hat{P}_{low}, \hat{P}_{up}\right]$ is referred to as $(1 - \alpha)$ confidence interval. This means that $\hat{P}_{low} < P_F < \hat{P}_{up}$ with a probability of $(1 - \alpha)$ and only with a probability of α the failure probability P_F lies out of the interval, see figure 2.4. Exemplarily, $\alpha = 0.02$ refers to the $(1 - \alpha) = 98\%$ confidence interval, which means that the true failure probability P_F lies within the estimated confidence interval $\left[\hat{P}_{low}, \hat{P}_{up}\right]$ with a probability of 98% and only with a chance of 2% outside. The evaluation of a single sample can be compared to a *Bernoulli* trial with exactly two possible outcomes: "failure" or "no failure". The failure probability is the same every time the experiment is conducted. The result of multiple *Bernoulli* trials follows a binomial distribution. According to the *De Moivre–Laplace* theorem, a binomial distribution can be approximated by a Gaussian distribution if the number of samples is large enough. A criteria for validity of the *De Moivre–Laplace* theorem is that $NP_F(1 - P_F) > 9$ [CH13b, p. 95]. Figure 2.5 shows the

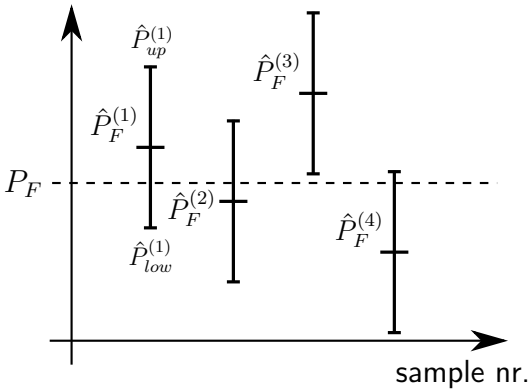


Figure 2.4: *Examples for double sided confidence intervals*

binomial distribution and its Gaussian approximation for a failure probability $P_F = 0.2$ and different number of samples N. Mean and standard deviation of the approximating Gaussian distribution are $\mu = NP_F$ and $\sigma = \sqrt{NP_F\left(1 - P_F\right)}$ respectively. To satisfy the validity criteria for $P_F = 0.2$, the number of samples must be higher than 56. For the displayed cases with more than 60 samples (lower two plots in figure 2.5), a very good approximation is recognizable.

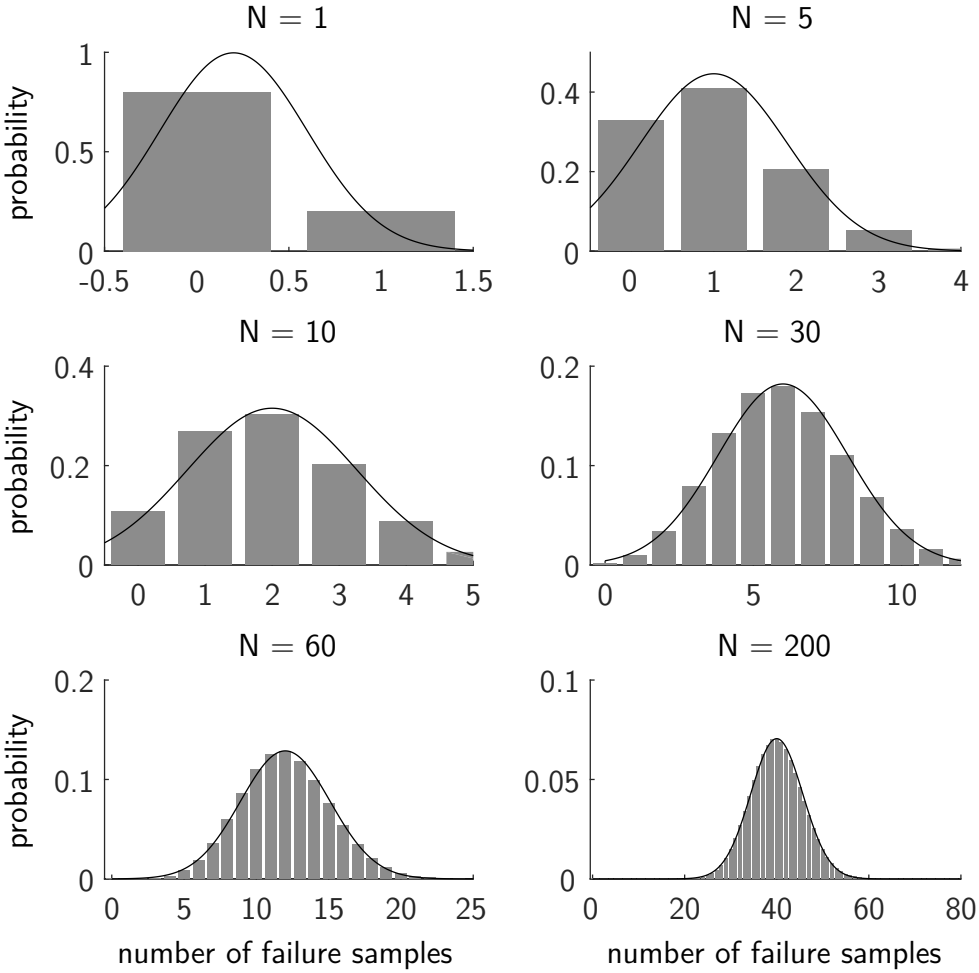


Figure 2.5: *Approximation of binomial distribution by normal distribution*

Given that the validity criteria is fulfilled, the standardized arithmetic mean of the random test

$$P_{F,s} = \frac{\hat{P}_F - P_F}{\sqrt{\mathrm{var}\left[\hat{P}_F\right]}} = \frac{\hat{P}_F - P_F}{\sqrt{\left(P_F\left(1 - P_F\right)\right)/N}} \tag{2.21}$$

is distributed according to a standard Gaussian distribution $\mathcal{N}\left(0;1\right)$. Symmetric bounds lead to the following definition of the confidence interval:

$$P\left(\left|\hat{P}_F - P_F\right| < c\right) = 1 - \alpha \tag{2.22}$$

where c denotes an (initially unknown) upper bound for the estimation error. $\left(\hat{P}_F - P_F\right)$ is Gaussian distributed, however it is not standard Gaussian distributed, which results in c being not only a function of α but also of the variance of $\left(\hat{P}_F - P_F\right)$. To cancel the dependency on the variance, equation (2.21) is substituted in (2.22), which results in a normalized description of the confidence interval:

$$\begin{aligned} P\left(\frac{\left|\hat{P}_F - P_F\right|}{\sqrt{\left(P_F\left(1 - P_F\right)\right)/N}} < \frac{c}{\sqrt{\left(P_F\left(1 - P_F\right)\right)/N}}\right) &= P\left(\left|P_{F,s}\right| < z_{1-\alpha/2}\right) \\ &= 1 - \alpha \end{aligned} \tag{2.23}$$

where $z_{1-\alpha/2}$ is the $(1-\alpha/2)$-quantile of the standard Gaussian distribution, which now only depends on α due to normalization by $\sqrt{\mathrm{var}\left[\hat{P}_F\right]} = \sqrt{\left(P_F\left(1 - P_F\right)\right)/N}$. The $(1-\alpha/2)$-quantile is the symmetric threshold $\pm z$ of the standardized Gaussian PDF $\phi\left(z\right)$ which is not exceeded with a probability higher than $\alpha/2$ at the lower and upper end of z, see figure 2.6. The quantile can be obtained using the inverse $\Phi^{-1}\left(z\right)$ of the *Cumulative Distribution Function* (CDF) of the standard Gaussian distribution:

$$\Phi\left(z\right) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{-t^2/2} \mathrm{d}t \tag{2.24}$$

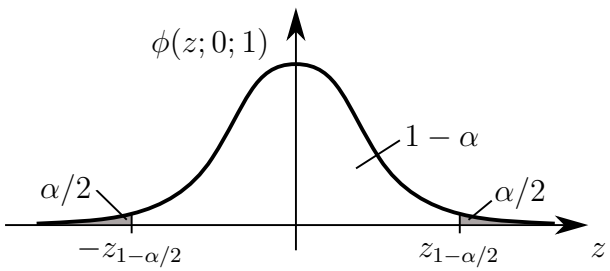


Figure 2.6: $1-\alpha/2$ *quantile for standard normal distribution*

Neither the CDF nor its inverse can be expressed in terms of elementary functions. A table for common values of $(1-\alpha)$ is given in appendix A.2.

Equation (2.23) can be rearranged to yield a quadratic inequality in P_F:

$$\begin{aligned} \frac{\left|\hat{P}_F - P_F\right|}{\sqrt{\left(P_F\left(1-P_F\right)\right)/N}} &< z_{1-\alpha/2} \\ \frac{\left(\hat{P}_F - P_F\right)^2}{\left(P_F\left(1-P_F\right)\right)/N} &< z_{1-\alpha/2}^2 \\ \left(\hat{P}_F - P_F\right)^2 &< \frac{P_F\left(1-P_F\right)}{N} z_{1-\alpha/2}^2 \end{aligned} \tag{2.25}$$

The solution of the inequality gives the following $(1-\alpha)$ confidence interval for P_F:

$$\begin{aligned} &\frac{N}{N+z_{1-\alpha/2}^2}\left(\hat{P}_F + \frac{1}{2N}z_{1-\alpha/2}^2 - \sqrt{\frac{z_{1-\alpha/2}^2}{N}\hat{P}_F\left(1-\hat{P}_F\right) + \left(\frac{z_{1-\alpha/2}^2}{2N}\right)^2}\right) < P_F \\ &< \frac{N}{N+z_{1-\alpha/2}^2}\left(\hat{P}_F + \frac{1}{2N}z_{1-\alpha/2}^2 + \sqrt{\frac{z_{1-\alpha/2}^2}{N}\hat{P}_F\left(1-\hat{P}_F\right) + \left(\frac{z_{1-\alpha/2}^2}{2N}\right)^2}\right) \end{aligned} \tag{2.26}$$

For estimation of failure probabilities, often only an upper bound is of interest, for the cases that an event happens more likely than estimated. The single-sided confidence interval is defined as (see also figure 2.7):

$$P\left(0 < P_F < \hat{P}_{up}\right) = 1-\alpha \tag{2.27}$$

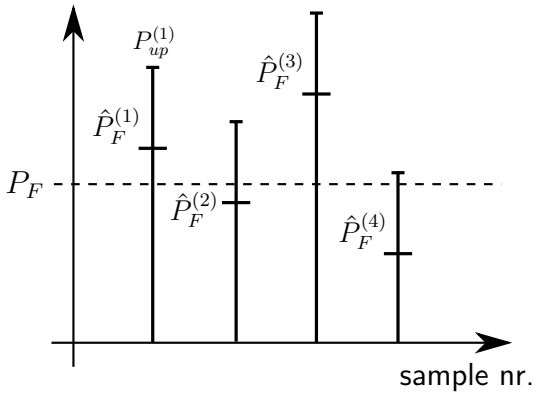


Figure 2.7: *Examples for single sided confidence intervals*

The equation for the upper bound can be derived analogously to the double-sided interval and is given by:

$$0 < P_F < \frac{N}{N+z_{1-\alpha}^2}\left(\hat{P}_F + \frac{1}{2N}z_{1-\alpha}^2 + \sqrt{\frac{z_{1-\alpha}^2}{N}\hat{P}_F\left(1-\hat{P}_F\right) + \left(\frac{z_{1-\alpha}^2}{2N}\right)^2}\right) \tag{2.28}$$

which is similar to equation (2.26) with the only difference that here the $z^2_{1-\alpha}$ confidence threshold is used instead of $z^2_{1-\alpha/2}$, which ensures that still with only a single-sided confidence interval the overall probability of exceeding the specified interval is α.

Reformulation of equation (2.23) and setting $z_{1-\alpha/2} = 1$, which corresponds to a single standard deviation with $(1-\alpha) = 68.3\%$, leads to

$$\left|\hat{P}_F - P_F\right| < \sqrt{\frac{P_F\,(1-P_F)}{N}} = P_F \cdot CoV \tag{2.29}$$

Hence, the coefficient of variation CoV corresponds to the standardized confidence interval $\left|\hat{P}_F - P_F\right| / P_F$ for $(1-\alpha) = 68.3\%$. Due to the unknown probability P_F, the CoV in equation (2.18) can only be approximated using the estimated probability $\hat{P}_F$. In contrast, the confidence intervals given in equations (2.26) and (2.28) only depend on known quantities. For both the CoV as well as the confidence intervals, the *De Moivre-Laplace* theorem must be fulfilled to obtain valid results.

A more general and easy to calculate upper bound for the estimation error is given by *Chebyshev*'s inequality. It is defined by [BN87, p. 219]:

$$P\left(\left|\hat{P}_F - P_F\right| < \varepsilon\right) \geq 1 - \frac{\sigma^2}{\varepsilon^2} \tag{2.30}$$

Substituting σ^2 by the equation for the variance of the binomial distribution, an upper bound for the estimated failure probability is obtained:

$$P\left(\left|\hat{P}_F - P_F\right| < \varepsilon\right) \geq 1 - \frac{P_F\,(1-P_F)}{N} \cdot \frac{1}{\varepsilon^2} \tag{2.31}$$

The main advantage of *Chebyshev*'s inequality is that it is valid for any arbitrary distribution, especially if it is very different from a Gaussian distribution. On the other hand, it is a very weak upper bound in case that samples are approximately Gaussian distributed as in the case of risk analysis. Figure 2.8 compares exemplarily the upper bound given by *Chebyshev*'s inequality to confidence intervals calculated by equation (2.26). The ε required for equation (2.31) is obtained by comparison of equations (2.22) and (2.30):

$$\varepsilon = c \Rightarrow \varepsilon = z_{1-\alpha/2}\sqrt{\frac{P_F\,(1-P_F)}{N}} \tag{2.32}$$

It can be seen that the intervals obtained by *Chebyshev*'s inequality are much higher, especially for high levels of confidence $(1-\alpha)$. Although *Chebyshev*'s bounds are easy to calculate, their conservativeness makes them impractical in the context of failure probability estimation.

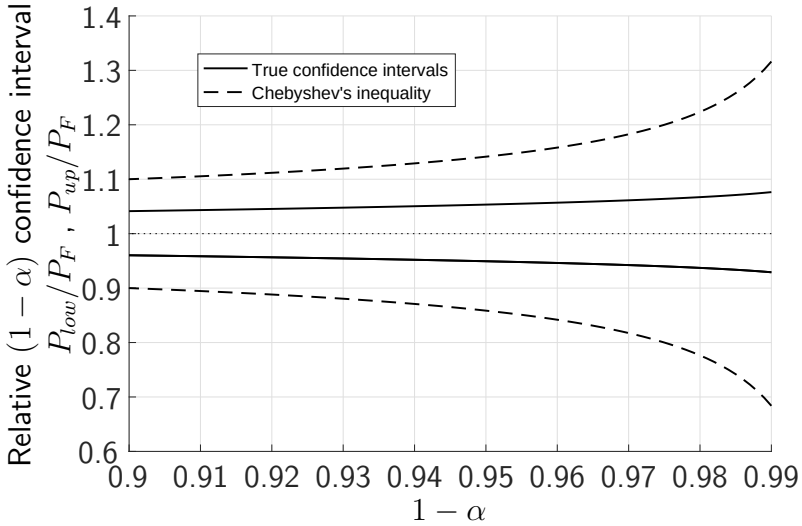


Figure 2.8: *Comparison between Chebyshev's inequality and confidence intervals for* $P_F = 10^{-6}, N = 10^9$

2.3.4 Conclusions on Standard Monte Carlo Simulation

Since its invention in the 1940s, Monte Carlo simulation has proven to be a valuable method for stochastic analysis. It is easy to understand and implement. Furthermore, it is a black box method: it is sufficient to consider only inputs and outputs of a system while no knowledge of its internal working is required. The algorithm allows a high level of parallelization, which makes it efficient for many applications. However, the required number of samples grows hyperbolically with decreasing probability, which makes conventional Monte Carlo simulation no viable option for estimation of small probabilities.

2.4 Introduction to Variance Reduction Techniques

The performance of stochastic simulation methods is determined by the product of cost and variance, i.e. lower computational cost and lower variance of the estimation characterize a more performant algorithm. Such algorithms are referred to as variance reduction techniques, where either the variance with respect to a given number of samples is reduced or the number of samples, which highly correlates with computational cost, for a desired variance. This can be achieved through a more efficient investigation of rare events by generating more failure samples. Variance reduction techniques can be divided into two groups:

- **Direct simulation**: More samples in the failure region are directly generated using preknowledge about the system and its failure modes.
- **Adaptive simulation**: New samples closer to failure region are adaptively generated in the course of evaluation based on older samples.

Loosing generality and robustness comes along with most variance reduction techniques. While there are for example efficient algorithms for evaluation of failure events for linear systems,

these algorithms are usually not applicable to the general case. The subsequent sections will describe variance reduction techniques that are of relevance for this research. For a general overview of enhanced stochastic algorithms beyond the ones presented here, see e.g. [RT09, GT13].

2.5 Importance Sampling

2.5.1 Algorithm

Importance Sampling (IS) is a representative of direct simulation methods. The idea is that certain realizations of uncertain parameters, i.e. regions in the uncertain parameter domain, have more impact on the estimated failure probability. Hence, instead of using the original distribution of the uncertain parameters $Q(\boldsymbol{\theta})$ for generation of samples, an importance sampling probability density function, also refereed to as importance sampling density $S(\boldsymbol{\theta})$ is chosen. This density utilizes pre-knowledge to directly generate more samples in relevant regions. Since this would lead to a biased failure probability estimation using the standard Monte Carlo estimator (2.8), the estimator is adjusted by weighting the individual samples with the likelihood ratio, which is the ratio of the probability of the original to the importance sampling density:

$$P_F = \int \frac{Q(\boldsymbol{\theta})}{S(\boldsymbol{\theta})} I_F(\boldsymbol{\theta})\, S(\boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta} = E_S\left[\frac{Q(\boldsymbol{\theta})}{S(\boldsymbol{\theta})} I_F(\boldsymbol{\theta})\right] \tag{2.33}$$

The subscript S of the expectation E_S denotes that the samples of the uncertain parameter vector $\boldsymbol{\theta}$ are distributed according to the importance sampling density $S(\boldsymbol{\theta})$. The probability can be estimated using equation (2.34) if the samples $\boldsymbol{\theta}_i, i = 1 \dots N$ are distributed according to $S(\boldsymbol{\theta})$:

$$\hat{P}_F = \frac{1}{N} \cdot \sum_{i=1}^{N} \frac{Q(\boldsymbol{\theta}_i)}{S(\boldsymbol{\theta}_i)} I_F(\boldsymbol{\theta}_i) \tag{2.34}$$

The importance sampling algorithm is very similar to the conventional Monte Carlo algorithm and can be formalized as follows:

Importance Sampling algorithm

1. Generate N independent and identically distributed uncertain parameter samples $\boldsymbol{\theta}_i, i = 1 \dots N$ distributed according to an importance sampling density $S(\boldsymbol{\theta})$.
2. For each sample, run a simulation to obtain the according system response $\mathbf{x}(\boldsymbol{\theta}_i)$, evaluate the indicator function $I_F(\boldsymbol{\theta}_i)$ and calculate the likelihood ratio $Q(\boldsymbol{\theta}_i)/S(\boldsymbol{\theta}_i)$.
3. Average the products of indicator function and likelihood ratio of all samples using equation (2.34).

2.5.2 Variance Estimation

According to [AW14, pp. 48-49], the coefficient of variation of the Importance Sampling failure probability estimation (2.34) can be calculated by

$$CoV_{IS} = \sqrt{\frac{CoV_F^2 + 1}{P_{F,S}} - 1} \tag{2.35}$$

where $P_{F,s}$ is the probability that a sample $\boldsymbol{\theta}$ drawn from the importance sampling density $S(\boldsymbol{\theta})$ is a failure sample, i.e. that the indicator function $I_F(\boldsymbol{\theta}) = 1$:

$$P_{F,s} = \int I_F(\boldsymbol{\theta})\, S(\boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta} \tag{2.36}$$

CoV_F describes the variability of the likelihood ratio in the failure region, where the expectation in the denominator is used for normalization of the variance:

$$CoV_F = \frac{\sqrt{\mathrm{var}\left[\frac{Q(\boldsymbol{\theta})}{S(\boldsymbol{\theta})} \mid F\right]}}{E\left[\frac{Q(\boldsymbol{\theta})}{S(\boldsymbol{\theta})} \mid F\right]} \tag{2.37}$$

Figure 2.9 shows exemplary importance sampling densities $S_1(\theta)$ and $S_2(\theta)$ where the response variable is chosen to be equal the uncertain parameter $r = \theta$ and the objective is to find $P(r > \theta_{limit})$. The shaded areas indicate the probability that a sample drawn from the importance sampling density lies in the failure region, which is calculated by (2.36). The importance sampling density $S_1(\theta)$ results in a lower variability of the likelihood ratio, which is the ratio between the original uncertain parameter distribution $Q(\theta)$ and the importance sampling density $S(\theta)$ in the failure region. The ratio $Q(\theta)/S_1(\theta)$ changes much less over θ than the ratio $Q(\theta)/S_2(\theta)$.

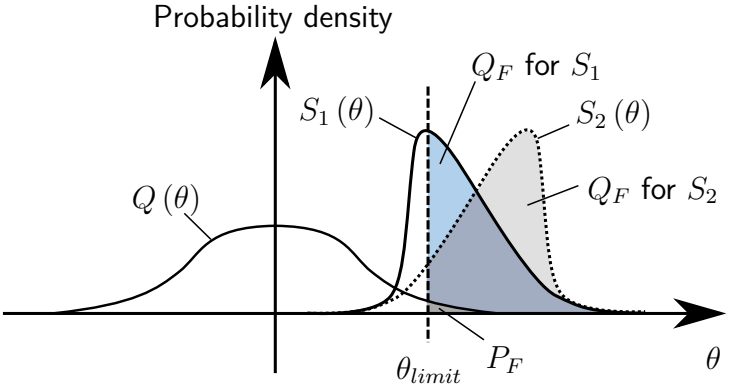


Figure 2.9: *Variability of the likelihood ratio in the failure region*

According to equation (2.35), the number of samples in the failure region must be maximized and the variability of the likelihood ratio minimized to minimize the variance of importance sampling estimations. Theoretically, the coefficient of variation CoV_{IS} of the estimated probability can be zero, i.e. the failure probability can be perfectly estimated without any error.

This is the case if the importance sampling density is the original parameter distribution $Q(\boldsymbol{\theta})$ conditional on the failure region:

$$S(\boldsymbol{\theta}) = Q(\boldsymbol{\theta}|F) = Q(\boldsymbol{\theta}) \frac{I_F(\boldsymbol{\theta})}{P_F} \tag{2.38}$$

which means that all samples lie in the failure domain and their relative distribution is equal to the original distribution. The ideal distribution is shown in figure 2.10. Unfortunately, samples cannot be directly drawn from this distribution since the probability P_F is usually unknown and the evaluation of $I_F(\boldsymbol{\theta})$ is computationally expensive. However, it underlines the requirement that an adequate importance sampling density $S(\boldsymbol{\theta})$ should lead to many failure samples and has a small variability of the likelihood ratio in the failure region. On the contrary, an unfavorable importance sampling density could lead to a worse variance than conventional Monte Carlo simulation.

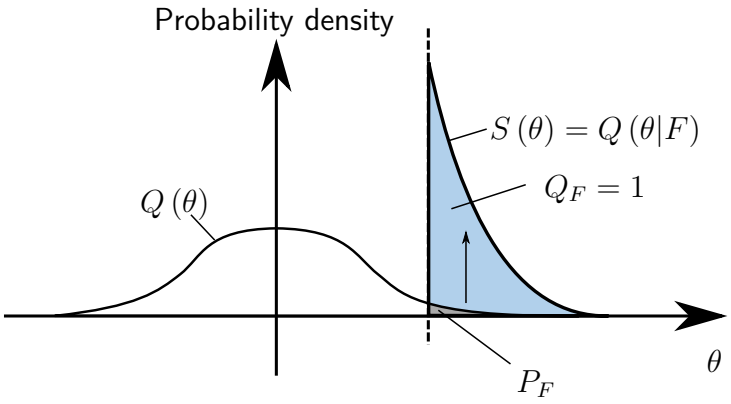


Figure 2.10: *Ideal importance sampling distribution*

2.5.3 Selection of the Importance Sampling Density

Selection of the Importance Sampling density is a challenging task and usually requires knowledge about probable failure regions. In civil engineering, methods are applied that approximate the failure region based on most probable points [ZO99, KC06]. These points in the uncertain parameter domain theoretically give the highest probability of failure occurrence. One idea for generation of an importance sampling density is to use densities centered on such most probable points. This concept is referred to as "shifting distribution", especially if the original parameter distribution is kept and only the mean is shifted to the location of the most probable points [DC00].

To illustrate this, consider the following example: The problem is described by two uncertain parameters θ_1 and θ_2. Both follow a standard Gaussian distribution, i.e. a Gaussian distribution with zero mean and unit variance. The failure domain is described by hyperbolas that fulfill the equation

$$\theta_1^2 - \theta_2^2 > 3^2 \tag{2.39}$$

For this example, the most probable points can be determined analytically, which are the points of the hyperbolas $\theta_1^2 - \theta_2^2 = 3^2$ that are closest to the origin. Figure 2.11 shows samples $\boldsymbol{\theta}_i$ generated by conventional Monte Carlo simulation in the upper plot and Importance Sampling in the lower plot, where the original parameter distribution is shifted to the two design points. While for conventional Monte Carlo simulation only very few samples lie in the failure region, with importance sampling the number of failure samples is much higher.

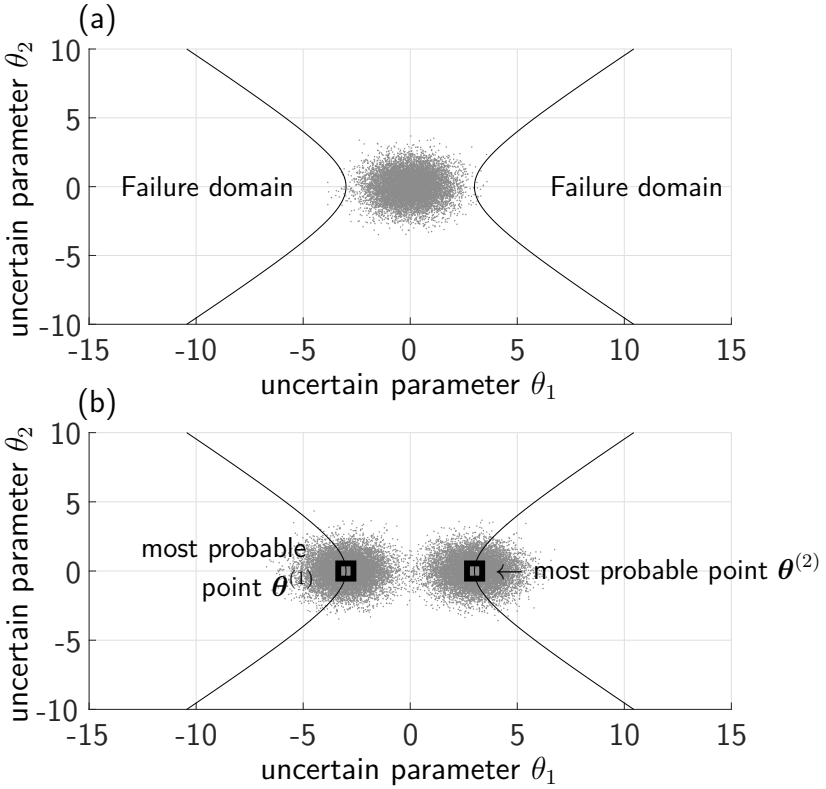


Figure 2.11: *Samples for conventional Monte Carlo and Importance Sampling. (a) Conventional Monte Carlo sampling; (b) Importance sampling*

Ideally, the importance sampling estimator has the same properties as the standard Monte Carlo estimator, i.e. the estimation is bias-free and convergence to the actual failure probability is given for large number of samples. However, this only holds as long as the importance sampling density is correctly chosen. Assume the case that in the previous example only the left most probable point $\boldsymbol{\theta}^{(1)}$ is considered for the Importance Sampling density. This leads to a total neglection of the right failure region and hence the estimated failure probability is biased. Figure 2.12 shows exemplary simulation histories for the estimated failure probability $\hat{P}_F$ versus number of samples for conventional Monte Carlo simulation as well as Importance Sampling with correct and incorrect importance sampling density. The estimated probability for Monte Carlo simulation shows a high variation due to the low number of samples in the failure region and converges only slowly to the actual failure probability of $P_F = 1.8e-3$. In contrast, the results for importance sampling both quickly converge to a certain value. However, the convergence of the importance sampling with only a single design point (dotdashed line) is deceptive since it converges to a wrong value.

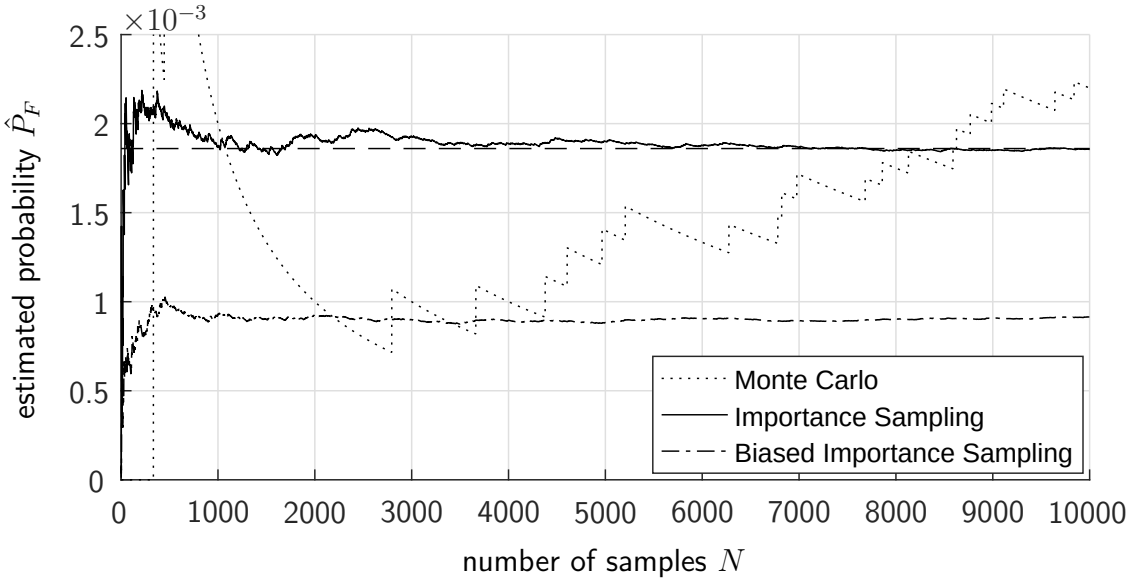


Figure 2.12: *Simulation histories for the estimated failure probability using Monte Carlo and Importance Sampling; dashed line is true failure probability* P_F

2.5.4 Conclusions on Importance Sampling

The importance sampling density must be carefully chosen to obtain correct results, especially since a wrong choice can result in a deceptive convergent result towards a biased probability. Although there exist methods for linear systems to identify the most probable points, this is not the case for general systems. There are attempts to apply importance sampling in high dimensions [AB03], however it is virtually impossible to design an importance sampling density for a high number of uncertain parameters that correctly covers the most probable areas in the high-dimensional parameter space. Nevertheless, importance sampling can be a useful tool for certain cases or to efficiently re-evaluate estimations after approximate failure regions were identified using other algorithms.

In the next chapter, a powerful method is described that does not need any preknowledge about the system and is nevertheless able to efficiently generate more samples in relevant regions. Although this method does not reach the performance of ideal Importance Sampling, it is much more robust and easier to apply.

2.6 Subset Simulation

2.6.1 Introduction

A high number of samples is required for conventional Monte Carlo simulation to obtain sufficient samples in the failure region and hence an adequate confidence in the estimated failure probability. The idea of Subset simulation is to describe a highly improbable event leading to a failure F by a sequence of conditional failure events with higher conditional

failure probabilities F_i:

$$P_F = \prod_{k=1}^{m} P(F_k|F_{k-1}) = \prod_{i=1}^{m} P_{F_k} \tag{2.40}$$

where m is the number of conditional failure levels and F_k are the conditional failure events with decreasing probability $F_0 = \mathbb{R}^k \supset F_1 \supset \ldots \supset F_m = F$. $P(F_k|F_{k-1})$ is the conditional failure probability, i.e. the probability that the event F_k happens given that the samples lie in the failure region F_{k-1}. The procedure is explained using figure 2.13. Subfigure a) shows the k-dimensional parameter domain, the failure domain F for which the probability of occurrence is looked for and samples generated by conventional Monte Carlo simulation distributed according to the distribution of the uncertain parameter vector $Q(\boldsymbol{\theta})$. Instead of increasing the number of samples to obtain samples in the failure domain F, a conditional failure domain $F_1 \supset F$ is introduced, which is a superset of F (subfigure b). The probability for the intermediate failure event F_1 is usually much larger than the probability of the actual failure event F. Hence considerably less samples are required to accurately estimate the conditional failure probability P_{F_1}. The first failure domain is also referred to as first subset, since it is a subset of the overall parameter space. In the next step, samples distributed according to $Q(\boldsymbol{\theta}|F_1)$ are generated based on the samples already lying in F_1, see subfigure c). The samples are distributed similar to their original distribution $Q(\boldsymbol{\theta})$ but conditional on the first subset, i.e. only samples lying in the first conditional failure region F_1 are generated. This distribution is described by

$$Q(\boldsymbol{\theta}|F_1) = Q(\boldsymbol{\theta}) \frac{I_{F_1}(\boldsymbol{\theta})}{P_{F_1}} \tag{2.41}$$

By definition, all of these samples lie within the failure region F_1 and hence are in average closer to the target failure region. If there are still not sufficient samples in the actual failure domain F, the procedure is repeated. This is shown exemplarily in subfigure d), where another intermediate failure domain $F_2 \subset F_1$ is introduced, which is a subset of the first conditional failure domain F_1 and hence is also referred to as second subset. Now, samples are generated conditional on the failure domain F_2 according to $Q(\boldsymbol{\theta}|F_2)$ (subfigure e). In the example, sufficient samples of the second subset lie in the actual failure domain F, see subfigure f).

In general, this procedure is repeated until there are sufficient samples in the target failure region and the conditional probability $P(F|F_k)$ is large enough. In practice, the intermediate failure events are defined by increasing limit values of the response variable r. This means if $P_{F_i} = P(r < r_{i,limit})$, then $r_{1,limit} < r_{2,limit} < \ldots < r_{k,limit} = r_{limit}$. Since the conditional probabilities are relatively high compared to the target probability, the number of samples required to obtain accurate estimates of the conditional failure probabilities and hence the total number of samples to estimate the failure probability is lower – this will be discussed in detail in section 2.6.4. Eventually, the actual failure probability P_F is the product of the conditional failure probabilities, see equation (2.40).

The high number of samples close to and in the failure region facilitates detailed failure

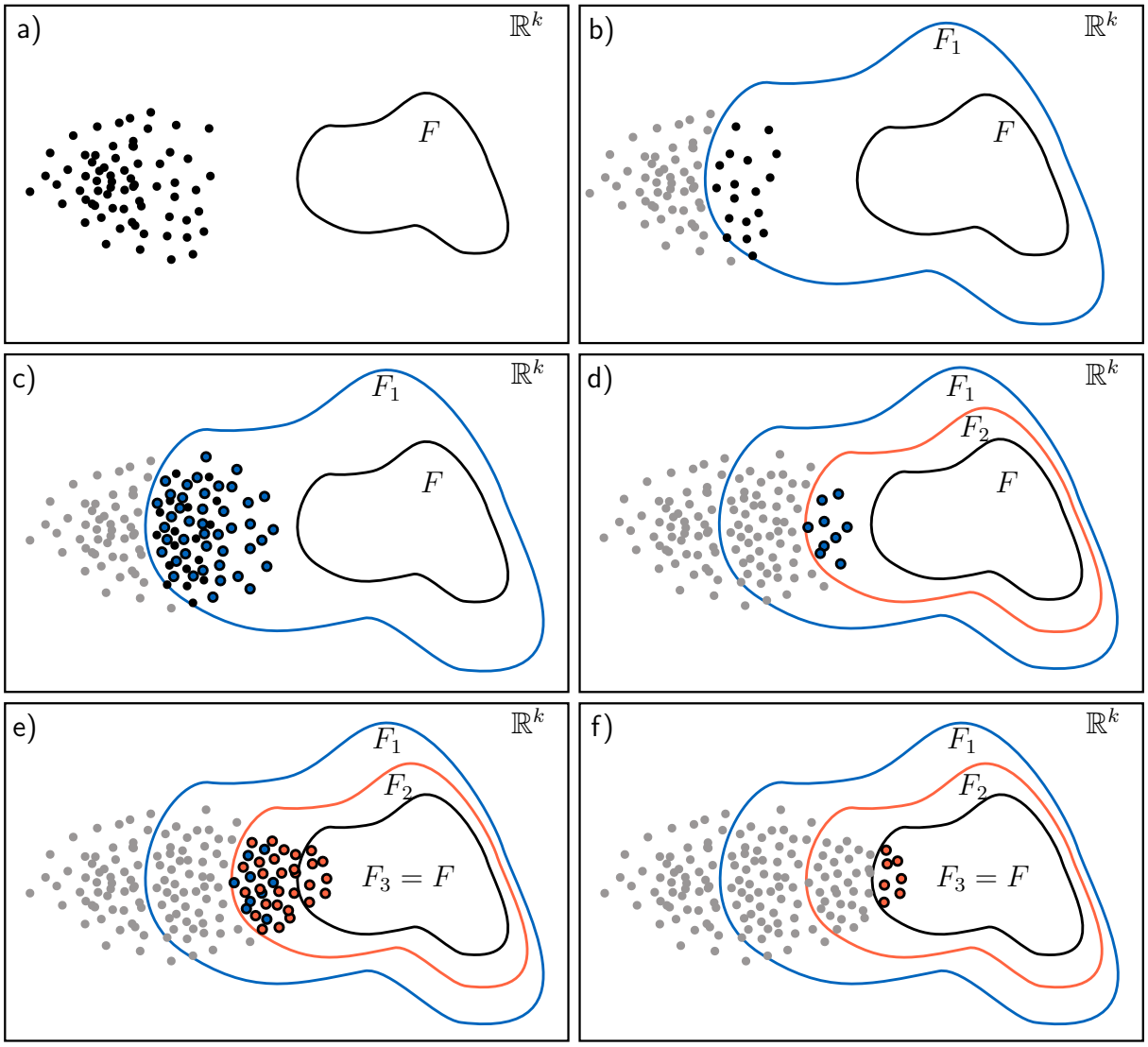


Figure 2.13: *Principle idea of Subset simulation*

analysis, i.e. evaluation of failure causes and consequences, since each sample is linked to an individual scenario leading to the failure event. Furthermore, the gradual generation of samples towards the failure region enables accurate estimation of the CDF and especially the *Complementary Cumulative Distribution Function* (CCDF) of the response variable. In contrast to the conventional CDF, which describes the probability that a random value lies within an interval from minus infinite to a certain threshold X, the CCDF gives the probability that a random variable lies above a particular threshold X. It is defined as

$$\bar{Q}(X) = P(x > X) = \int_X^\infty q(x)\,dx = 1 - Q(X) \tag{2.42}$$

where $Q(x)$ and $\bar{Q}(x)$ denote the cumulative and complementary cumulative distribution function respectively and $q(x)$ the probability density function, see also figure 2.14. The main advantage of the CCDF is the direct information about the probability of exceeding any arbitrary threshold X. Particularly, for risk analysis this means that for any threshold

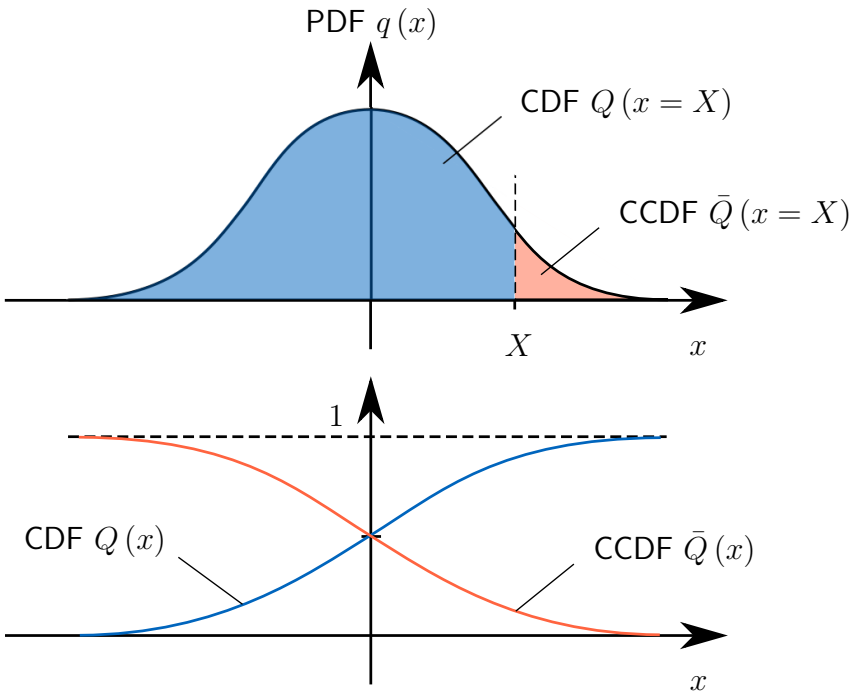


Figure 2.14: *Complementary cumulative distribution function*

X the according probability of exceedance can be directly obtained and vice versa, for a certain admissible failure probability the according threshold X directly results. Although the conventional CDF indirectly offers the same information, the CCDF enables direct access to this information, which is valuable for risk analysis. To obtain the CCDF from Subset simulation results, the definition of the response variable equation (2.3) is utilized: The probability of failure, or more general the probability that the response variable r exceeds a certain threshold r_{limit}, is determined by the ratio of samples lying above the threshold to the total number of samples. Since the actual response variable r of each sample is known, the comparison with arbitrary limit values r_{limit} can be performed by post-processing with low computational effort. To obtain the CCDF, consequently the limit value can be defined as $r_{limit} = x$, with $x = -\infty \dots X$, and by counting the samples lying above the individual threshold and division by the total number of samples, the according value of the CCDF $Q(x)$ is obtained.

The main challenge of Subset simulation is the generation of conditional failure samples that are distributed according to $Q(\boldsymbol{\theta}|F_i)$. These conditional distributions are usually no standard distributions and not a priori known, which renders direct generation of samples impossible. For that task, *Markov Chain Monte Carlo* (MCMC) simulations are used where conditional failure samples are generated based on previous samples. Hence, Subset simulation counts to the adaptive simulation methods among the variance reduction methods.

In the following, first MCMC with the underlying theory is introduced, followed by the description of the Subset algorithm and an analysis of the achievable estimation variance reduction.

2.6.2 Markov Chain Monte Carlo

Introduction

Markov Chain Monte Carlo simulation uses Markov chains for generation of samples from arbitrary distributions. It is especially used when samples cannot or not efficiently be generated directly from a known distribution. In a later step it will be explained how Markov Chain Monte Carlo supports rare event sampling, specifically the generation of conditional failure samples where the conditional failure distributions are actually unknown.

Markov Chains

A Markov chain is a stochastic process that satisfies the Markov property, which is also referred to as memoryless property. This property says that the future states of a Markov chain only depend on the current state. A Markov chain is fully described by its initial state and the transition probabilities that describe the transition from the i-th chain state $\theta^{(i)}$ to the $(i+1)$-th state $\theta^{(i+1)}$. For Markov chains with discrete state space, the transition probability is given by a transition matrix, see also figure 2.15 for an example with a Markov chain with only two possible states x_1 and x_2.

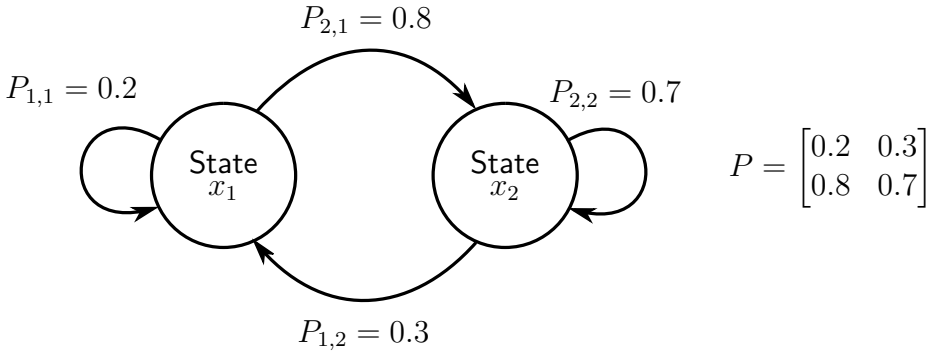


Figure 2.15: *Discrete-state Markov chain with two states*

The counterpart to the transition matrix in continuous state space is the transition probability density function. Figure 2.16 shows a single discrete time Markov chain with continuous state space. $\theta^{(i)}$ denotes the state of the Markov chain at the i-th step where the superscript in brackets denotes the step number and is not to be understood as exponent. Each chain element can take an arbitrary value within a continuous parameter space in contrast to the discrete chain example in figure 2.15, where the only possible values are x_1 and x_2. The transition from one chain element to the next – i.e. from one state to the next – is described by the conditional transition probability density function $P_{\theta^{(2)}|\theta^{(1)}}(b|a)$ that describes the probability of $\theta^{(2)}$ being equal to b given that $\theta^{(1)}$ is a.
For further analysis, it is useful to introduce an ensemble of Markov chains, which is the aggregation of multiple Markov chains with the same properties, i.e. same transition probability density functions, but with different initial values, see figure 2.17.

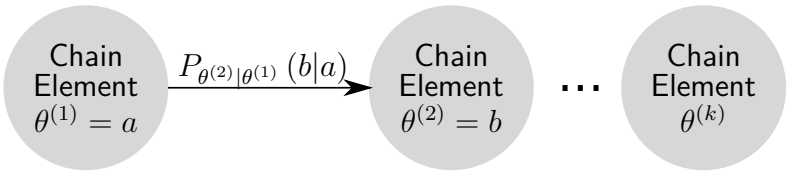


Figure 2.16: *Time descrete Markov chain with continuous states*

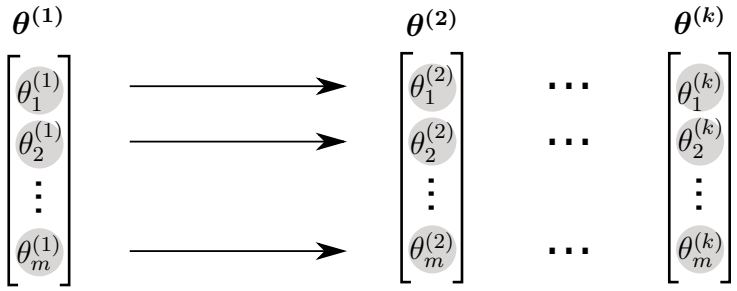


Figure 2.17: *Discrete time Markov chain with continuous states*

In the following, only discrete time Markov chains with continuous state space are considered since only those are suitable for generation of samples from arbitrary distributions. For more information on Markov chains with discrete state space it is referred to literature, e.g. [Beh00].

For generation of samples from an arbitrary distribution, a Markov chain is required that has the desired distribution as equilibrium distribution. For a single Markov chain, equilibrium distribution means that the aggregation of all samples after n burn-in steps $[\theta^{(n+1)}, \theta^{(n+2)}, \ldots]$ follows a constant distribution. For a chain ensemble, the same explanation holds, however if the size of the ensemble is large enough, also the components of the individual chain elements after the burn-in phase $\boldsymbol{\theta}^{(n+1)} = \left[{\theta_1}^{(n+1)}, {\theta_2}^{(n+1)}, \ldots\right]$ follow a constant distribution. This constant distribution is referred to as equilibrium distribution. A Markov chain does not necessarily have an equilibrium distribution. A sufficient condition for a Markov chain to converge to an equilibrium distribution is that it fulfills the detailed balance criterion and is ergodic.

Detailed Balance Criterion

The detailed balance criterion is also referred to as reversibility. If fulfilled, the transition rate is uniform between two arbitrary states when the Markov chain is in its stationary state. The detailed balance criterion can be written as

$$P_{\boldsymbol{\theta}^{(i+1)}|\boldsymbol{\theta}^{(i)}}(\mathbf{b}|\mathbf{a})\, P_e(\mathbf{a}) = P_{\boldsymbol{\theta}^{(i+1)}|\boldsymbol{\theta}^{(i)}}(\mathbf{a}|\mathbf{b})\, P_e(\mathbf{b}) \tag{2.43}$$

where P_e is the equilibrium distribution of the Markov chain. For a chain ensemble, this criterion reads as follows: If the components of the n-th element of a chain ensemble $\boldsymbol{\theta}^{(n)}$ are distributed according to the equilibrium distribution P_e, then the samples of all $\boldsymbol{\theta}^{(n+i)}, i =$

$1, 2, \ldots$ are distributed according to this distribution. Figure 2.18 shows an example where a chain ensemble with 100 components $\theta_1 \ldots \theta_{100}$ is initiated according to the equilibrium distribution, indicated by the black solid line. Note that this line is not the approximated distribution extracted from the samples but the target distribution, i.e. the distribution the samples should have. To obtain finer resolution of the histogram, plots were obtained using 40 individual chain ensembles with 100 components each. A histogram gives the number of samples in defined intervals, i.e. the height of the bars represent the number of samples lying within the according bin. Using a transition probability that ensures detailed balance (discussed later), also the second and tenth chain element follows the same equilibrium distribution. Deviations are caused by statistical outliers.

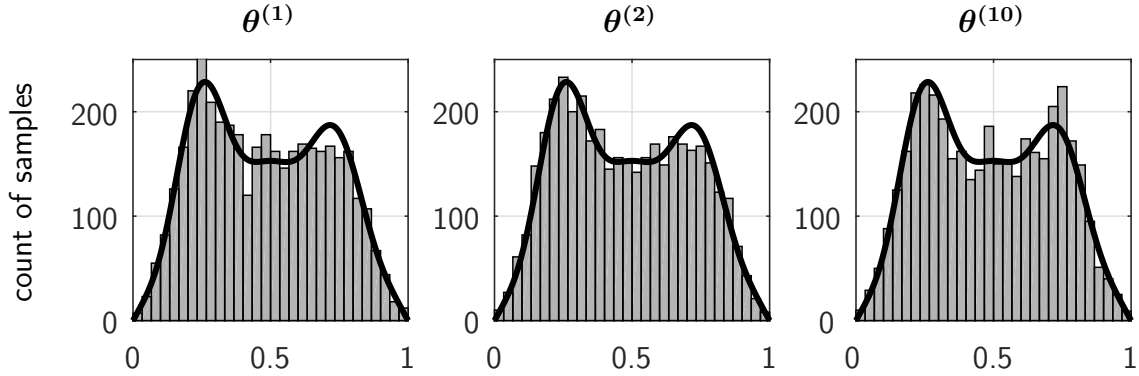


Figure 2.18: *Markov chain ensemble with stationary initiation that fulfills detailed balance condition; plots are for first, second and 10th step; solid line indicates target distribution*

While the detailed balance criterion ensures that a Markov chain stays in its equilibrium state, it is not ensured that it converges to the equilibrium state for arbitrary initial conditions.

Ergodicity

A Markov chain with discrete states is called ergodic if it is possible to go from every state to every other state, however which must not necessarily be possible in one step. For a continuous state space, this means that there is a non-zero probability that the Markov chain visits any area in the state space. If a Markov chain is ergodic, then it is guaranteed that if the initial chain ensemble $\boldsymbol{\theta}^{(1)}$ is distributed according to any arbitrary distribution S, it converges to the equilibrium distribution P_e:

$$\lim_{n \to \infty} S^{\boldsymbol{\theta_n}}(\mathbf{a}) = P_e(\cdot) \tag{2.44}$$

where $S^{\boldsymbol{\theta_n}}(\mathbf{a})$ denotes the distribution of the chain states at the n-th step, if the chain is initialized with $\boldsymbol{\theta^{(1)}} = \mathbf{a}$. This criterion holds for a chain ensemble as well as for a single chain, where the aggregation of all chain elements after the n-th step are distributed according to the equilibrium distribution P_e. No algorithm exists that ensures ergodicity of a Markov chain. However, several examples are given below to show the contributing factors that influence ergodicity.

First, assume a chain ensemble with 10000 Markov chains that are all initialized with the

same value 0.4, see figure 2.19. After the first step, the components of the chain ensemble start to diversify. Already the components of the fifth chain element (i.e. after the fourth step) are almost perfectly distributed according to the equilibrium distribution, which is again indicated by the solid line. The steps until the chain reaches its equilibrium are called burn-in. The Markov chain used in this example is obviously ergodic. In the next example, a single

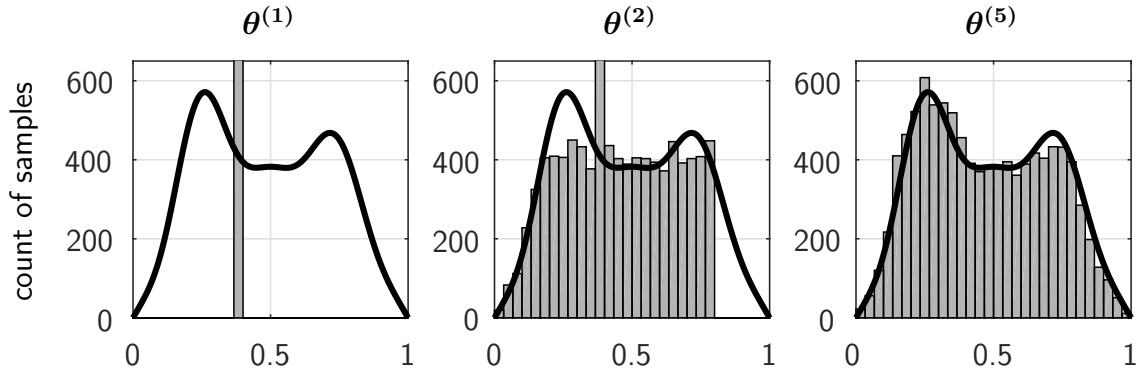


Figure 2.19: *Markov chain ensemble with non-stationary initiation that fulfills detailed balance criterion; plots are for first, second and fifth step; solid line indicates target distribution*

Markov chain is considered which is initiated with a value of $\theta^{(1)} = 0.5$. Figure 2.20 shows the histograms of the initial state as well as the aggregation of the states of the first 100 and first 500 steps, i.e. where a random vector is composed by all samples from the first to the 100th and 500th step respectively. The black solid line represents the target distribution while the blue solid line is the approximated probability distribution based on the values of the samples shown by the histogram. The higher the number of considered samples, the better the equilibrium distribution is approximated. The lower graph shows the history of the Markov chain for all steps. Here it becomes apparent that the Markov chain performs a random walk where more probable regions of the equilibrium distribution are more often visited than others. The shaded background is a contour plot for the the approximate distribution of the aggregated samples up to the respective chain index. It gives the approximated probability density indicated by the blue lines in the upper figure, where darker regions indicate higher probabilities and vice versa. While the approximation is inaccurate and changes rapidly for lower number of samples, there are no more substantial changes of the resulting distribution after approximately 400 samples, hence it can be said that in this example the stationary distribution is well sampled, if the number of considered samples is larger than 400.

The random walk process can cause non-ergodicity if the desired distribution has non-connected regions. This is the case for the example given in figure 2.21. The plots have the same meaning as in the previous example. However, in this case the desired distribution has two separated regions and a gap in the middle. Due to a relatively small step size of the random walk process, the Markov chain cannot leave one subspace and hence the distribution of the aggregated samples does not converge towards the desired distribution. However, if a chain ensemble is initiated according to the equilibrium distribution, the chain again converges towards the equilibrium. This is shown in figure 2.22, where the simplest case was chosen with a chain

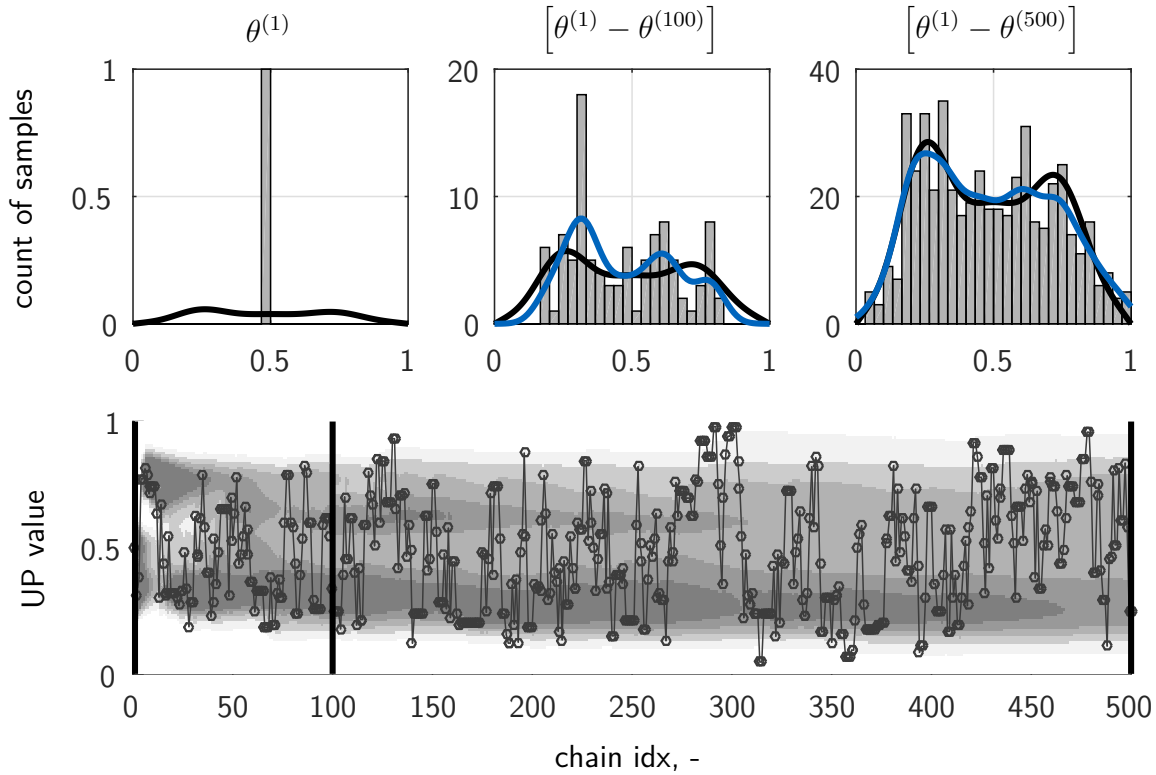


Figure 2.20: *Single ergodic Markov chain with cumulative histograms up to the first, 100th and 500th chain sample*

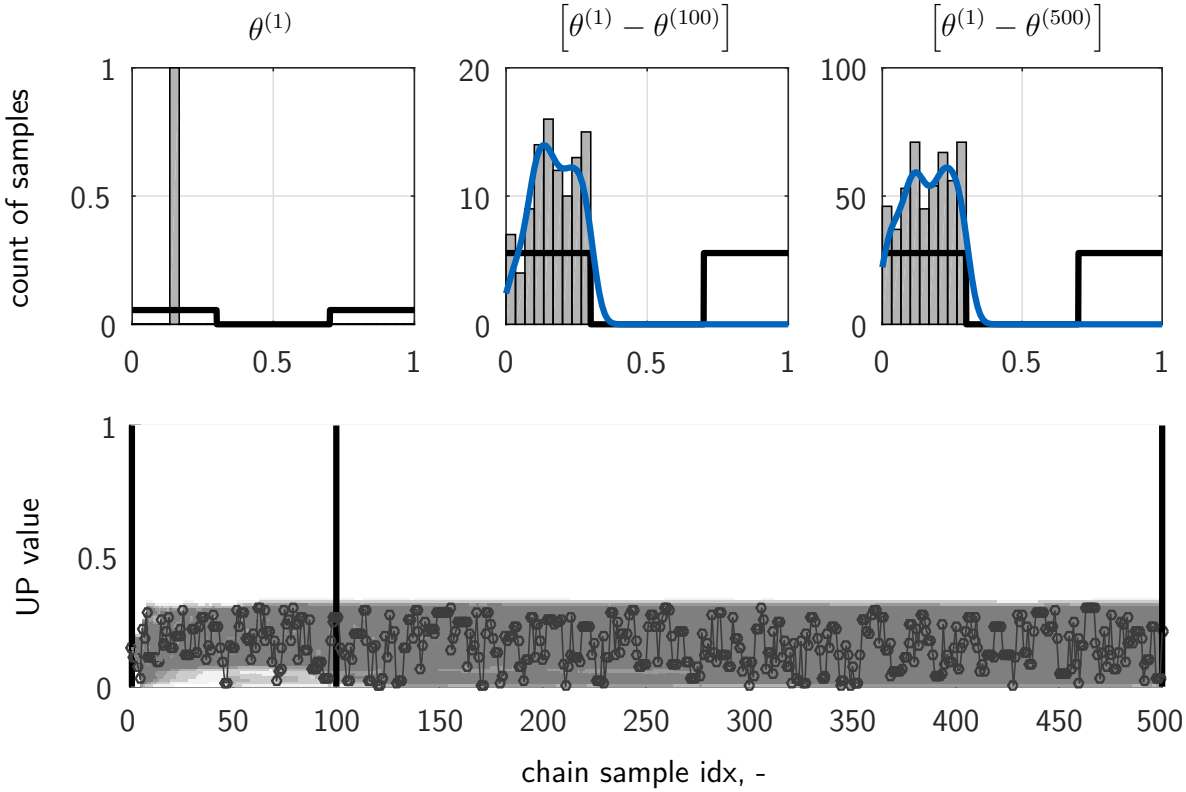


Figure 2.21: *Single non-ergodic Markov chain with cumulative histograms up to the first, 100th and 500th chain sample*

ensemble comprising two Markov chains, each initiated in one of the two separated regions. Furthermore, the average step size of the Markov chain is specified by the transition probability

density function. A wider transition PDF leads to bigger steps while a narrower transition PDF leads to smaller steps. Figure 2.23 give the results of the single chain example but with a

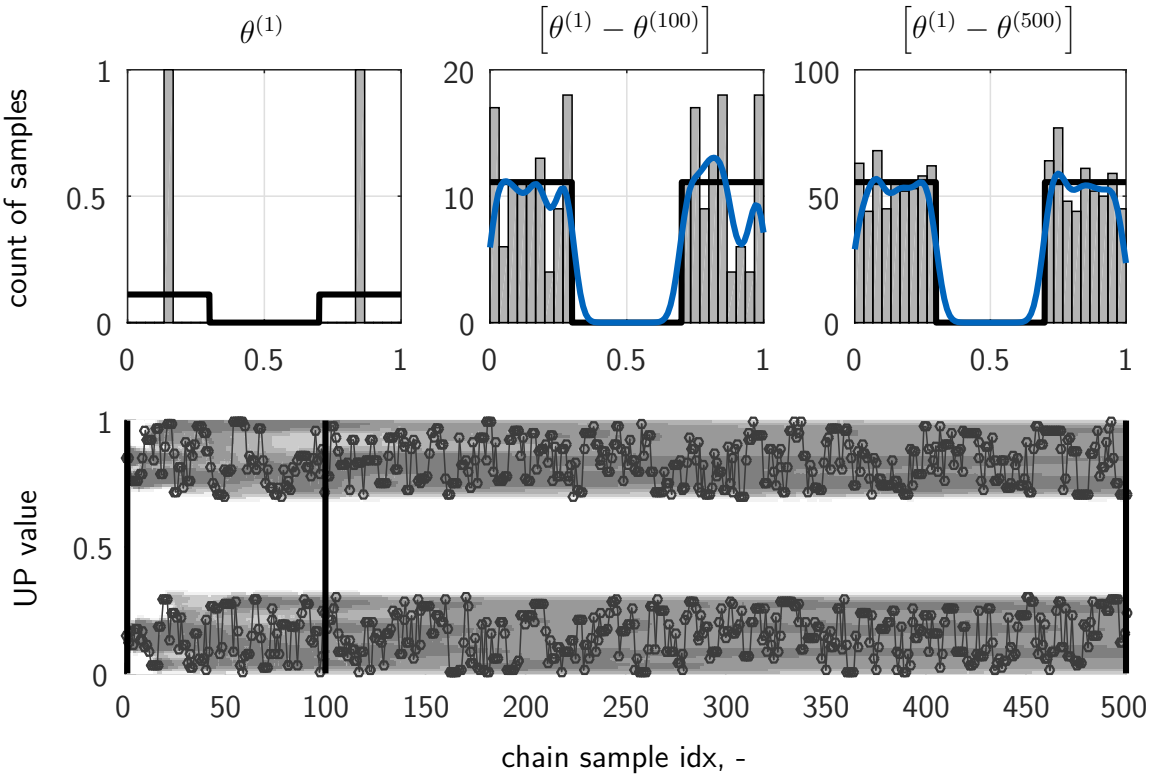


Figure 2.22: *Ensemble of two non-ergodic Markov chains initialized according to the stationary distribution; cumulative histograms up to the first, 100th and 500th chain sample*

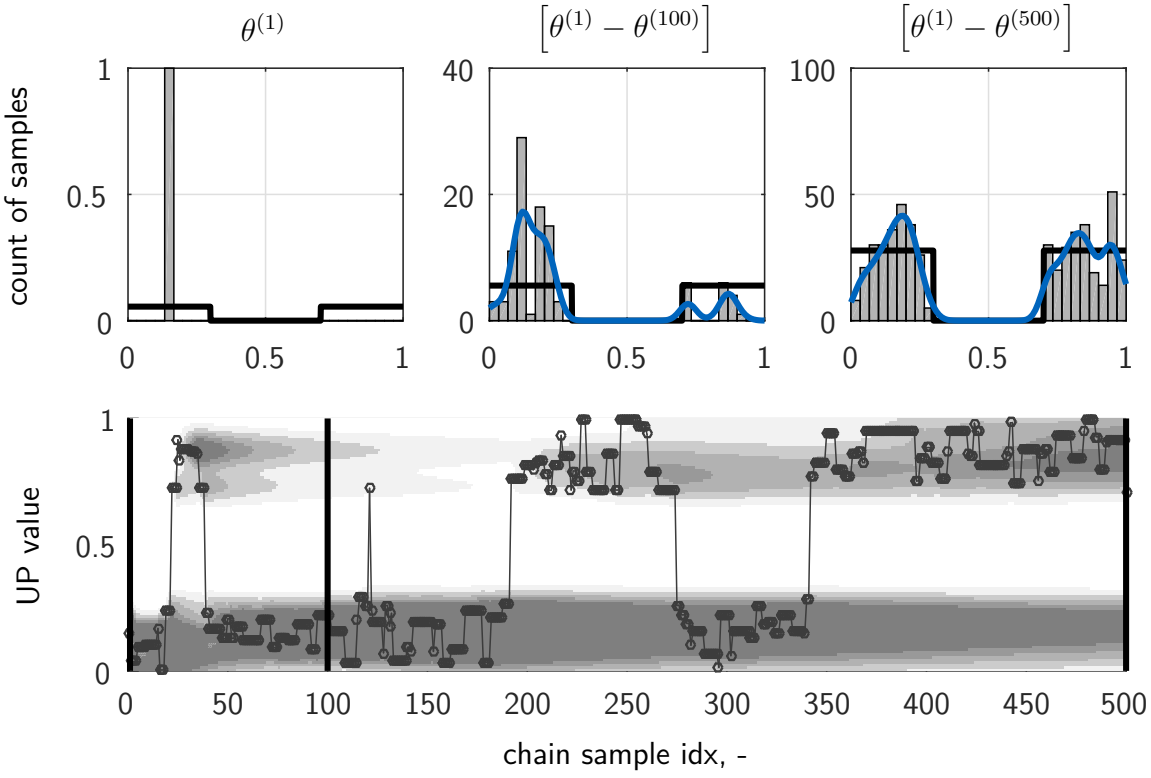


Figure 2.23: *Single ergodic Markov chain with higher step width; cumulative histograms up to the first, 100th and 500th chain sample*

bigger step size. Although this enables convergence to the equilibrium distribution with only one chain, bigger steps come along with other disadvantages, which will be discussed in the next section.
To summarize, ergodicity depends on the shape of the desired distribution as well as on the algorithm used to generate the transition probability density function. The desired distribution is usually connected, preventing non-ergodicity. The influence of the algorithm is discussed in the next chapter.

Generation of Markov Chains

There exist several ways to generate Markov Chains that fulfill the detailed balance criterion. The most common and fundamental method is the Metropolis algorithm [Met+53]. Many other algorithms are based on the original Metropolis algorithm, which is why it is explained in detail first.
The Markov chain fulfills a random walk where the next state only depends on the current state. The following Metropolis algorithm can be used to calculate the transition from one to the next state of a Markov chain, where the equilibrium distribution P_e is equal to the desired distribution D. The individual steps are discussed in detail below.

Metropolis algorithm for generation of samples from arbitrary distributions

1. Generate a candidate uncertain parameter vector $\boldsymbol{\theta}^{*(i+1)}$ from a proposal density $P^*\left(\cdot;\boldsymbol{\theta}^{(i)}\right)$, where $P^*\left(\cdot;\boldsymbol{\theta}^{(i)}\right)$ is a symmetric distribution about the current chain state $\boldsymbol{\theta}^{(i)}$.

2. Calculate the acceptance ratio r with the desired distribution D:

$$r = \min\left(\frac{D\left(\boldsymbol{\theta}^{*(i+1)}\right)}{D\left(\boldsymbol{\theta}^{(i)}\right)}, 1\right) \tag{2.45}$$

3. Set the new chain state: Generate a random number p from a uniform distribution in the range from 0 to 1.

$$\boldsymbol{\theta}^{(i+1)} = \begin{cases} \boldsymbol{\theta}^{*(i+1)} & \text{if } p < r \\ \boldsymbol{\theta}^{(i)} & \text{otherwise} \end{cases} \tag{2.46}$$

In the first step, a candidate uncertain parameter vector $\boldsymbol{\theta}^{*(i+1)}$ is generated where i is the current chain index and the asterisk denotes the candidate. The uncertain parameter vector of the current chain index $\boldsymbol{\theta}^{(i)}$ is referred to as seed vector, since it is the seed from which the next chain state is generated. The multivariate proposal density $P^*\left(\cdot;\boldsymbol{\theta}^{(i)}\right)$ describes the probability distribution of $\boldsymbol{\theta}^{*(i+1)}$ which is centered at the current chain state $\boldsymbol{\theta}^{(i)}$. This distribution must be symmetric about $\boldsymbol{\theta}^{(i)}$. The notation must not be confused with conditional distributions, where the arguments would be separated by a vertical bar $P^*\left(\cdot|\cdot\right)$. While the first argument of

the distribution denotes the k-dimensional uncertain parameter vector for which the probability of occurrence is calculated, the second argument – separated by a semicolon – represents a parameter of the distribution, which in this case is the center of the distribution. Often a Gaussian $\mathcal{N}$ or uniform distribution $\mathcal{U}$ is used where the mean is equal to $\boldsymbol{\theta}^{(i)}$, see figure 2.24.

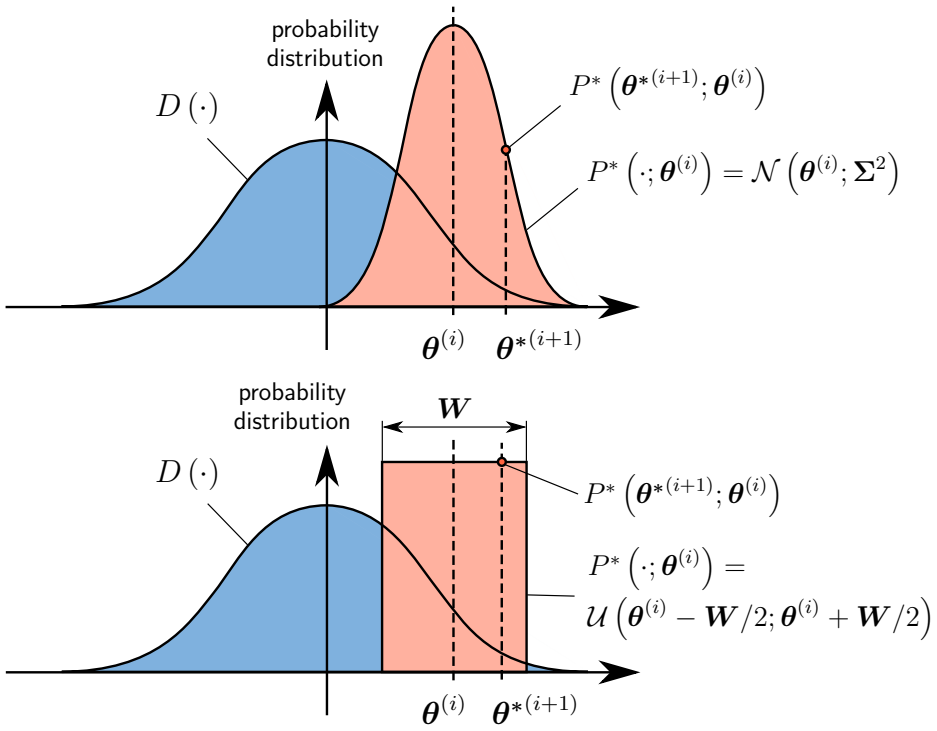


Figure 2.24: *Different proposal densities for the Metropolis algorithm; top: normal distribution, bottom: uniform distribution*

The width of the proposal distribution P^* can be interpreted as the step width of the random walk. Small step widths lead to slow progress of the chain. Hence, it can take many steps until the chain reaches its equilibrium distribution and the whole parameter space is adequately sampled. In contrast, bigger steps could lead to a faster convergence of the chain to its equilibrium distribution. However, the reasonable maximum step size is limited by the second step of the Metropolis algorithm.

In step two, the acceptance ratio r is calculated, which is the ratio of the probability of the desired distribution $D(\cdot)$ at the candidate $\boldsymbol{\theta}^{*(i+1)}$ and the seed $\boldsymbol{\theta}^{(i)}$, see figure 2.25. The acceptance ratio represents the probability with which the candidate is accepted in the third step. Since the probability of acceptance cannot be higher than one, the acceptance ratio r in equation (2.45) is bounded above by one. The candidate parameter vector is accepted with a probability r: This means, that using a random number distributed according to a uniform distribution between 0 and 1, the candidate is accepted if the random number is smaller than the acceptance ratio r. Otherwise, the seed is kept and is used as the new state. From the equation for the acceptance ratio follows that the candidate parameter is always accepted if it is more probable than the seed vector. As opposed to this, candidates that are less probable

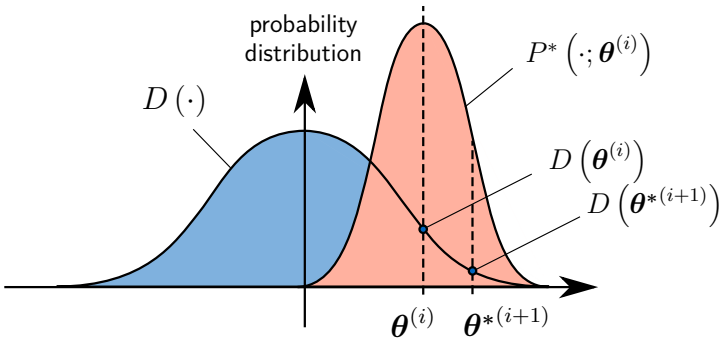


Figure 2.25: *Desired and proposal distribution for the calculation of the acceptance ratio*

than the seed are only accepted with a probability smaller than one. For large step widths, the acceptance ratio tends to zero and hence the chain does not move anymore. This is shown in figure 2.26. Larger step widths correspond to higher variances of the proposal density P^*. For

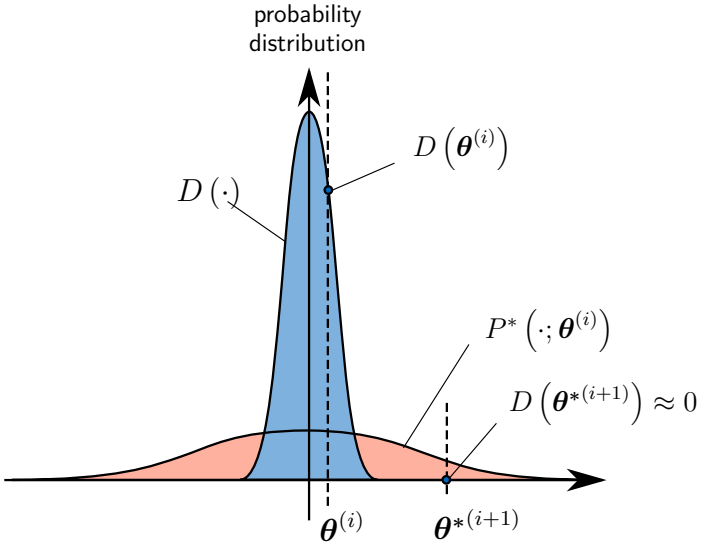


Figure 2.26: *Decreasing acceptance ratio with increasing width of proposal density P^**

large proposal widths – as shown in figure 2.26 – the probability that a candidate $\boldsymbol{\theta}^{*(i+1)}$ lies far from the desired distribution $D(\cdot)$ is high, which results in $D\left(\boldsymbol{\theta}^{*(i+1)}\right) \approx 0$ and consequently the acceptance ratio (2.45) tends to zero. A safe strategy is to choose the spread of the proposal density P^* to be of the same order as the spread of the desired distribution [AW14, p. 123].

The Metropolis algorithm generates Markov chains that fulfill the detailed balance criterion and where the equilibrium distribution is equal to the desired distribution $D(\cdot)$. This can be proven as follows: The transition probability density is denoted as $P_{\boldsymbol{\theta}^{(i+1)}|\boldsymbol{\theta}^{(i)}}(\mathbf{b}|\mathbf{a})$ and describes the probability distribution of the next uncertain parameter vector $\boldsymbol{\theta}^{(i+1)}$ given the current vector $\boldsymbol{\theta}^{(i)}$. The values $\mathbf{a}$ and $\mathbf{b}$ are exemplary realizations of the current and next uncertain parameter vector. This transition probability can be split up in two parts:

$$\begin{aligned} P_{\boldsymbol{\theta}^{(i+1)}|\boldsymbol{\theta}^{(i)}}(\mathbf{b}|\mathbf{a}) = {} & P_A(\mathbf{a}) \cdot P_{\boldsymbol{\theta}^{(i+1)}|\boldsymbol{\theta}^{(i)}}((\mathbf{b}|\mathbf{a})\,|A) \\ & + (1 - P_A(\mathbf{a})) \cdot P_{\boldsymbol{\theta}^{(i+1)}|\boldsymbol{\theta}^{(i)}}\left((\mathbf{b}|\mathbf{a})\,|\bar{A}\right) \end{aligned} \tag{2.47}$$

The first term on the right describes the case that a candidate is accepted as new state, which happens with an acceptance probability $P_A(\mathbf{a})$ while the second term stands for the case that a candidate is rejected and hence the new state is equal to the seed. The acceptance probability can be obtained by solving the integral of the product of proposal density $P^*(\cdot)$ and the acceptance ratio r:

$$P_A(\mathbf{a}) = \int P^*(\boldsymbol{\theta}; \mathbf{a}) \min\left(1, \frac{D(\boldsymbol{\theta})}{D(\mathbf{a})}\right) d\boldsymbol{\theta} \tag{2.48}$$

$P_A(\mathbf{a})$ can be interpreted as probability that a candidate $\mathbf{b}$ is accepted given that the current chain state is $\mathbf{a}$, see figure 2.27. The transition probability given that the candidate vector $\mathbf{b}$

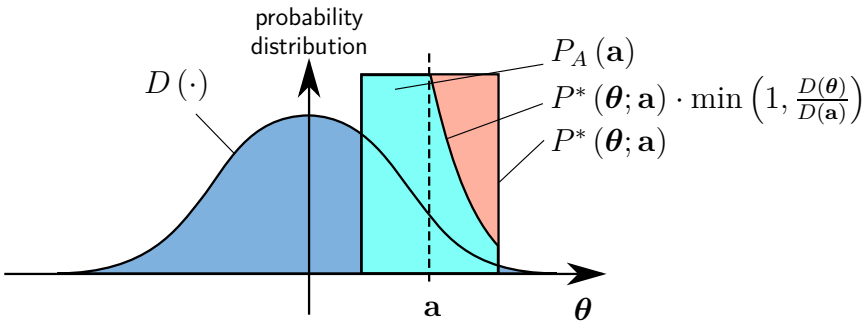


Figure 2.27: *Acceptance probability of the Metropolis algorithm*

is accepted can be written as

$$P_{\boldsymbol{\theta}^{(n+1)}|\boldsymbol{\theta}^{(n)}}((\mathbf{b}|\mathbf{a})|A) = P_A(\mathbf{a})^{-1} P^*(\mathbf{b}; \mathbf{a}) \min\left(1, \frac{D(\mathbf{b})}{D(\mathbf{a})}\right) \tag{2.49}$$

where the inverse of the acceptance probability ensures that the transition probability distribution $P_{\boldsymbol{\theta}^{(i+1)}|\boldsymbol{\theta}^{(i)}}((\cdot|\mathbf{a})|A)$ integrates to one. The transition probability given that the candidate is rejected is a Dirac delta function $\delta(x)$, which is $+\infty$ if $x = 0$ and zero otherwise:

$$P_{\boldsymbol{\theta}^{(n+1)}|\boldsymbol{\theta}^{(n)}}\left((\mathbf{b}|\mathbf{a})|\bar{A}\right) = \delta(\mathbf{a} - \mathbf{b}) \tag{2.50}$$

Inserting equations (2.49) and (2.50) in (2.47) gives the transition probability $P_{\boldsymbol{\theta}^{(i+1)}|\boldsymbol{\theta}^{(i)}}(\mathbf{b}|\mathbf{a})$.

$$P_{\boldsymbol{\theta}^{(i+1)}|\boldsymbol{\theta}^{(i)}}(\mathbf{b}|\mathbf{a}) = P^*(\mathbf{b}; \mathbf{a}) \min\left(1, \frac{D(\mathbf{b})}{D(\mathbf{a})}\right) + (1 - P_A(\mathbf{a})) \cdot \delta(\mathbf{a} - \mathbf{b}) \tag{2.51}$$

To prove that the Markov chain constructed using the Metropolis algorithm is reversible, the following condition must be fulfilled:

$$P_{\boldsymbol{\theta}^{(i+1)}|\boldsymbol{\theta}^{(i)}}(\mathbf{b}|\mathbf{a}) D(\mathbf{a}) = P_{\boldsymbol{\theta}^{(i+1)}|\boldsymbol{\theta}^{(i)}}(\mathbf{a}|\mathbf{b}) D(\mathbf{b}) \tag{2.52}$$

The proof for the rejection case, i.e. $\mathbf{a} = \mathbf{b}$ is trivial. To prove the acceptance case, equation

(2.51) is multiplied by the distribution $D(\cdot)$:

$$\begin{aligned} P_{\boldsymbol{\theta}^{(i+1)}|\boldsymbol{\theta}^{(i)}}(\mathbf{b}|\mathbf{a})\, D(\mathbf{a}) &= P^*(\mathbf{b};\mathbf{a}) \min\left(1, \frac{D(\mathbf{b})}{D(\mathbf{a})}\right) D(\mathbf{a}) \\ &= P^*(\mathbf{a};\mathbf{b}) \min\left(1, \frac{D(\mathbf{a})}{D(\mathbf{b})}\right) D(\mathbf{b}) \\ &= P_{\boldsymbol{\theta}^{(i+1)}|\boldsymbol{\theta}^{(i)}}(\mathbf{a}|\mathbf{b})\, D(\mathbf{b}) \end{aligned} \tag{2.53}$$

where for the second line symmetry of the proposal distribution $P^*(\mathbf{b};\mathbf{a}) = P^*(\mathbf{a};\mathbf{b})$ as well as the fundamental identity $\min(1, x/y) \cdot y = \min(1, y/x) \cdot x$ are used. This proves detailed balance of the Markov chain generated by the Metropolis algorithm.

In case that the candidate uncertain parameter vector is not accepted, the Metropolis algorithm requires to use the seed vector as new chain state $\boldsymbol{\theta}^{(i+1)}$, i.e. the old chain state is kept as new state. This leads to a higher correlation between samples which is unfavorable for statistical estimation. One could have the idea to modify the Metropolis algorithm in a way that new candidate uncertain parameter vectors are repeatedly generated until a candidate is accepted, which leads to the following (wrong) algorithm:

Wrongly modified Metropolis algorithm using repeated sampling

1. Generate a candidate uncertain parameter vector $\boldsymbol{\theta}^{*(i+1)}$ from a proposal density $P^*\left(\boldsymbol{\theta}^{*(i+1)};\boldsymbol{\theta}^{(i)}\right)$, where $P^*\left(\boldsymbol{\theta}^{*(i+1)};\boldsymbol{\theta}^{(i)}\right)$ is a symmetric distribution about the current chain state $\boldsymbol{\theta}^{(i)}$.

2. Calculate the acceptance ratio r with the desired distribution D:

$$r = \min\left(\frac{D\left(\boldsymbol{\theta}^{*(i+1)}\right)}{D\left(\boldsymbol{\theta}^{(i)}\right)}, 1\right) \tag{2.54}$$

3. Set the new chain state: Generate a random number p from a uniform distribution in the range from 0 to 1.

$$\boldsymbol{\theta}^{(i+1)} = \begin{cases} \boldsymbol{\theta}^{*(i+1)} \text{ if } p < r \\ \text{otherwise repeat steps 1-2 with old seed} \end{cases} \tag{2.55}$$

In this case, the transition probability density $P_{\boldsymbol{\theta}^{(i+1)}|\boldsymbol{\theta}^{(i)}}(\mathbf{b}|\mathbf{a})$ is the transition probability of the original Metropolis algorithm conditional on the case that the candidate is accepted, see (2.49).

$$P_{\boldsymbol{\theta}^{(i+1)}|\boldsymbol{\theta}^{(i)}}(\mathbf{b}|\mathbf{a}) = P_{\boldsymbol{\theta}^{(i+1)}|\boldsymbol{\theta}^{(i)}}((\mathbf{b}|\mathbf{a})\,|A) = P_A(\mathbf{a})^{-1} P^*(\mathbf{b};\mathbf{a}) \min\left(1, \frac{D(\mathbf{b})}{D(\mathbf{a})}\right) \tag{2.56}$$

Similar to the proof for the original Metropolis algorithm, multiplication by the desired distri-

bution gives

$$\begin{aligned} P_{\boldsymbol{\theta}^{(n+1)}|\boldsymbol{\theta}^{(n)}}(\mathbf{b}|\mathbf{a})\, D(\mathbf{a}) &= P_A(\mathbf{a})^{-1} P^*(\mathbf{b};\mathbf{a}) \min\left(1, \frac{D(\mathbf{b})}{D(\mathbf{a})}\right) D(\mathbf{a}) \\ &= P_A(\mathbf{a})^{-1} P^*(\mathbf{a};\mathbf{b}) \min\left(1, \frac{D(\mathbf{a})}{D(\mathbf{b})}\right) D(\mathbf{b}) \end{aligned} \tag{2.57}$$

where in the second line the same transformations are made as for equation (2.53). By comparison of (2.57) and the detailed balance condition (2.52) one can see that detailed balance is only ensured for the trivial case that $\mathbf{a} = \mathbf{b}$, i.e. $P_A(\mathbf{a}) = P_A(\mathbf{b})$. For the general case, it is not ensured that $P_A(\mathbf{a}) = P_A(\mathbf{b})$ holds and therefore the incorrect algorithm with resampling in case of rejection leads to Markov chains that do not fulfill the detailed balance condition. Hence it is not ensured that samples generated by such a wrongly modified algorithm are distributed according to the desired distribution $D(\cdot)$. However, there exist algorithms that increase the acceptance probability and which ensure detailed balance. Since they require many additional evaluations of the desired distribution, which is usually computationally expensive in the scope of failure probability estimation, such algorithms are not further discussed here. For an example of such an algorithm, see e.g. [GM01].

Metropolis-Hastings Algorithm

Hastings generalized the Metropolis algorithm to also allow for non-symmetric proposal distributions P^* [Has70]. The resulting Metropolis-Hastings algorithm is as follows:

Metropolis-Hastings algorithm for generation of samples from arbitrary distributions

1. Generate a candidate uncertain parameter vector $\boldsymbol{\theta}^{*(i+1)}$ from a proposal density $P^*\left(\cdot;\boldsymbol{\theta}^{(i)}\right)$, where $P^*\left(\cdot;\boldsymbol{\theta}^{(i)}\right)$ is an arbitrary distribution centered at the current chain state $\boldsymbol{\theta}^{(i)}$.

2. Calculate the acceptance ratio r with the desired distribution D:

$$r = \min\left(\frac{D\left(\boldsymbol{\theta}^{*(i+1)}\right)}{D\left(\boldsymbol{\theta}^{(i)}\right)} \frac{P^*\left(\boldsymbol{\theta}^{(i)};\boldsymbol{\theta}^{*(i+1)}\right)}{P^*\left(\boldsymbol{\theta}^{*(i+1)};\boldsymbol{\theta}^{(i)}\right)}, 1\right) \tag{2.58}$$

3. Set the new chain state: Generate a random number p from a uniform distribution in the range from 0 to 1.

$$\boldsymbol{\theta}^{(i+1)} = \begin{cases} \boldsymbol{\theta}^{*(i+1)} & \text{if } p < r \\ \boldsymbol{\theta}^{(i)} & \text{otherwise} \end{cases} \tag{2.59}$$

The algorithm is very similar to the original Metropolis algorithm. For symmetric proposal distributions $P^*(\cdot;\cdot)$, the algorithm becomes the original Metropolis algorithm. The only difference is that the acceptance ratio also takes the ratio between the proposal density centered

at the proposal vector and evaluated at the seed and the proposal density centered at the seed and evaluated at the proposal vector into account. The Metropolis-Hastings algorithm fulfills the detailed balance criterion which can be shown analogously to the original Metropolis algorithm. The transition probability for the acceptance case becomes

$$P_{\boldsymbol{\theta}^{(n+1)}|\boldsymbol{\theta}^{(n)}}\left(\left(\mathbf{b}|\mathbf{a}\right)|A\right) = P_A\left(\mathbf{a}\right)^{-1} P^*\left(\mathbf{b};\mathbf{a}\right) \min\left(1, \frac{D\left(\mathbf{b}\right)}{D\left(\mathbf{a}\right)} \frac{P^*\left(\mathbf{a};\mathbf{b}\right)}{P^*\left(\mathbf{b};\mathbf{a}\right)}\right) \tag{2.60}$$

with the acceptance probability

$$P_A\left(\mathbf{a}\right) = \int P^*\left(\boldsymbol{\theta};\mathbf{a}\right) \min\left(1, \frac{D\left(\boldsymbol{\theta}\right)}{D\left(\mathbf{a}\right)} \frac{P^*\left(\mathbf{a};\boldsymbol{\theta}\right)}{P^*\left(\boldsymbol{\theta};\mathbf{a}\right)}\right) d\boldsymbol{\theta} \tag{2.61}$$

Inserting equation (2.60) into (2.47) and multiplication by the desired distribution $D\left(\cdot\right)$ proves detailed balance of the Metropolis-Hastings algorithm for the non-trivial case, i.e. $\mathbf{a} \neq \mathbf{b}$.

$$\begin{aligned} P_{\boldsymbol{\theta}^{(n+1)}|\boldsymbol{\theta}^{(n)}}\left(\mathbf{b}|\mathbf{a}\right) D\left(\mathbf{a}\right) &= P^*\left(\mathbf{b};\mathbf{a}\right) \min\left(1, \frac{D\left(\mathbf{b}\right)}{D\left(\mathbf{a}\right)} \frac{P^*\left(\mathbf{a};\mathbf{b}\right)}{P^*\left(\mathbf{b};\mathbf{a}\right)}\right) D\left(\mathbf{a}\right) \\ &= P^*\left(\mathbf{b};\mathbf{a}\right) \min\left(1, \frac{D\left(\mathbf{a}\right)}{D\left(\mathbf{b}\right)} \frac{P^*\left(\mathbf{b};\mathbf{a}\right)}{P^*\left(\mathbf{a};\mathbf{b}\right)}\right) \frac{P^*\left(\mathbf{a};\mathbf{b}\right)}{P^*\left(\mathbf{b};\mathbf{a}\right)} D\left(\mathbf{b}\right) \\ &= P^*\left(\mathbf{a};\mathbf{b}\right) \min\left(1, \frac{D\left(\mathbf{a}\right)}{D\left(\mathbf{b}\right)} \frac{P^*\left(\mathbf{b};\mathbf{a}\right)}{P^*\left(\mathbf{a};\mathbf{b}\right)}\right) D\left(\mathbf{b}\right) \\ &= P_{\boldsymbol{\theta}^{(n+1)}|\boldsymbol{\theta}^{(n)}}\left(\mathbf{a}|\mathbf{b}\right) D\left(\mathbf{b}\right) \end{aligned} \tag{2.62}$$

where again the fundamental identity $\min\left(1, x/y\right) \cdot y = \min\left(1, y/x\right) \cdot x$ is used.

Figure 2.28 shows an asymmetric proposal distribution $P^*\left(\cdot;\cdot\right)$ and exemplarily the acceptance probability $P_A\left(\mathbf{a}\right)$ represented by the shaded area. Given that the proposal distribution is adequately chosen, it is possible to obtain a higher acceptance probability for a given step width. This theoretically reduces the correlation between samples and hence increases the estimation accuracy. However, the proposal distribution should follow standard distributions from which samples can be efficiently drawn. There exist no expression for the optimal proposal distribution. Furthermore this optimal distribution is likely no standard distribution from

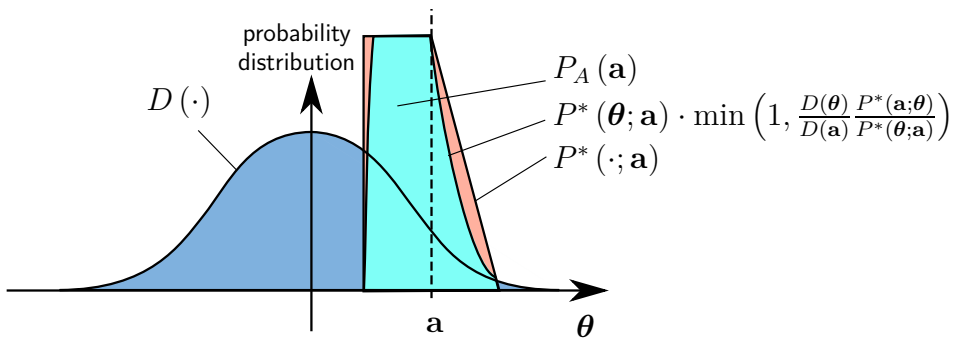


Figure 2.28: *Asymmetric proposal distribution and acceptance probability of the Metropolis-Hastings algorithm*

which samples can be efficiently generated. Therefore, mostly symmetric proposal distributions are chosen. Note that only for a uniform desired distribution $D(\cdot)$ that completely overlaps a uniform proposal distribution $P^*(\cdot;\cdot)$, an acceptance probability of $P_A(\mathbf{a}) = 1 \forall \mathbf{a}$ can be achieved. For all other desired distributions there exist no proposal distribution so that the acceptance probability (2.61) is one. For very small step sizes, i.e. narrow proposal distributions, very high acceptance ratios can be achieved, since in this case the value of the desired distribution $D(\boldsymbol{\theta})$ does not change much between the state $\boldsymbol{\theta} = \mathbf{a}$ and $\boldsymbol{\theta} = \mathbf{b}$ and hence it can be considered approximately as uniform distribution within the range of the step width of the proposal distribution. However, this would result in a very slow random walk, very high correlations between the samples and it would require very many steps to evaluate the whole desired distribution. An optimal acceptance rate of 0.234 under quite general conditions has been found for the multidimensional Metropolis algorithm [RGG97].
Unfortunately, the Metropolis as well as the Metropolis-Hastings algorithm are imperfect for higher dimensions of the uncertain parameter vector $\boldsymbol{\theta}$, i.e. for increasing numbers of uncertain parameters. This problem is related to the "curse of dimensionality". With increasing size of the uncertain parameter space the acceptance ratio decreases. Figure 2.29 exemplarily shows the decreasing ratio for a multivariate Gaussian distribution as proposal density P^*. This effect

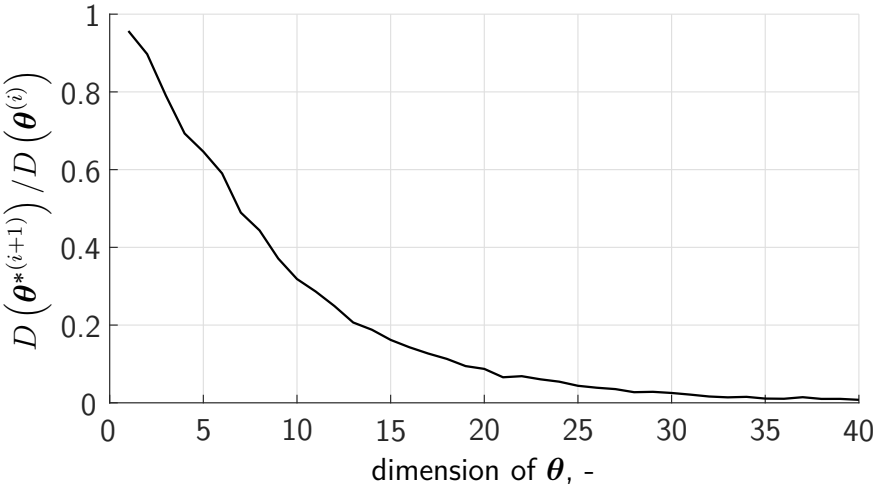


Figure 2.29: *Acceptance ratio of Metropolis algorithm vs. number of uncertain parameters; averaged over 20000 independent samples*

can be described by the increasing distance of the candidates from the seed with increasing size of the parameter space. Assume that the proposal distribution P^* is a multivariate Gaussian distribution centered at the seed with unit standard deviation. For Gaussian distributed candidates around the seed, the distance of each candidate from the seed given by the Euclidean norm of the k-dimensional vector $\boldsymbol{\theta} - \boldsymbol{\theta}^*$ exactly represents the definition of a χ-distribution with k degrees of freedom [Hol13, p. 96]. The mode, i.e. the most frequent value of such a χ-distribution, is given by $\sqrt{k-1}$. This means that for high dimensions k, the mean distance of a candidate is approximately proportional to the square root of the dimension. The variance of the χ-distribution is given by

$$\sigma^2 = k - \mu^2 \tag{2.63}$$

where the mean of this distribution is obtained by

$$\mu = \sqrt{2}\frac{\Gamma\left((k+1)/2\right)}{\Gamma\left(k/2\right)} \tag{2.64}$$

$\Gamma(z)$ denotes the Gamma function which can be calculated for z with positive real part according to

$$\Gamma(z) = \int_0^\infty x^{z-1}e^{-x}dx \tag{2.65}$$

Figure 2.30 shows the mean and according 99% confidence intervals for the Euclidean distance of the candidate $\boldsymbol{\theta}^*$ from the seed $\boldsymbol{\theta}$. Additionally, estimates of these quantities using 200 independent random samples are plotted to underline the theoretical results.

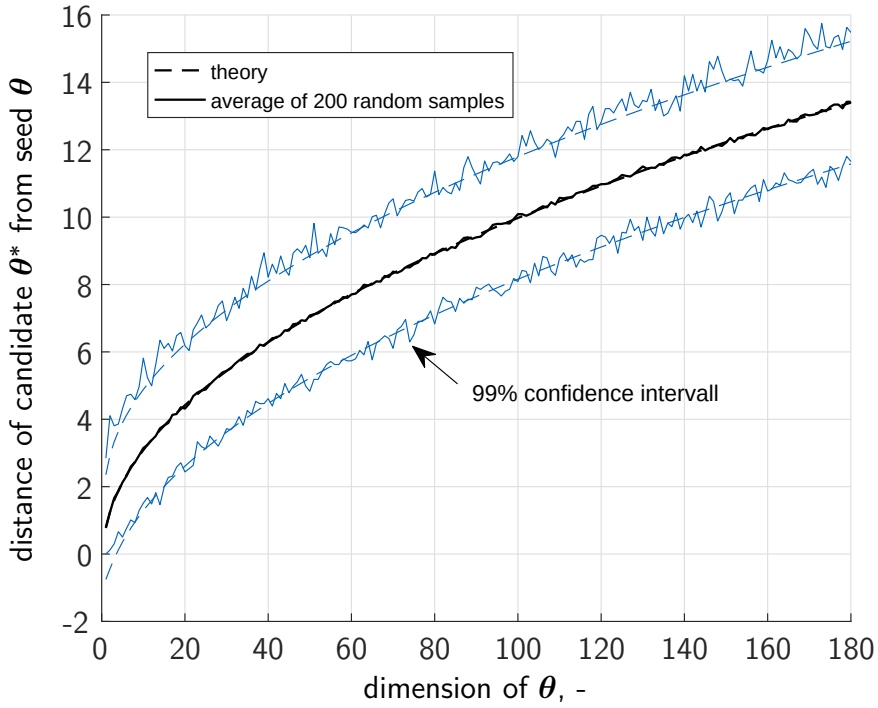


Figure 2.30: *Average distance of candidate from seed vector vs. number of uncertain parameters*

It can be seen that with increasing dimension of the uncertain parameter vector $\boldsymbol{\theta}$ also the average distance increases. Since the variance does not increase in the same order, virtually no samples are generated close to the seed for higher dimensions. With increasing distance from the seed also the acceptance ratio decreases. The increasing distance with increasing dimension could be encountered and compensated by tighter proposal densities P^*. However, this is no option for very high dimensions (>1000), since in this case the individual uncertain parameters would almost not change. This would lead to immobile Markov chains which cannot evaluate the whole parameter domain.

The convergence of the acceptance ratio to zero can also be proven theoretically ([AW14, p. 149]): Suppose that the multivariate proposal density is a product of one-dimensional

probability density functions:

$$P^*\left(\boldsymbol{\theta}^*;\boldsymbol{\theta}\right)=\prod_{j=1}^{k}P_n^*\left(\theta_j^*;\theta_j\right) \tag{2.66}$$

where the chain indices in superscript were skipped for the sake of clarity. Furthermore, the proposal density is the same for every parameter, i.e. $P_i^*=P_1^*$. The uncertain parameters are also assumed to be independent with similar distribution, leading to the joint probability density

$$D\left(\boldsymbol{\theta}\right)=\prod_{j=1}^{k}D_1\left(\theta_j\right) \tag{2.67}$$

Given that the seed $\boldsymbol{\theta}$ is distributed according to the target distribution D and the candidate $\boldsymbol{\theta}^*$ is obtained by the proposal density P^*, the acceptance ratio can be calculated by

$$r=\prod_{j=1}^{k}\frac{D_1\left(\theta_j^*\right)}{D_1\left(\theta_j\right)}\frac{P^*\left(\theta_j;\theta_j^*\right)}{P^*\left(\theta_j^*;\theta_j\right)}=\prod_{j=1}^{k}r_j \tag{2.68}$$

Note that the $\min$ expression used for the acceptance ratio in equation (2.58) is omitted here since it will be shown later that $1/r\to 0$ for $k\to\infty$ and hence the upper saturation for r is not decisive for the derivation of the convergence of r. The acceptance ratio r denotes the probability that a candidate is accepted. Hence, small r means that the candidate will probably be rejected. For the subsequent explanation, the following transformation is made:

$$\frac{1}{k}\ln\left(r\right)=\frac{1}{k}\sum_{j=1}^{k}\ln\left(r_j\right) \tag{2.69}$$

Since for this proof $P_j^*=P_1^*$ and $D_j=D_1$ are chosen and hence the r_j are independent and identically distributed, it follows from the law of large numbers that

$$\frac{1}{k}\ln\left(r\right)\to E\left[\ln\left(r_j\right)\right] \tag{2.70}$$

Jensen's inequality says that $E\left[f(r_j)\right]<f(E\left[r_j\right])$ for any strictly concave function f [Nee93]. Since the natural logarithm is a strictly concave function, an upper bound for $E\left[\ln\left(r_j\right)\right]$ results:

$$E\left[\ln\left(r_j\right)\right]<\ln\left(E\left[r_j\right]\right) \tag{2.71}$$

The expectation of r_j is obtained using equation (2.68),

$$\begin{aligned}E\left[r_j\right]&=\int\int r_j f\left(\theta_j,\theta_j^*\right)d\theta_j d\theta_j^*\\&=\int\int\frac{D_1\left(\theta_1^*\right)}{D_1\left(\theta_j\right)}\frac{P_1^*\left(\theta_j;\theta_j^*\right)}{P_1^*\left(\theta_j^*;\theta_j\right)}D_1\left(\theta_j\right)P_1^*\left(\theta_j^*;\theta_j\right)d\theta_j d\theta_j^*\end{aligned} \tag{2.72}$$

$$= \int \int D_1\left(\theta_j^*\right) P_1^*\left(\theta_j; \theta_j^*\right) d\theta_j d\theta_j^* = 1$$

since θ_j and θ_j^* are distributed according to the joint probability density function $f\left(\theta_j, \theta_j^*\right) = D\left(\theta_j\right) P_1^*\left(\theta_j^*; \theta_j\right)$ which integrates to one. From this follows that $E\left[\ln\left(r_j\right)\right] < \ln\left(1\right) = 0$, hence

$$E\left[\ln\left(r_j\right)\right] = -C \tag{2.73}$$

with the constant $C > 0$. Back transformation gives the following expression for the acceptance ratio:

$$r^{1/k} = e^{-C} < 1 \text{ for } k \to \infty \tag{2.74}$$

This implies that r is proportional to a number smaller than 1 to the power of k which converges towards zero for $k \to \infty$.

Component-Wise Metropolis-Hastings Algorithm

To overcome the curse of dimensionality, a component-wise Metropolis-Hastings algorithm can be used [AB01]. Instead of calculating the acceptance ratio r for the whole uncertain parameter vector $\boldsymbol{\theta}$, the ratio is calculated individually for every uncertain parameter θ_j and the accept/reject step is applied for each parameter instead of collectively for the whole vector. This requires independence of the uncertain parameters and hence of the components of the desired density $D\left(\mathbf{a}\right) = \prod_{j=1}^{k} D_j\left(a_j\right)$. However, this is no limitation of generality since dependent random variables can always be modeled by independent ones, see appendix A.1.2.

Component-wise Metropolis-Hastings algorithm for generation of samples from arbitrary distributions

1. Generate a candidate uncertain parameter vector $\boldsymbol{\theta}^{*(i+1)}$ from a proposal density $P^*\left(\cdot; \boldsymbol{\theta}^{(i)}\right)$, where $P^*\left(\cdot; \boldsymbol{\theta}^{(i)}\right)$ is an arbitrary distribution centered at the current chain state $\boldsymbol{\theta}^{(i)}$.

2. For each uncertain parameter $\theta_j, j = 1 \ldots k$:

   - Calculate the acceptance ratio r_j with the desired distribution D:

$$r_j = \min\left(\frac{D_j\left(\theta_j^{*(i+1)}\right)}{D_j\left(\theta_j{}^{(i)}\right)} \frac{P_j^*\left(\theta_j{}^{(i)}; \theta_j^{*(i+1)}\right)}{P_j^*\left(\theta_j^{*(i+1)}; \theta_j{}^{(i)}\right)}, 1\right) \tag{2.75}$$

   - Set the j-th element of the new chain state: Generate a random number p from a uniform distribution in the range from 0 to 1.

$$\theta_j{}^{(i+1)} = \begin{cases} \theta_j^{*(i+1)} & \text{if } p < r \\ \theta_j{}^{(i)} & \text{otherwise} \end{cases} \tag{2.76}$$

The first step is similar to the original Metropolis-Hastings algorithm. However, the component-wise acceptance/rejection in step 2 ensures with a very high probability that at least some elements θ_j^* of the candidate uncertain parameter vector $\boldsymbol{\theta}^*$ are accepted and hence no repetition of the seed occurs. This algorithm generates adequate Markov chains also in high dimensions. The component-wise Metropolis-Hastings algorithm also fulfills the detailed balance condition, which can be proven as follows: Due to independence of the uncertain parameters, the transition probability is given by

$$P_{\boldsymbol{\theta}^{(n+1)}|\boldsymbol{\theta}^{(n)}}\left(\mathbf{b}|\mathbf{a}\right) = \prod_{j=1}^{k} P_{\theta_j{}^{(n+1)}|\theta_j{}^{(n)}}\left(b_j|a_j\right) \tag{2.77}$$

Each single uncertain parameter is generated using the standard Metropolis-Hastings algorithm. Hence at least component-wise the detailed balance condition is fulfilled:

$$P_{\theta_j{}^{(n+1)}|\theta_j{}^{(n)}}\left(b_j|a_j\right) D_j\left(a_j\right) = P_{\theta_j{}^{(n+1)}|\theta_j{}^{(n)}}\left(a_j|b_j\right) D_j\left(b_j\right) \tag{2.78}$$

Multiplying equation (2.77) by the desired density $D\left(\mathbf{a}\right) = \prod_{j=1}^{k} D_j\left(a_j\right)$ gives

$$\begin{aligned} P_{\boldsymbol{\theta}^{(n+1)}|\boldsymbol{\theta}^{(n)}}\left(\mathbf{b}|\mathbf{a}\right) D\left(\mathbf{a}\right) &= \prod_{j=1}^{k} P_{\theta_j{}^{(n+1)}|\theta_j{}^{(n)}}\left(b_j|a_j\right) D_j\left(a_j\right) \\ &= \prod_{j=1}^{k} P_{\theta_j{}^{(n+1)}|\theta_j{}^{(n)}}\left(a_j|b_j\right) D_j\left(b_j\right) \\ &= P_{\boldsymbol{\theta}^{(n+1)}|\boldsymbol{\theta}^{(n)}}\left(\mathbf{a}|\mathbf{b}\right) D\left(\mathbf{b}\right) \end{aligned} \tag{2.79}$$

where equation (2.78) is used in the second line. This proves detailed balance of the whole k-dimensional Markov chain.

Generation of Conditional Failure Samples

The component-wise Metropolis-Hastings algorithm can be used to generate samples from an arbitrary high-dimensional distribution. For the case of generating conditional failure samples, i.e. samples of a known distribution conditional on the failure region F, the desired distribution $D\left(\boldsymbol{\theta}\right)$ of the uncertain parameter vector $\boldsymbol{\theta}$ is given by the original distribution of the uncertain parameter vector $Q\left(\boldsymbol{\theta}\right)$ conditional on the failure event F according to *Bayes*' theorem:

$$D\left(\boldsymbol{\theta}\right) = Q\left(\boldsymbol{\theta}|F\right) = \frac{P\left(F|\boldsymbol{\theta}\right) Q\left(\boldsymbol{\theta}\right)}{P\left(F\right)} \tag{2.80}$$

$P\left(F|\boldsymbol{\theta}\right)$ corresponds to the indicator function $I_F(\boldsymbol{\theta})$, i.e. it is 1 if $\boldsymbol{\theta}$ lies in the failure region, zero otherwise. Samples from this distribution cannot be drawn directly since the failure probability $P\left(F\right)$ is the unknown to be determined later. However, when using the Metropolis-Hastings algorithm, only the shape of the probability distribution must be known, i.e. the desired distribution $D\left(\boldsymbol{\theta}\right)$ must only be known up to a constant scaling factor, which is $P\left(F\right)$ in

this case. This can be inferred from the acceptance ratio of the Metropolis-Hastings algorithm in equation (2.58) where only the ratio of the desired distribution is required. Plugging in of the desired distribution equation (2.80) into the acceptance ratio for the Metropolis-Hastings algorithm shows that the unknown failure probability $P(F)$ cancels out:

$$r = \min\left(\frac{I_F\left(\boldsymbol{\theta}^{*(i+1)}\right) Q\left(\boldsymbol{\theta}^{*(i+1)}\right)}{I_F\left(\boldsymbol{\theta}^{(i)}\right) Q\left(\boldsymbol{\theta}^{(i)}\right)} \frac{P^*\left(\boldsymbol{\theta}^{(i)}; \boldsymbol{\theta}^{*(i+1)}\right)}{P^*\left(\boldsymbol{\theta}^{*(i+1)}; \boldsymbol{\theta}^{(i)}\right)}, 1\right) \tag{2.81}$$

The component-wise Metropolis-Hastings algorithm cannot be directly applied, since the indicator function is only defined for the whole uncertain parameter vector, while the modified Metropolis-Hastings algorithm would require a component-wise evaluation. However, since the value of the indicator function is either 0 or 1, equation (2.81) can be reformulated to

$$r = \min\left(\frac{I_F\left(\boldsymbol{\theta}^{*(i+1)}\right)}{I_F\left(\boldsymbol{\theta}^{(i)}\right)}, 1\right) \min\left(\frac{Q\left(\boldsymbol{\theta}^{*(i+1)}\right)}{Q\left(\boldsymbol{\theta}^{(i)}\right)} \frac{P^*\left(\boldsymbol{\theta}^{(i)}; \boldsymbol{\theta}^{*(i+1)}\right)}{P^*\left(\boldsymbol{\theta}^{*(i+1)}; \boldsymbol{\theta}^{(i)}\right)}, 1\right) \tag{2.82}$$

This equation can be further simplified: Since the first term has an upper bound of one, the result of this term is always similar to $I_F\left(\boldsymbol{\theta}^{*(i+1)}\right)$ no matter if the seed $\boldsymbol{\theta}^{(i)}$ lies in the failure region or not. Anyway, for the application of the component-wise Metropolis-Hastings algorithm for generation of samples distributed according to (2.80) the seed must lie in the failure domain and hence be a failure sample itself since otherwise the newly generated sample would not be a failure sample in the rejection case. This finally leads to the modified Metropolis-Hastings algorithm for generation of conditional failure samples:

Component-wise Metropolis-Hastings algorithm for generation of conditional failure samples

1. Generate a candidate uncertain parameter vector $\boldsymbol{\theta}^{*(i+1)}$ from a proposal density $P_j^*\left(\cdot; \boldsymbol{\theta}^{(i)}\right)$, where $P_j^*\left(\cdot; \boldsymbol{\theta}^{(i)}\right)$ is an arbitrary distribution centered at the current chain state $\boldsymbol{\theta}^{(i)}$.

2. For each uncertain parameter $\theta_j, j = 1 \ldots k$:

   - Calculate the acceptance ratio r_j with the distribution of the j-th uncertain parameter Q_j:

$$r_j = \min\left(\frac{Q_j\left(\theta_j^{*(i+1)}\right)}{Q_j\left(\theta_j{}^{(i)}\right)} \frac{P_j^*\left(\theta_j{}^{(i)}; \theta_j^{*(i+1)}\right)}{P_j^*\left(\theta_j^{*(i+1)}; \theta_j{}^{(i)}\right)}, 1\right) \tag{2.83}$$

- Correct the j-th element of the candidate uncertain parameter vector: Generate a random number p from a uniform distribution in the range from 0 to 1.

$$\theta_j^{*(i+1)} = \begin{cases} \theta_j^{*(i+1)} & \text{if } p < r_j \\ \theta_j{}^{(i)} & \text{otherwise} \end{cases} \tag{2.84}$$

3. Accept the updated candidate vector $\boldsymbol{\theta}^{*(i+1)}$ if it lies in the failure region, otherwise use seed, i.e.

$$\boldsymbol{\theta}^{(i+1)} = \begin{cases} \boldsymbol{\theta}^{*(i+1)} & \text{if } I_F\left(\boldsymbol{\theta}^{*(i+1)}\right) = 1 \\ \boldsymbol{\theta}^{(i)} & \text{otherwise} \end{cases} \tag{2.85}$$

Usually, either a uniform or Gaussian distribution is chosen for the proposal distribution. Still, the width of this distribution must be adequately selected. Since a-priori knowledge about the optimal scaling is usually not available, the width can only be adjusted during evaluation based on already obtained samples. *Zuev* et al. found an overall acceptance ratio between 30% and 50% as optimal range to minimize the variance of the estimated probability [Zue+12]. This enables the following adaption rule for the proposal distribution width: If the acceptance ratio lies above 50%, the width of the proposal distribution is increased, while in the case for acceptance ratios below 30%, the width of the proposal distribution is decreased. This tuning task increases the efforts required for parallelization of the modified Metropolis-Hastings algorithm. *Pellissetti* suggested chain-wise parallelization as well as speculative computing [Pel09], where in many cases a linear speedup growth with number of processors can be achieved.

Infinity Sampling

Infinity sampling is a novel advancement for generation of conditional failure samples recently developed independently by different researchers [Pap+15, PA15, AP16]. The idea is based on the observation that the performance of the component-wise Metropolis-Hastings algorithm is independent of the number of uncertain parameters, where numerical experience even reveals better performance with higher number of uncertainties. This is mainly caused by a higher variation of $\boldsymbol{\theta}$ for higher number of uncertain parameters [AB01].

To artificially increase the number of uncertain parameters, each parameter θ_j is theoretically described by a combination of k' independent uncertain parameters $\theta_{jh}, h = 1 \dots k'$. These additional parameters are referred to as hidden parameters since they are not directly visible in $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots]$. For that, assume that the uncertain parameters θ_j are independent from each other and the distribution $Q_j(\cdot)$ is standard Gaussian, i.e. Gaussian distributed with zero mean and unit variance. This is no limitation of generality since correlated random variables with arbitrary distributions can usually be constructed from independent standard Gaussian variables, see A.1. Under these assumptions, θ_j can be described by a linear combination of

k' standard Gaussian variables θ_{jh}:

$$\theta_j = \frac{1}{\sqrt{k'}} \sum_{h=1}^{k'} \theta_{jh} \tag{2.86}$$

where the scaling by the square root k' ensures unit variance. Inserting this extension into the modified Metropolis-Hastings algorithm results in the following theoretical algorithm:

Component-wise Metropolis-Hastings algorithm with hidden parameters for generation of conditional failure samples

1. For each uncertain parameter $\theta_j, j = 1 \dots k$:
   For each hidden variable $\theta_{jh}, h = 1 \dots k'$:

   - Calculate a candidate uncertain parameter θ_{jh} from a Gaussian proposal density $P^*\left({\theta^*_{jh}}^{(i+1)}; {\theta_{jh}}^{(i)}\right)$, which is centered at ${\theta_{jh}}^{(i)}$ with unit variance.
   - Calculate the acceptance ratio r_{jh}:

$$r_{jh} = \min\left(\frac{Q_j\left({\theta^*_{jh}}^{(i+1)}\right)}{Q_j\left({\theta_{jh}}^{(i)}\right)} \frac{P^*_j\left({\theta_{jh}}^{(i)}; {\theta^*_{jh}}^{(i+1)}\right)}{P^*_j\left({\theta^*_{jh}}^{(i+1)}; {\theta_{jh}}^{(i)}\right)}, 1\right) \tag{2.87}$$

   - Generate a random number p from a uniform distribution in the range from 0 to 1.

$${\theta^*_{jh}}^{(i+1)} = \begin{cases} {\theta^*_{jh}}^{(i+1)} & \text{if } p < r_{jh} \\ {\theta_{jh}}^{(i)} & \text{otherwise} \end{cases} \tag{2.88}$$

   Set the j-th uncertain parameter θ_j:

$$\theta_j^{*(i+1)} = \frac{1}{\sqrt{k'}} \sum_{h=1}^{k'} {\theta^*_{jh}}^{(i+1)} \tag{2.89}$$

2. Accept the updated candidate vector $\boldsymbol{\theta}^{*(i+1)}$ if it lies in the failure region, otherwise use seed, i.e.

$$\boldsymbol{\theta}^{(i+1)} = \begin{cases} \boldsymbol{\theta}^{*(i+1)} & \text{if } I_F\left(\boldsymbol{\theta}^{*(i+1)}\right) = 1 \\ \boldsymbol{\theta}^{(i)} & \text{otherwise} \end{cases} \tag{2.90}$$

In [AP16] it is shown that for $k' \to \infty$, the transition probability of step 1 becomes

$$P_{\theta_j^{*(i+1)}|{\theta_j}^{(i)}}\left(b_j|a_j\right) = \frac{1}{\sqrt{2\pi}\sigma_j} \exp\left[-\frac{1}{2\sigma_j^2}\left(b_j - c_i a_j\right)^2\right] \tag{2.91}$$

i.e. it is normally distributed with variance σ_j^2 and mean $c_j a_j$. Note that this transition probability does not directly depend on the hidden uncertain parameters θ_{jh}. The overall

algorithm must still fulfill the detailed balance criterion (2.43). Since step 2 ensures that all samples lie in the failure region, it suffices to only check detailed balance for failure samples, i.e. where the desired distribution $D(\boldsymbol{\theta}) = Q(\boldsymbol{\theta}|F) = Q(\boldsymbol{\theta})$ with $\boldsymbol{\theta} \in F$. Furthermore, since the individual components D_i are independent and identically distributed, it suffices to evaluate detailed balance only in one dimension. It must be proven that

$$P_{\theta_j{}^{(n+1)}|\theta_j{}^{(n)}}(b_j|a_j)\, D_j(a_j) \overset{!}{=} P_{\theta_j{}^{(n+1)}|\theta_j{}^{(n)}}(a_j|b_j)\, D_j(b_j) \tag{2.92}$$

Since only failure samples are considered, the transition probability of the first step (2.91) is equal to the transition probability of the overall algorithm $P_{\theta_j{}^{(n+1)}|\theta_j{}^{(n)}}(b_j|a_j)$. All uncertain parameters θ_j are standard Gaussian distributed, the desired distribution $D(\theta_j)$ is a Gaussian distribution with zero mean and unit variance $\phi(\cdot, 0, 1)$:

$$D_{a_j} = \phi(a_j, 0, 1) = \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{1}{2}a_j^2\right] \tag{2.93}$$

Inserting equations (2.91) and (2.93) into (2.92) gives the following expression:

$$\begin{aligned}\frac{1}{2\pi\sigma_j} \exp\left[-\frac{1}{2\sigma_j^2}(b_j - c_i a_j)^2\right] \exp\left[-\frac{1}{2}a_j^2\right] &\overset{!}{=} \frac{1}{2\pi\sigma_j} \exp\left[-\frac{1}{2\sigma_j^2}(a_j - c_j b_j)^2\right] \exp\left[-\frac{1}{2}b_j^2\right] \\ -\frac{1}{2\sigma_j^2}\left(b_j^2 - 2a_j b_j c_j + c_i^2 a_i^2\right) - \frac{1}{2}a_j^2 &\overset{!}{=} -\frac{1}{2\sigma_j^2}\left(a_j^2 - 2a_j b_j c_j + c_j^2 b_j^2\right) - \frac{1}{2}b_j^2\end{aligned} \tag{2.94}$$

This equation is true for $c_j = \sqrt{1 - \sigma_j^2}$. This results in the Infinity Sampling algorithm, where "infinity" refers to the concept that each uncertain parameter is theoretically modeled by an infinite number of hidden uncertain parameters [AP16]:

Infinity Sampling algorithm for generation of conditional failure samples

1. Generate a candidate uncertain parameter vector $\boldsymbol{\theta}^{*(i+1)}$ as a Gaussian vector with independent components, mean vector $\left[c_1\theta_1^{(i)}, \ldots, c_k\theta_k^{(i)}\right]$ and variances $[\sigma_1^2, \ldots, \sigma_k^2]$, where $c_j = \sqrt{1 - \sigma_j^2}$ and $\sigma_j^2 \leq 1, j = 1 \ldots k$.
2. Accept the updated candidate vector $\boldsymbol{\theta}^{*(i+1)}$ if it lies in the failure region, otherwise use seed, i.e.

$$\boldsymbol{\theta}^{(i+1)} = \begin{cases} \boldsymbol{\theta}^{*(i+1)} & \text{if } I_F\left(\boldsymbol{\theta}^{*(i+1)}\right) = 1 \\ \boldsymbol{\theta}^{(i)} & \text{otherwise} \end{cases} \tag{2.95}$$

Clearly, this algorithm is simpler than the modified Metropolis-Hasting algorithm since now only one tuning parameter exists which is the variance of the candidate samples. $\sigma_j^2 \leq 1$ results in $0 \leq a_j \leq 1$. The larger the variance σ_j^2, the closer the proposal mean is to the origin. The smaller the variance, the closer the proposal mean is to the seed $\theta_j^{(i)}$. Figure 2.31

shows the candidate distributions for a one-dimensional example for different variance values σ^2, where the seed lies at $\theta^{(i)} = 1$, indicated by the black dotted line. The black line shows the distribution of the uncertain parameter, however not conditional on the failure domain, since the failure criteria is only evaluated in the second step. The blue line gives the resulting candidate distributions. In the upper left plot, a variance of $\sigma^2 = 1$ leads to $c = 0$, i.e.

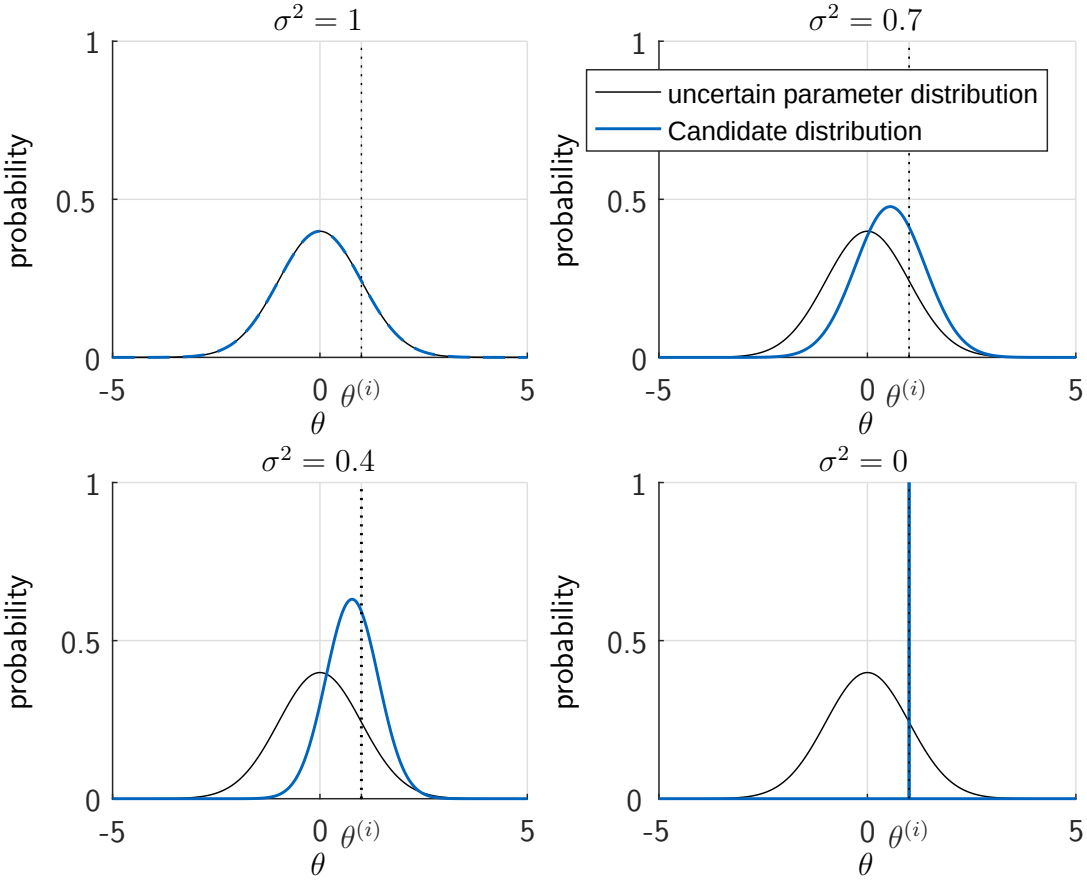


Figure 2.31: *Probability distribution of proposal $\theta^{(i+1)}$ of infinity sampling for different proposal variances σ^2; seed at $\theta^{(i)} = 1$*

the candidate distribution has zero mean with unit variance and hence is equal to the actual parameter distribution. In this case, the Infinity Sampling algorithm becomes equal to the generation of failure samples by conventional Monte Carlo simulation, also known as rejection method [Dev86, p. 40]. However, this choice of σ is very unfavorable since the acceptance ratio would be very small for small failure probabilities. The other extreme is shown in the lower right plot with $\sigma = 0$ and therefore $c = 1$. This means that the new sample is exactly the same as the seed, since it is generated from a Gaussian distribution with the mean being equal to the seed and zero standard deviation. This case is similar to the modified Metropolis-Hastings algorithm with zero proposal width. The Markov chain would be immobile, since the step width of the random walk would be zero. Eventually, the remaining plots show the cases with standard deviations between zero and one where it can be seen that the lower the variation of the candidate, the closer it is to the seed. Numerical experience suggests that a variance $0.4 < \sigma_j^2 < 0.6$ provides best results for generation of conditional failure samples

[PA15].
In contrast to the modified Metropolis-Hastings algorithms, Infinity Sampling does not require any dynamic adjustment of tuning parameters based on already generated samples. Although additional effort is required to transform uncertainties into standard Gaussian form, this still makes the Infinity Sampling algorithm the first choice for generation of conditional failure samples, especially if scalability on parallel machines is desired [PA15].

2.6.3 Subset Simulation Algorithm

In the previous section, methods are described that can be used for generation of conditional failure samples. Applying this to the idea of Subset simulation described in section 2.6.1, leads to the following Subset simulation algorithm. Details on the individual steps are given below the algorithm.

Subset simulation algorithm

1. Subset level $j = 0$: Pure Monte Carlo
   (a) Generate N samples $\boldsymbol{\theta}_{(0)}^{(i)}, i = 1 \dots N$ distributed according to $Q(\boldsymbol{\theta})$
   (b) Calculate corresponding values of response variable $r\left(\boldsymbol{\theta}_{(0)}^{(i)}\right)$
   (c) Determine the first conditional failure threshold $r_{(1),limit}$ as the $P_0 N$-th largest value of $r\left(\boldsymbol{\theta}_{(0)}^{(i)}\right)$
   (d) Determine number of samples in final failure domain N_F, i.e. count of samples with $r\left(\boldsymbol{\theta}_{(0)}^{(i)}\right) > r_{limit}$
2. While $N_F < P_0 N$, i.e. as long as there are not enough samples in the final failure domain
   (a) Increase subset level: $j = j + 1$
   (b) Generate N conditional samples $\boldsymbol{\theta}_{(j)}^{(i)}, i = 1 \dots N$ distributed according to $Q(\boldsymbol{\theta} \mid F_j)$ where $F_j = \{r | r > r_{(j),limit}\}$
   (c) Calculate corresponding values of response variable $r\left(\boldsymbol{\theta}_{(j)}^{(i)}\right)$
   (d) Determine the $(j+1)$-th conditional failure threshold $r_{(j+1),limit}$ as the $P_0 N$-th largest value of $r\left(\boldsymbol{\theta}_{(j)}^{(i)}\right)$
   (e) Determine number of samples in final failure domain N_F, i.e. count of samples with $r > r_{limit}$
3. Calculate failure probability:

$$P_F = P_0^j \frac{N_F}{N} \tag{2.96}$$

The first step is similar to pure Monte Carlo simulation, i.e. N samples for the uncertain param-

eter vector are generated according to a known distribution $Q(\boldsymbol{\theta})$, followed by the evaluation of the response variable $r\left(\boldsymbol{\theta}_{(0)}^{(i)}\right)$ using the generated uncertain parameter vectors. While the subscript denotes the subset level j, which is zero here, the upper index denotes the sample $i = 1 \dots N$. The intermediate failure thresholds are not a priori known. Instead, they are adaptively determined for a desired conditional failure probability $P_{F_{(1)}} = P\left(F_1 \,|F_0\right) = P_0$. In step 1.(c), the first conditional failure threshold $r_{(1),limit}$ is calculated. First, the values of the response variables $r\left(\boldsymbol{\theta}_{(1)}^{(i)}\right), i = 1 \dots N$ are sorted in descending order. To obtain a conditional failure probability $P_{F_{(1)}} = P_0$, the threshold is selected as the mean of the $P_0 N$-th and the $(P_0 N + 1)$-th largest value of the response variable, see also figure 2.32 a). Hence, the $P_0 N$-th largest sample lies in the failure domain while the sample with the $(P_0 N + 1)$-th largest response value does not, i.e. there are $N_C = P_0 N$ samples in the newly defined conditional failure domain $F_1 = \{r | r > r_{(1),limit}\}$. This results in a conditional failure probability of $P_{F_{(1)}} = P_0 N / N = P_0$ which fulfills the required condition. Note that P_0 and N must be selected so that the product $N_C = P_0 N$ is integer. Finally for the first step, the number of samples N_F in the actual failure domain $F_1 = \{r | r > r_{limit}\}$ is determined.

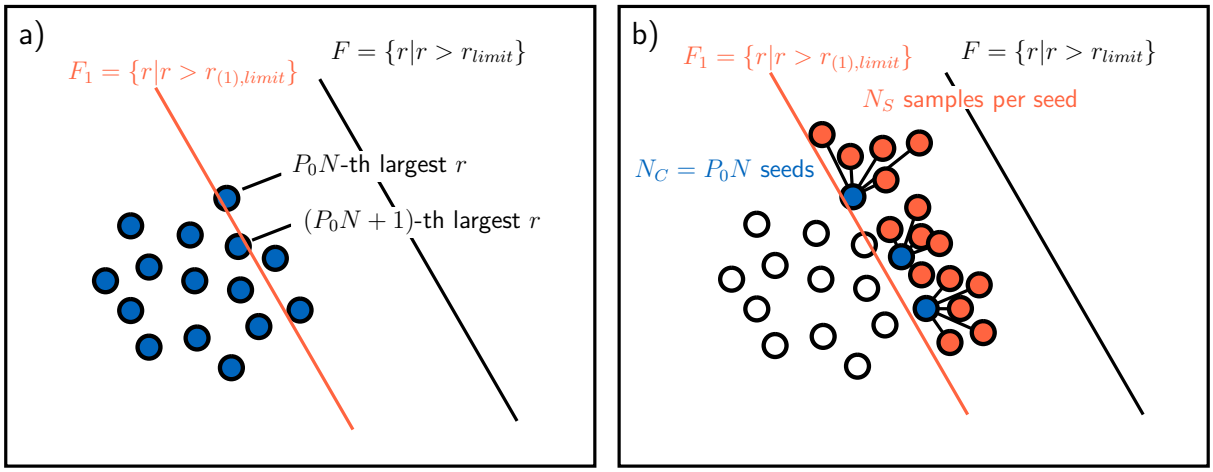


Figure 2.32: *Determination of the conditional failure domain and generation of new samples*

In step 2, conditional failure samples are generated, i.e. samples that lie in the conditional failure domain $F_j = \{r | r > r_{(j),limit}\}$. For that, each of the $N_C = P_0 N$ conditional failure samples from the previous level is used as seed for generation of $N_S = 1/P_0$ samples using Markov chains, see figure 2.32 b). N_C stands for the number of chains per subset level, while N_S is the number of samples per chain. For example, a conditional failure probability $P_0 = 0.2$ and a total number of samples $N = 1000$ results in $N_C = P_0 N = 200$ chains with $N_S = N/N_C = 1/P_0 = 5$ samples per chain. Although this selection restricts the possible conditional probabilities P_0 to those for which $1/P_0$ is integer, this choice ensures that the number of samples $N = N_C N_S$ remains constant with increasing subset level. For each seed, a Markov chain is constructed using either the modified Metropolis-Hastings algorithm or Infinity Sampling (see section 2.6.2). The indicator function $I_F\left(\boldsymbol{\theta}^{*(i+1)}\right)$ required for the

construction of the Markov chain for the j-th subset level is defined by

$$I_F\left(\boldsymbol{\theta}^{*(i+1)}\right) = \begin{cases} 1 & if \quad r\left(\boldsymbol{\theta}^{*(i+1)}\right) > r_{(j),limit} \\ 0 & else \end{cases} \tag{2.97}$$

Note that the samples used as seeds are by construction distributed according to the stationary distribution $Q(\boldsymbol{\theta} \mid F_j)$ and hence no burn in of the Markov chain is required and samples generated from these seeds are distributed according to the target distribution. Furthermore, for reasonable number of samples N, ergodicity is almost sure as shown in the previous section. Steps 2.(c) - 2.(e) are similar to 1.(b)-1.(d). Step 2 is repeated until there are sufficient samples in the actual failure domain F. Common criterion for sufficiency is that the number of samples in the failure domain is larger than $P_0 N$. This is similar to the condition that the last conditional failure threshold $r_{k,limit}$ is higher than the actual failure threshold r_{limit}.

Finally, in step 3 the failure probability $P_F = P(r > r_{limit})$ is calculated. According to construction, the failure probability is defined as the product of the conditional failure probabilities:

$$P_F\left(\boldsymbol{\theta}\right) = \prod_{j=1}^{m} P(F_j | F_{j-1}) \tag{2.98}$$

For all subset levels except of the last, the conditional failure probability is given by $P_{F(j)} = P\left(F_j \,|F_{j-1}\right) = P_0$. For the last subset m, the conditional failure probability is given by $P\left(F_m \,|F_{m-1}\right) = N_F/N$ where N_F is the number of samples from the last subset that lie in the failure region. Substituting this into equation (2.98) leads to the failure probability for the described Subset algorithm, which is given by (2.96).

For illustration of the Subset simulation algorithm, the same example is used as in section 2.5.3, where a problem is described with two uncertain parameters θ_1 and θ_2. Both follow a standard Gaussian distribution, i.e. a normal distribution with zero mean and unit variance. The failure domain is described by hyperbolas that fulfill the following equation:

$$\theta_1^2 - \theta_2^2 > 3^2 \tag{2.99}$$

A reasonable choice for the response variable for this example is $r = \theta_1^2 - \theta_2^2$, which results in $r_{limit} = 3^2 = 9$. Figure 2.33 shows in detail the individual steps of the Subset simulation algorithm applied to this example.

First, samples are distributed according to $Q(\boldsymbol{\theta})$ which in this case is a two-dimensional Gaussian distribution with zero mean and unit variance. In subfigure a), the failure region is indicated by black solid lines. The first conditional failure region is calculated according to step 1.c) of the Subset algorithm. This is shown in subfigure b). The samples, which lie in the first subset, are the seeds for generating samples distributed according to $Q(\boldsymbol{\theta} \,|F_1)$, which are depicted in subfigure c). There are already a few samples in the failure region, however, not sufficient to fulfill the termination criteria, less than P_0 percent of all samples of this subset are failure samples with respect to the final failure region F. Hence, step 2 of the Subset algorithm

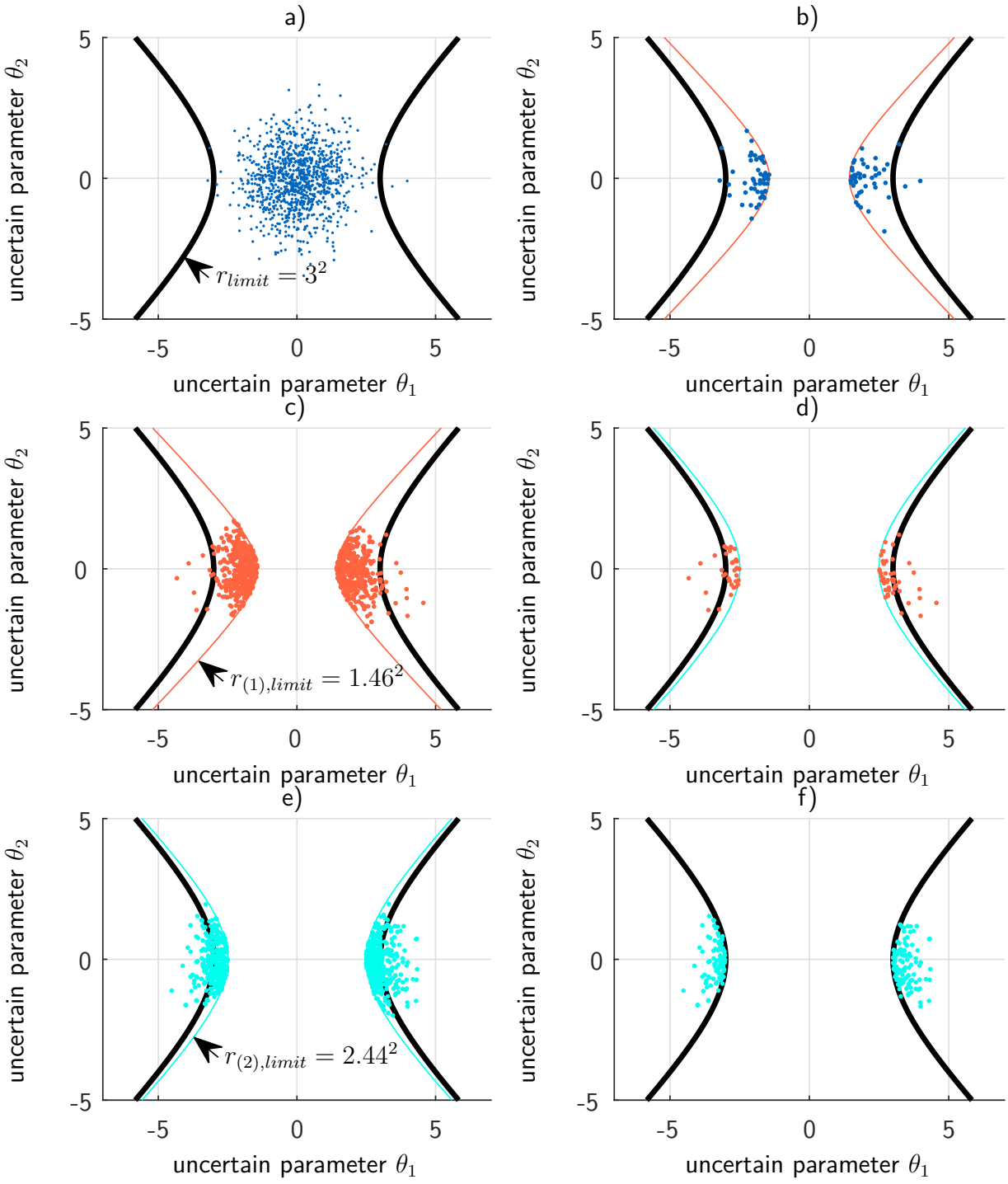


Figure 2.33: *Subset simulation example: Different steps; Algorithm: Subset simulation with* $P_0 = 0.1, N = 4000$ *using Infinity sampling with* $\sigma^2 = 0.4$

is repeated. The resulting conditional failure domain and samples are shown in subfigures d) and e). Finally, there are sufficient samples in the failure region, which is shown in f). Figure 2.34 shows all samples together with the conditional failure bounds. Comparing this with the results for the same example using Importance Sampling shown in figure 2.11, it can be seen that using Subset simulation, a lot of samples in and close to the failure region are generated

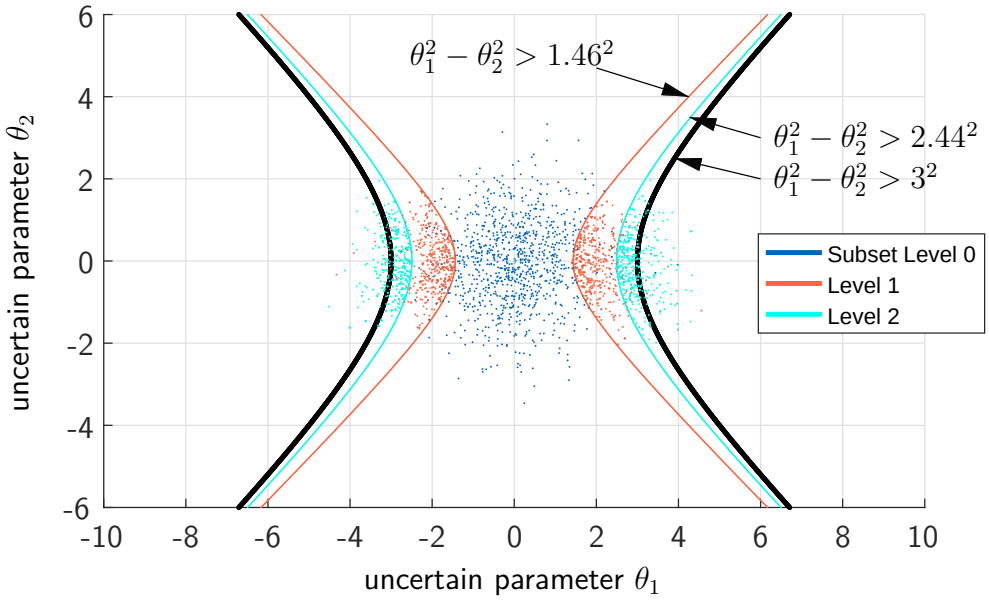


Figure 2.34: *Samples and intermediate failure thresholds for Subset simulation*

similar to Importance Sampling, however without the need of applying preknowledge about the failure region.

Figure 2.35 shows the histogram of the values of the response variable for different subsets. The black dashed lines indicate the conditional failure thresholds. It can be seen that no samples are generated that have a response value lower than the respective conditional threshold. Comparing the histograms of individual subsets, it can be seen that the distributions of the samples for one subset can be interpreted as more frequent sampling of the distribution tail of the previous subset, i.e. the less probable regions are sampled more accurately, which is exactly what is required for analysis of unlikely events. Finally, figure 2.36 shows the resulting probability distributions of the two uncertain parameters for the different subsets. While the distribution of the second parameter θ_2 shown in the lower plot does not change much with increasing subset level, the distribution of the first parameter θ_1 shown in the upper plot

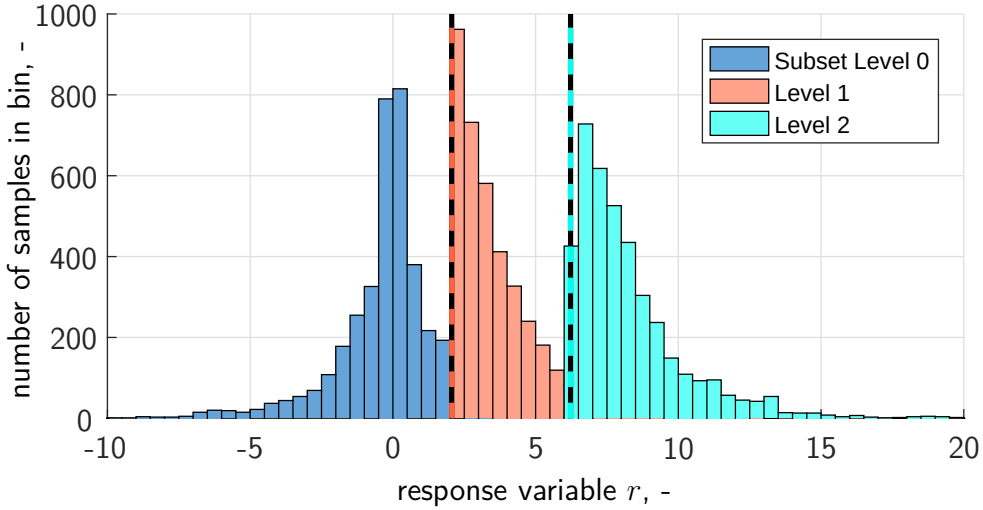


Figure 2.35: *Samples and intermediate failure thresholds for Subset simulation*

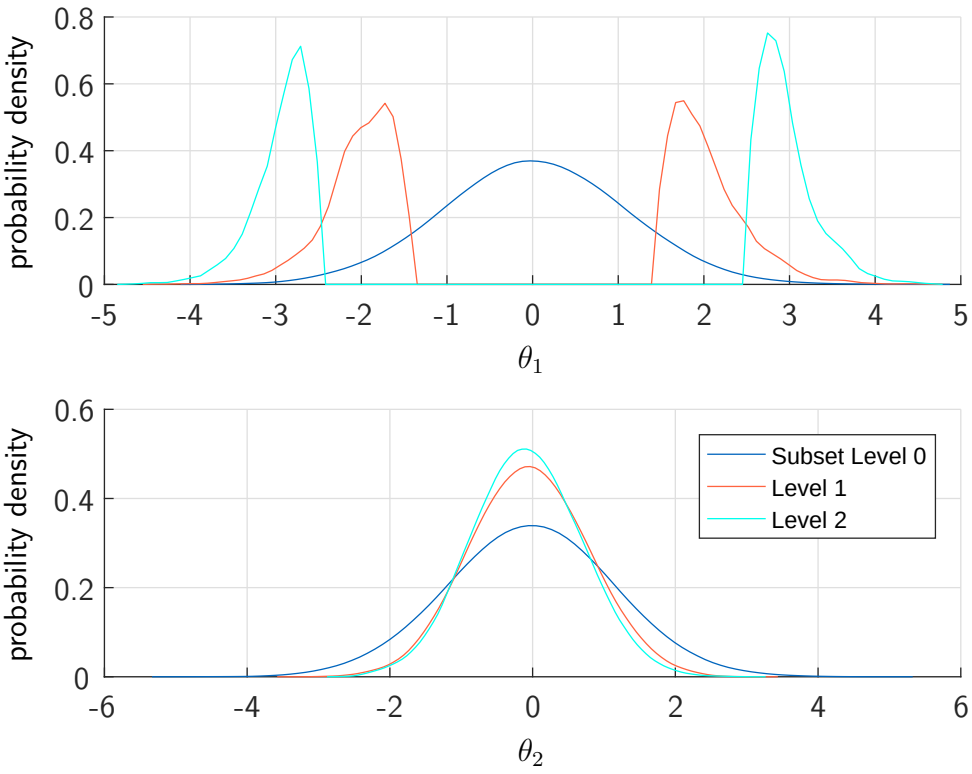


Figure 2.36: *Probability distributions of sample uncertain parameters*

changes significantly. From equation (2.99) it follows that θ_1 must always be larger than

$$\theta_1^2 > r_{(j),limit} + \theta_2^2 > r_{(j),limit} \rightarrow |\theta_1| > \left|\sqrt{r_{(j),limit}}\right| \tag{2.100}$$

For the given example, the first and second conditional failure threshold are given by $r_{(1),limit} = 1.46^2$ and $r_{(2),limit} = 2.44^2$ which matches well the distribution of θ_1 with increasing subset level.

The selection of the desired conditional failure probability P_0 is a trade-off between sample correlation and number of subsets. While smaller P_0 enables quicker evaluation of small failure probabilities, the correlation between the samples increases since for each subset a higher number of $1/P_0$ samples are generated from one sample. Much research has been done on the optimal selection of the conditional failure probability P_0. *Au* suggests P_0 to be 0.1 for good efficiency [AB01]. Zuev et al. made an exhaustive analysis of Subset simulation and figured out that choosing $0.1 \leq P_0 \leq 0.3$ practically leads to similar efficiency and hence fine tuning of the conditional probability for performance enhancement is not necessary [Zue+12]. To prove that Subset simulation is superior to conventional Monte Carlo simulation for small failure probabilities, performance of the Subset algorithm is evaluated in the next section.

2.6.4 Variance Estimation

Subsequent explanations are valid for both introduced algorithms for generation of conditional failure samples using Markov chains, i.e. the Metropolis-Hastings algorithm as well as Infinity Sampling. For the zeroth subset level, samples are independent and identically distributed equal to Monte Carlo simulation, hence also the variance is similar (see chapter 2.3.2):

$$\operatorname{var}\left[\hat{P}_{F(0)}\right] = \frac{1}{N^2} \cdot \sum_{k=1}^{N} \operatorname{var}\left[I_{F(0)}\left(\boldsymbol{\theta}_k\right)\right] = \frac{P_{F(0)}\left(1 - P_{F(0)}\right)}{N} \tag{2.101}$$

For the other subset levels, the generated samples are no longer statistically independent. The variance of these samples is derived based on the variance of correlated samples of a single Markov chain. The variance of the $k = 1 \dots N_S$ samples of the j-th chain in the i-th subset is obtained by [Gey05, p. 8ff]

$$\begin{aligned} \operatorname{var}\left[\sum_{k=1}^{N_S} I_{F(i),jk}\right] &= \sum_{k=1}^{N_S} \sum_{m=1}^{N_S} \operatorname{cov}\left[I_{F(i),jk}, I_{F(i),jm}\right] \\ &= \sum_{k=1}^{N_S} \operatorname{var}\left[I_{F(i),jk}\right] + 2 \sum_{k=1}^{N_S-1} \sum_{m=k+1}^{N_S} \operatorname{cov}\left[I_{F(i),jk}, I_{F(i),jm}\right] \end{aligned} \tag{2.102}$$

where $I_{F(i),jk} = I_{(i)}\left(\boldsymbol{\theta}_{jk}\right)$ denotes the indicator function of the i-th subset level, evaluated for the k-th element of the j-th chain, N_S the number of samples per chain and $\operatorname{cov}\left[\cdot,\cdot\right]$ the covariance of two arguments. The second line results from the symmetry of the covariance matrix. Since Markov chains used for Subset simulation are already in a stationary state, the equation can be simplified:

$$\operatorname{var}\left[\sum_{k=1}^{N_S} I_{F(i),jk}\right] = N_S \operatorname{var}\left[I_{F(i),jn}\right] + 2 \sum_{k=1}^{N_S-1} \left(N_S - k\right) \operatorname{cov}\left[I_{F(i),jn}, I_{F(i),j,n+k}\right] \tag{2.103}$$

where the right-hand side does not change with the chain index n due to stationarity, i.e. n is an arbitrary element of the considered Markov chain. A visual representation of the right side of equation (2.103) is given in figure 2.37. Since the covariance matrix is symmetric, only the upper triangular matrix is considered. The pairs with similar color combination have the same variance since the Markov chain is stationary. The multiplier $(N_S - k)$ emerges from the finite length of the considered Markov chain. In the shown four dimensional example, the sum of all different covariance combinations results in three times the covariance with lag $k = 1$ (orange) plus two times with $k = 2$ (cyan) and only one time with $k = 3$ (red), where lag k means the distance along the chain. The left side of the equation can be further refined using the estimated failure probability of the j-th chain in the i-th subset level, i.e.

$$\hat{P}_{F(i),j} = \frac{1}{N_S} \sum_{k=1}^{N_S} I_{F(i),jk} \tag{2.104}$$

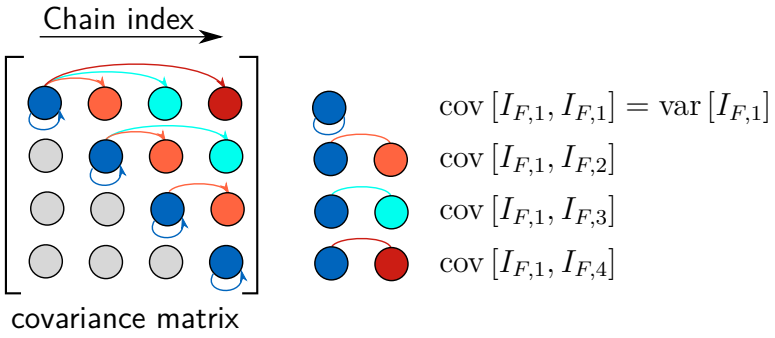


Figure 2.37: *Covariance matrix of stationary Markov chain, Subset and chain index skipped for the sake of clarity*

and using the following identity for the variance

$$\begin{aligned} \mathrm{var}\left[N_S \hat{P}_{F(i),j}\right] &= E\left[\left(N_S \hat{P}_{F(i),j} - N_S P_{(i),j}\right)^2\right] \\ &= N_S^2 E\left[\left(\hat{P}_{F(i),j} - P_{(i),j}\right)^2\right] \\ &= N_S^2 \mathrm{var}\left[\hat{P}_{F(i),j}\right] \end{aligned} \tag{2.105}$$

Inserting equation (2.105) in (2.103) and division by N_S^2 leads to

$$\mathrm{var}\left[\hat{P}_{F(i),j}\right] = \frac{1}{N_S}\left(\mathrm{var}\left(I_{F(i),jn}\right) + 2\sum_{k=1}^{N_S-1}\left(1 - \frac{k}{N_S}\right)\mathrm{cov}\left[I_{F(i),jn}, I_{F(i),j,n+k}\right]\right) \tag{2.106}$$

For simplified notation, the autocovariance with lag k for the stationary Markov chain is defined by $\gamma_{(i),k} = \mathrm{cov}\left[I_{F(i),jn}, I_{F(i),j,n+k}\right]$ which again is independent of n due to stationarity. This leads to the variance of the estimated conditional failure probability of a single chain:

$$\mathrm{var}\left[\hat{P}_{F(i),j}\right] = \frac{1}{N_S}\left(\gamma_{(i),0} + 2\sum_{k=1}^{N_S-1}\left(1 - \frac{k}{N_S}\right)\gamma_{(i),k}\right) \tag{2.107}$$

For analysis it can be assumed that the samples of different chains from a single subset are uncorrelated [AB01], i.e.

$$\mathrm{var}\left[\sum_{j=1}^{N_C}\hat{P}_{F(i),j}\right] = N_C \mathrm{var}\left[\hat{P}_{F(i),j}\right] \tag{2.108}$$

where the index j on the right hand side can be an arbitrary chain index since the different chains are generated using the same algorithm and hence have the same variance. The overall estimated failure probability for the i-th subset is obtained by

$$\hat{P}_{F(i)} = \frac{1}{N_C}\sum_{j=1}^{N_C}\hat{P}_{F(i),j} \tag{2.109}$$

which is the average failure probability over all N_C chains of the (i)-th subset. Inserting the unity for the variance $\mathrm{var}\left[N_C \hat{P}_{F(i)}\right] = N_C^2 \mathrm{var}\left[\hat{P}_{F(i)}\right]$ and the relation $\sum_{j=1}^{N_C}\hat{P}_{F(i),j} = N_C \hat{P}_{F(i)}$

from equation (2.109) as well as the variance of a single chain (2.107) into (2.108), the following expression for the variance of the estimated conditional failure probability of the i-th subset level results:

$$\begin{aligned} N_C^2 \mathrm{var}\left[\hat{P}_{F(i)}\right] = N_C \mathrm{var}\left[\hat{P}_{F(i),j}\right] = N_C \frac{1}{N_S}\left(\gamma_{(i),0} + 2\sum_{k=1}^{N_S-1}\left(1-\frac{k}{N_S}\right)\gamma_{(i),k}\right) \\ \rightarrow \mathrm{var}\left[\hat{P}_{F(i)}\right] = \frac{1}{N}\left(\gamma_{(i),0} + 2\sum_{k=1}^{N_S-1}\left(1-\frac{k}{N_S}\right)\gamma_{(i),k}\right) \end{aligned} \tag{2.110}$$

using $N = N_C N_S$ in the second line. The autocovariance with zero lag $k = 0$ is the variance of a single Bernoulli trial (compare chapter 2.3.2):

$$\gamma_{(i),0} = P_{F(i)}\left(1 - P_{F(i)}\right) \tag{2.111}$$

where $P_{F(i)} = P\left(F_{(i)} \mid F_{(i-1)}\right)$ is the conditional failure probability of the i-th subset. Although the conditional probability is not exactly known, the autocovariance with zero lag can be approximated by $\gamma_{(i),0} \approx \hat{\gamma}_{(i),0} = \hat{P}_{F(i)}\left(1 - \hat{P}_{F(i)}\right)$ where $\hat{P}_{F(i)}$ is similar to the desired conditional probability P_0 for all subset levels except of the last. For the last subset level, the conditional failure probability is equal to N_F/N (compare equation (2.96)). The autocorrelation with lags $k = 1 \ldots N_S - 1$, i.e. the covariance between different samples of a single chain, can be obtained by averaging over all chains [AB01]:

$$\gamma_{(i),k} \approx \hat{\gamma}_{(i),k} = \left(\frac{1}{N - kN_C}\sum_{j=1}^{N_C}\sum_{m=1}^{N_S-k} I_{F(i),jm} I_{F(i),j,m+k}\right) - \hat{P}_{F(i)}^2 \tag{2.112}$$

To simplify further analysis, the influence of correlated samples is condensed using a scaled autocorrelation function $R_{(i)}$:

$$R_{(i)} = 2\sum_{k=1}^{N_S-1}\left(1-\frac{k}{N_S}\right)\frac{\gamma_{(i),k}}{\gamma_{(i),0}} \tag{2.113}$$

Substituting equation (2.113) into (2.110) yields:

$$\mathrm{var}\left[\hat{P}_{F(i)}\right] = \frac{\gamma_{(i),0}}{N}\left(1 + R_{(i)}\right) \tag{2.114}$$

If the samples are not correlated, which results in the autocorrelation $R_{(i)} = 0$, the variance of the estimated probability of the i-th subset $\mathrm{var}\left[\hat{P}_{F(i)}\right]$ becomes the variance of pure Monte Carlo simulation (given in (2.13)). For correlated samples of a Markov chain, usually $R_{(i)} > 0$, i.e. there is a positive correlation between the samples which leads to an increased estimation variance. There exists the possibility that the autocorrelation is negative, which means that if one sample lies in the failure region, the next sample more likely lies out of the failure region. However, usually $\gamma_{(i),k}$ only becomes negative for some k and especially only for longer chains,

i.e. higher values of k and the effect is averaged out by the sum in equation (2.113). Hence, a negative autocorrelation is a theoretic case in this context.

The coefficient of variation of a single subset, which gives the relative uncertainty of the estimated conditional failure probability, can be defined similarly to the Monte Carlo estimator (2.17), which results in:

$$CoV\left[\hat{P}_{F(i)}\right] = \sqrt{\frac{1 - P_{F(i)}}{N P_{F(i)}}\left(1 + R_{(i)}\right)} \tag{2.115}$$

According to the strong law of large numbers, which is applicable to independent and identically distributed stochastic variables, $\hat{P}_{F(0)} \to P_{F(0)}$ for $N \to \infty$. The same holds for ergodic Markov chains [Gey05, p. 7], hence $\hat{P}_{F(i)} \to P_{F(i)}$ for $N \to \infty$. This results in $\hat{P}_F \to P_F = \prod_{i=1}^{m} P_{(i)}$ for $N \to \infty$, i.e. $\hat{P}_F$ is asymptotically unbiased. *Au* [AB01] derived an upper bound for the overall coefficient of variation for the estimated failure probability $\hat{P}_F$ as follows:

$$CoV\left[\hat{P}_F\right]^2 = E\left[\frac{\hat{P}_F - P_F}{P_F}\right]^2 \leq \sum_{i,j=1}^{m} CoV\left[\hat{P}_{F(i)}\right] CoV\left[\hat{P}_{F(j)}\right] + o\left(1/N\right) \tag{2.116}$$

$o\left(1/N\right)$ is the "small-o" notation also known as small Landau symbol, which means that the error is asymptotically negligible, since it grows slower than $1/N$. For the ideal case that samples of different subsets are not correlated, the lower bound for the overall coefficient of variation results:

$$CoV\left[\hat{P}_F\right]^2 = E\left[\frac{\hat{P}_F - P_F}{P_F}\right]^2 = \sum_{i=1}^{m} CoV\left[\hat{P}_{F(i)}\right]^2 \tag{2.117}$$

For evaluation of the derived variation bounds, again the example given in the previous section is used. 1000 independent Subset simulation runs using Infinity Sampling are conducted to calculate the actual variance of the estimated quantities, which are compared to the analytic bounds. The resulting individual CCDF curves are shown in figure 2.38a), where compared to the original example, the final failure response r_{limit} is increased to also evaluate smaller failure probabilities. The distinct colors indicate different subset levels. Subfigure b) gives the mean and variance of the family of curves together with the analytic variance bounds. The dashed and dotted lines depict the upper and lower estimated covariance bounds respectively, i.e. for the case of full correlation between the samples of different subsets according to equation (2.116) and no correlation according to (2.117). The variance bounds are evaluated for the failure probabilities $P_F = [10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}]$, which for the selected algorithm parameters correlates to a total number of samples $N = [1000, 2000, 3000, 4000, 5000]$.

Figure 2.39 gives the coefficient of variation for this example, again with the estimated lower and upper variation bounds. It can be seen that for lower Subset levels, i.e. higher probabilities, the actual variation of the results is closer to the lower, uncorrelated variation bound while with decreasing probability, the actual variation lies somewhere between the two bounds. This can be explained by the increasing dependence and hence correlation of samples with increasing

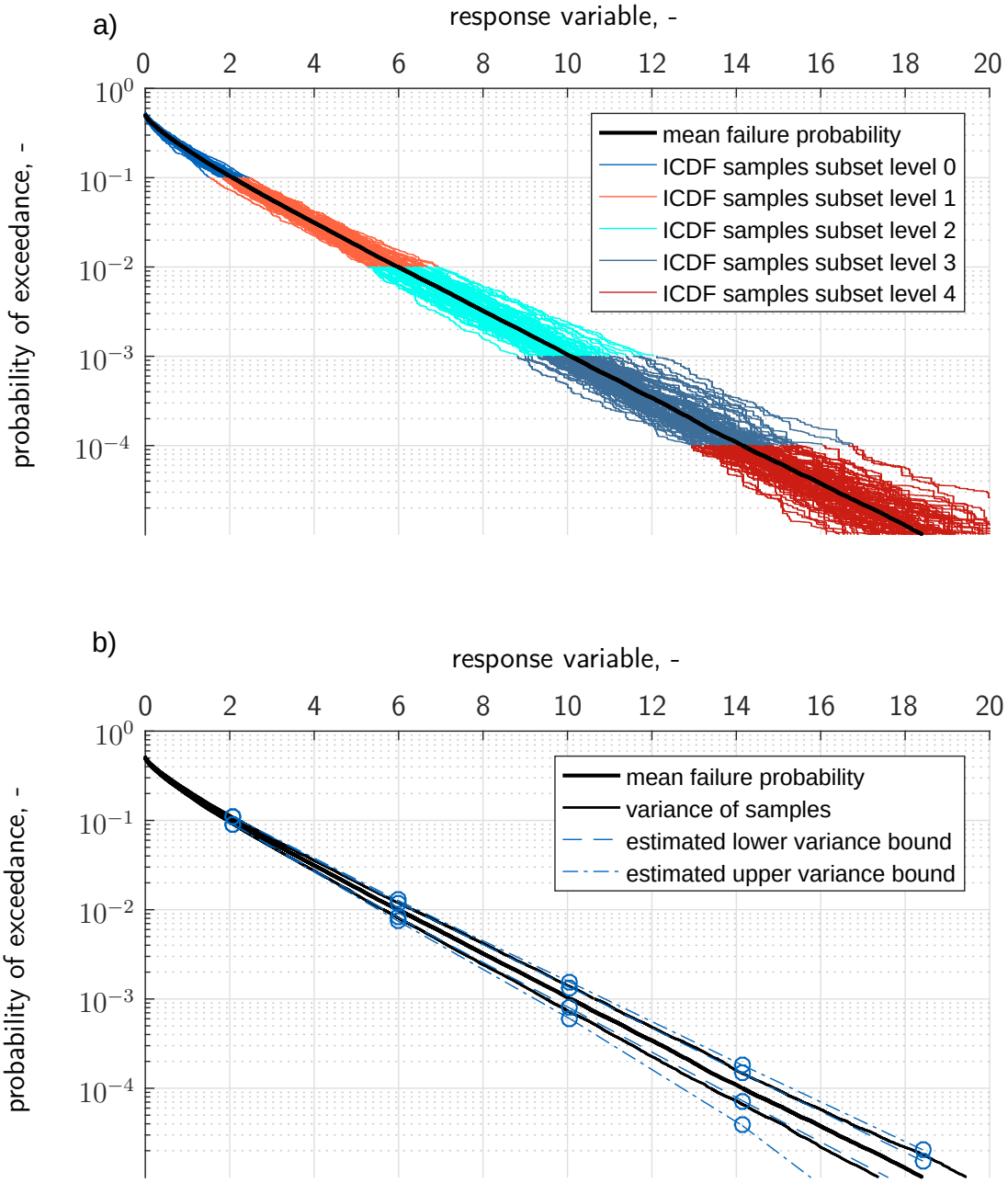


Figure 2.38: *Variance of Subset simulation; a): Complementary cumulative distribution functions for 1000 independent Subset simulation runs, b): Comparison of sample variance and estimated variance*

subset level. In the figure, also the resulting variance using Monte Carlo simulation is given where it is assumed that for $P_F = [10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}]$ the same number of samples as for Subset simulation $N = [1000, 2000, 3000, 4000, 5000]$ is used. For the probability $P_F = 10^{-1}$, variance of Subset and Monte Carlo simulation is similar. This is expectable since in the zeroth level of Subset simulation, the samples are generated using independent and identically distributed samples similar to Monte Carlo simulation, i.e. the coefficient of variation for both is given by equation (2.18). With decreasing probability, the coefficient of variation for Monte Carlo simulation increases exponentially, while it only increases linearly for Subset simulation.

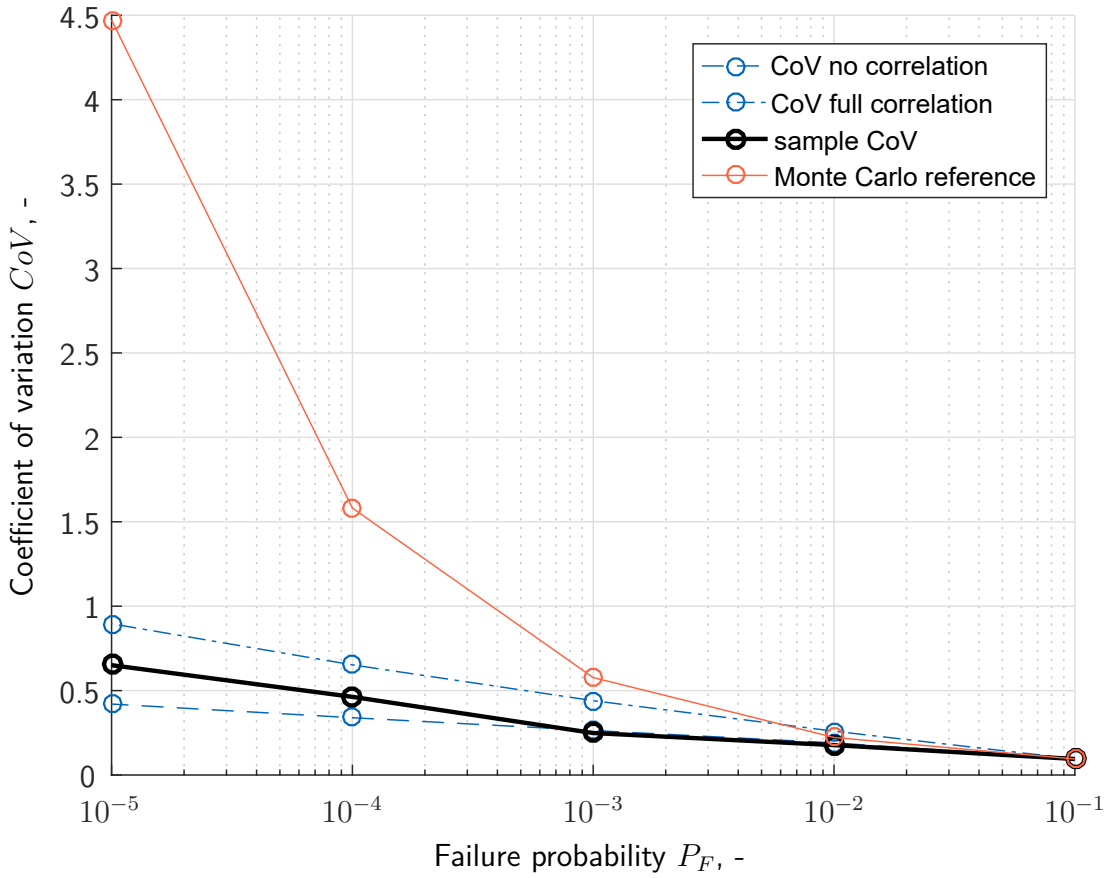


Figure 2.39: *Variance of Subset simulation compared to Monte Carlo simulation*

Figure 2.40 gives the number of samples for Subset simulation and the number of samples that would be required for pure Monte Carlo simulation to obtain the same coefficient of variation as achieved in the given example by Subset simulation. For $P_F = 10^{-5}$, almost 50 times more samples would be required for Monte Carlo simulation to achieve the same accuracy.

The number of samples required to achieve a certain estimation accuracy cannot be determined a priori since it depends on the correlation between the samples which itself depends on the tuning of the algorithm and the specific application. However, a rough estimation can be obtained. Consider same conditional probabilities $P_{F(i)} = P_0$ for all subset levels $i = 1 \dots m$, as well as same autocorrelation and hence coefficient of variation $CoV\left[P_{F(i)}\right] = CoV\left[P_{F(1)}\right]$. Using $P_F = \prod_{i=1}^{m} P_{F(i)} = P_0^m$, the required number of subsets for a given failure probability P_F and conditional probability $P_{F(1)}$ is given by

$$m = \log(P_F)/\log(P_0) \tag{2.118}$$

The admissible variance for individual subset levels is obtained by substituting the assumptions into equation (2.117) which results in

$$CoV\left[\hat{P}_F\right]^2 = \sum_{i=1}^{m} CoV\left[\hat{P}_{(i)}\right]^2 = mCoV\left[P_0\right] \tag{2.119}$$

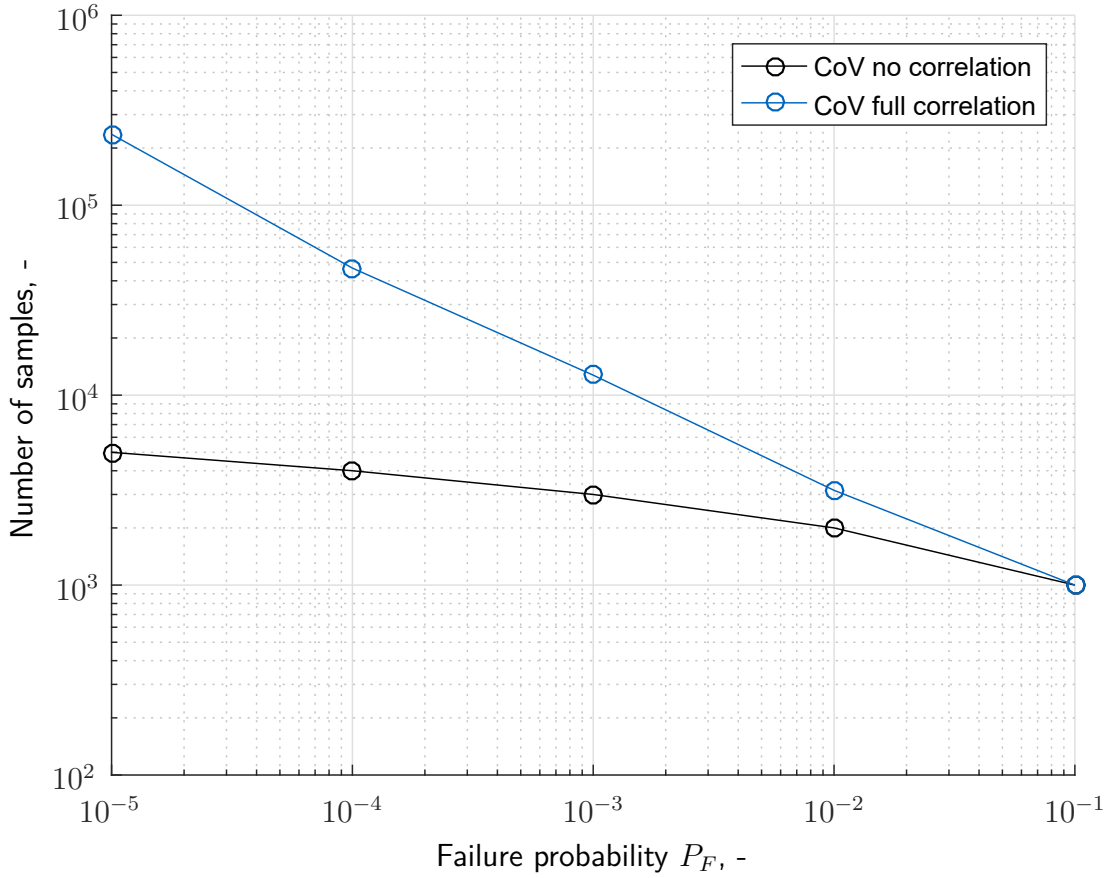


Figure 2.40: *Comparison of number of samples for Subset and Monte Carlo simulation*

Inserting equations (2.118) and (2.119) into (2.115) and reformulation yields

$$N \approx \frac{\log\left(P_F\right)}{\log\left(P_0\right) CoV_{des}\left[\hat{P}_F\right]^2} \frac{1-P_0}{NP_0}\left(1+R_{(1)}\right) \tag{2.120}$$

where CoV_{des} denotes the desired overall coefficient of variation for the estimated failure probability P_F. A target $CoV_{des} \approx 0.3$ has been derived in section 2.3.2. This means that – given that there is no correlation between the samples of different subsets – the number of samples required grows only with the logarithm of the failure probability, i.e. $N \propto |\log\left(P_F\right)|$. Opposed to conventional Monte Carlo simulation, where the number of samples $N \propto 1/P_F$ (see equation (2.18)), this is a significant enhancement, especially for very small probabilities. *Au* [AB01] suggested an enhanced metric that takes into account the correlation between the samples of different subset levels by a scalar scaling factor $r \leq 3$, where higher values stands for higher correlation:

$$N \approx \frac{\left|\log\left(P_F\right)\right|^r}{\left|\log\left(P_{(1)}\right)\right|^r CoV\left[\hat{P}_F\right]^2} \frac{1-P_{(1)}}{NP_{(1)}}\left(1+R_{(1)}\right) \tag{2.121}$$

This results in $N \propto |\log\left(P_F\right)|^r$, which still surpasses the performance of conventional Monte Carlo simulation.

2.6.5 Conclusions on Subset Simulation

Subset simulation has been found as powerful tool for stochastic evaluation of complex functions. Opposed to specialized algorithms like importance sampling, no knowledge about the function must be input to the algorithm. This black box behavior is beneficial for the ease of application for engineers. Although the underlying theory of Markov Chains is complex and hard to understand, no detailed knowledge is required for a successful application of the Subset simulation method.
The samples of Subset simulation obtained by both presented algorithms, Markov Chain Monte Carlo and Importance Sampling, are correlated in contrast to the uncorrelated samples of basic Monte Carlo simulation. This usually comes along with lower estimation accuracy. However, since Subset simulation generates more samples in the failure region, this algorithm is superior to any other black box stochastic evaluation algorithm, especially for low probabilities.
The samples of Subset simulation are generated gradually by random walks towards more critical regions. This leads to the theoretical drawback that the solution space must be connected or not too far separated either with respect to the probability space or the parameter space. In this context, not connected means that there is a region between two solution subspaces, where the probability that the value of an uncertain parameter lies there is close to zero. This is shown in figure 2.41, where the contours of the probability distribution of a single uncertain parameter θ versus a decreasing failure probability is shown. In subplot a), the parameter space is separated with respect to the parameter θ, however it is connected with respect to the probability leading to a valid parameter distribution for Subset simulation. In the first step of Subset simulation, samples are generated according to the original parameter distribution, which is similar to the probability distribution for a failure probability of $P_F = 1$ shown on the upper end of the plots. Hence, samples in both separated spaces are generated with decreasing probability by the random walk of the Subset samples. Subplot b) shows the case that although the space is separated with respect to the uncertain parameter for higher probabilities, this is not the case for lower probabilities, which results in a legitimate distribution. Only for the third case shown in subplot c), there is neither a connection in probability nor parameter space: Only at very low probabilities, a region occurs that significantly contributes to the failure probability. Since there is no connection to the other parameter subspace, no samples are generated in this region by Subset simulation and hence its contribution to the overall failure probability is neglected. However, to the best knowledge of the author, for physical systems this is only possible for discrete failure events, which suddenly occurs at lower probabilities. Since discrete events can be well covered by established methods like fault trees, this is no severe limitation of the introduced method and hence Subset simulation is the first choice for reliability analysis in the context of this work.

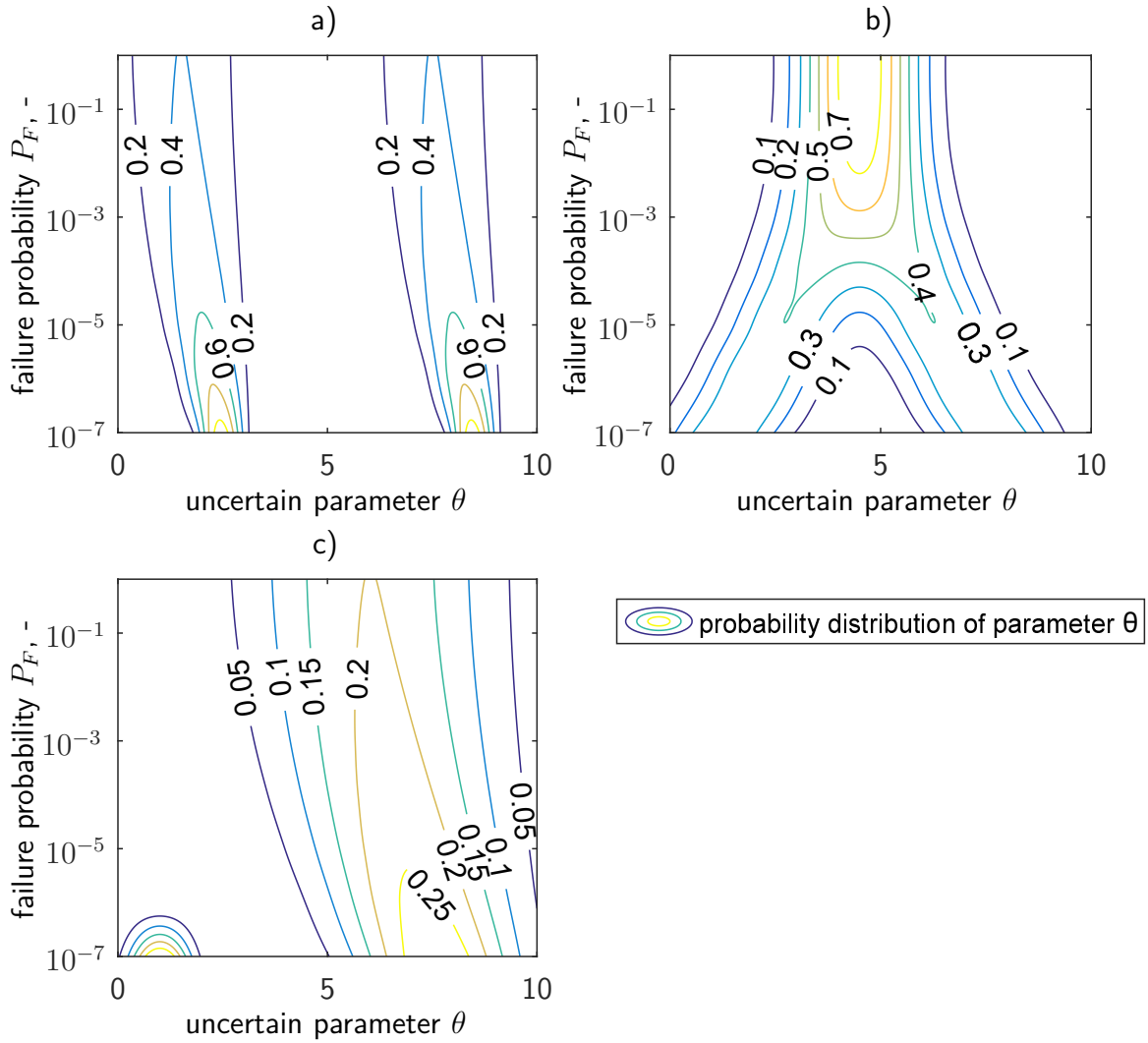


Figure 2.41: *Interpretation of connected spaces for Subset simulation*

3

Total Capability Approach

3.1 Introduction

In this chapter, a novel development approach for safety-critical functions is introduced, which particularly addresses the challenge of developing systems in a way that quantitative safety guarantees can be given. For that, first terms and definitions are given that are essential to understand the description of the common aircraft and systems development process, which is explained afterwards. Eventually, the concept of the *Total Capability Approach* (TCA) is introduced, the benefits are highlighted, and arising challenges for application of the new development approach are discussed. The principle idea of the TCA was first presented and published by the author of this dissertation on the 13th International Conference on Probabilistic Safety Assessment and Management in October 2016 [LMH16b]. However, the descriptions given in this chapter are much more thorough and detailed.

3.1.1 Terms and Definitions

In chapter 1, specific terms of a development process were used without preceding definition, e.g. function, validation and verification. The definitions are common for the aircraft and systems development process. However, they are given here to ensure an equal understanding. The explanations given below follow the definitions used in the *Aerospace Recommended Practice* (ARP) *for development of civil aircraft and systems* ARP4754A [SAE10].

- **Safety**: "*Safety is the state in which risk is acceptable.*" [SAE10, p. 13]
- **Risk**: Risk is the combination of severity of an event and its probability of occurrence.
- **System**: A system is the combination of interrelated items required for and used to perform specific functions. On top level, the system is the aircraft itself, while lower-levels refer to specific systems of the aircraft, e.g. the flight control system. For *Unmanned Aerial Systems* (UASs), the top level also includes the ground segment and data link.

- **Function**: A function is the desired behavior of a product. It is specified regardless of the implementation. A function must be completely specified by a set of requirements.
- **Requirement**: A requirement is "*an identifiable element of a function specification that can be validated and against which an implementation can be verified*" [SAE10, p. 13].
- **Validation**: Validation is the process of ensuring correctness and completeness of requirements for a product. It is to answer the question whether we are building the right aircraft/system/function/item. Hence, validation ensures that the requirements are sufficient and suitable to describe the intended functionality. In a broader sense, validation can be understood as the assurance that a specified functionality meets the needs defined by customers and stakeholders.
- **Verification**: Verification is the assurance that the implemented functionality meets all requirements. It is to answer the question whether we build the aircraft/system/function/item right.
- **Item**: An item is "*a hardware or software element having bounded and well-defined interfaces*" [SAE10, p. 12].
- **Model**; A model is "*an abstract representation of a given set of aspects of a system/function/item that is used for analysis, simulation and/or code generation and that has an unambiguous, well defined syntax and semantics*" [SAE10, p. 12].

Eventually, a safety-critical function is a function that potentially has an influence on safety of the considered system and operation and hence sufficient performance of the function is required to achieve an acceptable level of safety.

3.2 Aircraft Development Process

Many requirements are put on aircraft and systems to be developed. Requirements are mainly motivated by two stakeholders: customer and certification authorities. The first ensures that the developed product is useful for the customer and the envisaged operation. The second ensures that the final product is considered as safe and hence certifiable for the intended application. In this research, especially requirements arising from the second aspect are of relevance.

According to article 31 and 33 of the convention on international civil aviation, "*every aircraft engaged in international navigation shall be provided with a certificate of airworthiness issued or rendered valid by the State in which it is registered*" [Int06, p. 14] and this certificate "*shall be recognized as valid by the other contracting States*" [Int06, p. 15]. Already in the introduction, sources for technical requirements, which must be fulfilled to obtain a certificate of airworthiness, were listed, e.g. the certification specifications for small and large aircraft CS23/25 published by the EASA [Eur17d, Eur17b] for the European area as well as the

Table 3.1: *Relationship between severity of the effects and classification of failure conditions [Eur17b, p. 2-F-46]*

Severity of the Effects						
	Effect on Aeroplane	No effect on operational capabilities or safety	Slight reduction in functional capabilities or safety margins	Significant reduction in functional capabilities or safety margins	Large reduction in functional capabilities or safety margins	Normally with hull loss
	Effect on Occupants excluding Flight Crew	Inconvenience	Physical discomfort	Physical distress, possibly including injuries	Serious or fatal injury to a small number of passengers or cabin crew	Multiple fatalities
	Effect on Flight Crew	No effect on flight crew	Slight increase in workload	Physical discomfort or a significant increase in workload	Physical distress or excessive workload impairs ability to perform tasks	Fatalities or incapacitation
Classification of Failure Conditions		No Safety Effect	Minor	Major	Hazardous	Catastrophic

counterpart for the United States 14 CFR Part 23/25 published by the FAA [Fed67, Fed64]. Those standards are well harmonized and also many other countries outside Europe and the US rely on them for the certification of civil aircraft. The requirements given in these specifications cover all technical aspects that must be taken into account for a system to be considered as safe and certifiable. Almost all of the given safety-driven functional, performance and implementation requirements are deterministic and prescribe specific design solutions to ensure safety. For example, CS25.145 gives requirements for longitudinal control of aircraft for various specific configurations, e.g. with landing gear or wing-flaps retracted or extended. The only exception is paragraph CS 25.1309 - *Equipment, systems and installations*. This specification and the corresponding *Acceptable Means of Compliance* (AMC) give specific admissible risk levels for according levels of criticality. As an example, if a failure condition can cause a catastrophic event, which is normally characterized by a hull loss and multiple fatalities, the admissible failure probability, i.e the maximum probability that this condition occurs, is $P_{limit} = 10^{-9}$. Tables 3.1 and 3.2 gives in detail the classifications for criticality and related admissible failure probabilities according to AMC CS 25.1309 [Eur17b, Appendix F]. Certainly, the more severe the effect of a failure condition, the lower the acceptable probability for this failure condition must be. This requirement must not be understood as replacement for the prescriptive design requirements given in the certification specifications to ensure safety. It is written in CS25.1309 that "*the requirements of this paragraph (. . .) are applicable, in addition to specific design requirements of CS-25, to any equipment or systems as installed in the aeroplane. Although this paragraph does not apply to the performance and flight characteristic requirements of Subpart B (. . .), it does apply to any system on which compliance with any*

Table 3.2: *Relationship between classification of failure conditions and probabilities [Eur17b, p. 2-F-47]*

Classification of Failure Conditions	No Safety Effect	Minor	Major	Hazardous	Catastrophic
Allowable Qualitative Probability	No Probability Requirement	Probable	Remote	Extremely Remote	Extremely Improbable
Allowable Quantitative Probability: Average Probability per Flight Hour on the Order of:	No Probability Requirement	$< 10^{-3}$	$< 10^{-5}$	$< 10^{-7}$	$< 10^{-9}$

of those requirements is dependent" [Eur17b, p. 1-F-4]. Hence, this "safety paragraph" only ensures that the mostly conservative, prescriptive design requirements are implemented in a manner that ensures a high level of availability and integrity of the implemented functions. For example, a flight control system used to fulfill conservative requirements put on aircraft control must be designed in a manner that failures of components of the flight control system leading to non-compliance with those requirements do not happen with a probability higher than the threshold value.

A different classification of criticality is found in some military specifications. For example, in the SAE standard AS94900 [SAE07], which is a general specification for design, installation and test of *Flight Control System* (FCS) of piloted military aircraft, different levels of FCS performance degradation are classified by "*Operational State I (Normal Operation)*" to "*Operational State V (Controllable to an Evacuable Flight Condition)*". Standards that use these classifications usually impose different performance requirements dependent on the operational state. Using this approach, the criticality of functions is already considered to a certain extend in the function description. Common to both approaches is the classification of functions dependent on their criticality, which is important for development of safety-critical functions.

3.2.1 Current Aircraft and Systems Development Process

During the last two decades, several guidelines were developed to provide standardized processes for the development of aircraft, systems, and components. Figure 3.1 gives an overview of most common standards and their focus.

These guidelines cover top-level aspects like the aircraft and system development process and related safety processes as well as lower-level aspects like electronics and software development. Especially of relevance for this this research is the ARP4754A. The guidelines given in there were developed in context of development of large transport aircraft (14CFR Part 25 and CS-25), but may also be applicable to other regulations for development of different aircraft categories and helicopters (CS-23, CS-27, CS-29). It describes how aircraft and systems can

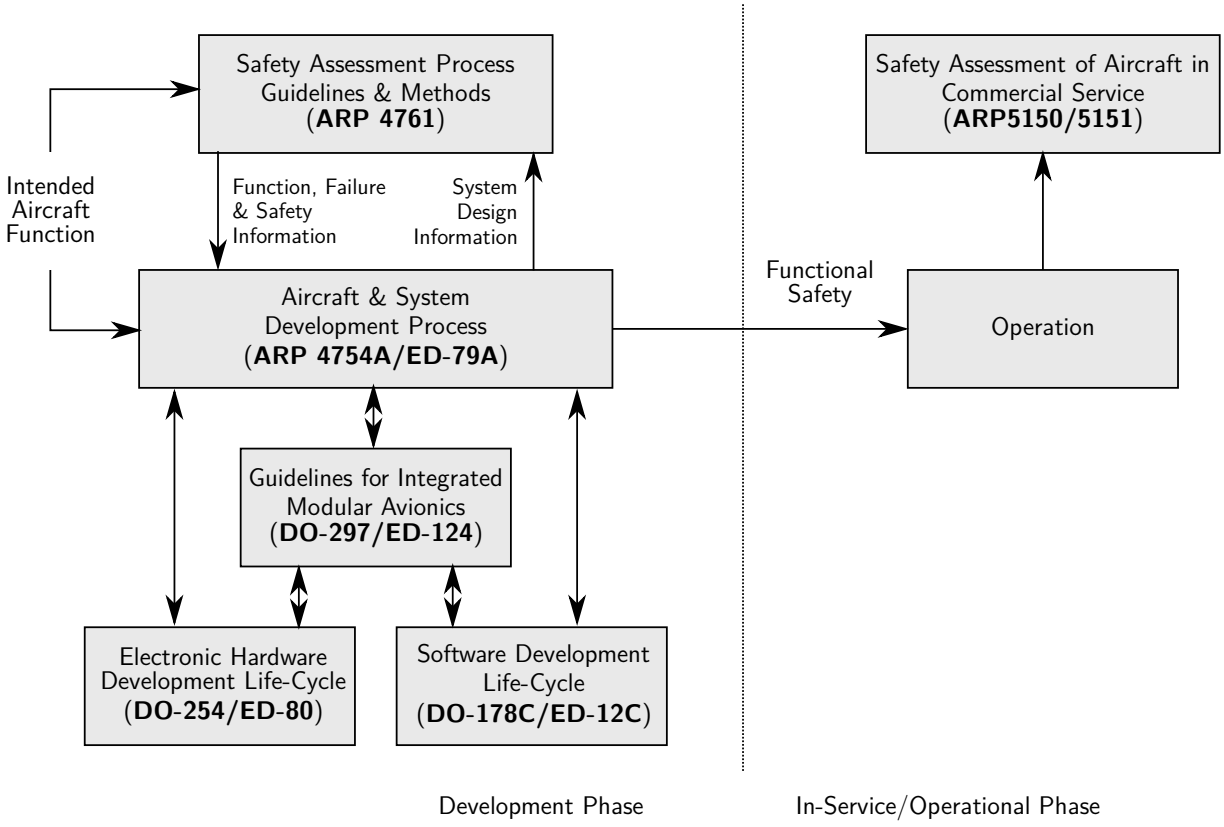


Figure 3.1: *Guideline documents covering development and in-service/operational phases [SAE10, p. 6]*

be developed especially to fulfill the probabilistic requirements put on the implementation of safety-critical functions by taking into account the aircraft operating environment and all functions contributing to the intended operation. An overview of the aircraft and system develop process is given in figure 3.2. It spans the whole development from the initial concept and definition of aircraft functions to the implementation and documentation. Two main processes can be divided:

- **Aircraft/system development process**: This process comprises the engineering tasks, i.e. the actual development of functions and its implementation. Top level functions are derived from a concept, followed by a break-down of aircraft functions, their allocation to systems and items, and eventually their implementation.

- **Integral processes**: The integral processes influence all steps of the development process. The focus lies on ensuring safety by a transparent, well documented, and consistent process. The integral processes are not absolutely necessary to obtain a working implementation but cause a big share of the overall development time and hence costs. Still, they are essential to ensure safety of the resulting product and to obtain certification.

The intended aircraft and system functions are captured by requirements. In the safety assessment according to ARP4761 [SAE96], these functions are evaluated with their associated failure modes and effects on the aircraft.

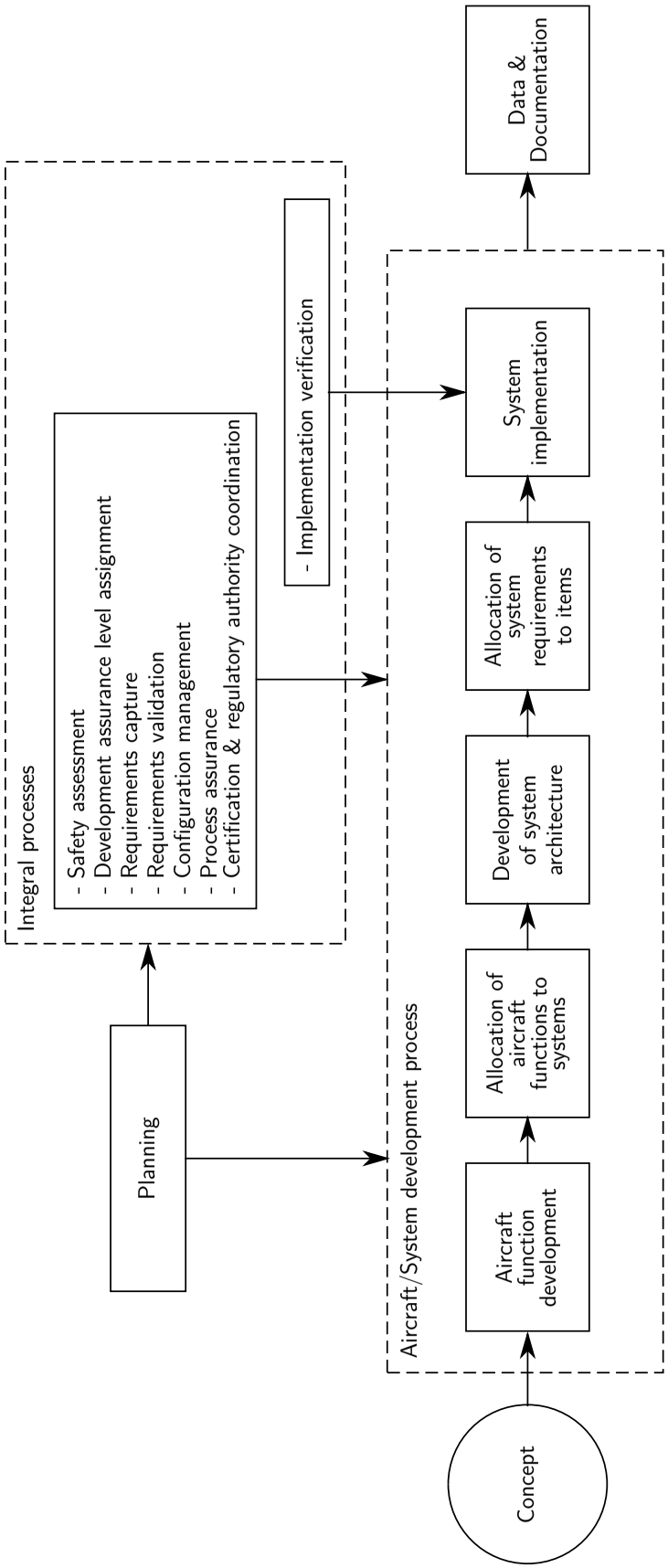


Figure 3.2: *Aircraft or system development process model [SAE10, p. 21]*

Requirements and related hazards are essential and form the basis for the integral processes. For the novel development approach for safety-critical functions presented in this theses, particularly the way how requirements are derived, validated and verified is of importance. There are many types of requirements, e.g. safety, functional, performance, customer, installation, and interface requirements. Especially the first three types are essential for this research:

- **Functional requirements**: Every function to be implemented is captured in a functional requirement, which describes in detail the intended functionality. There are several sources for functional requirements, for example, from customer desires, operational limitations, implementation restrictions and regulatory authorities.
- **Safety requirements**: Safety requirements give minimum performance constraints for the availability and integrity of functions. The basis for those requirements are safety assessments of the related functions. Dependent on the criticality of each function, the admissible failure probabilities can be very low, see table 3.2. Safety requirements should be uniquely identified and traceable to the according functions.
- **Performance requirements**: Performance requirements specify the attributes of aircraft and system functions. Again, there are several sources for performance requirements, e.g. from customer and safety considerations.

Note that different sources of requirements could also lead to inconsistent requirements, for example, if the customer intentionally wishes to have a function or a certain performance range that violates safety requirements. In the following, the process of capturing safety-related functional and performance requirements is further analyzed. To indicate the origin of such requirements and to clearly distinguish them from requirements from other sources, they are hereinafter referred to as "safety-driven" functional and performance requirements.

3.2.2 Safety of Current Approach

The aircraft and systems development process is usually strictly top-down: From the definition of aircraft functions down to item level. This fulfills the regulatory demand that functions must be traceable through all levels of development, i.e. the function of every item is associated to a system and therefore also to an aircraft level function. This should also be true for safety requirements: Classification of top-level failure conditions lead to safety requirements for top-level functions. Lower-level requirements that are defined to prevent these failure conditions and to provide safety-related functions are derived from these top-level safety functions and are therefore traceable to the top-level safety functions. However, when using today's regulatory standards like the afore mentioned EASA CS25, the process is rather bottom up: The prescriptive safety-driven design requirements given in those specifications already describe certain functions and their performance on a very detailed, i.e. low level. It is supposed that the experience-based collection of safety-driven design requirements are sufficient to ensure safety on aircraft level. Unfortunately, this is a very qualitative trace, since there is no real

link between the different levels of requirements except of the experience gained during one century of aircraft development. Furthermore, safety is only ensured on a qualitative level, with no quantification of the actual risk.

Safety assessment outputs the criticality of each function and their associated critical events. Unfortunately today this information of criticality is only used to make sure that the implementation of these functions has a high availability and integrity, which actually only ensures that functions do not fail due to implementation issues, like unreliable hardware or software. This does not consider other sources influencing the performance and hence safety of the function, like for example uncertainties and disturbances. Today's safety approach applied to the prescriptive safety-driven design requirements given in current certification specifications only ensures that the mostly conservative lower-level functions are implemented in a safe manner, so that the related failure conditions do not occur with a probability higher than accepted.

The current approach does not guarantee that the overall risk for a failure condition is smaller than the established maximum admissible failure probability. The difference between the overall risk and the failure probability obtained by today's approach is the risk that a function is inadequately specified or that disturbances and uncertainties are not adequately covered by functional and performance requirements. If the function and its performance lead to an unsafe behavior under certain conditions neglected during specification, an implementation of this function with highest availability and integrity would not increase safety. For example, if wind shears would not be adequately covered in the safety-driven requirements for an autopilot for automatic landing, this could lead to a crash, although established development and safety assessment processes were used to develop a "safe" autopilot. Fortunately, today's safety-driven prescriptive design requirements given in certification specifications are conservative enough that the major influence on safety arises from implementation. Since implementation is well covered by established development and safety assessment processes, today's approach is still a practicable way for developing safety-critical functions for conservative applications for which many decades of experience is available.

3.2.3 Drawbacks of Current Approach

Although the current approach with prescriptive design requirements is well established and usable for conventional aircraft, it comes along with several drawbacks, that impede its application for novel aviation applications:

- Top-level safety requirements are linked to admissible probabilities of failure for related top-level functions, while safety-driven lower-level requirements from certification specifications are deterministic. There is no physically motivated or quantitative relation between safety requirements of different levels. Today, this is compensated by conservative certification specifications, which come along with sacrifice of possible total system performance.

- Today's certification specifications usually follow the principle "*one system for one function*" and split up performance budges to subfunctions, for example the total performance of a flight control system into admissible control and navigation errors. Furthermore, current specifications often prescribe certain design solutions, which prevents usage of novel state-of-the-art solutions.

- Current safety-driven performance requirements are usually only formulated for specific, deterministic conditions. Although the implemented function pretends to be safe, it is only as safe as the function specification and the completeness of expected conditions. Hence, the current approach does not guarantee safety of the overall system.

- Safety-driven requirements from certification specifications are based on past experience. Using those requirements based on conventional concepts for novel aviation applications can be inappropriate or even adverse for ensuring safety. Hence, the classical approach cannot be applied for systems with modern or disruptive technologies like electric propulsion and the wide variety of configurations of UAV resulting thereof.

A shift in paradigm must take place to enable development of aircraft with non-conventional topologies or operations.

3.3 Shift in Paradigm

Today, major mitigation means to reduce risk of safety-critical functions are to deactivate a non-compliant function and shift the responsibility to the pilots (e.g. in case of a non-compliant autopilot behavior during cruise flight) or to initiate an automatic abort function that ensures a safe transition to less critical states (e.g. for abort of an automatic approach). These mitigation means reduce the availability of functions but are acceptable for conventional aircraft operations. Future applications will probably rely on highly automated systems where no pilots will be available as fallback option and availability must be very high to allow for reliable operations. Furthermore, today criticality of failure conditions is mainly driven by aircraft safety and the influence of incidents on crew and passengers. However, separation between aircraft safety and operation becomes increasingly difficult. For example, it is less critical if a heavy UAV crashes on an empty filed compared to into a crowd of people. While at the beginning of the research, this separation was not yet an issue, it became of major importance for safe operations of systems using disruptive technologies. Therefore, conservative approaches as of today are no viable option for future applications, where it will be important to consider the influence of aircraft and operation on safety in an integrated manner.
The current development process for aircraft and system described in the previous section will also be essential in future since it gives a standardized framework for the development of safety-critical applications, as it is well established and yielded good and safe systems in past. Anyway, especially the capture and derivation process of safety-driven requirements does

not meet future needs for the development of novel aircraft topologies and operations with complex functions and tightly coupled components. The TCA introduced in this thesis supports a shift towards an integral, probabilistic consideration of the overall system performance during system specification to reduce conservativeness but still ensuring a high level of safety. One attempt of an integrated, performance-based approach is the *Performance Based Navigation* (PBN), where a *Total System Error* (TSE) is defined as the sum of the *Flight Technical Error* (FTE), the *Navigation System Error* (NSE), and the *Path Definition Error* (PDE) [Int13]. PBN prescribes monitoring algorithms that ensure that the total error does not exceed certain limits during operation. However, while this approach only focuses on the operational aspect, the TCA aims at the overall development and operation of safety-critical functions and already starts with integrated considerations during requirements derivation.

3.3.1 Concept of the Total Capability Approach

The idea of the TCA rest on two pillars: First, the capability of each item and the resulting performance of each function contribute to the performance and hence also to the safety of the overall system. Instead of separately considering each component during development and especially specification, its effect on the total system performance in combination with all other components and its influence on safety is considered. A simple example for this idea is the special case that only two functions contribute to the overall performance of a system. Following conventional, conservative approaches, each function is specified and designed independently and hence deteriorating performance of one function, which is non-compliant with conventional requirements put on this function, cannot be compensated by better performance of the other function. Following the idea of the TCA, the total performance of the resulting system is decisive for successful and safe operation and hence degraded performance of one function can be compensated by better performance of another function, see figure 3.3. This could be for example the total performance of a flight controller, which is composed by the control and the navigation performance. Following this total performance idea, it is possible to achieve higher availability without the necessity of using components with better performance, which are usually more expensive.

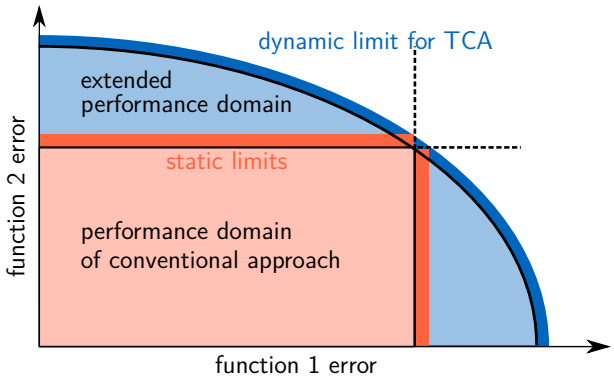


Figure 3.3: *Independent (conventional) versus joint performance limits (TCA)*

The second pillar is the consideration of the performance of each function in a probabilistic framework: Today, hard requirements are specified that must be met under certain specific, deterministic conditions. These conditions include for example specific test configurations (e.g. flap and gear position, airspeed) but also specific disturbances like wind. However, in reality disturbances do only occur with a certain probability. Furthermore even less favorable disturbances can occur, although with a lower probability, but are still possible. While this case is not covered by today's approach, the TCA allows for consideration of uncertainties, disturbances, and failures in a probabilistic way, that means by taking their intensity with related probability of occurrence into account during determination of the overall risk, which eventually results in a development approach for which risk and hence the level of safety can be quantified.
The term "*Total Capability Approach*" is motivated by those pillars: The performance-based approach with probabilistic requirements and integrated consideration of all contributing components is able to fill the gap between top-level probabilistic requirements and specific safety-driven design requirements by taking the total capability of each component and consequentially system and aircraft into account during the safety-driven development process. This novel approach is enabled by a physically driven, model-based development process.

3.3.2 Field of Application

The novel development approach presented in this thesis is directed toward systems that have aircraft-level functions with failure modes that potentially affect safety of the aircraft. Furthermore, it must be possible to adequately describe the items, functions, and resulting systems together with their uncertainties, disturbances, and failures that contribute to the failure modes by models. However, this is not a severe limitation compared to today's development approach: Models and simulations are often used during the design process. The novelty of the TCA is the application of models throughout the whole development process from requirements formalization to verification to enable the integral and probabilistic consideration of the performance of all components.
Models are used to evaluate and quantify the influences of uncertainties, disturbances, and failures of components on risk and hence on safety. The idea of the TCA is to derive safety-driven performance requirements based on this knowledge. There is often a high influence of uncertainties on the performance of a function and, therefore, on safety: Whenever a risk results from an underperforming function, safety-driven performance requirements must be established that provide admissible performance ranges.
The TCA is especially useful if there are many disturbances and uncertainties influencing the performance of each component and consequentially of the whole system. Although discrete failures can be well modeled and simulated, it is recommended to stick to well established methods like fault tree analysis for such failures. Further descriptions will therefore not explicitly discuss the failure case. Nevertheless, since failures are nothing but discrete disturbances, the presented methods are theoretically also applicable for the evaluation of failure cases.

3.4 Steps of the Total Capability Approach

Although the TCA is motivated by the need for a more reasonable derivation of safety-driven requirements, the shift in paradigm also impacts other steps of the aircraft and systems development process. An overview of the different steps of the TCA is given in figure 3.4, detailed descriptions follow in the next subsections.

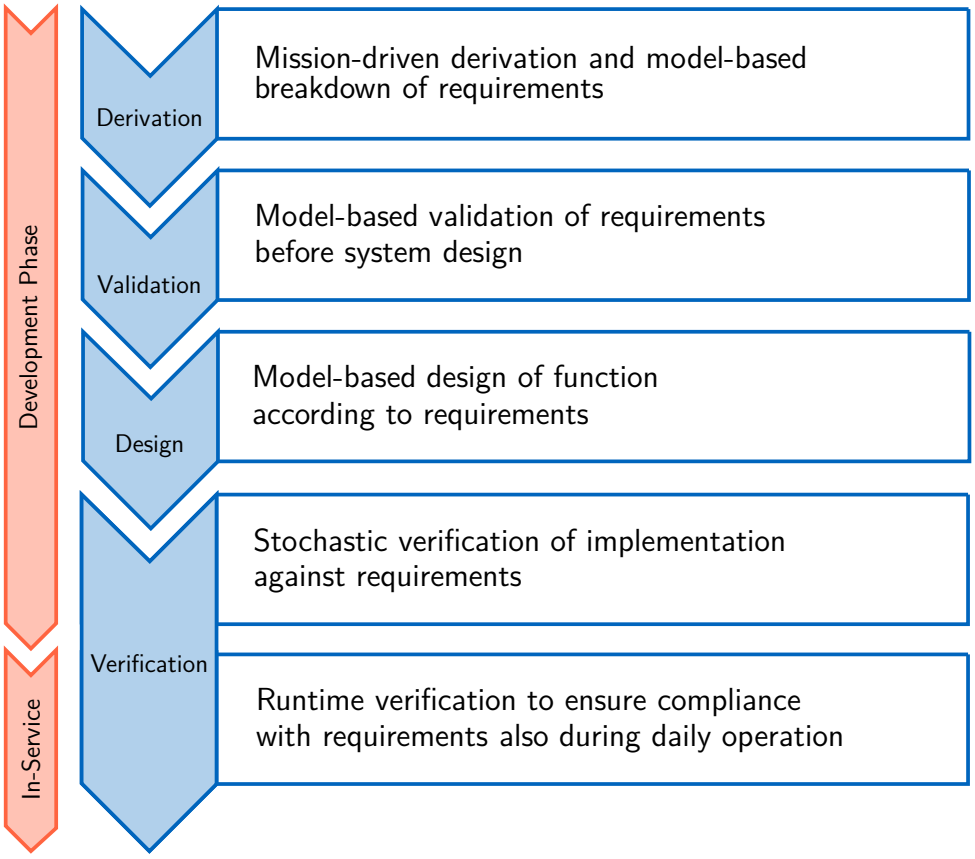


Figure 3.4: *Steps influenced by and main contribution of the TCA*

3.4.1 Requirements Derivation

The derivation of more reasonable safety-driven requirements compared to today's conservative, experience-based specifications is the main motivation for the TCA. Requirements derivation in this context is to answer the question "*How good must a function perform to result in a safe system?*". For that, top-level safety requirements are obtained by analysis of top-level functions and their criticality in a safety assessment. A model-based breakdown of these safety requirements to lower-levels results in safety-driven functional and performance requirements, which ensure that the specified function does not lead to a failure condition with a probability higher than acceptable.

The break-down process is based on models that use knowledge available at this early development stage, e.g. about the environment and already developed components. Note that during

this phase, usually no implementation of the intended function is available and no implementation must be assumed or anticipated since this would limit possible design solutions to those which use the assumed implementation. Instead, specification models are expected to model the behavior of the intended functions. For example, for development of a flight controller, instead of assuming a control method or architecture, only the desired behavior of the closed loop system is modeled. The process of deriving requirements in this framework is discussed in chapter 4.

3.4.2 Requirements Validation

Validation is ideally conducted before design and implementation of functions to prevent possible useless development efforts due to wrongly specified functions. Wrong specifications could for example arise from unnoticed inconsistent, missing or wrong requirements.

Today, validation is often only possible in a qualitative manner: A group of experts evaluate a catalog of requirements and try to figure out to their best knowledge and belief whether the given requirements are correct, non-contradicting and sufficient to completely specify the intended functionality and also eliminates the room for unintended functionality. The behavior models used for requirements derivation can facilitate the validation process since it allows to simulate the intended function before implementation. Conflicts like missing or inconsistent requirements can be easily determined, which can greatly ease the validation process for the experts and increase confidence in the function specification.

Note that this model-based validation is not only possible for derived safety-driven requirements but for all requirements which can be simulated, e.g. also customer requirements. Having simulations of the intended functionality before implementation also gives the customers more confidence in what they can expect from the resulting system and can initiate changes or additions at an early project phase, if necessary. This can save total development time and hence costs. More details on validation are given in section 4.4.

3.4.3 Design and Implementation

The design and implementation process for specified functions is very similar to today's approach: Requirements still uses classical metrics, only additionally linked to probabilities. This similarity is important to facilitate the implementation of the TCA. However, since safety for certification is no longer established by prescriptive design requirements but instead by a quantitative proof of safety of own design solutions, the increased design freedom also allows for application of novel, disruptive technologies and state-of-the-art solutions.

Furthermore, the simulation framework used for requirements derivation and validation allows for automatic testing of implementations against requirements. By that, a function can be designed and implemented in a way that it does not perform best under ideal and usually rather safe conditions but instead to result in highest safety or availability.

Methods and benefits of design an implementation in the context of the TCA are presented in chapter 5.

3.4.4 Verification

Verification is the step of ensuring that the implementation complies with the established requirements. This is especially of importance for the proof of safety, which is essential for certification. During this last phase of development, usually very detailed models are available that are also well verified against already available prototypes. Proof of safety is even more important for the TCA, since the less conservative requirements also reduce the natural buffer inherent to the current approach to mitigate effects of uncertainties and universality of certification specifications.

Today, it suffices to verify a new design only once to obtain certification of a system or aircraft. Following the TCA, offline verification is conducted similar to the classical approach, however, enhanced stochastic methods must be applied for model-based verification to efficiently prove the top-level safety requirements, which are usually linked to very low probabilities, especially for critical failure conditions.

Verification results are only as good as the uncertainties and disturbances assumed during the certification. Since this knowledge is certainly not perfect, it could happen in reality that some important sources of uncertainties were not considered in an adequate manner. Since there is less conservatism in the TCA, it is advisable to ensure compliance with requirements also in service. This is usually referred to as runtime verification. Although this is not a part of the development process, it can be used as one means to increase confidence in the estimated safety and hence might be essential to obtain certification using this novel probabilistic development approach. This is especially true if operation is tightly coupled to safety like for UAV, where the area of application strongly influences criticality and hence the required level of safety and availability. Furthermore, uncertainties assumed for development and verification can be updated using measurements from in-service operation. Although not further discussed in this thesis, this a-posteriori analysis can additionally increase the confidence in the estimated safety margins [Wan+14, Dre17].

The different steps of the TCA rely on similar models, however with increasing level of detail with development progress. Since adequate models are essential for the TCA, special attention is given to them in the chapters describing the individual steps.

3.5 Influence on Today's Development Process

Important for the correct interpretation of the TCA is the fact that the well established top-down development process accoring to ARP-4754A is kept. This also includes all the related standards, especially the safety assessment process (ARP-4761 [SAE96]) as well as the detailed guidelines on development assurance and considerations for electronic hardware, software, and

avionics design (DO-254 [RTC00], DO-178 [RTC11a], DO-297 [RTC05]). Following these standards in the future is highly recommended since they are the foundation for successful certification.
Hence, the TCA must be seen as addition to the current processes, where a more reasonable, physically motivated approach is used to obtain safety-driven lower-level requirements opposed to today's rather conservative, experience-based requirements catalogs issued by certification authorities. The shift towards probabilistic requirements comes along with some more complex verification routines, which however does not change the process itself but only requires additional means for verification.
To highlight the major differences, development using the conventional approach and the TCA are compared next.

3.5.1 Comparison with Total Capability Approach

Major similarities and differences during the different development phases are shown in figure 3.5. At some points of the comparison, the example of developing a flight control system for a fixed wing aircraft is used to enrich explanations with practical examples.
The development starts with the identification of aircraft functions. Safety assessment of these functions results in related top-level safety requirements. These steps are similar for both approaches and lie the foundation for the subsequent development. The main difference arises in the way how lower-level safety-driven functional and performance requirements are derived, to answer the question "*how to specify systems and functions to fulfill top-level safety requirements*". For the conventional approach, there are several paragraphs in the certification specifications for large transport aircraft CS-25 [Eur17b] that influences the design of a flight control system. Requirements on controllability, maneuverability and stability as well as on hardware are given in this standard. Although the requirements are quite specific regarding to "*how a function must look like*", not many performance requirements are given to answer the question "*how well must a function perform*". Due to missing additional civil guidance material, one often fall back to military standards, like – for the given example – the SAE standard AS94900 [SAE07]. It gives requirements on nominal closed-loop behavior for commands and certain disturbances. Although this is a military standard, it is considered to result in a safe flight behavior. In comparison, the TCA might also partly rely on today's certification specifications where reasonable. This is for example the case for standardization of the interaction between the flight control system and the pilots, which is essential to ensure a safe operation. However, instead of using conservative, experience-based performance requirements from military standards, a direct link between the top-level safety requirements and safety-driven lower-level requirements is established by a physically motivated, model-based requirements breakdown, taking into account all kind of uncertainties, disturbances and failures. By that, no explicit requirements for disturbance behavior are required since they are implicitly covered by the integrated consideration of all uncertainties. Furthermore, while standardized requirements follow the "*one system for one function*" idea, the TCA allows

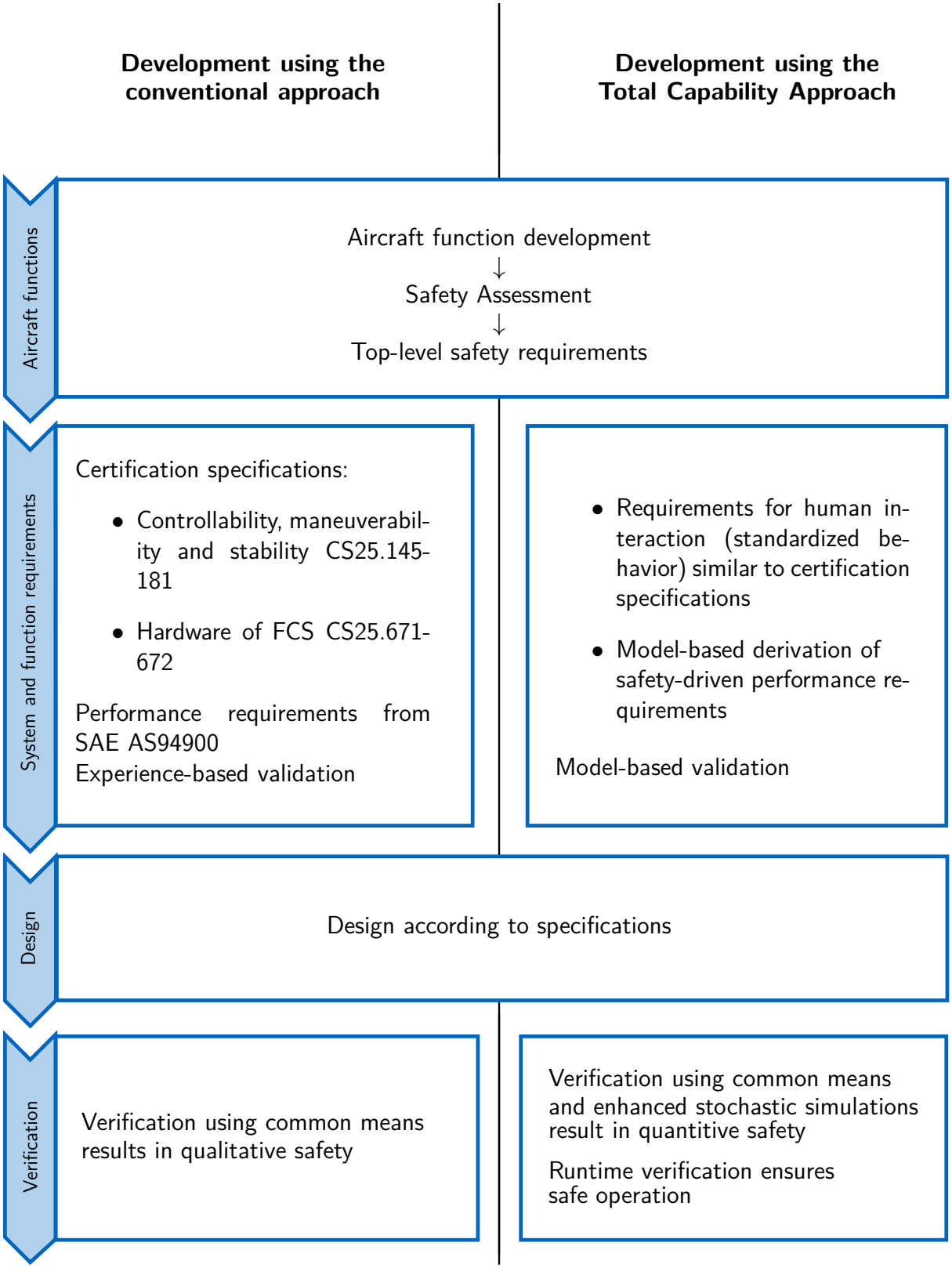


Figure 3.5: *Comparison of conventional approach and TCA using the example of developing a flight control system (simplified)*

an integrated consideration of the total flight control system performance, composed in this example by the control and navigation error.

The availability of models during that early development phase also allows to validate the set of requirements by simulation (e.g. by flight simulator experiments), to ensure that the specified flight control system behaves as desired. This is a major benefit of the TCA.

The design process is similar between the two methods – a flight control system is designed according to specifications, so that the imposed requirements are fulfilled. However, since for the TCA the requirements do not prescribe design solutions as it is often the case for conventional certification specifications, this allows for application of novel control methods like adaptive control. Furthermore, the integrated consideration of all components allows for a dynamic trade-off between the control and navigation performance, which usually results in a system with higher availability compared to the individual specification and design of components. Additionally, the physical link between implementation and top-level safety requirements allows for optimization of safety and availability in the presence of uncertainties and disturbances.

Eventually, the implemented system is verified against the requirements. While the task is the same for both approaches, the result is different: While for the conventional approach, the compliance with requirements only qualitatively ensure safety of the implemented flight control system, the application of enhanced stochastic methods in the scope of the TCA results in statements for the actual risk of failure conditions and hence gives a quantitative feedback about safety of the implemented function. Since this statement heavily relies on the assumptions made for uncertainties and disturbances, it is recommended to evaluate conform behavior of the flight control system also during daily operation, e.g. by online monitoring.

Note that the given comparison is very simplified and many steps that are essential for certification were omitted, since they are mostly similar between the two approaches. Furthermore, for the specific example of developing a flight control system for conventional applications, the experience-based requirements from certification specifications usually serve well for this purpose. Quantitative safety is often not required, since safety is established by adequate, prescribed fallback solutions provided to the pilots. However, for future applications, pilots might not be available as fall-back, which results in higher demands put on the (automatic) flight control system, where the TCA can act out its full potential.

The TCA shift the responsibility for safety from the certification authorities to the design organizations, since safety is no longer established by design solutions prescribed by the authority and is supposed to be safe, but by design solutions developed by the design organization, which then must also provide evidence that the suggested design is safe. However, this must not count against the TCA, since already today there are efforts of the certification authorities to shift the responsibility to the design organizations, see for example the latest revision of the certification specifications for small aircraft CS-23 [Eur17d]. Hence, the TCA must be seen as pioneering solution to tackle future challenges in aircraft development and certification.

3.6 Main Challenges

The TCA promises many benefits. However, several challenges must be met to enable successful implementation. The main challenges are presented below. Detailed discussions and possible solutions are presented in the referred chapters.

- **Model-based approach**: The success of the TCA stands and falls with the availability and quality of the used models. This does not only include models for the dynamics of the involved components, but also for the environment, disturbances, and uncertainties of all contributing components. The quality and extend of the models certainly increase with development progress: At the beginning, usually fewer uncertainties are known, no implementations of the envisaged functions are available and also the environment might not yet be known in detail. However, at the latest for verification, sophisticated models are available that are well verified against reality. Required properties of the models usable for the individual development phases are given in the chapters on requirements derivation and verification.

- **Requirements breakdown**: The model-based derivation of safety-driven requirements is a big challenge: Although no implementation is available at early development phases and also no assumptions must be made thereon, the behavior must be simulated to an extend that allows to make statements on the performance required to fulfill probabilistic top-level requirements. Usually, the design space is large and there are many degrees of freedom, while the probabilities related to critical failure conditions are very low. Both circumstances usually increase the computational effort. The challenges related to the requirements breakdown in the context of the TCA are addressed in chapter 4, where also one possible solution is presented.

- **Confidence in novel method**: The requirements, functions and implementations developed following the TCA are only as good as the models used for and assumptions made during development. Runtime verification is a powerful mean to increase confidence in the developed system and to ensure that models and assumptions were correct and operation is safe. Chapter 6 highlights the benefits of runtime monitoring and provides a model-based online monitoring algorithm to prove compliance for one common type of requirements.

- **Stochastic analysis**: Safety requirements are related to admissible probabilities for occurrence of individual failure conditions. Dependent on criticality, the probability can be very low, see table 3.2. The computational challenge can be encountered by enhanced stochastic methods described in chapter 2. However, the adequate application of these methods and the interpretation of results is often not trivial. Furthermore, it is important for any analysis of uncertain quantities that each estimation is supported by an according accuracy measure that gives the level of confidence one can have in the estimation. This is why the correct application and interpretation of the results is discussed for each step

of the TCA where these methods are applied, especially for requirements derivation and verification.

4

Requirements Derivation and Validation

The core of this chapter is the introduction of a novel method for derivation of safety-driven lower-level requirements based on probabilistic top-level requirements. The foundation for this is model-based requirements engineering, which is described first, together with a literature study on past and current efforts in this field. The two subsequent sections evaluate the challenges of model-based break down of probabilistic top level requirements and describe a method, which is developed taking into account the usually very low probabilities of top level safety requirements. Finally, it is shown how the proposed model-based approach can be utilized to enhance the validation of requirements.

4.1 Model-Based Requirements Engineering

4.1.1 Introduction

Today, using models and simulations is common in many disciplines, e.g. in engineering, physics, biology and medicine. The triumph of models already started with the development of the first high-level programming language *FORTRAN* in 1956 [IBM56, Ste12]. With increasing computational performance, the problems to be solved by models also became more complex. In the early nineties, efforts began to use models to enhance the whole life cycle of a product, from specification to disposal [Wym93]. One aspect thereof, which is especially treated in this chapter, is the model-based requirements engineering, which became more prominent in the last few years, also boosted by the introduction of the standardized general-purpose modeling language SysML in 2006 [OMG07].

In general, model-based requirements engineering follows the idea that models are used to support specification, validation, analysis, design and verification of systems. This is motivated by moving from a document-centric towards a model-centric approach, where artifacts required to prove correct development of a product are mainly generated automatically based on models instead of manually. This shift from informal documents to digital models comes along with the advantage that one is able to better understand the impact of design changes on the overall system and can, therefore, also identify shortcomings in a specified product earlier. This is

enabled by the availability of executable models during early development phases, which usually leads to lower costs for corrections.
The main driver of model-based requirements engineering is the software development branch, where formalization of the former informal development process is simpler due to the high level of standardization available in software engineering. Only during recent years, the benefits of model-based software development were recognized by the aviation industry and standardization bodies. Model-based software development for aerospace applications is enabled by the latest revision of the guidelines for "*Software Considerations in Airborne Systems and Equipment Certification*" DO-178C [RTC11a]. The RTCA standard DO-331 is a model-based development and verification supplement to DO-178C and DO-278A, which describes in detail how models can be used for development of airborne software [RTC11b].
Although the TCA focuses on system development, the standards given for model-based software development provide a comprehensive framework for a general model-based development process. The foundation for consistent model-based requirements engineering is requirements capture and formalization.

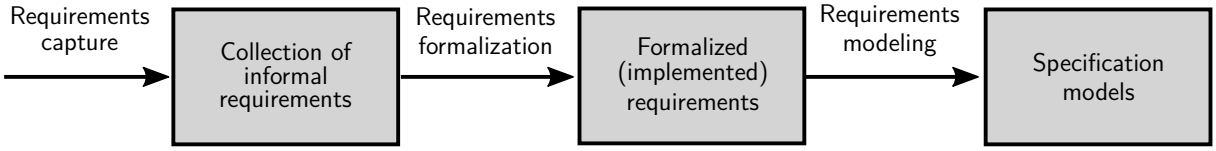


Figure 4.1: *Model-based requirements engineering – capture and formalization*

Figure 4.1 shows the initial steps of model-based requirements engineering. The output of requirements capture is a collection of informal, usually textual requirements. Subsequent formalization and modeling leads to a machine-processable, formal requirement with corresponding behavior model. These essential steps for consistent requirements engineering are addressed subsequently.

4.1.2 Requirements Capture

Requirements capture is the process of collecting all requirements that are necessary to fully describe the intended functionality of a product. The capture process usually starts with the identification of stakeholders and their requirements. In principle, stakeholder is "*anyone who has significant power over, influence of, or interest in our system*" [Lop15, p. 259]. While the most decisive stakeholder groups are customers and certification authorities, further stakeholder can arise by considering the whole product life cycle from development via operation up to disposal. This could, for example, lead to further top-level requirements by users and operators. The process of capturing top-level requirements is usually manual work, while certainly experience from past projects and standards facilitate this task. Referring to the aircraft development process in figure 3.2, the capture of top-level requirements match with the identification of aircraft functions, according functional interfaces and safety requirements. With

increasing development progress, further decisions are made, like the selection of an architecture and allocation of aircraft functions to systems and items. This comes along with further requirements that must be captured.
Requirements can be captured in different formats, e.g. in text or graphics format. Regardless of the format, the captured set of requirements must be consistent and lower-level requirements must be traceable to top-level requirements or be allocated to decisions made during the development process (denoted as derived requirements). To capture and organize requirements and ease traceability, usually requirements management tools are used, e.g. Polarion® Requirements™ [Sie17] or IBM® Rational® DOORS® [IBM17].
The TCA especially focuses on safety-driven performance requirements, which are a subset of lower-level requirements. Today, such requirements are often taken from guidance material provided by legal entities or standardization bodies. Traceability to top-level safety requirements is then ensured by the use of these accepted standards, which are expected to provide an adequate level of safety for the developed product. There are two major differences during the capture process when applying the TCA:

- **Probabilistic requirements**: Top-level requirements are still defined by classical metrics, however, they are linked to probabilities: Today, aircraft functions are defined in a deterministic manner. Although safety requirements are derived from a safety assessment of top-level functions, which results in admissible probabilities of failures corresponding to the criticality of the possible failure conditions, this classification is only used to determine the reliability the implementation must have (see chapter 3.2). When applying the TCA, the level of criticality is also applied to the function itself and resulting lower-level requirements, i.e. the respective function must not cause the failure conditions determined in the safety assessment with a probability higher than acceptable. Although this change seems to be subtle, it enables the physically motivated derivation and especially quantification of lower level requirements.

- **Quantification of lower-level requirements**: Metrics for lower-level requirements are still conventional, however their quantification is no longer obtained from accepted standards but derived from top-level functions and their criticality. This leads to safety-driven performance requirements that are tailored to the actual product and operation. Hence, the resulting requirements are usually less restrictive than the quantification from standards which have to take into account a wide variety of different aircraft topologies and operations.

An addition to the quantification step is the introduction of different performance levels:

- **Adequate Performance** provides quantified metrics for acceptable behavior at verification and hence describes the minimum performance required to ensure safety. If the integrated system meets the "adequate performance limits" during verification, the design is acceptable.

- **Desired Performance** provides quantified metrics for the design objectives, i.e. every time direct design goals are required, the "desired performance limits" are to be taken.

To result in consistent specifications according to these definitions, desired performance must always be a subset of adequate performance and hence be more strict or equal. The distinction is motivated by the fact that during design, clear design objectives are required. However, it is clear that due to "reality effects", e.g. implementation, tolerances, and uncertainties, these objectives cannot be achieved perfectly. Hence, the adequate performance describes the region, in which the resulting design is still acceptable. Specifically for safety-driven performance requirements considered in this work, this means that although a function must only comply with wide performance limits to ensure safety and hence obtain certification, tighter limits can arise from further stakeholders. This could include technical aspects (e.g. "*What would be the best or most desirable performance?*"), but also non-technical issues (e.g. "*Which performance range could be achieved with limited system cost?*"). Consider for example a requirement on the deceleration performance of an aircraft after landing. If safety in this example would only be motivated by the stop distance of the aircraft, there is only a lower limit for the deceleration capabilities while any higher deceleration would be acceptable from a safety point of view. Due to the lack of specific design targets, the technical solution could be anything between a simple breaking system that exactly satisfies the adequate performance requirement and an extreme solution using drag parachutes and brake rockets. In this case, the desired requirement would also provide an upper limit, motivated by further considerations of adequacy, cost, passenger comfort, etc.

4.1.3 Requirements Formalization

Requirements formalization is the transfer from an informal to a formal requirement. It establishes a link between text or diagram based requirements and the corresponding models, which can be automatically evaluated. Formalized specifications are formal in the sense that they have a syntax and they can be used to obtain information about the compliance of requirements.

In the first place, the formalized requirement can only be used to check whether a designed system represented by models complies with requirements, i.e. it is a function, that is applied to models and which results in a statement about compliance with or violation of the associated requirement.

In the context of formalization, the model-based development and verification supplement for software DO-331 introduces "specification models", which are defined as follows: "*A Specification Model represents high-level requirements that provide an abstract representation of functional, performance, interface, or safety characteristics of software components. The Specification Model should express these characteristics unambiguously to support an understanding of the software functionality. It should only contain details that contributes to this understanding and does not prescribe a specific software implementation or architecture except*

for exceptional cases of justified design constraints.". This concept can also be extended to systems and function design: The desired behavior of a system or function given by requirements can be modeled leading to behavior models without applying knowledge about actual implementations. Note that this must not be confused with the formalized requirement: While specification models reproduce the function specified by requirements, formalized requirements only result in a statement about compliance. Still, specification models can be utilized for requirements formalization by comparing the actual implementation with the reference behavior given by specification models and thereof derive a statement about compliance.
The concept of formalized requirements and specification models can be applied to different levels of the development process from aircraft to item level. In this case, the specification model of one level represents the requirement for the lower level. For example, suppose that the higher-level requirement specifies that an aircraft has to be able to maintain altitude with a certain accuracy in the presence of uncertainties and disturbances. The formalized requirement would be, in this example, a simple comparison of compliance with predefined altitude bounds. The specification model would reproduce the desired behavior in the presence of uncertainties and disturbances that just fulfills the requirement. Eventually, this specification model becomes the requirement for the (lower level) actual control design, where then the objective is that the closed-loop behavior with a controller must not be worse than the limit behavior given by the specification model of the higher level.
The actual process of formalization highly depends on the content of the requirement. For common requirement types, e.g. which only prescribe thresholds or admissible intervals of observable values, formalized requirement templates can be used. For more complex requirements, usually manual programming is required. This also holds for specification models, which – in contrast to usually simple comparisons done by formalized requirements – highly depend on the target system and environment. Current research in this field try to automate the formalization process by identification of catchwords in informal requirements and then selecting an appropriate template from a library [PH12, WL13, ASA17]. However, these approaches are still limited to specific types of requirements.
Specification models describe the behavior of functions and usually depend on other functions, the environment and inherent uncertainties and disturbances, which is discussed next.

4.1.4 Models and Environment

In the context of model-based requirements derivation and validation, a model is an abstract representation of relevant aspects of a system or function that is used for analysis and simulation, to obtain insight and understanding of the specified functions and their interactions among themselves and with the environment. With respect to the system or function to be developed, the model must not include any information about the actual (planned) implementation except for justified design constraints. However, the models used for requirements derivation and validation do not only include the specification models, but also models for the environment and its interfaces. For example, when the objective is to develop a flight control

system, usually models of the used aircraft, environment and off-the-shelf components to be used are readily available and their influence on the system and functions to be specified must be taken into account. In the subsequent descriptions, all components, which could have an influence on the system to be developed but which cannot be influenced or modified by the actual development, are summarized as "system environment". Figure 4.2 shows the scheme of the developed simulation environment for requirements derivation.

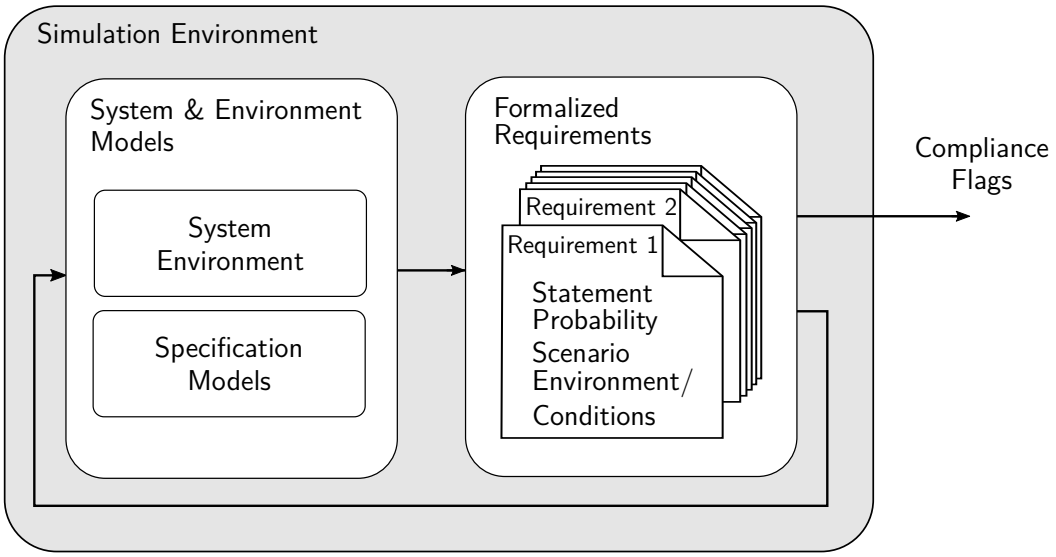


Figure 4.2: *Simulation environment for model-based requirements derivation*

According to definition, specification models should be as simple as possible and only as complex as necessary. On the other hand, the system environment should be modeled as accurately as possible using all knowledge available at the early development stage. That applies all the more to requirements derivation in the context of the TCA, where this simulation environment is used to derive safety-driven lower level requirements based on probabilistic top-level requirements.

Adequate simulation models are required to allow for reasonable derivation of safety-driven requirements based on probabilistic top-level requirements. Models suitable for that task must provide the following properties and functionalities:

- **Representation of environment**: The models must correctly represent the system environment that cannot be influenced or modified. Certain dynamics are inherent to the principal system and cannot be altered by any inputs and hence cannot be influenced by any function to be specified. This is for example the case for system kinematics and external disturbances. Also dynamics of subsystems that are not in the reach of the envisaged development, e.g. off-the-shelf components, must be correctly modeled, including uncertainties.

- **Tunable specification models**: The models must provide means to adjust the relevant dynamics part of the model that can be influenced. This is usually ensured by adequate specification models.

- **Effects of uncertainties and disturbances**: The models must reproduce the effects of disturbances and uncertainties in a manner that allows for a stochastic evaluation of their influences on the system response. Often, adequate disturbance models and admissible system responses are defined by certification authorities or can be obtained from experiments.
- **Interfaces**: The models must provide interfaces for relevant inputs, outputs and for variation of uncertain parameters. Relevant signals in this context are signals that are required for requirements evaluation, i.e. for execution of formalized requirements.

It is necessary to take all kinds of uncertainties and disturbances into account to result in valid lower-level requirements. Major sources of uncertainties are:

- **Model uncertainties**: For model uncertainties due to simplifications or lack of knowledge, often an upper threshold can be specified. The effects of model uncertainties must be adequately represented using a sufficient number of uncertain parameters.
- **Initial assumptions**: When there is initially only the knowledge that a parameter is uncertain but no quantification is available, assumptions based on experience, specification sheets, etc. must be made.
- **Parameter identification**: Models are verified using experiments. The model parameters are tuned so that the errors between model and experiment are minimized. The initially guessed uncertainties are refined using e.g. Bayesian inference, leading to realistic probability density functions for uncertain parameters (see e.g. [Che+13]).
- **Stochastic uncertainties**: Stochastic processes are used for modeling of time-dependent random phenomena. Such processes describe the sequence of mostly correlated uncertain parameters. Stochastic models are either given by standards or must be established by experiments.

Although models are usually problem-dependent and require considerable development effort, this additional effort must not be accounted to the TCA, since models of the system environment are usually already used today during system and function design and optimization. Hence, the effort is only shifted to an earlier development stage.

4.2 Determination of Design Parameter Boundaries

Model-based break-down of requirements in the scope of the TCA goes beyond what has been done with model-based requirements engineering up to now. Today, formalized requirements and specification models are used to support understanding of the specified system and functions, while safety-driven performance requirements usually do not have a physical link to the top-level safety objectives. Although high-level safety requirements can be formalized, this knowledge is not further used, mainly due to two reasons: First, probabilities related to

top-level safety requirements are usually very low and therefore evaluation is computationally expensive. Second, since this complexity inhibited physically motivated evaluations in the past, prescriptive certification specifications were derived during the last seven decades, which are applicable to conventional aircraft topologies and operations. Since these led to safe aircraft, there was no necessity in the past to change this established procedure. However, in the third chapter the overdue shift in paradigm was motivated.
One major enabler of this shift is a physically motivated derivation of safety-driven lower-level requirements, where compliance with top-level safety requirements is established by a hard link instead of the weak compliance with top-level safety requirements by prescriptive and experience based certification specifications. Following this approach, the result of requirements derivation and validation is no longer limited to the statement that a set of requirements is reasonable to result in a safe system, but it leads to the "best possible" solution, where best is defined concerning safety and availability of the prescribed function under certain conditions.
In the following subsections, the challenges of model-based determination of safety-driven performance requirements is discussed and a new method is described, which can be used to break down top-level probabilistic safety requirements. This novel approach was only enabled during the last years by development of enhanced stochastic algorithms (see chapter 2). The knowledge about the admissible design parameter bounds obtained by the proposed algorithm is the prerequisite for selection of reasonable design parameter intervals, which is discussed in section 4.3.
The algorithm described in this section was first presented and published by the author of this thesis on the EURO GNC 2017 - 4th CEAS Specialist Conference on Guidance, Navigation & Control in April 2017 [LH17].

4.2.1 Problem Description

The idea of model-based break-down of requirements is to use specification models, i.e. models that reproduce the desired closed-loop behavior of a system when no implementation is available yet. A common design task is the determination of an optimal design that minimizes the weighted effects of all requirements. For that, adequate methods exist, e.g. [BS07] for general design optimization and [Kou16] for high-dimensional, stochastic design problems. Opposed to design optimization, here the implementation-independent models should be used to find the complete range of acceptable ranges of lower-level functional behavior for the safety-driven performance requirements that fulfill the probabilistic top-level requirements. For deterministic requirements or when only considering nominal conditions, this break down could be approached with common engineering knowledge, since the deterministic behavior in the absence of uncertainties gives a clear bound between acceptable and unacceptable behavior, i.e. there is no chance that the top-level requirements are violated if the lower-level requirements are specified correctly.
In the presence of uncertainties and disturbances, the break down is no longer trivial, since it requires the additional simultaneous consideration of possible uncertainties and disturbances

that could lead to violation of top-level safety requirements. In reality, there are always uncertainties, e.g. arising from unknown system dynamics, inaccurately known model parameters, and environment. All these uncertainties contribute to the overall system performance and could lead to an – although unlikely, but still possible – violation of top-level requirements. This implies that for a single specified behavior and the according specification model, which is used to simulate the desired lower-level behavior, a very high number of simulations is already required to estimate the failure probability with a sufficiently high accuracy and hence high level of confidence. Consequently, the estimation of admissible bounds for specified behavior within which it is ensured that the top-level safety requirements are not exceeded with a probability higher than a usually very small threshold in the presence of uncertainties, adds an additional layer of computational complexity. This makes the probabilistic model-based derivation virtually impossible when using conventional stochastic algorithms already for a low number of design degrees of freedom, i.e. a low number of tuning parameters of the specification model for which admissible intervals must be found.
An additional challenge arises from the fact that the design bounds cannot be exactly determined due to the stochastic nature of the uncertainties, since the bounds can usually not be determined analytically but only by stochastic algorithms. The results of such stochastic approximations change from one to another simulation even if the model is not changed at all. This requires determination of confidence intervals for the estimated quantity within which the true solution is supposed to lie with a certain probability. This must be taken into account during derivation of admissible design bounds to prevent misinterpretations of the resulting acceptable design domains.
Recall that the objective is not to find "new" metrics for lower-level requirements, but to find physically motivated and hence more reasonable quantifications of common metrics. In subsequent explanations, the modifiable parameters, which can be influenced by the designer to fulfill the top-level safety requirements, are denoted as design parameters. With increasing complexity of the requirement and hence of the specification model, the degrees of freedom and hence number of design parameters increase.

4.2.2 Intuitive Approach using Rastering

The probability of violating a requirement for a given design parameter combination can be determined by stochastic simulations for an environment subject to uncertainties and disturbances. However, for determination of acceptable ranges of design parameter combinations during requirements derivation, it is required to find all combinations of design parameters for which the top-level safety requirements are fulfilled. More specific, the objective is to find the design parameter bounds usually in the form of hypersurfaces, which divide the design parameter space into acceptable and unacceptable designs. At each bound, for at least one requirement, the actual probability of exceedance is equal to the allowed failure probability, while the probabilities of exceeding the other requirements are smaller than the admissible threshold.

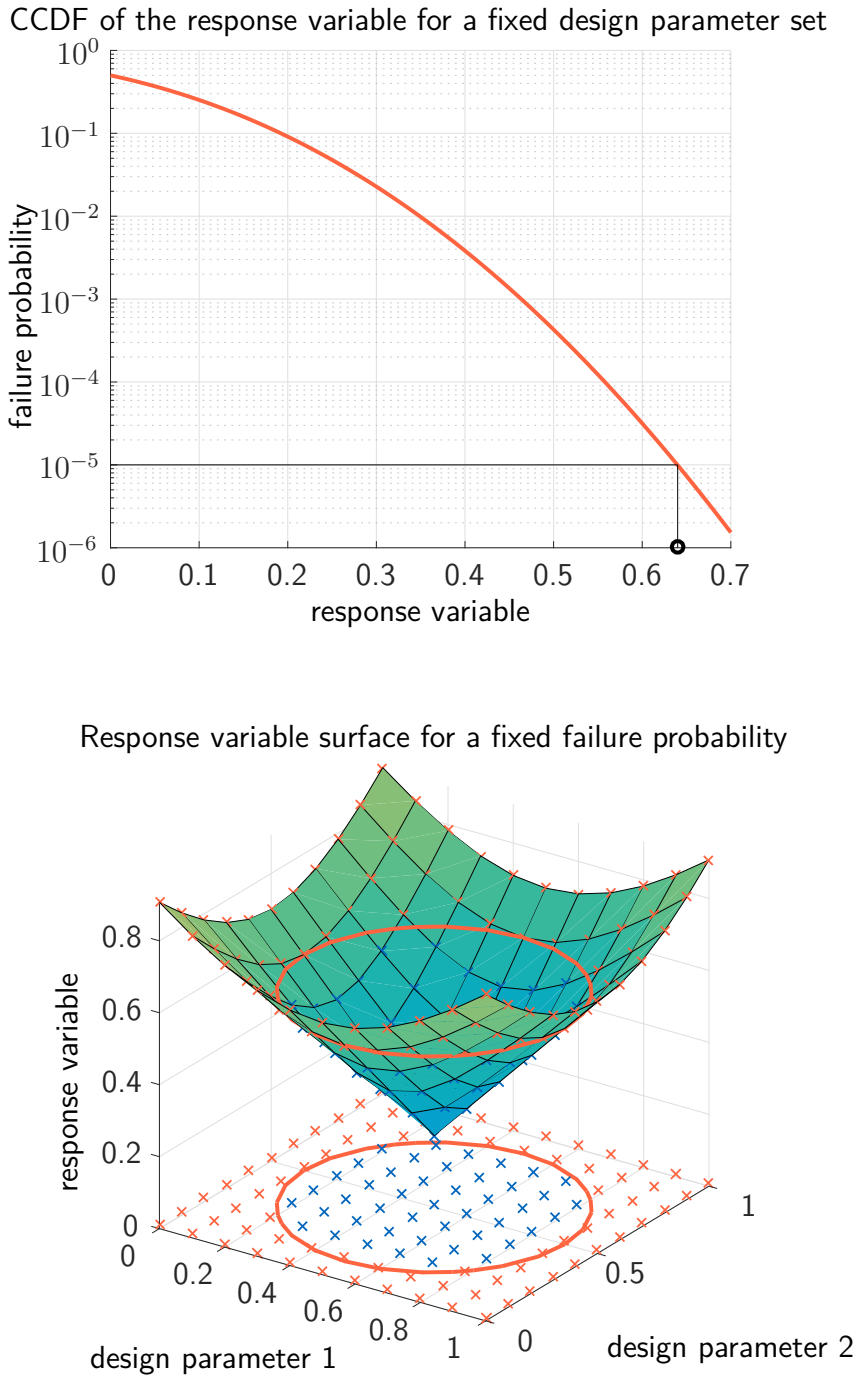


Figure 4.3: *Determination of design parameter bounds using grid samples*

Figure 4.3 gives a generic example for the case with two design parameters. A requirement is quantified by a scalar response variable (see chapter 2.2). For a fixed combination of design parameters, the threshold for the response variable is determined that is not exceeded with a probability higher than acceptable. This is shown in the upper plot with a failure probability of $P_F = 10^{-5}$ in the considered example. The CCDF required for this evaluation is obtained by (enhanced) stochastic analysis. This determination is repeated for all combinations of design parameters obtained by rastering over the whole design parameter space. The result is a response variable surface with constant failure probability, where response values lying above this surface are less likely to happen than the admissible failure probability. This surface is

shown in the lower plot. A probabilistic requirement prescribes the admissible probability that a specific value of the response variable is exceeded. The definition and corresponding critical value of the response variable depends on the respective failure condition. For the example shown, the value of the system response must not exceed 0.6 with a probability higher than 10^{-5}. Samples which comply with this requirement are drawn in blue, violations in orange. By interpolation between design parameter combinations with acceptable and non-acceptable response, i.e. by intersection of the response surface for constant failure probability with a plane of constant response variable, an approximation of the design parameter boundary for this requirement is obtained, which is a circle in the example. Unfortunately, this is the most inefficient way to obtain the design parameter bounds, since the number of samples required grows exponentially with the number of design parameters. Hence, rastering only makes sense if the boundary must only be evaluated once and sufficient computational power and time is available. Furthermore, the CCDF is actually not exactly known since it is obtained by stochastic simulations (see section 2.6.4). This is shown in figure 4.4. The uncertainty of the CCDF results in an uncertain estimate of the response variable that is not exceeded with a specific probability. Hence, also the response surface that is not exceeded with a certain probability, which is actually a deterministic surface, is not exactly known. This approximation error results in an uncertain design parameter boundary. Zero variability of the design parameter boundary can only be obtained for the theoretical case of an infinitely large sample size used for stochastic analysis.

4.2.3 Efficient Determination of Uncertain Boundaries

The idea of the proposed algorithm is to use a pseudo-inverse algorithm where – starting from a single point on the design parameter boundary – the whole boundary is evaluated in a more efficient way by walking along this boundary using a gradient method. This promises to require a lower number of samples, see figure 4.5. Although the general idea of walking along the boundary is simple, there are several challenges related to the application of this idea to the estimation of design boundaries, which can only be approximated. Major challenges when using gradient-based methods in this context are

- Finding an adequate initial point on the parameter boundary
- Calculation of the gradient, when the functional value is not accurately known due to estimation errors
- Efficient generation of samples along the uncertain design boundary

These points are discussed in detail in the following paragraphs.

Finding a Single Point on the Design Parameter Boundary

Without initial knowledge about the design parameter boundary, it can be challenging to efficiently find a starting point on this boundary. There are two principle methods that can

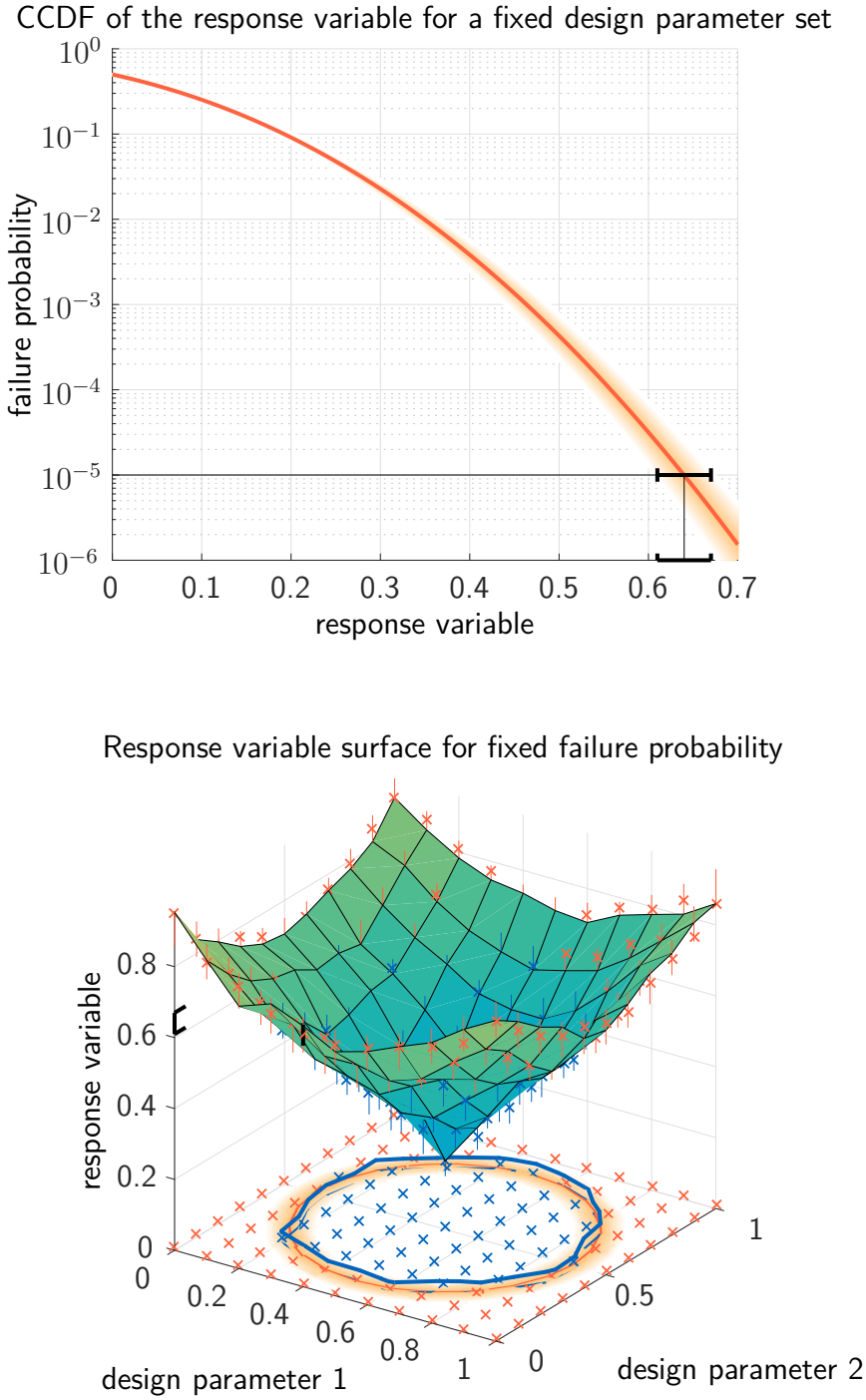


Figure 4.4: *Determination of uncertain design parameter bounds using grid samples*

be used to identify a starting point for the gradient-based algorithm, if no foreknowledge is available.

The first method starts from a design parameter combination in the midst of the parameter domain. Then, additional samples are generated from the center point by only varying a single design parameter from its minimum to maximum value. If all or no samples comply with the considered requirement, the procedure is repeated with another design parameter until a dimension is found where the variation of a single design parameter leads to a transition between accepted and non-accepted samples. By interpolation between the samples closest to

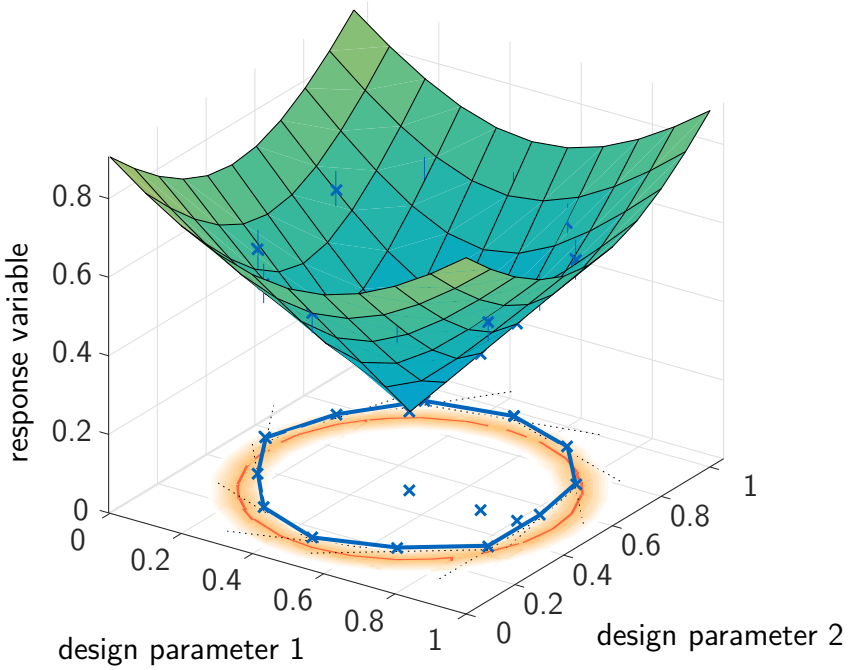


Figure 4.5: *Determination of design parameter bounds using stochastic gradients*

the boundary, an approximation of a single point of the design parameter boundary is obtained. By fixing the other parameters in the midst of the parameter domain, the resulting initial point is still very much in the middle of the parameter domain and hence is a good starting point for the gradient based method.

The second method relies on gradient calculations (see next subsection) and is hence very similar to optimization algorithms that tries to find the fastest way to an extreme value. First, the multi-dimensional gradient of the system response with respect to the design parameters is calculated for an initial design parameter combination in the midst of the parameter domain. Using the difference between the estimated system response value and the target value specified in the requirement, an iterative approximation towards the design parameter boundary can be conducted similar to the Newton–Raphson method [Kel03]. Note that this method could converge to local solutions and hence prevent a transition to the design parameter boundary. Both methods lead to a single point on the design parameter boundary that can be used as seed for the gradient based evaluation of the boundary.

Estimation of the Uncertain Gradient

Calculation of the gradient is a trivial task for a deterministic response. However, since an analytical solution of the probabilistic design problem is in general not available, the system response is only inaccurately known. Depending on the required accuracy of the gradient, usually either a single sided or a double-sided difference quotient is used for approximation of the local derivative. The first only requires one additional function evaluation in addition to the center point and the approximation has an order of $\mathcal{O}(\Delta\lambda)$, where $\Delta\lambda$ denotes the step width. The second requires two function evaluations per dimension but has therefore

an approximation order of $\mathcal{O}(\Delta\lambda^2)$ [Bec13]. The challenge here is the uncertain system response, which is usually obtained by stochastic simulations. Figure 4.6 gives the example for the double-sided difference quotient with uncertain system responses, where λ_i denotes the i-th design parameter and y and $\hat{y}$ the actual but unknown and the estimated system response respectively. The vertical bars represent variance bounds for the individual estimations, which result from stochastic simulations, and the dotted lines indicate upper and lower variance bounds for the estimated difference quotient.

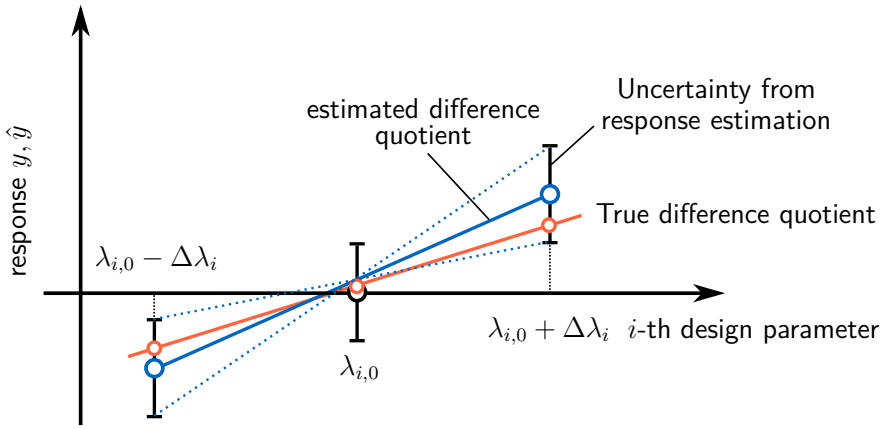


Figure 4.6: *Difference quotient of uncertain functions*

The double-sided difference quotient for the i-th design parameter is estimated by

$$\hat{g}_i(\lambda_{i,0}) = \frac{\hat{y}(\lambda_{i,0} + \Delta\lambda) - \hat{y}(\lambda_{i,0} - \Delta\lambda_i)}{2\Delta\lambda_i} \tag{4.1}$$

for any positive, non-zero step width $\Delta\lambda_i$, where the linear calculation rules for variances are used. The index i is skipped in the following for the convenience of reading. The variance of this quotient, calculated based on the variance of the individual response estimates $\hat{y}$, is

$$\operatorname{var}[\hat{g}(\lambda_0)] = \frac{\operatorname{var}[\hat{y}(\lambda_0 + \Delta\lambda)] - \operatorname{var}[\hat{y}(\lambda_0 - \Delta\lambda)]}{2\Delta\lambda} \tag{4.2}$$

where it is assumed that there is no correlation between $\operatorname{var}[\hat{y}(\lambda_0 + \Delta\lambda)]$ and $\operatorname{var}[\hat{y}(\lambda_0 - \Delta\lambda)]$, which is usually the case since the random samples for estimation of $\hat{y}(\lambda_0 + \Delta\lambda)$ are always independently generated from the samples for estimation of $\hat{y}(\lambda_0 - \Delta\lambda)$. Opposed to the difference quotient for the deterministic case, which converges towards the gradient of $y(\lambda)$ for infinitesimal step widths $\Delta\lambda$, this is not true for the difference quotient of uncertain responses $\hat{y}$. According to equation (4.2), the variance of the gradient becomes infinite for non-zero variances of the estimations $\hat{y}$. A trade-off is required between small step sizes for better approximation of the local response y and larger step sizes for lower impact of estimation errors. Since the step size cannot be arbitrarily increased to reduce the variance of the estimated gradient, the estimates of the response $\hat{y}$ must have an adequate accuracy, i.e. a relatively low variance $\operatorname{var}[\hat{y}(\lambda_0 \pm \Delta\lambda)]$. There exists an optimum step width for this trade-off. Two factors influence the accuracy of the estimated gradient: The stochastic

estimation error and the approximation error due to non-infinitesimally small step widths. The first error can be described by

$$\operatorname{var}\left[\hat{g}\left(\lambda_0\right)\right] = E\left[\left(\hat{g}\left(\lambda_0\right) - g\left(\lambda_0\right)\right)^2\right] = E\left[\left(\hat{g}\left(\lambda_0\right) - y'\left(\lambda_0\right)\right)^2\right] \tag{4.3}$$

where $y'\left(\lambda_0\right)$ is the first derivative of $y\left(\lambda\right)$ with respect to λ evaluated at $\lambda = \lambda_0$, which exactly corresponds to the gradient $g\left(\lambda_0\right)$. Equalizing (4.2) with (4.3), rearrangement and application of the norm gives the following description for the single standard deviation bound of the estimated gradient as a function of the step width $\Delta\lambda$:

$$\left|\hat{g}\left(\lambda_0\right) - y'\left(\lambda_0\right)\right| = \sqrt{\frac{\operatorname{var}\left[\hat{y}\left(\lambda_0 + \Delta\lambda\right)\right] - \operatorname{var}\left[\hat{y}\left(\lambda_0 - \Delta\lambda\right)\right]}{2}}\Delta\lambda^{-0.5} = c_1\Delta\lambda^{-0.5} \tag{4.4}$$

c_1 is a positive constant that only depends on the variance of the response estimator $\hat{y}$. Using stochastic simulations, the estimation $\hat{y}$, the corresponding estimated probability of exceedance $\hat{P}\left(\hat{y}\right)$ and the variance of this expected probability $\operatorname{var}\left[\hat{P}\left(\hat{y}\right)\right]$ are obtained (e.g. by (2.13) for conventional Monte Carlo simulation or (2.114) for Subset simulation). Unfortunately, the required variance of the response for the considered failure probability $\operatorname{var}\left[\hat{y}\left(\hat{P}_F\right)\right]$ does not directly result from simulation, but only the variance of the estimated failure probability for the determined threshold $\operatorname{var}\left[\hat{P}_F\left(\hat{y}\right)\right]$. To approximate the variance of $\hat{y}$, the CCDF is used (see 2.6.1). By using the CCDF as a mapping from $\Delta\hat{P}_F$ to $\Delta\hat{y}$, the variance of $\hat{y}$ can be estimated from the known variance of $\hat{P}_F$, see figure 4.7. This is only an approximative mapping, since the CCDF is an approximation itself obtained by stochastic simulations. However, the results are still useful for approximation of c_1 and hence for quantification of the stochastic estimation error of the gradient $g\left(\lambda_0\right)$.

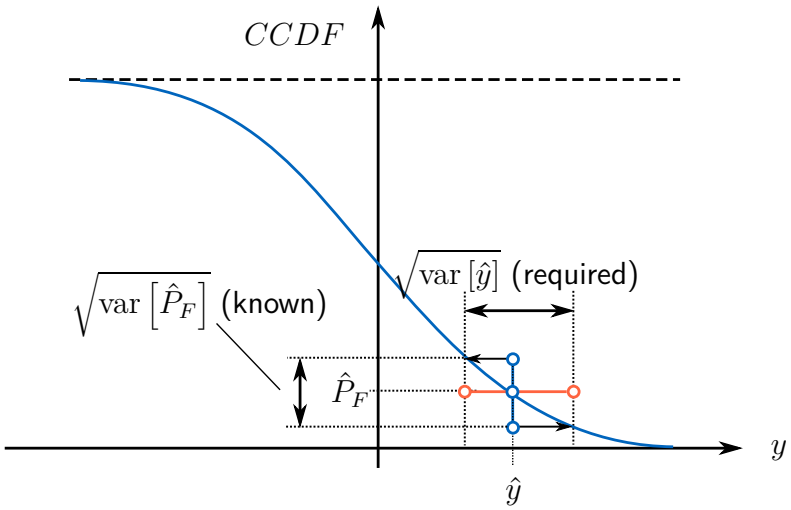


Figure 4.7: *Mapping from variance of estimated probability to variance of response*

For the derivation of the approximation error due to non-infinitesimal step widths, consider the third order *Taylor* approximation of the exact response value (neglecting approximation

errors) for the design parameter variation $\lambda_0 \pm \Delta\lambda$, which is

$$y(\lambda_0 \pm \Delta\lambda) = y(\lambda_0) \pm y'(\lambda_0)\,\Delta\lambda + y''(\lambda_0)\,\frac{\Delta\lambda^2}{2} \pm y'''(\xi)\,\frac{\Delta\lambda^3}{6} \tag{4.5}$$

According to the mean value theorem, this equation is exactly true for at least one value of $\xi \in [\lambda_0 - \Delta\lambda, \lambda_0 + \Delta\lambda]$ [Bro06, p. 405]. The commas in superscript denote the order of derivation with respect to σ. Inserting this into the equation for the double-sided difference quotient gives

$$g(\lambda_0) = \frac{\hat{y}(\lambda_0 + \Delta\lambda) - \hat{y}(\lambda_0 - \Delta\lambda)}{2\Delta\lambda} = y'(\lambda_0) + \frac{y'''(\xi_1) + y'''(\xi_2)}{6}\Delta\lambda^2 \tag{4.6}$$

Hence, the gradient equals to the true derivative plus an additive error proportional to $\Delta\lambda^2$, which also proves the afore made statement that the double-sided difference quotient has an approximation order of $\mathcal{O}(\Delta\lambda^2)$. Rearrangement and application of the norm gives the following expression for the estimation error as function of the step width:

$$|\hat{g}(\lambda_0) - y'(\lambda_0)| = \left|\frac{y'''(\xi_1) + y'''(\xi_2)}{6}\right| \Delta\lambda^2 = c_2\Delta\lambda^2 \tag{4.7}$$

where c_2 is a positive constant that depends only on the curvature of the response function. Figure 4.8 shows the error of the gradient due to the stochastic estimation error (4.4) and the approximation error (4.7) for exemplary c_1 and c_2, both as function of the step width. The influence of the stochastic error can be well quantified from the results of stochastic simulations.

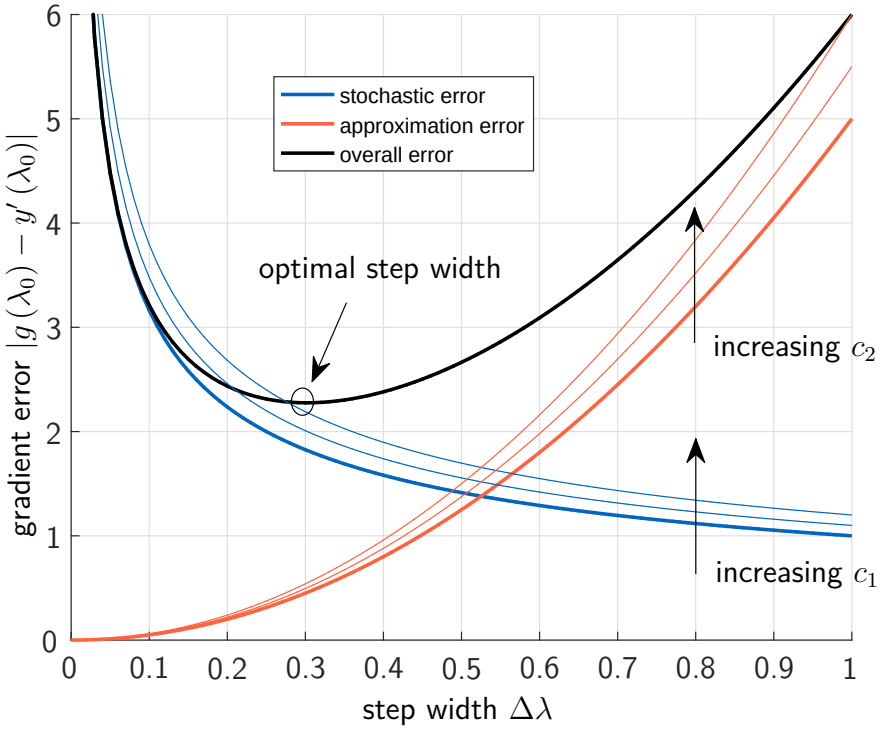


Figure 4.8: *Dependencies of different gradient errors on absolute step width for exemplary values of c_1 and c_2*

With increasing stochastic error, i.e. increasing c_1, the optimal step width increases. However, the optimal step width cannot be calculated in advance, since the shape of the system response $y(\lambda)$ and hence also the third derivative $y'''(\lambda)$, which directly influences the coefficient c_2, is not a priory known, since it is actually the information looked for. Nevertheless, a general statement can be derived from (4.7): The less the curvature of the system response $y(\lambda)$ changes with λ, the bigger is the optimal step width. In the special case that the curvature of the bound is constant, c_2 becomes zero and hence step width can be maximized to the limits of the considered design parameter λ to obtain the best approximation of the local gradient.

Generation of Samples along the Parameter Boundary

In the previous section it was shown how the gradient of an uncertain response value $\hat{y}(\boldsymbol{\lambda})$ with respect to a p-dimensional parameter vector $\boldsymbol{\lambda}$ can be estimated. The response surface for constant probability of exceedance can be estimated by a first order approximation of the hypersurface using the stochastic gradient vector $\boldsymbol{g}(\boldsymbol{\lambda})$ derived in the previous section. Neglecting stochastic uncertainties in a first step, the *Taylor* approximation is given by

$$y(\boldsymbol{\lambda_0} \pm \Delta\boldsymbol{\lambda_S}) = y(\boldsymbol{\lambda_0}) \pm \frac{dy}{d\boldsymbol{\lambda}}(\boldsymbol{\lambda_0})\,\Delta\boldsymbol{\lambda_S} + \sum_{i=1}^{p}\sum_{j=1}^{p}\frac{d^2y}{d\lambda_i d\lambda_j}(\boldsymbol{\zeta})\,\Delta\lambda_{S,i}\Delta\lambda_{S,j} \tag{4.8}$$

where the last term approximates the curvature of the hypersurface by a Jacobi matrix and the approximation is exact for at least one $\boldsymbol{\zeta} \in [-\Delta\boldsymbol{\lambda_S}, \Delta\boldsymbol{\lambda_S}]$ according to the mean value theorem. Note that $\Delta\boldsymbol{\lambda_S}$ is the step width used for generation of new samples along the hypersurface, which is usually different to the step width $\Delta\boldsymbol{\lambda}$ used for estimation of the stochastic gradient. Estimating the Jacobi matrix for the p-dimensional parameter vector using symmetric finite differences requires $2p^2$ evaluations of the response function y which is computationally very expensive. Furthermore, due to the product of two small numbers in the denominator of the Jacobi matrix, the negative influence of stochastic uncertainties highly outweighs the benefits obtained by the higher-order approximation. Hence, only a first order approximation of the hypersurface is considered for generation of new samples along the surface, which has an approximation order of $\mathcal{O}\left(\Delta\lambda_{i,S}^2\right)$:

$$\hat{y}^*(\boldsymbol{\lambda_0} \pm \Delta\boldsymbol{\lambda_S}) = \hat{y}(\boldsymbol{\lambda_0}) \pm \hat{\boldsymbol{g}}^T(\boldsymbol{\lambda_0})\,\Delta\boldsymbol{\lambda_S} \tag{4.9}$$

where $\hat{y}^*$ denotes the linearly extrapolated value of $\hat{y}$ based on the estimated gradient $\hat{\boldsymbol{g}}$. Inserting $\Delta\boldsymbol{\lambda_S} = \boldsymbol{\lambda} - \boldsymbol{\lambda_0}$ into (4.9) and rearranging leads to the point normal form of a hyperplane which ensures $\hat{y}^*(\boldsymbol{\lambda}) - \hat{y}^*(\boldsymbol{\lambda_0}) = 0$, where $\boldsymbol{\lambda_0}$ represents the parameter vector at the initial support point:

$$(\boldsymbol{\lambda} - \boldsymbol{\lambda_0})^T\,\hat{\boldsymbol{g}}(\boldsymbol{\lambda}_0) = 0 \tag{4.10}$$

Equation (4.10) is used to generate new sample points on the hyperplane for $\hat{y}^*(\boldsymbol{\lambda}) = y_{limit} =$ const., which is a local approximation to the actual parameter boundary. If there are p

parameters $\lambda_i, i = 1 \dots p$, the hyperplane has an order of $p-1$, i.e. all but one of the p design parameters can be freely modified (in theory) and the remaining parameter can be calculated as function of the others. Very high variations of the sample point position on the surface could arise from steep gradients even in combination with relatively low step widths of individual design parameters. To prevent the associated numerical problems, it is essential to limit the absolute step width, which is possible by scaling the individual parameter step widths by the respective component of the gradient.

For each newly generated sample point $\boldsymbol{\lambda}^{(j)}, j = 1 \dots m$, the value of the response $\hat{y}\left(\boldsymbol{\lambda}^{(j)}\right)$ is estimated and compared to the approximation $\hat{y}^*\left(\boldsymbol{\lambda}^{(j)}\right)$. If at the m-th step the estimated system response deviates too far from the desired value $y\left(\boldsymbol{\lambda_0}\right) = y_{limit}$, i.e. if $\hat{y}^*\left(\boldsymbol{\lambda}^{(m)}\right) - y\left(\boldsymbol{\lambda_0}\right)$ becomes too large, an updated gradient vector is calculated at the point $\boldsymbol{\lambda} = \boldsymbol{\lambda}^{(m)}$. Afterwards, new samples are generated that fulfills the following equation:

$$\left(\boldsymbol{\lambda}^{(m+1)} - {\boldsymbol{\lambda_0}}^{(m)}\right)^T \hat{\boldsymbol{g}}\left(\boldsymbol{\lambda}^{(m)}\right) = \hat{y}\left(\boldsymbol{\lambda}^{(m)}\right) - y_{limit} \tag{4.11}$$

The value on the right side of the equation in contrast to the null in (4.10) compensates the offset of the m-th point from the considered hypersurface with $y\left(\boldsymbol{\lambda}\right) = y_{limit}$.

The optimal step width $\Delta\boldsymbol{\lambda_S}$ for the steps $\boldsymbol{\lambda}^{(j+1)} = \boldsymbol{\lambda}^{(j)} + \Delta\boldsymbol{\lambda_S}$ for any arbitrary step j is not known at this stage. A reasonable initial step width must be chosen. For physical problems, an initial step width of approximately 10% of the overall parameter width has been found as suitable. To find a rule to dynamically adjust the step width to be close to an optimum, the error components contributing to the overall error are looked at. Inserting equations (4.4) and (4.7) into (4.8) gives

$$\hat{y}\left(\boldsymbol{\lambda_0} \pm \Delta\boldsymbol{\lambda_S}\right) = \hat{y}\left(\boldsymbol{\lambda_0}\right) + \hat{\boldsymbol{g}}\left(\boldsymbol{\lambda_0}\right)\Delta\boldsymbol{\lambda_S} + \boldsymbol{c_1}\Delta\boldsymbol{\lambda}^{-0.5}\Delta\boldsymbol{\lambda_S} + \boldsymbol{c_2}\Delta\boldsymbol{\lambda}^2\Delta\boldsymbol{\lambda_S} + \boldsymbol{c_3}{\Delta\boldsymbol{\lambda_S}}^2 \tag{4.12}$$

where $\boldsymbol{c_3}{\Delta\boldsymbol{\lambda_S}}^2$ represents the error arising from neglecting the curvature of the hypersurface. Inserting (4.9) and rearrangement gives the following error estimation for $\hat{y}^*\left(\boldsymbol{\lambda}\right) - \hat{y}\left(\boldsymbol{\lambda}\right)$, where $\boldsymbol{\lambda} = \boldsymbol{\lambda_0} + \Delta\boldsymbol{\lambda_S}$:

$$\left|\hat{y}^*\left(\boldsymbol{\lambda}\right) - \hat{y}\left(\boldsymbol{\lambda}\right)\right| \leq \left|\boldsymbol{c_1}\Delta\boldsymbol{\lambda}^{-0.5}\Delta\boldsymbol{\lambda_S}\right| + \left|\boldsymbol{c_2}\Delta\boldsymbol{\lambda}^2\Delta\boldsymbol{\lambda_S}\right| + \left|\boldsymbol{c_3}{\Delta\boldsymbol{\lambda_S}}^2\right| \tag{4.13}$$

Figure 4.9 depicts the effects of c_1 and c_2, i.e. stochastic estimation error and approximation error, on the response error $\left|\hat{y}^*\left(\boldsymbol{\lambda}\right) - \hat{y}\left(\boldsymbol{\lambda}\right)\right|$, exemplarily for the scalar case with same step width for gradient calculation and hypersurface approximation $\Delta\boldsymbol{\lambda} = \Delta\boldsymbol{\lambda_S}$. For the general case where $\Delta\boldsymbol{\lambda} = k\Delta\boldsymbol{\lambda_S}$, the shapes of the curves are similar, only the scaling coefficients $\boldsymbol{c_1}$, $\boldsymbol{c_2}$, and $\boldsymbol{c_3}$ becomes $k\boldsymbol{c_1}$, $k\boldsymbol{c_2}$, and $k^2\boldsymbol{c_3}$ respectively. It can be seen that there is no optimal step with that minimizes the estimated response error. This is obvious since for very small step widths also the estimation error is very low and for the limit case that $\boldsymbol{\lambda} = \boldsymbol{\lambda_0}$ and hence $\Delta\boldsymbol{\lambda} = \boldsymbol{0}$, the error is zero since the new value is similar to the value of the support point of the approximation. However, still for small step width the stochastic error

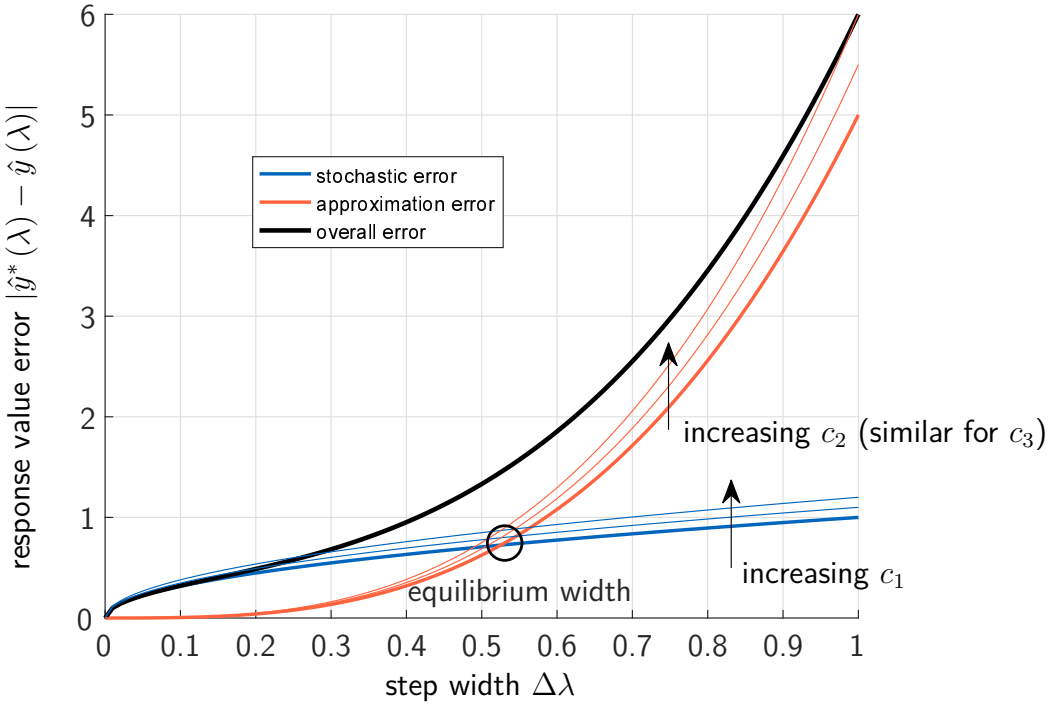


Figure 4.9: *Dependencies of estimation error of the response value on step width (depicted for the scalar case)*

is the major contributor to the approximation error of the response plane while for large step widths the effect of planar approximation is determining. The error arising from the first order approximation of the hypersurface $k^2\boldsymbol{c_3}\Delta\boldsymbol{\lambda}^2$ behaves very similarly to the approximation error of the gradient $k\boldsymbol{c_2}\Delta\boldsymbol{\lambda}^3$, i.e. it starts at zero and only slowly grows with increasing step width while the estimation error $k\boldsymbol{c_1}\Delta\boldsymbol{\lambda}^{-0.5}$ grows faster for smaller step width and slower for higher step width. This knowledge can be utilized to dynamically adapt the step width during generation of new parameter samples along the parameter boundary. For further description, the step width for which the stochastic error is equal to the approximation error is referred to as equilibrium width (see figure 4.9). Up until now, only a qualitative criterion was given for the necessity of calculating a new gradient vector if the response error becomes too high. Now, a quantitative statement is derived: Step widths up to the equilibrium width are considered to be acceptable for the approximation of the hypersurface since this error size would also occur for any other non-gradient method due to the stochastic nature of $\hat{y}$. The equilibrium width cannot be determined a-priori since the influence of the approximation errors ($\boldsymbol{c_2}$ and $\boldsymbol{c_3}$) cannot be quantified. However, from figure 4.9 it can be seen that for step widths below the equilibrium width mainly the estimation error contributes to the overall estimation error. The single standard deviation bound of the estimation error can be obtained using (4.4). The error arising from linear extrapolation $|\hat{y}^*(\boldsymbol{\lambda}) - \hat{y}(\boldsymbol{\lambda})|$ is calculated for each step. If the actual error is lower than two times the estimation error, it is very likely that the current step width is below the equilibrium width or that the approximation error compensates the stochastic error due to different signs, which is also acceptable. On the other hand, if the true error is much bigger than the estimation error, the current step width is probably above the equilibrium

width. While in the first case the step width can be increased to enable faster evaluation of the hypersurface, in the second case the step width must be reduced to ensure an adequate estimation of the hypersurface. Note that the stochastic error is Gaussian distributed (see section 2.3.3) and $n\boldsymbol{c_1}\Delta\boldsymbol{\lambda}^2$ is a single standard deviation bound, i.e. if only this error would be decisive, approximately 68% of all samples would lie within this bound while still 32% of all samples would lie outside. To prevent premature reduction of the step width due to exceedance, either double standard deviation bounds $2n\boldsymbol{c_1}\Delta\boldsymbol{\lambda}^2$ can be used as reference, or the average of multiple independent samples can be considered.

To allow quantification of the performance increase by walking along the boundary using gradients, it is assumed that in average each parameter direction is evaluated by N_{1dim} samples. At worst, a new gradient vector must be calculated for each evaluated point on the design parameter boundary, leading to a required number of function evaluations of

$$N = 2pN_{1dim}^{p-1} + N_{1dim}^{p-1} = (2p+1)\,N_{1dim}^{p-1} \tag{4.14}$$

where N_{1dim} is the number of samples per uncertain parameter direction and p is the number of design parameters. The factor $2p$ account for the double sided gradient calculation, i.e. two evaluations are required for each of the N_{1dim} sample points. The factor one account for the evaluation of the response $\hat{y}$ at each point on the approximated hypersurface. Dependent on the number of samples per uncertain parameter direction, there is a maximum of one order gain in performance compared to simple rastering of the whole parameter space, which is intuitive since by using the described algorithm samples are generated along a grid on a $(p-1)$-dimensional hypersurface while in the reference method a grid over the whole p-dimensional space is evaluated. Higher increase in performance can be obtained if the effects of different design parameters on the response are (almost) not correlated. This enables a sparse grid of samples. Figure 4.10 shows exemplarily how a sparse grid of samples is generated for the case of three design parameters, which corresponds to two parameters λ_1 and λ_2 that can be freely chosen and one that depends on the other two parameters $\lambda_3 = fctn\,(\lambda_1, \lambda_2)$ to fulfill (4.9).

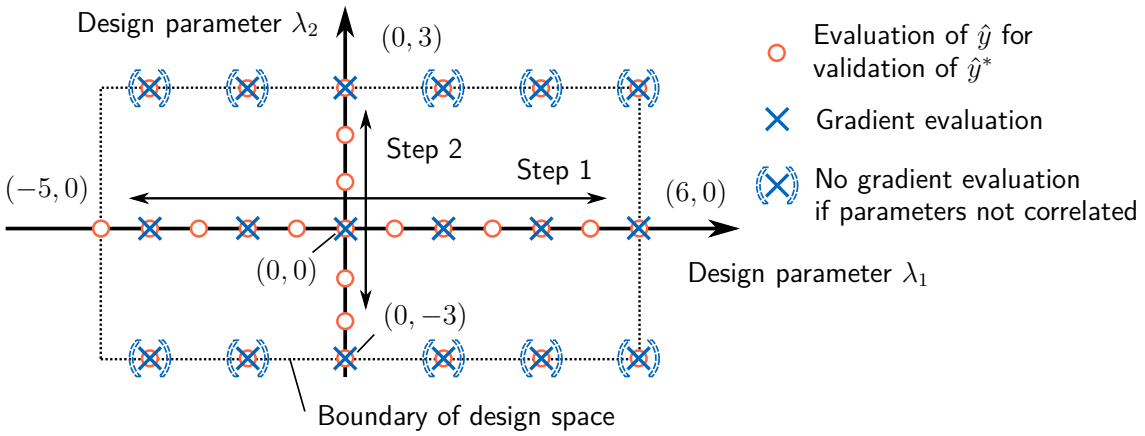


Figure 4.10: *Distribution of sample points for validation and gradient calculation*

Circles denote evaluations of $\hat{y}(\boldsymbol{\lambda})$ for validation of $\hat{y}^*(\boldsymbol{\lambda})$, crosses denote points where the gradient vector with respect to design parameters is recalculated to correct too high errors of the response value $\hat{y}^*(\boldsymbol{\lambda}) - \hat{y}(\boldsymbol{\lambda})$. First, the design parameter boundary is evaluated along one parameter dimension, in the shown example in direction of design parameter 1, where the step width is not necessarily constant but depends on the gradient, to ensure a constant step width on the design parameter plane, i.e. if the design parameter λ_3 would change much with change in λ_1, the step width in λ_1 would be lower and vice versa. Second, the next parameter direction is evaluated. If it is detected that the gradient in direction of the first parameter is similar to or only changes uniformly for different values of parameter 2, fewer samples can be used for validation of the approximation plane. In the case that certain parameter dimensions are not or only slightly correlated, gradients in direction of the non-correlated design parameters can be extrapolated based on adjacent sample points. Figure 4.11 shows exemplarily two different hypersurfaces, in the upper plot for the case without any correlation between design parameter 1 and 2, and with full correlation between the parameters in the lower plot. It is apparent that in case of no correlation, considerably less samples are required for evaluation of the gradient, but also for validation of the approximation surface.

For detection of only slight correlation, assume that two freely selectable design parameters are not or only slightly correlated with respect to the response $\hat{y}$, i.e. if a is a realization of the first parameter λ_1, then

$$\left.\frac{d\hat{y}}{d\lambda_1}\right|_{\lambda_1=a,\lambda_2=b} \approx \left.\frac{d\hat{y}}{d\lambda_1}\right|_{\lambda_1=a,\lambda_2=c} \approx \text{const.} \forall b, c \in [\lambda_{2,min}, \lambda_{2,max}] \tag{4.15}$$

This means that the change of system response $\hat{y}$ with design parameter λ_1 does only depend on the value of λ_1 and not on the value of λ_2. A similar criterion is the derivative of $\hat{y}$ with respect to the design parameter λ_2: If it is zero or constant over the whole parameter range λ_1, this corresponds to no correlation. In this case, it is theoretically sufficient to evaluate the gradient only for one specific value of λ_2. This is highlighted in figure 4.10 by points in brackets – for those points no gradient needs to be calculated in case of no correlation, which reduces the number of samples required to

$$N = 2\,(p-1)\,N_{1dim}^{p-1} + N_{1dim}^{p-1} = (2p-1)\,N_{1dim}^{p-1} \tag{4.16}$$

Compared to (4.14), only the first term changes and not the second, since only the evaluation of the gradient is not needed while still it is required to evaluate the values at all points to ensure that the approximation is valid. Also the number of validation points can be reduced in case of no correlation, however no quantification can be given since the reduction highly depends on the level of confidence one requires in the approximation. Although this does not reduce the order of number of samples required, there is still a certain reduction, e.g. in the example with three parameters where two are not correlated, the number of samples can be reduced by approximately 29% compared to the case that a gradient is calculated at each

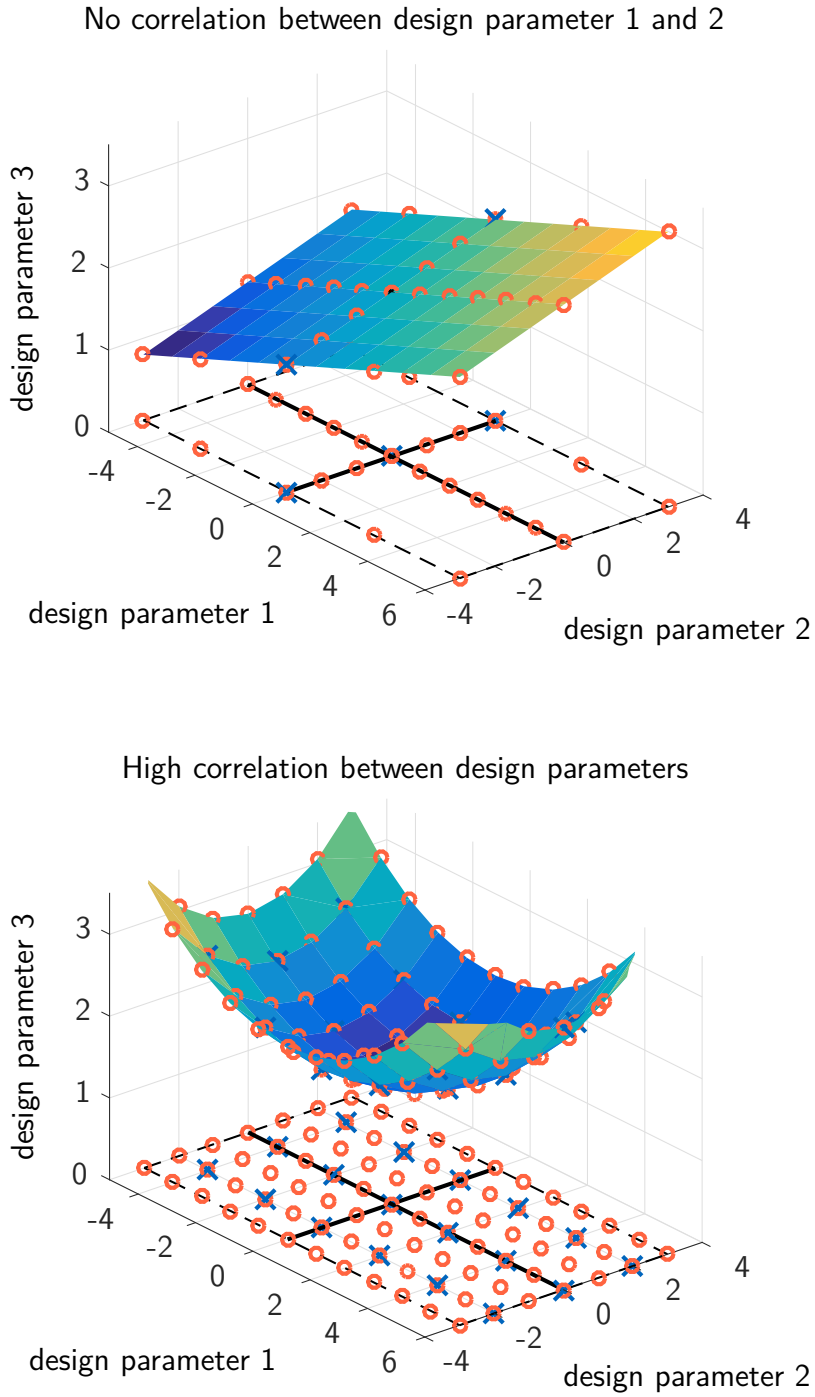


Figure 4.11: *Influence of design parameter correlation on sample points*

evaluation point, for the case where two parameters out of four are not correlated, the required number of samples is still reduced by 22%. To detect the case that two directions are not or only slightly correlated, equation (4.15) together with the findings in the previous section can be used: After the gradient in one parameter direction is evaluated, gradients along the second parameter are generated for one fixed value of the first parameter. In the example given in figure 4.10, starting from the initial point $(0, 0)$, first the hypersurface is evaluated along parameter λ_1, i.e. the points $(-5, 0)$ to $(6, 0)$ are evaluated. In the second step, the points along the second dimension are looked at, i.e. $(0, -3)$ to $(0, 3)$. Next, the gradients

with respect to the first parameter generated for different values of the second parameter are compared to the gradient value at the initial point $(0, 0)$. If the deviations are in the range of the stochastic gradient error (4.4), it can be inferred that the gradient in direction of one parameter does not change with the second parameter. This is certainly not guaranteed for the whole range of the first parameter, hence still the complete surface must be evaluated, however it is no longer necessary to evaluate the gradient for every point on the surface, which saves a lot of evaluations. Note that further considerable reduction of number of samples could arise from the limited design parameter space. For example, if the hypersurface only cuts a small corner of a cubic design space, the resulting area is relatively small. Figure 4.12 shows an example for a more complex design parameter surface for the case that only designs above the indicated surface are admissible and design parameter 3 is limited above by $\lambda_{3,max} = 5$.

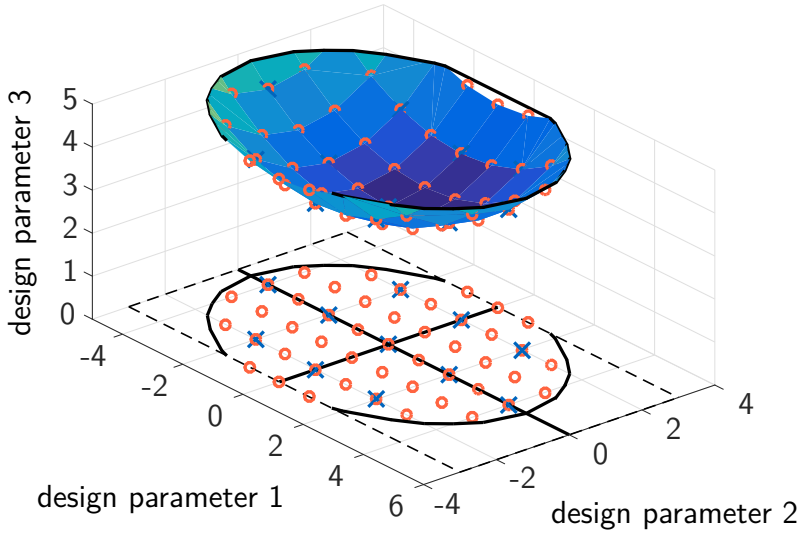


Figure 4.12: *Reduced number of samples due to limited design parameter space*

A regular grid of points ease the approximation of the actual design parameter boundary using a cubic interpolation. For the scalar case, i.e. by treating the design parameters independently, the approximation can be described by:

$$\hat{y}(\lambda) = a + b(\lambda - \lambda_{low}) + \frac{1}{2}c(\lambda - \lambda_{low})^2 + \frac{1}{6}d(\lambda - \lambda_{low})^3 \tag{4.17}$$

where $\hat{y}(\lambda)$ is the approximated design parameter boundary for a given λ, λ_{low} the closest grid point with $\lambda_{low} < \lambda$ and a,b,c and d are coefficients calculated using the following boundary conditions:

$$\begin{aligned} \hat{y}(\lambda_{low}) &= \hat{y}_{low} \\ \hat{y}(\lambda_{up}) &= \hat{y}_{up} \\ \left.\frac{d\hat{y}}{d\lambda}\right|_{\lambda=\lambda_{low}} &= \hat{g}(\hat{y}_{low}) \\ \left.\frac{d\hat{y}}{d\lambda}\right|_{\lambda=\lambda_{up}} &= \hat{g}(\hat{y}_{up}) \end{aligned} \tag{4.18}$$

which leads to

$$\begin{aligned} a =& \hat{y}_{low} \\ b =& \hat{g}\left(\hat{y}_{low}\right) \\ c =& \frac{6}{\left(\lambda_{up}-\lambda_{low}\right)^2}\Big(\hat{y}_{up}-\hat{y}_{low}-\hat{g}\left(\lambda_{up}-\lambda_{low}\right)- \\ & \frac{1}{3}\left(\lambda_{up}-\lambda_{low}\right)\left(\hat{g}\left(\hat{y}_{up}\right)-\hat{g}\left(\hat{y}_{low}\right)\right)\Big) \\ d =& \frac{2}{\left(\lambda_{up}-\lambda_{low}\right)^2}\left(\hat{g}\left(\hat{y}_{up}\right)-\hat{g}\left(\hat{y}_{low}\right)-c\left(\lambda_{up}-\lambda_{low}\right)\right) \end{aligned} \tag{4.19}$$

In case that sample points are generated in a non-uniform grid, alternative interpolation methods can be used, e.g. Shepard interpolation [She68]. This case is not further discussed.
Note that the descriptions given in this chapter shall highlight the challenges related to the use of uncertain gradients for determination and approximation of hypersurfaces. The given equations facilitate a proof of principle, which is given for the example of determination of design parameter bounds in section 7.3.1, while a universal routine and implementation of this method still requires further research.

4.3 Selection of Adequate Design Parameter Intervals

4.3.1 Objectives

The previous section introduced high-dimensional design parameter boundaries and one possible method for the challenging task of identification of these bounds for stochastic top-level requirements. Usually, design parameters are not only limited by one requirement, but by several, which leads to intersections of different boundaries that determine the actual feasible design parameter space, i.e. the regions of design parameter combinations that satisfy all requirements. This section answers the question how adequate design parameter ranges within this typically complex solution space can be found.

4.3.2 Available Methods

During early development phases, many design degrees of freedom exist, from which adequate design ranges must be selected. This is a common problem, which is why several approaches for identification of possible designs are readily available. Note that the major objective is not to identify the "best possible" solution, but instead to find a wide range of possible designs that all satisfy the imposed requirements to provide the designers a function with as much freedom as possible.
The challenging task in this context is to describe the feasible design space in a suitable manner, so that it is usable for specification and design. For that, set-based approaches were developed in the past (e.g. [SWL99, AIA+09]), where the solution space is approximated

by sets of feasible design parameter combinations. The state-of-the-art for the description of sets is a box-shaped solution space, where the biggest-possible box is identified within the overall solution space. Each edge of the box represents a feasible and hence acceptable interval for one design parameter. This method is well researched and also applied in industry ([Gra13, Fen+14, GHZ16]). The fitting of the box into the entire solution space is often done using stochastic methods. This is no limitation for its application in the scope of the TCA, since for this step, the design bounds are readily available in a mathematical formulation or numerical description, which allows an efficient evaluation of the boundaries and therefore the identification of good and bad design combinations.
The box-shaped solution space has the advantage that well established algorithms are available for the identification of feasible design parameter ranges. Since the interval for each parameter is decoupled and hence independent from the other parameters when using a box, the specification of a system or function is simple and also the function designer does not have to take into account the parameters possibly selected by other designers. However, this comes along with the big disadvantage that usually a high fraction of the possible design space is neglected, since a box usually only represent a small subspace of the feasible design space, even more for high dimensions, see figure 4.13.

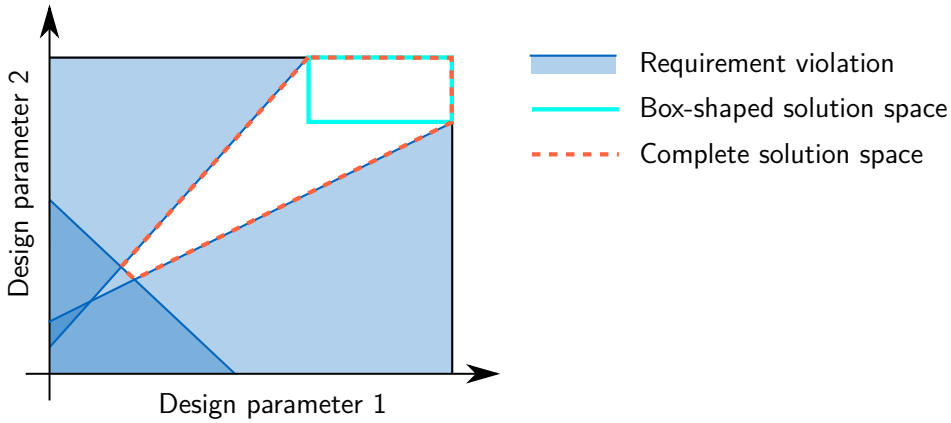


Figure 4.13: *Box-shaped versus complete solution space for two design variables (based on [Ers+17])*

For many design problems, the dependency of two design parameters can be well handled, e.g. if the design of the function, which determines a pair of parameters, is under supervision of one designer or development team. This was the motivation for beyond state-of-the-art research currently conducted by Erschen et al. [Ers+17, Ers18], where the solution space is described by two-dimensional solution regions (referred to as 2D-spaces). A comparison between box-shaped and 2D spaces is shown in figure 4.14 for a simple example with only one boundary that fulfills $\lambda_1^2 + \lambda_2^2 + \lambda_3^2 < 1.5$. Although still only a subset of the complete solution space is covered, this approach results in an exponential growth of size of the solution space with increasing number of design parameters compared to the box approach [Ers+17]. However, the required computational effort is much higher, even with the assumption of linear or almost linear dependencies of individual boundaries on design parameters made in the

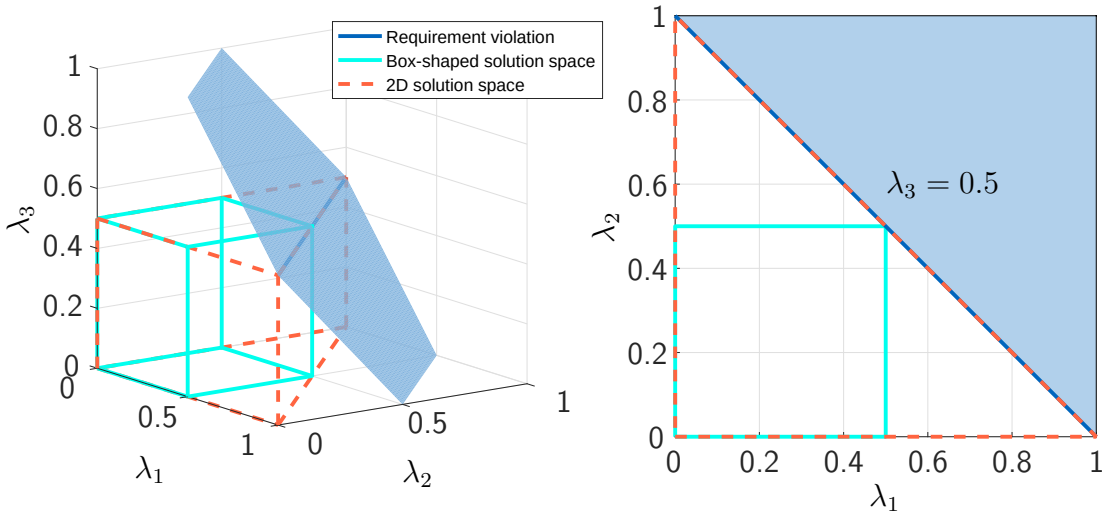


Figure 4.14: *Box-shaped vs. 2D solution space for three design variables (example from [Ers+17])*

procedure described by Erschen. Furthermore, no method exist yet for nonlinear boundaries. 2D solution spaces are still object of ongoing research. Nevertheless, already box approaches result in physically more reasonable and less conservative design parameter intervals compared to conventional parameter intervals prescribed by today's certification standards and hence the referred methods can be applied in the scope of the TCA, where, however, future research can lead to even more performance.

4.3.3 Specification Trade-Off

No matter if box-shaped or 2D solution spaces are used for approximation, in both cases the best solution space under the given constraints is looked for. However, the meaning of "best" has not yet been defined. One option would be to choose the box or combination of 2D spaces that has the biggest volume, certainly given that each design parameter is normalized to allow a reasonable comparison. Although this is one common way, it does not take into account how hard it might be to achieve specific design parameters or combinations. Recall that design parameters in the current context are not parameters used for actual design but instead prescribe admissible ranges for designs established in corresponding requirements. Hence, the specific selection of design parameter ranges can have a significant influence on the complexity and effort that is required during later development to achieve designs which comply with these requirements. Consider the example given in figure 4.15. Although the volume of the solution space obtained using the 2D method is less than the optimal volume indicated in figure 4.14, an increased interval for design parameter $\lambda_3 = 0 \ldots 0.7$ might ease design and implementation much more than the consequences of reduced intervals for λ_1 and λ_2. A weighting of the individual dimensions as a function of the parameter value can be used to account for parameter-dependent effort, whether it be cost, time, or further influences. By that, a trade-off can be made between the efforts resulting from specific design parameter interval selections.

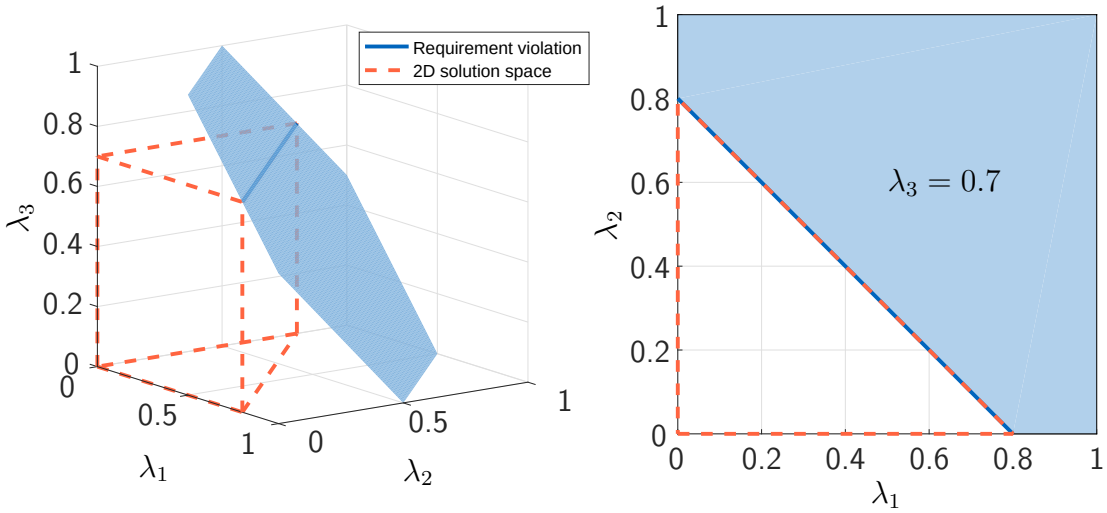


Figure 4.15: *Trade-off for the size of the solution space*

Any solution resulting from the aforementioned algorithms leads to a safe system since in no case any bound is violated and hence any design within the specified solution spaces complies with all requirements. Referring to the different performance levels introduced in section 4.1.2 – adequate and desired performance – the obtained solution spaces correspond to adequate performance, i.e. the performance range which must be fulfilled to achieve a safe system.

For the desired performance, which gives specific design goals required during development, further considerations can be added. For example, instead of only considering the boundary which divides the solution space in acceptable and unacceptable designs, the combination of design parameters could be looked for that leads to the theoretically safest system. Furthermore, sensitivity of the design objective with respect to the individual design parameters can be taken into account. Using sensitivities, the range of design parameters could be identified where a design objective does not change much within the intervals and hence the resulting system behavior is more predictable during early development phases. Additionally, a low sensitivity with respect to design parameter changes makes the resulting system more robust against uncertainties of parameters, which helps to achieve a more consistent system behavior in the presence of uncertainties. While set-based approaches are well suitable for the specification of the adequate performance, for the determination of the desired performance also (robust) design optimization methods can be used, see e.g. [BS07] for general design optimization and [Kou16] for optimization of high-dimensional, stochastic design problems. Certainly, also other aspects besides safety could be decisive for selection of the desired performance. For example, an aggressive autopilot for a passenger aircraft could result in higher safety of the aircraft, however, passenger comfort would be drastically reduced which might no be desired. Note that robust design optimization is not suitable for determination of adequate performance ranges, since it usually only results in one optimal solution with respect to the assumed uncertainties of the individual design parameters, while for the determination of the feasible design space, the design parameters are not uncertain but instead the widest possible range of parameter combinations is looked for that satisfies all requirements. Furthermore, a weighting used during

design optimization does usually not ensure compliance with all requirements.

4.4 Validation of Requirements

4.4.1 Task Description

Validation is the assurance that the requirements for a product are correct as well as complete and hence answers the question whether "*we build the right aircraft, system and function*". It can be also understood as the assurance that specifications meet the needs defined by stakeholders, which are, for example, customers, operators, users and certification authorities. Different stakeholders have different intentions and hence objectives during validation: Certification authorities focus on safety and, therefore, constrain undesired and hence unsafe behavior. Customers want to ensure that the specified product meets the objectives driven by its application and hence can be used for the intended operation. Operators and users may focus on implications of specified functionalities on the actual (daily) use and operation of a product, where especially high availability is a major concern.
Independent of the motivation for validation, the common objective is to ensure that requirements are correct and complete. Correctness is "*the degree to which an individual requirement is unambiguous, verifiable, consistent with other requirements and necessary for the requirement set*" [SAE10, p. 57]. Completeness is defined as "*the degree to which a set of correct requirements, when met by a system, satisfy the interests of customers, users, maintainers, certification authorities as well as aircraft, system and item developers under all modes of operation and lifecycle phases for the defined operating environment*" [SAE10, p. 57].
If validation is conducted correctly, it reduces risk for unintended functionality. It is possible to identify errors or omissions already early during development, which reduces risk for subsequent redesign or inappropriate system behavior. Since redesign is usually related to additional development effort and cost, validation highly contributes to keeping development costs down and hence is not only motivated by the function but also by economy.
Driven by the importance of validation, usually a validation plan is required that describes methods used for validation, collected data, and schedule. Adequate guidelines exist for validation planning (e.g. [SAE10, chapt. 5.4] for aerospace applications). Therefore, the remaining section focuses on methods for validation and how the TCA with the model-based approach can enhance the validation process. Note that validation must be conducted at each level of development, i.e. starting from aircraft functions down to item specifications. The explanations given below are applicable to any phase of development.

4.4.2 State of the Art

The validation effort is left to the developer. Certification authorities usually only require a structured process that should be captured in a validation plan. The question how to

actually conduct validation of specific as well as sets of requirements to ensure correctness and completeness is up to the design organization.
Validation is conducted with respect to "own" requirements, i.e. those that are captured for the actual development and which are in the scope of the developed system, but also with respect to other systems. This is covered by a thorough analysis of the interfaces to other functions and systems. It is inevitable to make assumptions, especially during early development phases. These assumptions are usually challenged by independent reviewers, together with related justifications and interpretations.
Today, usually a collection of questions, templates and checklists are used for validation. Those means are mainly based on knowledge and lessons learned and are used by experts and stakeholders to assess correctness and prevent incompleteness. However, these means are usually either very specific but only applicable to particular functions and systems or rather general and hence less powerful but, therefore, applicable to more general products.
Ideally, requirements should be validated before actual design and implementation of a system, function and item. In practice however, this is usually not possible, especially for complex and integrated systems, where understanding of the specified functionality only emerges during implementation. Hence, validation is often only possible parallel to design and implementation, which results in a staged validation throughout the development cycle. Only with increasing design progress, confidence in the correctness and completeness of requirements grows. This could even lead to the case that testing of the implemented function serves both, validation and verification.
For development of aerospace functions, ARP4754A lists the following validation methods [SAE10, chapt. 5.4.6]: Traceability, analysis, modeling and test and experience. At early development phases, mainly traceability, experience and qualitative analysis can be used and it hence heavily relies on expert knowledge. With design progress and implementation, more quantitative validation methods like modeling and test can be used.

4.4.3 Model-Based Support of Validation

Especially completeness of a set of requirements is, by its nature, usually difficult to prove before design and implementation. The model-based approach of the TCA allows to close the gap between early qualitative and late quantitative validation. This is possible by using models and simulations already during early development phases and particularly before design and implementation.
In section 4.1.4, specification models are introduced, which reflect the behavior of a requirement without using knowledge about any possible implementation. For requirements derivation, the specification models are tuned to identify the possible range of behavior that fulfills top-level safety requirements. Now, the identified acceptable and desired behavior is assumed to be fixed and simulations can be used to support experts and stakeholders to check correctness and identify incompleteness of requirements. Although simulations have already been used in the past for validation, the important difference is that the specification models are

readily available prior to design and implementation and hence allow a very early validation with the associated benefits of early identification of incorrect or incomplete requirements. Certainly, correct representation of the specified behavior and the environment are essential. This is also true for models used for derivation of requirements and hence there is no additional burden for model-based validation. The simulation environment developed for requirements derivation can be directly used for validation. While the system and environment models remain the same and can be used without additional effort, adequate validation methods must be applied to evaluate the results obtained from analysis and simulation using the system models, see figure 4.16.

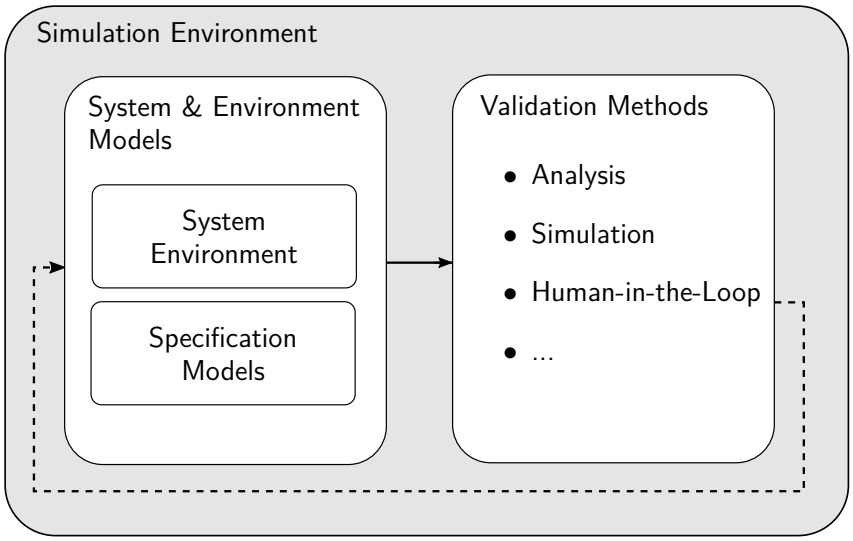


Figure 4.16: *Simulation environment for model-based requirements validation*

Note that this procedure is not limited to safety-driven performance requirements for which already specification models exist from requirements derivation. Further requirements can also be evaluated, e.g. customer functional or performance requirements. For any requirement that can be represented by a specification model, model-based validation during early development phases is possible. The advantages can be concluded as follows:

- **Early quantitative validation**: Simulations and analysis are already used today for validation and verification, however this usually requires an (initial) implementation. Using specification models allows for an early quantitative validation, which significantly helps to assess correctness and completeness and prevent later design changes or undesired behavior.

- **Confidence in specified functionality**: Simulations usually provide a vivid representation of the specified functionality, which increases confidence. Furthermore, since simulation results do not necessarily require deep knowledge about technical aspects, they are also useful for decision makers to determine whether the specified system matches the envisaged product.

- **No additional effort**: If following the TCA, virtually no additional effort is required to utilize the specification models already developed and implemented for requirements derivation. No novel or special methods are required for model-based validation.

Certainly, care should be taken to obtain correct validation results. The specification models usually rely on assumptions and still require thorough analysis of these and also of the interfaces and non-technical aspects. Hence, model-based validation methods can only be one additional, even though powerful tool for determining correctness and completeness of requirements.

4.5 Open Challenges

Model-based requirements derivation and validation highly rely on the availability of formalized requirements and corresponding specification models for simulation of the specified behavior before actual design and implementation. This chapter discusses, how models can be used for derivation and validation especially of safety-driven performance requirements. Furthermore, a method is proposed for derivation of design parameter boundaries based on probabilistic top-level requirements. Although the given descriptions and methods suffice for an initial application and proof of principle, the TCA has even more potential that can be achieved if the following main challenges are solved:

- **Formalization and modeling efforts**: Although templates can be used for formalization of requirements and implementation of according specification models, those are only available for standard requirements. More complex specifications still require additional effort for that task.

- **Size of design problem**: Although the proposed methods reduce the computational effort for requirements derivation, they are still limited to smaller design problems, with a maximum of 5-6 design parameters based on experience. This is no show-stopper since for many physical problems, the overall design problem can be split up into independent subproblems with lower numbers of design parameters. However, more efficient algorithms would allow for a more integrated consideration of all contributing parameters or quicker evaluation, which both would increase practicability of the proposed approach.

- **Multidisciplinary considerations**: The descriptions given in this chapter mainly focus on safety-driven performance requirements and technical aspects. For determination of the optimal design, further influences might be decisive, e.g. motivated by operation and economics. Using multidisciplinary models would further automate the task of identifying a range for ideal behavior, which would be highly beneficial.

The model-based derivation approach for safety-driven lower level requirements requires a different thinking: Requirements derivation is no longer about ensuring that specifications taken from traditional, established standards make sense and trace well bottom-up to top-level

safety requirements but instead it is the opposite approach to look for specifications that results in a safe system in a top-down fashion. The posed open challenges can be mainly approached by applying the proposed ideas to the development of a real product. This is currently subject of research conducted by another researcher at the FSD of the TUM [Mum18].
Exemplary applications of the described ideas and introduced algorithms are given in chapter 7.3.1 for an example in the context of developing an *Automatic Flight Control System* (AFCS) for close formation flight of fixed-wing aircraft.

5

Design and Verification

In the previous chapter, the top-down requirements derivation and validation processes were discussed. The subsequent steps in the development process are design, implementation, and verification, which are conducted bottom up from item to aircraft level. This process model is well known as "V-Model", shown in figure 5.1. The bottom up steps and especially the influences and enhancements of the TCA on the design and verification phase are discussed in this chapter. Although these steps are different in principle, the impacts of the TCA and also the applied methods are similar, which is why design and verification are considered together in this chapter.

The development process is a highly parallel and iterative process. At lower levels, system development is usually split up into several subsystem development processes in parallel. Integral processes are applied to every subsystem development. Typical aircraft (sub)systems are for example flight and ground controls, engine control, guidance and navigation. Furthermore, although ideally the development is straight forward, design and verification go hand in hand, where a non-compliance with requirements identified during verification on item, subsystem and system level necessitates a redesign. This leads to the common iterative design and verification process.

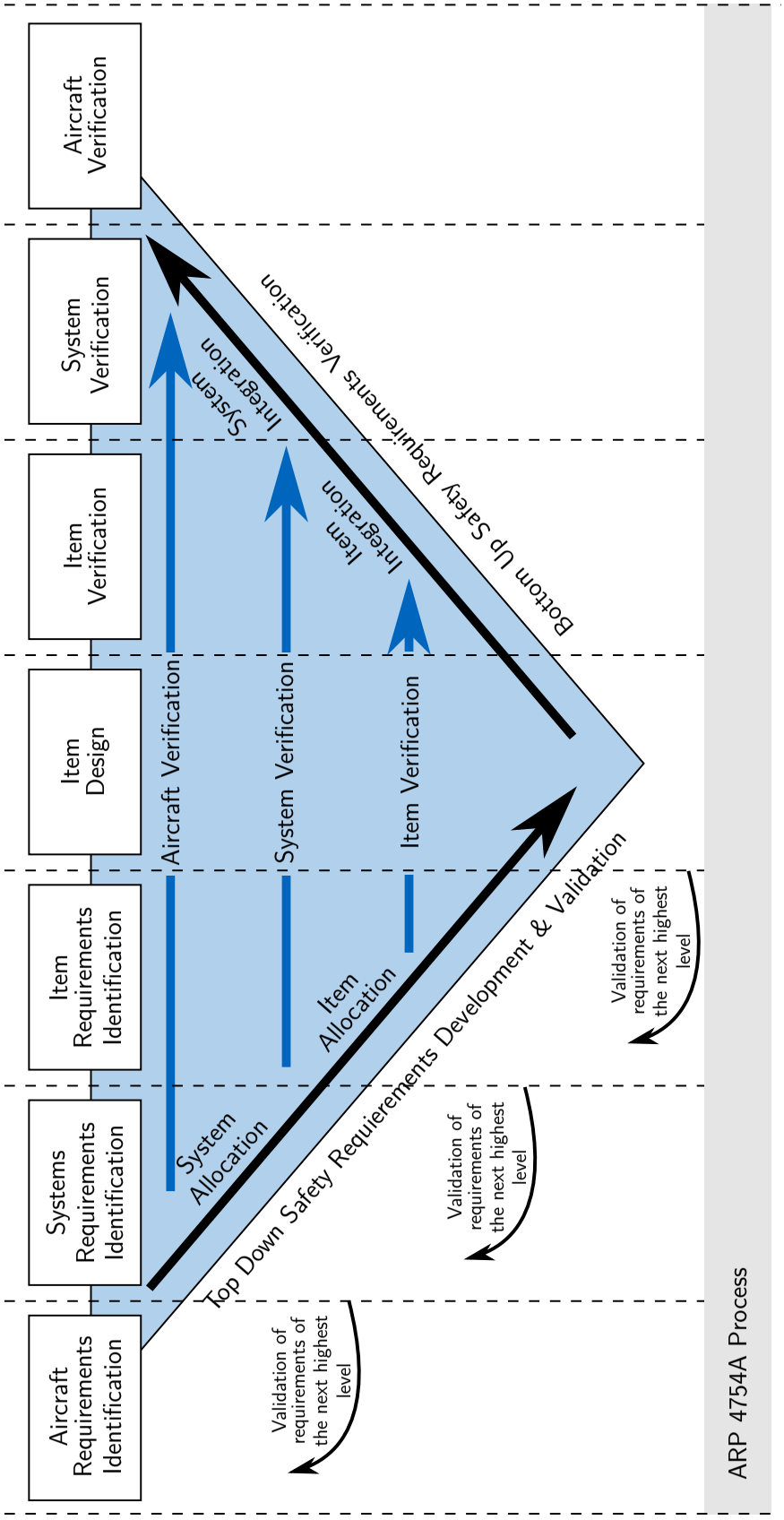


Figure 5.1: *Intersection between Safety and Development Process [SAE10, p. 24]*

5.1 Design Process

5.1.1 Introduction

This section addresses design and implementation of functions specified using the methods described in the previous chapter and especially answers the question how stochastic methods can enhance the design and implementation of systems and functions.
The objective of design and implementation in the considered context is to realize safety-critical functions and systems according to specifications to yield a safe product. Besides the functions, also the items that realize the higher level functions certainly have a high impact on the safety of the resulting product and hence admissible failure probabilities must be considered during the design on item level and appropriate rigor must be applied to the process of selecting or designing items. The TCA aims at ensuring safe functions and systems, hence there is no direct impact of the TCA on item design. Therefore only minor attention is given to the actual implementation of functions to real hard- and software since for that, already well established processes and standards are available, which establish safety on item level by ensuring high levels of reliability. For an overview of related aerospace standards, see figure 3.1.

5.1.2 State of the Art

Similar to the specification of safety critical functions, also their design is heavily driven by past experience. Changes of classical design principles are usually only motivated by insights obtained from incidents and accidents. Still today, uncertainties and disturbances are treated very conservatively. To understand the reasons, it is necessary to consider developments during the past century. This in done in this section exemplarily for the development of flight control functions. However, the principle of how uncertainties are considered during development of arbitrary safety-critical functions is similar for many disciplines.
Flight controllers, which represent the implementation of specified and designed flight control functions, are usually based on linear control theory. The first autopilot developed back in 1912 was a gyroscopic autopilot (also known as Sperry autopilot), which used mechanical links between gyroscopes and aircraft control surfaces [Wil06]. These links can be considered as linear feedbacks of attitude errors to control surfaces and hence represent the beginning of using linear flight control laws. Different technologies have been used for implementation of flight control functions since then, starting from mechanical and hydraulic, via electric and electronic systems, up to today's common digital autopilots. However, still today, mainly linear control functions are used, also driven by the usually low level of complexity of automation tasks, e.g. wings leveler and altitude hold functions.
The experience gained during one century of application of linear control design and evaluation almost only led to requirements based on linear control theory that must be fulfilled for certification. If requirements are already formulated using linear theory, it is natural to also develop linear controllers to satisfy those requirements. Note that this is not necessarily a drawback –

for conventional autopilot functions, such prescriptive design requirements adequately answer the purpose of ensuring safe designs. Even if today non-linear control theory is used, the capability of non-linear controllers is approximated by linear controllers for which linear verification procedures are available assuming that in each approximation region, the nonlinear controller behaves similar to the linear approximative controller. For that, well-known tools are available for linear robust multi-input multi-output control, which allows analysis of steady-state and transient performance and which also provide robustness margins for the case of uncertainties and disturbances[FYN17].

While the level of conservatism arising from applying conventional linear control theory is acceptable for common flight control functions, it is no longer reasonable for highly-automated or autonomous vehicles, for which a high level of safety and availability must be achieved. Another source of conservatism of the state-of-the-art approach for design of flight control functions is the way how uncertainties are treated. Conventional design of control functions means design for best performance under nominal conditions and acceptable disturbance behavior for certain known disturbances specified in today's requirements. Uncertainties are usually treated by robust control design methods, where uncertain parameters are represented by parameters with bounded magnitude [Mac04, chapt. 12], i.e. with hard deterministic bounds, which however is usually only a weak representation of the reality. The outcome of robust control and analysis methods is the guarantee that no combination of bounded parameters violate requirements imposed on the respective functions.

The modeling of uncertainties with interval bounds comes along with two major disadvantages. First, uncertainties from physical processes usually follow a distribution with a high probability close to an expected mean value with decreasing probability the further the actual value lies afar from the assumed mean. Hence, for any selection of hard interval bounds, there is a certain probability that a value lies out of the bounds. Depending on the selected width of the bounds, two extreme cases can be identified, see also figure 5.2. On the one hand, wide bounds can be chosen that include almost all possible realizations of the uncertain parameter, with only a very small probability that the actual value lies out of the specified interval. This selection is very conservative, since also very unlikely values are considered without using the possibly available information about the likelihood of such values. On the other hand, tight bounds would ease design of a function according to requirements due to a smaller variation of uncertainties. However, such relaxed bounds are closely related to a high probability of exceedance in reality and hence guaranteed compliance with requirements under such tight bounds could cause deceptive confidence in actual system performance.

The second principal disadvantage of hard interval bounds is the loss of information due to modeling of uncertainties by deterministic intervals, which is especially of relevance for the combination of many uncertain parameters. While it is obvious that, when considering multiple uncertain parameters, a combination of improbable parameter values is even less probable, this accumulation effect is not considered for robust control approaches. This could lead to overestimation of the criticality of very unlikely combinations of uncertain parameters, even if

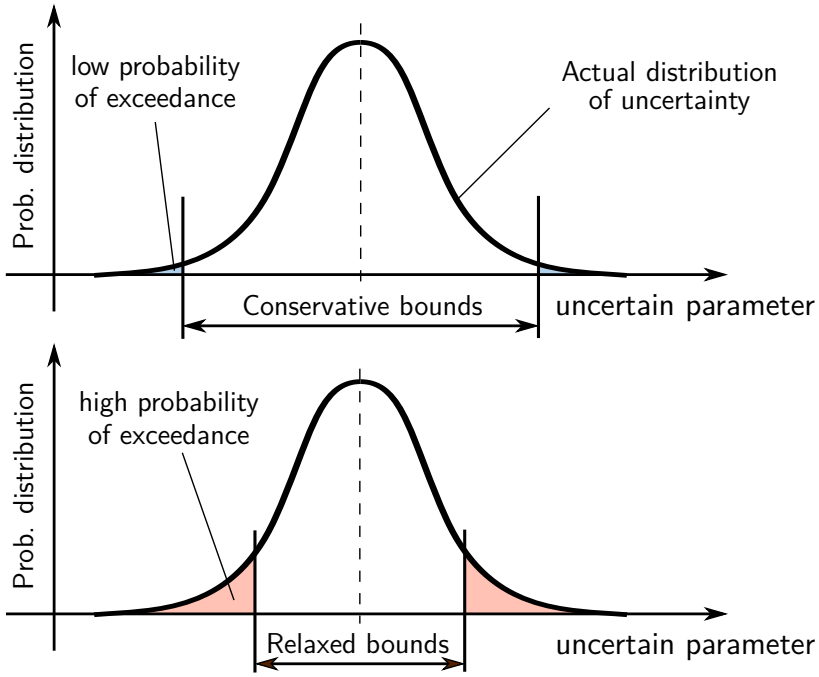


Figure 5.2: *Hard interval bounds for approximation of uncertainties*

those parameters would be modeled by relaxed bounds.

All these disadvantages can be addressed using a physically driven approach, which matches the reality much better. This is enabled by modeling uncertainties using the whole available knowledge, e.g. about the distribution of individual uncertainties. In this case, there is no longer a 100% guarantee of complying with conventional (deterministic) requirements, since there is always a very small probability of violation. However, formal compliance is enabled by the shift from deterministic to probabilistic requirements in the scope of the TCA.

5.1.3 Model-Based Support of Design

There is a high potential when using the whole knowledge about uncertainties and disturbances also during system design in order to exploit the total capabilities of a given system or setup. No new method for this task is given in this section, since such methods usually depend much on the design objective. Instead, the consideration of uncertainties in the development phase is motivated and current efforts as well as challenges are worked out.

Robustness analysis of complex systems started in the eighties, where deterministic descriptions of uncertainties were used to analyze the effect especially of parametric uncertainties on controlled systems [Bar94]. Only during the last decade, research started on probabilistic and randomized methods, where uncertainties affecting the system are considered during control design with their probabilistic nature. The objective is to obtain probabilistic robustness margins, usually using randomized algorithms like Monte Carlo simulation. For a detailed literature study and evaluation of the state-of-the-art, refer [CDT11] and [TCD13]. The motivation for past research on probabilistic methods is the desire to break the computation complexity barrier

suffered by deterministic worst case approaches [CD06, pp 381-414]. In the referred literature, stochastic approaches are proposed that minimize the likelihood of violating requirements for the important case of uncertain plant parameters, where the full probabilistic knowledge of these uncertainties is used. These approaches already reduce conservatism, however, they focus on relatively high admissible failure probabilities, opposed to the usually very small failure probabilities inherent to probabilistic safety requirements. Still, these methods can be a valuable starting point for future research.

Much effort is currently also put in researching on novel control methods and development of related performance guarantees. Certification of adaptive controllers is one example for that [Jac08, LCK10]. Especially approaches that ensure performance of an uncertain plant and that provides probabilistic performance bounds, i.e. bounds that are only exceeded with a certain probability, must be mentioned in this context (see e.g. [MCJ12, CH13a, Müh18]).

The TCA provides the perfect foundation for any new method that is able to provide guaranteed probabilistic performance bounds for uncertain plants, mainly due to the probabilistic requirements inherent to the TCA. Worst case considerations with estimation of the related failure probabilities is essential to ensure a safe implementation, however, this is usually computationally very expensive. While dedicated methods for specific problems and control methods can provide even higher gains for evaluation performance, the universal approach given by the application of enhanced stochastic methods (described in chapter 2) already enables efficient stochastic worst case analysis for arbitrary implementations thanks to the black box principle of the introduced methods. Using worst case analysis, the designed and implemented functions do not only work well in average or with bounded uncertainties, but also under unlikely conditions, however, where the probability of exceedance is known opposed to worst case scenarios obtained by robust methods. If enhanced stochastic methods are used, the simulation environment introduced for requirements derivation and already utilized for validation, can also be used to support design, see figure 5.3. The major extension to the simulation environment

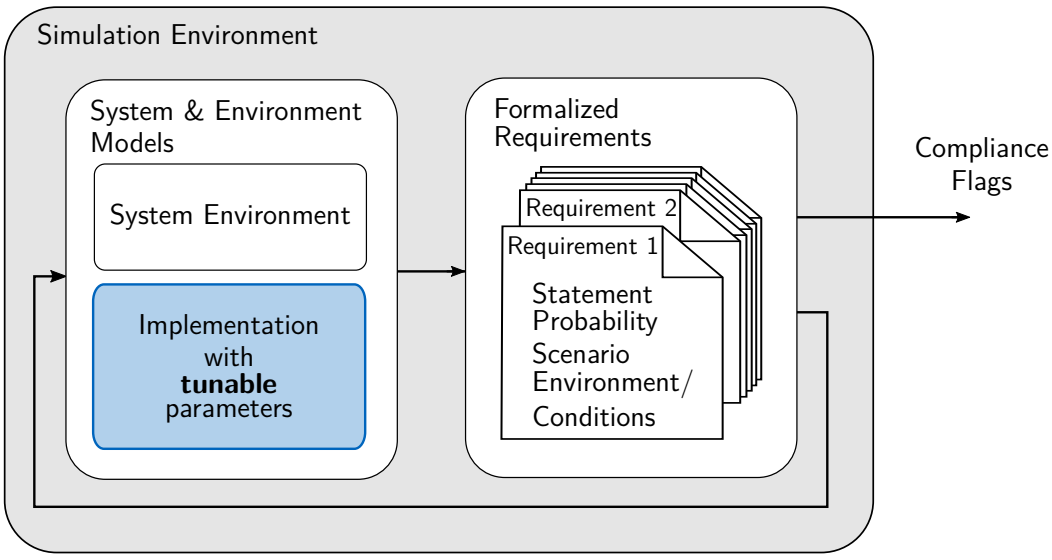


Figure 5.3: *Simulation environment for model-based support of design and implementation*

is the replacement of the specification models by models of the actual implementation. With advances in development, also system environment models are usually updated with better models for uncertainties, while the formalized requirements remains the same to enable continuous analysis of compliance parallel to function design and implementation. Using enhanced stochastic methods, available degrees of freedom of an implementation (denoted as "tunable parameters" of the implementation) can be chosen in a way that all requirements are fulfilled.

Note that although the consideration of all different kinds of uncertainties and worst case analysis during design and implementation imply a shift in paradigm, the required system performance is still defined by classical metrics. This is vital, since it ensures easy understanding and hence a high level of acceptance of the probabilistic approach for today's design engineers. By also following the TCA during the design phase, it is ensured that a system is developed that complies with requirements under all admissible conditions. For example, a flight controller solely optimized for ideal conditions can use very high feedback gains to obtain a good performance under ideal conditions, while it probably leads to very poor performance in disturbed and uncertain environments. Using an integrated approach with consideration of all possible uncertainties, disturbances, and failures during system design leads to the safest system, which does not necessarily show the best performance under ideal conditions.

The possibility of continuous evaluation of requirements during design and implementation by efficient simulation methods allows an early identification and evaluation of scenarios that could lead to unsafe conditions. This enables the derivation of mitigation means for safety-critical behavior and a design that ensures an acceptable level of safety early during development, which can reduce time and cost compared to the case that unsafe behavior is only identified during (final) verification. This task is especially supported if using enhanced stochastic methods, where usually many failure samples are obtained by simulation and each sample is related to an individual failure scenario. This is exemplarily shown in section 7.3.3 for an example in the context of development of an automatic flight control system for an aircraft formation flight. Another important benefit of parallel verification is the possibility to optimize a function design to find the best possible implementation. "Best" in this context could be for example the design that leads to the highest level of availability of a function for a given level of safety. This is very similar to the definition of adequate and desired performance in section 4.1.2, where the first constitutes the acceptable range of designs which fulfills all requirements (and hence leads to a safe product) and the second also takes into account non-safety driven aspects like economy and optimality. The challenges arising in the context of finding the optimal design are similar to identification of the desired performance range, where especially the number of degrees of freedom of the design are limited by available computational resources. In summary, it can be said that the TCA, characterized by the usage of probabilistic requirements and the integrated consideration of uncertainties and disturbances, also provides benefits for the design phase. Robust design methods, which are usually used to take uncertainties into account during design, can lead to deceptive deterministic performance guarantees. By using probabilistic bounds instead, the level of confidence in the designed and implemented

function can be quantified. This usually requires either dedicated design and analysis methods for specific control algorithms or application of enhanced stochastic methods to reduce computational effort usually related to evaluation of stochastic worst case scenarios. However, this is a solvable task, either by using novel methods currently researched at or developed in near future or by application of enhanced stochastic methods described in chapter 2. The TCA highly benefits from using the whole knowledge about uncertainties and disturbances throughout the development process, also during the design step. This is exemplarily demonstrated in section 7.3.3. Note that the argumentation in this section uses thinking in black and white terms to highlight the major differences. Certainly, there are attempts already today to a more realistic consideration of uncertainties during system design, however, the application lack in the availability of a direct link to probabilistic safety-driven requirements, which is one major novelty and advantage of the TCA.

5.2 Verification

5.2.1 Introduction

Already parallel to the design and implementation phase, verification-like tests are used to check whether the product at its current state is likely to fulfill all requirements put on it. Actual verification takes place as soon as an item, function, system, or overall product reaches its final state to ensure that the implemented functions meet all requirements, i.e. it is the proof that the specifications are correctly and completely translated into a real product.
Assumptions are made throughout the whole development. At latest during verification, it must be proven that they are valid and hence the requirements derived under these assumptions are valid and appropriate to ensure safety. Although this is already of relevance following conventional certification approaches with prescriptive design requirements, it is essential to ensure safety using the TCA with its model-based break down of probabilistic top-level requirements. During the requirements derivation step, necessarily assumptions and simplifications must be made due to the lack of information and implementation during early development. Although the safety-driven requirements derived using the model-based approach ensure safety under the assumptions and simplifications made, this is not guaranteed for the actual implementation. Hence, during verification safety of the actual implementation must be proven, using the whole knowledge about uncertainties, disturbances, and failures of the developed product. This is necessary at all levels of verification, starting from item level, where, for example, a selected sensor comes along with additional uncertainties and failure modes up to aircraft level, where the total performance is driven by the performance and uncertainties of all items, functions, and subsystems. While it is required also for the common development approach to verify that the assumptions made earlier during development are still valid, the reconsideration of safety of the overall product after replacing assumptions by actual solutions reduces the level of conservatism, which is one objective of the TCA.

In the subsequent sections, first it is discussed how verification especially of aerospace systems is conducted today and what the results are, followed by the motivation for a stochastic, model-based verification approach and the challenges inherent to that.

5.2.2 State of the Art

The verification of aircraft systems is well documented in ARP4754 [SAE10, chapt. 5.5]. This practice describes recommended steps, tools, and procedures for verification and certification. For example, it is described in detail, which documents must be generated to prove compliance with requirements, it describes a verification plan and matrix that specifies the means of compliance for each requirement and also the expected outcome. Furthermore, based on the criticality of each item, function, and system identified in the safety assessment and the resulting function and item development assurance level (FDAL and IDAL), the required verification rigor is specified, i.e. how "good" compliance with a requirement must be shown. This is a well documented, practice-proven process, which should also be maintained for the verification in the scope of the TCA. Therefore, the focus of the remainder of this section lies on the means used for verification, where a major contribution of the TCA can be identified, while the principle process of verification is unchanged and hence not further discussed.
According to the referred recommended practice, means for verification can be classified into four principal categories:

- **Inspections and reviews**: Experts check the resulting product, documents and drawings to ensure design and implementation according to specification. Such evaluations are especially applicable for requirements, where compliance can be easily identified by inspection of the product and its documentation. This is, for example, the case for requirements that prescribe certain properties (e.g. "*The warning light must be red*") or the presence of items (e.g. "*There must be an indication for battery voltage*").

- **Analysis**: For analysis, models are used to check the compliance of items, functions, and systems in normal and abnormal conditions. This could be, for example, mechanical models for *Finite Element Method* (FEM) analysis or flight dynamics models for performance and stability analysis. For that, models are required that correctly represent the actual product. This is usually ensured by verifying models against the real product by tests (see next bullet point). This type of verification is usually used for requirements that cannot or only partially be checked by tests due to too high risk or cost.

- **Tests and demonstration**: In this category, the actual product is tested, its functions are evaluated and it is checked whether it behaves according to requirements and does not show any unintended behavior. Verification by tests and demonstration requires a thorough identification and description of test cases to ensure that all requirements are adequately checked. The description of test cases must, for example, include the

function to be tested, required inputs, measured outputs, and admissible responses. Tests are also used to verify models required for analysis.

- **Similarity / service experience**: If a similar item, function, or system has already been developed and implemented before, service experience can be used to prove that the chosen solutions adequately fulfill requirements put on the current development.

5.2.3 Model-Based Support of Verification

The power and potential of the TCA rely on the integrated consideration of all known uncertainties and disturbances and the use of probabilistic requirements. This also necessitates new or enhanced means for verification to show compliance with probabilistic requirements.
Similar to the other steps of the development process, also verification in the scope of the TCA is based on models. Evaluation methods for stochastic assessment can be seen as additions to and enhancements of the methods already used today for analysis. Probabilistic requirements are linked to low acceptable violation probabilities. Compliance with safety-driven requirements can often not be shown by tests and demonstration, since safety-critical events must only occur with very low probabilities and are hence hard or even impossible to test. Furthermore, tests of such requirements could cause severe damage to the product and the environment. Hence, analysis is the only reasonable way to prove compliance with such safety-driven probabilistic requirements.
Analysis of requirements with low failure probabilities is already a challenge during requirements derivation. For verification, this is even more challenging since the models used for verification are usually very sophisticated and contain – in contrast to the specification models used for requirements derivation – detailed models of items, functions, and subsystems of the final product with many additional uncertainties arising from these components. Although models are already used today, their usage is extended by the application of enhanced stochastic methods, which enable an efficient evaluation of small failure probabilities, even for complex models with many uncertainties. The simulation environment used throughout the whole development can also be used for verification. If models are consistently used also during design from item to system level, detailed models for all implemented functions are readily available for verification, see figure 5.4. The formalized requirements are available from requirements derivation. Models of the implemented functions for the final design result from the design phase, where also uncertainties of all components and the environment are modeled with an adequate level of detail. Latest for verification, all models of the environment and the product itself, must be verified, e.g. by using tests of the real product in the real environment, to obtain valid results of model-based analysis for verification. Due to the consistency of the simulation environment throughout all development steps, enhanced stochastic algorithms can be easily applied to prove compliance with safety-driven probabilistic requirements. Even discrete failures could be included in the stochastic evaluation by the probability of exceeding nominal ranges of uncertainties. However, it is recommended to use established methods like *Fault Tree*

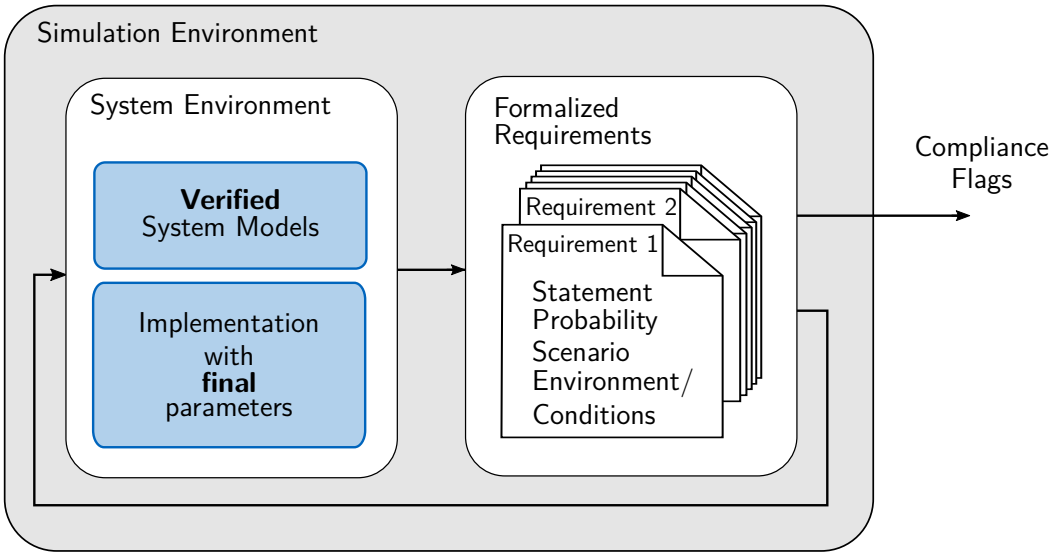


Figure 5.4: *Simulation environment for model-based requirements verification by analysis*

Analysis (FTA) for explicit consideration of failures instead of the implicit consideration by inclusion into the modeling of uncertainties, since this provides more efficiency and reduces the risk that related failure conditions are not identified due to the way how samples from stochastic simulation are generated (e.g. by a random walk process for Subset simulation). This is especially the case, if the failure conditions triggered by failure events are very different to the behavior for the non-failure case.

Up to now, the description focused on verification using models of the real product and the environment, which is usually referred to as model-in-the-loop simulation. Nevertheless, the simulation environment can also be used in conjunction with coded software or even the actual hardware using *Software-in-the-Loop* (SIL) and *Hardware-in-the-Loop* (HIL) simulations. Still enhanced stochastic methods can be used to vary uncertainties of the environment and by that excite the software or hardware embedded in the closed-loop in a way that failure conditions occur. Referring to figure 5.4, in his case the "system environment" is enhanced by SIL or HIL setups with adequate interfaces to the uncertainties of the models used to evaluate software and hardware.

Confidence in Results of Enhanced Stochastic Analysis

When proving small failure probabilities by simulation, the accuracy of the analysis and hence the level of confidence one can have in the results certainly rely on the models used to describe the various types of uncertainties (see 4.1.4). However, this is no exclusive problem of the probabilistic approach inherent to the TCA, but can also be found today for the well established and accepted FTA. The failure probability of functions and systems analyzed using a fault tree solely relies on failure probabilities of individual items and functions and their interaction. Hence, the assessment is only as good as the assumptions made for the failure probabilities of individual items (for FTA) and for uncertainties (for model-based analysis in the scope of

the TCA). Fortunately, uncertainties that can have a significant influence on safety are usually well known which is why FTA is accepted today and which is therefore also a legitimate argumentation for the use of enhanced stochastic methods for verification. Furthermore, similar to documentation and analysis that must be made for FTA to prove the correctness of the failure probabilities of individual items, the confidence in the correctness of results obtained by enhanced stochastic simulations is increased by using well verified and justified models of the product and the environment.

Apart from the definition of uncertainties, another source for lack in confidence, which is unique to the proposed approach, is the confidence one can put in the accuracy of the results of stochastic simulations. By nature, the results of stochastic analysis are uncertain, i.e. they are not deterministic but vary from one evaluation to another. When estimating a failure probability by stochastic simulation, this probability is an uncertain quantity itself. Dependent on the selected parameters of the stochastic algorithm, e.g. the number of samples used, the accuracy of the estimation varies. As shown in chapter 2 and already extensively used for quantification of uncertainties during requirements derivation, for every estimation also confidence intervals results. Given that a failure probability $P_{F,req}$ must be ensured by stochastic simulations and that a certain level of confidence must be achieved, the probability that must be shown by simulation can be determined, see figure 5.5.

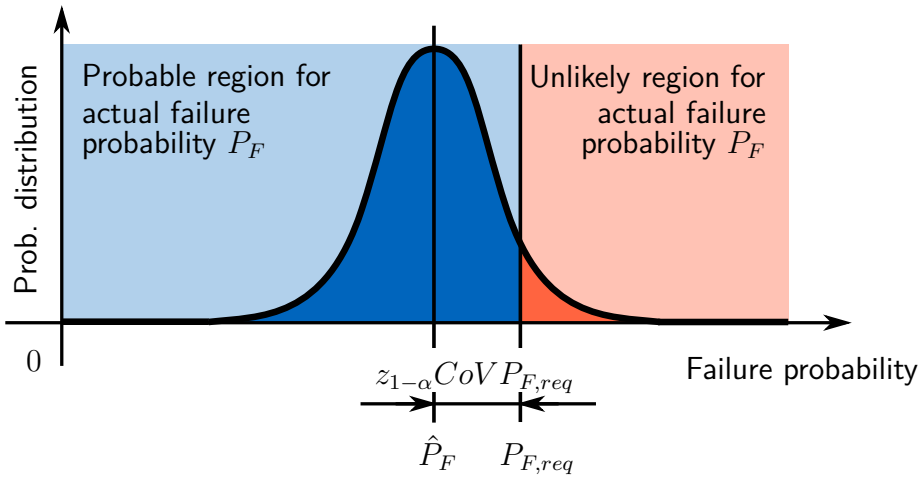


Figure 5.5: *Consideration of confidence intervals for probabilistic verification*

Recall that the CoV is by definition normalized by the actual failure probability P_F and not the estimated quantity $\hat{P}_F$:

$$CoV\left[\hat{P}_F\right] = E\left[\frac{\hat{P}_F - P_F}{P_F}\right] \tag{5.1}$$

Input $P_F = P_{F,req}$ and using the value of $CoV\left[\hat{P}_F\right]$ from the actual estimation, which can be calculated based on the Subset simulation results (2.112)-(2.117), the estimated failure probability to be proven by stochastic simulation for a desired $(1-\alpha)$ quantile is

$$\hat{P}_F = P_{F,req}\left(1 - z_{1-\alpha}CoV\left[\hat{P}_F\right]\right) \tag{5.2}$$

where the $(1-\alpha)$-quantile according to the definitions given in section 2.3.3 is used, with values for $z_{1-\alpha}$ given in appendix A.2. For example, consider that the acceptable probability for the case that the actual failure probability P_F is larger than the probability to be proven $P_{F,req} = 10{-}6$ is 2%, i.e. $\alpha = 0.02$, which corresponds to $z_{1-\alpha} = 2.05$, see table A.1. If in the example the coefficient of variation obtained by Subset simulation is $CoV = 0.3$, then the estimated probability must be smaller than $P_{F,req}\left(1 - z_{1-\alpha} CoV\left[\hat{P}_F\right]\right) = 3.85 \cdot 10^{-7}$.
A big advantage of enhanced stochastic analysis is the high number of samples which lie close to the failure domain. Each sample is linked to a specific combination of realizations of uncertain parameters. Due to the high number of critical samples it is possible to determine the sensitivity of the estimated failure probability with respect to each uncertain parameter and by that the influence of each uncertain parameter on specific failure conditions. For example, if the distribution of realizations of a specific uncertain parameter close to the failure domain, i.e. at higher subset levels, is similar to the original parameter distribution, it can be concluded that its influence on the considered failure condition is small. On the other hand, a significantly different distribution than the original distribution of the uncertain parameter indicate a high influence of the specific parameter and hence reduction of this uncertainty could reduce the failure probability.
Enhanced stochastic simulations are a powerful tool to test safety-critical functions by analysis and hence are a useful addition to the means of compliance for verification alongside many different other methods that are required to check the various different kinds of requirements. Certification requires an agreement with authorities on proposed means of compliance for each requirement, where the application of enhanced stochastic methods for analysis of small failure probabilities promises a high potential.
Verification of safety-critical functions using stochastic methods relies on the conformation of the conditions considered during verification and those met in reality. For conventional certification of aircraft as it is done today, a product is considered to be safe if certified, and this is only reconsidered if the opposite is shown by recorded incidences or accidents from daily operation. This is possible due to the high level of conservatism of today's approach, which usually provides a large buffer for uncertainties and disturbances. However, solely trusting on one-time verification for certification is not reasonable when applying a probabilistic development approach on novel aircraft topologies and operations due to several reasons. This will be discussed in the next chapter, where it is shown how runtime verification can facilitate even higher levels of safety and availability when following the TCA.

6

Runtime Verification

6.1 Introduction

Following a conventional development process, the development ends with demonstration of compliance with all requirements during verification phase. For the development of safety-critical functions of the aircraft, this also includes the compliance with the certification specifications and hence proves safety. Only after verification, the resulting system can be used for regular operation.

As opposed to this, runtime verification extends the verification efforts beyond the conventional development process towards regular operation, i.e. the correct functioning of a product according to requirements is (also) checked during its use. In general, runtime verification is defined as "*the discipline (...) that deals with the study, development and application of those verification techniques that allow checking whether a run of a system under scrutiny satisfies or violates a given correctness property*"[LS09]. For safety-critical aircraft functions, correctness properties refer to safety requirements, which ensure safe operation of the considered function. For conventional offline verification, theoretically all possible combinations of system and function states, uncertainties, disturbances and failures must be evaluated to ensure compliance with the established requirements for any possible condition that could be encountered in reality. Opposed to this, for runtime verification, only compliance of the system under the prevailing conditions is evaluated, which includes the current system states, environment and uncertainties.

One important difference between offline and runtime verification are the means used for verification. For many requirements, especially for the development of flight control systems, analysis is the preferred method for offline verification (see 5.2.2). Analysis heavily relies on the models of the implemented function, uncertainties, disturbances and failures. When using models, perfect information is available. Opposed to this, runtime verification can only utilize limited information to yield a certain verdict about compliance. While theoretically infinite measurements could be used for runtime verification, in practice only a finite number of uncertain measurements is available. Furthermore, limitations in computational performance restrict the range of usable algorithms to ensure real-time capability of runtime verification.

In a wider context, runtime verification does not only include monitoring of compliance, but it also deals with making the right decisions based on monitored results and triggering the right countermeasures to mitigate the effects of non-compliant and possibly unsafe system behavior. Figure 6.1 shows the chain of tasks for runtime verification. First, system states are captured by measurements. An online monitor uses this data to extract relevant system information, which could be, for example, calculated based on a series of measurements using operations in time or frequency domain. The objective of this task is to detect non-compliant behavior. System, components and the resulting behavior are diagnosed in case of abnormalities to determine the reason of the violation, e.g. specific failures, disturbances, or uncertainties. Finally, adequate countermeasures are identified and initiated by the mitigation task. This could be for example a change to another control mode or operation, which is specifically designed to cope with the identified problem.

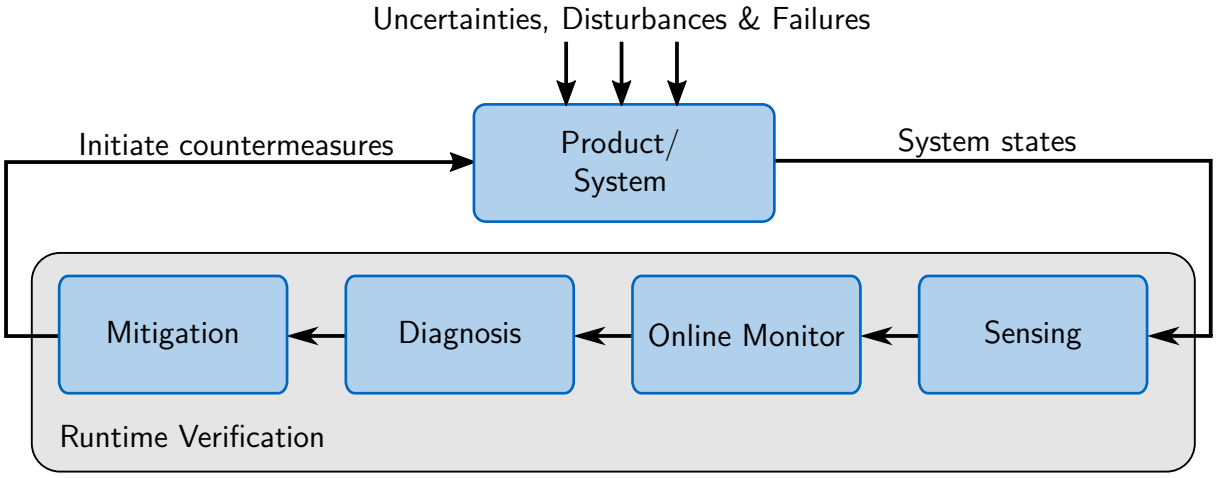


Figure 6.1: *Tasks of runtime verification*

Runtime verification is certainly only applicable to non-essential functions, i.e. for functions, which are not vital for operation and where adequate mitigation means or operations like alternative control modes are available. For fail-operative functions, which must still be able to operate adequately in case of failures, a detection of non-compliance would not be usable to prevent critical conditions due to the lack of adequate mitigation means. Hence, safety of essential functions cannot be ensured by runtime verification.

In this chapter, first runtime verification is motivated, followed by the introduction of objectives of online monitoring and a literature review of the state-of-the-art for runtime verification and online monitoring. Eventually a high level online monitoring algorithm is proposed that can be used to verify dynamic behavior of systems in the presence of uncertainties and disturbances, which is especially of relevance for detecting non-compliant behavior of controlled (aerial) systems.

6.2 Motivation

Although the concept of runtime verification sounds reasonable, it has not been used much in the past, mainly since conventional offline verification was the only admissible mean to prove safety and achieve adequate levels of availability of considered functions and operations. The achieved levels of safety using the conventional approach are quite qualitative and the conventional approach is limited to standard operations, which, however, is a showstopper for future applications. Three main motivators for runtime verification can be identified:

1. Novel control methods: Runtime verification is essential for novel control methods, which cannot be fully evaluated offline, e.g. for adaptive controllers, where the controller adapt to the real plant, environment, uncertainties and failures. For certain conditions, adaptation might fail, which could cause uncontrollability. For runtime verification, it is often sufficient to prove the correct system behavior only for the current aircraft state, uncertainties and disturbances. Hence, it is one promising mean to ensure a safe behavior of a controller using such novel control methods.

2. Novel operations with varying safety goals: Safety goals for conventional aircraft operations are invariant. Depending on the effects of failures of safety critical functions, a safety assessment results in a constant admissible failure probability and required availability of these functions. Hence, safety is mainly driven by the effects on aircraft, crew and passengers. For novel operations, the required level of safety could also be linked to operations. For example, it is less critical if an unmanned aerial vehicle crashes in an uninhabited region, which only causes an aircraft and hence financial loss, compared to a crash into a crowd of people, which could cause fatalities. While certain disturbances, uncertainties and failures might be still acceptable in one case, they might not be in other cases. Hence, it would be beneficial if everything is not only verified once during (offline) verification to prove static safety goals, but instead also during operation by runtime verification. This allows to dynamically adapt the actual required level of safety to operation and hence can significantly contribute to increased availability of a product, which would not be the case under worst-case assumptions for the required level of safety applied during offline verification.

3. System degradation: The performance of a system can degrade over time, e.g. due to wear or undetected defects. Continuous monitoring by runtime verification can detect non-compliant behavior before failures occur and hence can contribute to higher reliability of the monitored system. To some extend, this is already done today by condition-based maintenance, where measured data is used to schedule maintenance tasks. For *Flight Operational Quality Assurance* (FOQA), data of individual flights collected by a *Quick Access Recorder* (QAR) is analyzed, however, usually only offline after a flight is finished. Early work for online monitoring especially focused on health monitoring in rotorcraft industry, where *Health and Usage Monitoring System* (HUMS) are used to improve safety, mainly by monitoring caution and warning information together with vibration data from rotor and drivetrain components [Hes05]. Runtime verification can facilitate a sophisticated monitoring of system health, es-

pecially when using a model-based approach to identify non-compliant conditions.

6.3 Challenges and Objectives

In the previous sections, the limitations of offline verification are highlighted, which is especially the inability to test every possible combination of system states, uncertainties, disturbances and failures for complex systems. At first glance, runtime verification seems to be simpler, since safety must only be determined for the current condition. However, there are several challenges that must be addressed when implementing runtime verification. Certainly, runtime verification is only useful, if the level of confidence in the specific monitor implementation is high. The challenges of high-assurance runtime verification were evaluated in detail by Goodloe [Goo16]. The identified challenges can be grouped into the following categories:

- **Source of specification**: Safety can only be achieved, if the monitor is specified correctly and hence can detect undesired or unintended behavior. Furthermore, the right actions must be specified and performed in case of non-compliance.

- **Observability**: Ideally, monitor specifications can be directly derived from system requirements. However, usually the signals required to yield a verdict can only be inaccurately measured or are not even observable. Furthermore, there is a high number of uncertainties that can impede detection of non-compliant behavior. Formalized requirements used during function design and verification cannot be used directly for online monitoring, since there is no perfect information about all system states, which is the case during model-based development.

- **Traceability**: If runtime verification is used as a verification mean to prove compliance with (safety) requirements, it is inevitable that specifications of online monitoring algorithms are traceable to safety requirements. According to [Goo16], monitor specifications should derive from system level requirements and assumptions that have been validated by experts.

- **Fault-tolerant runtime verification**: An online monitor for runtime verification is useless, if there is only a low confidence in the correctness of the monitoring result. According to [But08], runtime verification should be fault-tolerant, i.e. it must be able to work even if there is a single failure. Furthermore, monitor and monitored function should not have common failure modes, like overflows of measurements.

- **No harm**: In no case, runtime verification should do harm to the system due to wrong verdicts.

- **Correct monitors**: The specified monitors must be implemented in a correct and robust way.

The trade-off between level of conservatism and level of confidence and hence safety can be depicted by the decision matrix shown in figure 6.2. Ideally, non-compliance is correctly detected by the monitor ("true positive") and it does not lead to a wrong verdict about violation ("true negative"). However, it is also possible that the system complies with requirements while the monitor detects a violation ("false positive") as well as the opposite that the system does not behave as intended but this non-compliance is not detected ("false negative"). Wrong verdict of a compliant system reduces availability of the considered function since mitigations, which usually influences performance, would be triggered that are not necessary. However, the most critical case is the missed detection, since this can heavily influence safety. Hence, the probability for "false negative" must be very low to obtain an adequate confidence in the monitor, while the probability for "false positive" should be low to achieve high availability. Unfortunately, these requirements are usually contradictory – decreasing the level of conservatism by reducing the probability for "false positive" often leads to an increase of non-detected adverse behavior. Hence, the challenge is to achieve a certain desired level of safety, while maximizing the level of availability.

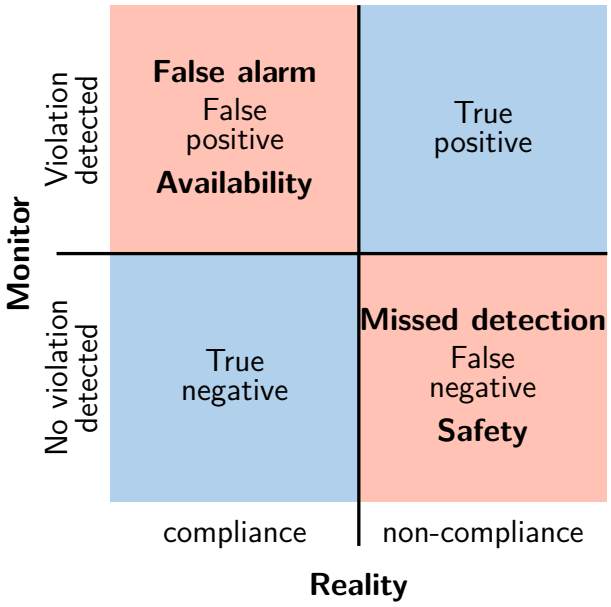


Figure 6.2: *Decision matrix of an online monitor*

Despite the high number of challenges, [Goo16] concluded that the advantages of high-assurance runtime verification outweigh the efforts required to tackle these challenges.
In the context of developing safety-critical functions according to the TCA, especially the following objectives can be identified:

- **Detection of non-compliant behavior**: The principle objective of runtime verification is to detect deviations between the observed behavior and the specified and admissible behavior. The non-compliance could be caused by software or hardware failures, uncertainties and disturbances. Depending on the source of non-compliant behavior, different mitigation strategies are required. The identification of non-compliant behavior is linked

to the monitoring task of runtime verification, identification of reasons and adequate countermeasures is related to diagnosis and mitigation.

- **Justification of assumptions**: Throughout the whole development process, assumptions are used to specify, design, implement and verify safety-critical functions. Runtime verification should prove that the assumptions made during the development were correct and, therefore, the system behaves as intended. A byproduct of this task is the extensive collection of data, which could be used to derive more accurate uncertainty models for future developments.

- **Determination of the current level of safety**: Online estimation of the current safety level enables variable safety goals driven by different operational needs. This can significantly increase the availability of the system under observation.

- **Predictive monitoring**: the applied monitor must have an adequate prediction horizon to be able to detect non-compliant behavior before a dangerous situation occurs and, therefore, be able to initiate adequate countermeasures and prevent critical situations.

If the listed challenges and objectives are considered during development of runtime verification means, safety also of novel aerial applications and operations can be guaranteed. Furthermore, the higher level of confidence in the system function and performance together with the lower level of conservatism of this approach promises a higher level of availability compared to conventional development approaches with only offline verification of the implemented functions.
The focus of the remainder of this chapter lies on monitoring algorithms, since all other tasks, especially sensing and mitigation, highly depend on the actual function and can, therefore, not be discussed in general.

6.4 Monitoring Algorithms

6.4.1 State of the Art

In the beginnings of online monitoring, simple comparisons of measurements with thresholds were made, e.g. for early HUMS [Hes05], which resulted in warning or alert annunciations. The application of such monitoring approaches are limited to very specific types of criteria.
Since the early 1990s, model-based monitoring approaches [DK89, HCK92] are widely applied in *Diagnosis* (DX) and *Fault Detection and Isolation* (FDI) in many technical applications, also in aviation [Kar+93, PCN11]. Model-based approaches for online monitoring significantly increase the range of non-compliant behavior that can be detected, since the description of such behavior is no longer limited to simple comparisons of measurements. This opens a wide field of applications. *Bateman* et. al. [Bat+05] split up high risk components of the control system of a UAS, where the high risk components include functions that cannot be

verified offline, which in, the specific example, is an adaptive controller. A big increase in success rate for simulated shipboard landings could be achieved by using a run-time safety monitor that compares a priori estimates of expected aircraft response with the actual aircraft behavior. If required by the monitor, a switching is made to a less-sophisticated but fail safe controller, which is fully verified and validated at design time. *Bak* et. al. [Bak+14] researched on a verified simplex design, which again uses an adaptive controller with high performance and low level of conservatism together with a back-up controller that is fully verified during design time. Instead of monitoring the behavior of the sophisticated controller and switch to the fail-safe controller in case of non-compliant behavior, the referred approach monitors the current aircraft state and compares it to the envelope of the fail-safe controller, within which this controller is able to control the aircraft back to a safe condition. Only right before the boundary of this envelope is reached, the switching is conducted from the sophisticated controller to the fail-safe controller. For determination of the envelope, a mixture of offline analysis using linear matrix inequalities [SS99] and real-time reachability analysis [DM98] is used.

Monitoring can be applied to different levels, from item level to system level. While it is most important for safety to monitor the performance on system level, monitoring on function and item level can significantly ease diagnosis and decision making for adequate mitigation actions. An example for item level monitoring are redundant items with appropriate architectures and voting algorithms. Using such an approach can prevent non-compliant behavior already on item level before non-compliance is detected on system level. Goodloe [GPC10] researched on different architectures for monitoring of distributed real-time systems. Although this is beyond the scope of this thesis, such architectural considerations must be included during implementation of monitors, to prevent, for example, unintended common cause failures.

In the following, a new monitoring algorithm for system level requirements of controlled systems is proposed that can be used to monitor imprecise plants in the presence of uncertainties and disturbances. This is one important case in the context of developing safety-critical functions following the TCA.

6.4.2 Proposed Monitor Using Incremental Predictions

Models are suitable to monitor behavior that cannot be directly measured. They are used to simulate the admissible behavior of a system during runtime and the simulation results are continuously compared to the actual response. The algorithm proposed in this section is useful for reactive systems, i.e. systems where the input and error behavior can be quantified. Aircraft using an (automatic) flight control system are one example for reactive systems, where the desired input and error response are even specified in respective requirements. By comparing the actual response with the admissible range of specified response, non-compliant and hence possibly unsafe behavior can be detected.

If the system is developed following the TCA, models for description of the admissible system behavior are readily available from system specification, design and verification, which also

includes uncertainties, disturbances, etc. However, these models cannot be directly used, since for online monitoring there is no perfect information available, but instead only measurements of a limited number of signals, which are furthermore distorted by measurement uncertainties.

The main objective of the algorithm proposed in this section is to monitor the high-level response of a system in order to detect non-compliant behavior that can have significant influence on safety. Although the reasons for high-level violations might emerge from component or function errors, these cases can be well covered by conventional architectural decisions and according monitoring solutions, e.g. redundancy (duplex, triplex or dual-duplex) and respective voting concepts.

One major type of top-level requirements describes the admissible dynamic behavior of a system, i.e. the response to inputs and disturbances. The developed algorithm described in this section is applicable to this type of requirements, where a monitoring approach is used that utilizes the known admissible range of system behavior, specified in related requirements, and measurements from the partially observable plant dynamics as well as the knowledge about system uncertainties and stochastic processes. The approach is based on research by Rinner and Weiss [RW02, RW04] that combines FDI and DX methods for systems where the structure is known and the admissible dynamic range is defined within bounded, numerical intervals. Compared to the original approach, the derived algorithm described in this section allows continuous online monitoring of the compliance of a system with safety relevant requirements also in the presence of uncertainties and stochastic disturbances that cannot be adequately described by bounded intervals used solely in the underlying approach. The detailed descriptions given in this subsection focuses on the algorithms for monitoring and uncertainty propagation. Further activities related to uncertainty prognostics [SG14] like uncertainty representation, interpretation and quantification is beyond the scope.

First the monitoring algorithm is derived, uncertainty propagation is discussed as well as the update of propagated uncertainties by measurements. Subsequently, the principle function of the algorithm is explained using a simple generic system, while the application for a realistic plant is shown in 7.3.4. The algorithm described in this section was first presented and published by the author of this thesis on the AIAA Atmospheric Flight Mechanics Conference 2016 [LMH16a].

Imprecise Modeling

First, the principle idea of online monitoring using imprecise models based on the algorithms of Rinner and Weiss [RW02, RW04] is described, followed by the transition from bounded intervals to a probabilistic formulation that then also allows the incorporation of stochastic processes.

Imprecise models can be used to describe systems where the architecture is known but certain

parameters are not or not exactly known. Such models can be written as

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{p}) \\ \mathbf{y}(t) &= \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{p}) \end{aligned} \tag{6.1}$$

where $\mathbf{x}(t)$, $\mathbf{u}(t)$, $\mathbf{y}(t)$ and $\mathbf{p}$ denote the state, input, output and uncertain parameter vector, respectively, and $\mathbf{f}$ and $\mathbf{g}$ are vector functions. For imprecise modeling, it is assumed that the exact value of the uncertain parameter vector is unknown and only bounds are known. Therefore, $\mathbf{p}$ is replaced by an uncertain parameter interval vector

$$\tilde{\mathbf{p}} = [(p_{1,low}, p_{1,up}), (p_{2,low}, p_{2,up}), \ldots, (p_{k,low}, p_{k,up})]^T \tag{6.2}$$

where the indices "low" and "up" denote the lower and upper interval bounds for a k-dimensional uncertain parameter space. For this type of model, it is sufficient to only investigate the external surface of the uncertain parameter space [BB94]. However, this would still require an infinite amount of trajectories to be evaluated. Under certain conditions, which are discussed soon, it suffices to propagate the system trajectories only at the 2^k corner points of the parameter space since all other trajectories are guaranteed to lie between these bounding trajectories. This leads to the following system description, where $i = 1, 2 \ldots 2^k$.

$$\begin{aligned} \mathbf{X}(\mathbf{t}) &= \{\mathbf{x}_i(t) : \dot{\mathbf{x}}_i(t) = \mathbf{f}(\mathbf{x_i}(t), \mathbf{u}(t), \mathbf{p_i})\} \\ \mathbf{Y}(\mathbf{t}) &= \{\mathbf{y}_i(t) : \mathbf{y}_i(t) = \mathbf{g}(\mathbf{x_i}(t), \mathbf{u}(t), \mathbf{p_i})\} \end{aligned} \tag{6.3}$$

In the original approach presented in [RW04], it is assumed that the initial state $\mathbf{x}(0)$ is precisely known. In case only imprecise initial states are available, which is usually the case due to measurement uncertainties, those can be described as additional uncertain parameters. The evaluation only at corner points is valid as long as the states are monotonic with respect to the uncertain parameters $\mathbf{p}$. For determination of monotonicity, the gradient matrix $\mathbf{W}(t, \mathbf{x}, \mathbf{p})$ with the elements $w_{ij}(t, \mathbf{x}, \mathbf{p})$ is required:

$$w_{ij}(t, \mathbf{x}, \mathbf{p}) = \frac{\partial x_i(t, \mathbf{x}, \mathbf{p})}{\partial p_j} \tag{6.4}$$

w_{ij} represents the change of the i-th state with respect to the j-th uncertain parameter p_j. Monotonicity is only given if all w_{ij} have the same sign. For non-trivial cases, monotonicity can be calculated using the derivative of $\dot{\mathbf{x}}_i$ with respect to $\mathbf{p}$:

$$\begin{aligned} \frac{d\dot{\mathbf{x}}(t, \mathbf{x}, \mathbf{p})}{d\mathbf{p}} &= \frac{\partial \dot{\mathbf{x}}(t, \mathbf{x}, \mathbf{p})}{\partial \mathbf{x}} \frac{\partial \mathbf{x}(t, \mathbf{x}, \mathbf{p})}{\partial \mathbf{p}} + \frac{\partial \dot{\mathbf{x}}(t, \mathbf{x}, \mathbf{p})}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \mathbf{p}} + \frac{\partial \dot{\mathbf{x}}(t, \mathbf{x}, \mathbf{p})}{\partial \mathbf{p}} \\ &= \mathbf{A}(t, \mathbf{x}, \mathbf{p}) \mathbf{W}(t, \mathbf{x}, \mathbf{p}) + \mathbf{V}(t, \mathbf{x}, \mathbf{p}) \end{aligned} \tag{6.5}$$

where in the second line independence of $\mathbf{u}$ from $\mathbf{p}$ and the Hessian matrix $\mathbf{A}(t, \mathbf{x}, \mathbf{p})$ as well

as the parameter sensitivity matrix $\mathbf{V}(t, \mathbf{x}, \mathbf{p})$ are used, which have the following components:

$$\begin{aligned} a_{ij}(t, \mathbf{x}, \mathbf{p}) &= \frac{\partial \dot{x}_i(t, \mathbf{x}, \mathbf{p})}{\partial x_j} \\ v_{ij}(t, \mathbf{x}, \mathbf{p}) &= \frac{\partial \dot{x}_i(t, \mathbf{x}, \mathbf{p})}{\partial p_j} \end{aligned} \tag{6.6}$$

Given that the partial derivatives $\dot{\mathbf{x}}$ themselves are differentiable, which is usually the case for dynamic systems, then $d\dot{\mathbf{x}}(t, \mathbf{x}, \mathbf{p})/d\mathbf{p}$ equals $d\mathbf{W}(t, \mathbf{x}, \mathbf{p})/dt$ with w_{ij} from (6.4). The interchange of the order of derivatives is usually only possible for partial differential equations according to the Schwarz integrability condition [Wes15, p. 406]. However, according to demonstrations of Moore indicated in [BB94], under the given conditions, this is also true for the total differential in (6.5). This yields a differential equation, which can be integrated over time during monitoring to determine the signs of $w_{ij}(t, \mathbf{x}, \mathbf{p})$ and hence monotonicity. For the differential equation, the initial condition $\mathbf{W}(0, \mathbf{x}(0), \mathbf{p}) = \mathbf{0}$ is used, which indicates that there is no possible variation at $t = 0$.
For obvious reasons, monotonicity is generally not given for arbitrary systems and integration intervals, e.g. for systems with complex conjugate eigenvalues. However, for not too large parameter intervals and small integration periods, i.e. higher measurement update rates, monotonicity is given for many relevant cases. This is demonstrated later using a simple generic example.

Intersection of Propagation and Measurements

The state vector trajectory for each corner point can be propagated using equation (6.3). The trajectories would diverge quickly if no update of the propagated uncertain state and output space is done. Therefore and to maintain monotonicity, a new initial state interval is set during each measurement update, hence limiting prediction time by the period between two measurements.
Figure 6.3 shows the principle process of the measurement update for a two-dimensional system. The box A depicts the initial state intervals for x_1 and x_2 at time $t - \Delta t$, where it is only known that the true state $\mathbf{x}(t - \Delta t)$ lies within the intervals. Each corner of the initial state space is predicted 2^k times, i.e. for each parameter limit combination. In the example shown in figure 6.3, there is only one uncertain plant parameter, i.e. $k = 1$. Hence, each corner point of box A is propagated for the lower and upper uncertainty parameter bound. This results in separate trajectory spaces for the individual corner points of the uncertain parameter space at time t. Box C represents an approximation of the overall trajectory space envelope, which includes all subspaces B. If the system response is monotonous with respect to the parameter, the response using any parameter between the limits lies inside box C. The updated initial state value interval for time t, which is the basis for the next prediction step, is given by box E, which results from the intersection of the measurement uncertain space D and the predicted state space C. When there is no overlapping between the predicted state space

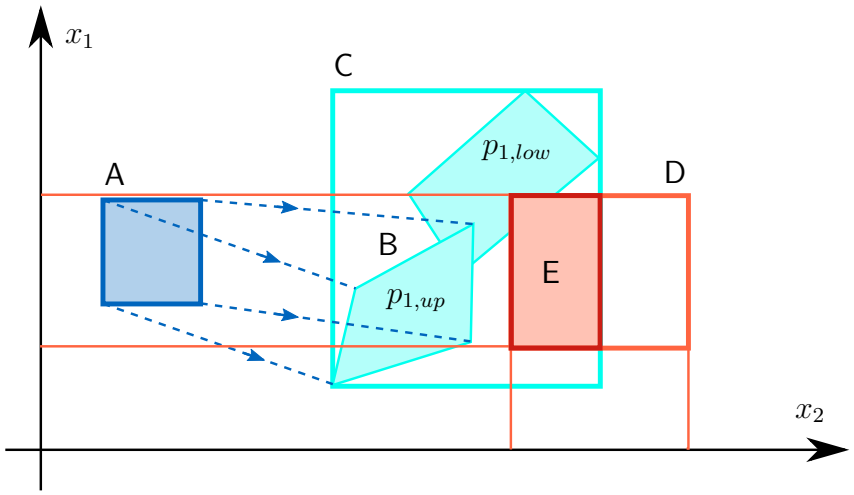


Figure 6.3: *Generating new initial state intervals (based on [RW04])*

and the measurement space and the predictions are monotonic with respect to the uncertain parameters at time t, non-compliant system behavior is detected. For the case that x_i is not monotonic with respect to an uncertain parameter, e.g. due to too long propagation times, no statement about the validity of the predicted bounds can be made. Hence the measurement space D must be used as new initial state space. In this case, no verdict about compliance can be made.

Uncertainty Propagation

With the algorithm described so far, only uncertain parameters of the plant dynamics with bounded intervals can be examined and neither realistic distributions of uncertainties nor disturbances can be taken into account. This does not represent many physical systems where uncertain parameters are defined by distributions and stochastic processes affect the overall system dynamics. Hence, the original approach is extended so that it also allows for consideration of stochastic uncertainties.

Usually uncertain parameters are not uniformly distributed as supposed by the above-described algorithm but they are rather distributed about a mean with decreasing probabilities for higher deviations from the mean. Hence, by only propagating corner points, unlikely combinations are overestimated. To overcome this disadvantage, this modification propagates the mean of the state vector together with the corresponding covariances. While the original concept of imprecise modeling is kept for design parameters, where the admissible interval is usually specified in requirements, only the mean of the state vector instead of the boundaries of state errors is propagated:

$$\begin{aligned} \mathbf{X}(\mathbf{t}) &= \{\mathbf{x}_i(t) : \dot{\mathbf{x}}_i(t) = \mathbf{f}(\mathbf{x_i}(t), \mathbf{u}(t), \mathbf{p_i}, \mathbf{v}(t))\} \\ \mathbf{Y}(\mathbf{t}) &= \{\mathbf{y}_i(t) : \mathbf{y}_i(t) = \mathbf{g}(\mathbf{x_i}(t), \mathbf{u}(t), \mathbf{p_i}, \mathbf{v}(t))\} \end{aligned} \tag{6.7}$$

where $\mathbf{x_i}$ is the mean state vector for the i-th uncertain parameter corner. In contrast to

equation (6.3), there is an additional Gaussian process noise vector $\mathbf{v}(t) \in \mathbb{R}^{n_v}$, which allows incorporation of uncertain stochastic processes.

Uncertainty propagation of nonlinear equations require the solution of the Fokker-Planck equation, which is a partial differential equation that can only be solved analytically for special cases [Ris96]. Approximation schemes for all other cases often require high computational effort (e.g. for stochastic simulations) or are limited to the determination of the steady state uncertainties. Both are no option here. However, for small integration times, consideration of linearized dynamics at least for uncertainty propagation is usually acceptable. The admissible dynamics is even often already specified using linear dynamics, in which case the Fokker-Planck equation degrades to a linear optimization problem that can be solved using linear filtering and prediction commonly applied for Kalman filters [Kal60]. Reducing (6.7) to linear state space models yields the following expression:

$$\delta\dot{\mathbf{x}}_i(t) = \mathbf{A}_i(t)\,\delta\mathbf{x}_i(t) + \mathbf{B}_i(t)\,\delta\mathbf{u}(t) + \mathbf{G}_i(t)\,\delta\mathbf{v}(t) \tag{6.8}$$

where the state, input, and disturbances matrix $\mathbf{A}_i(t)$, $\mathbf{B}_i(t)$ and $\mathbf{G}_i(t)$ are obtained by linearization at each time step, with $\delta\mathbf{x}_i$ being the offset from $\mathbf{x}_i$ obtained by nonlinear simulation using (6.7) [SC08, p. 244]. For the numerical implementation, (6.8) is discretized:

$$\delta\mathbf{x}_{i,k+1} = \mathbf{\Phi}_{i,k}\delta\mathbf{x}_{i,k} + \mathbf{\Gamma}_{i,k}\delta\mathbf{u}_k + \mathbf{\Upsilon}_{i,k}\delta\mathbf{v}_k \tag{6.9}$$

where $\mathbf{\Phi}_{i,k}$, $\mathbf{\Gamma}_{i,k}$, $\mathbf{\Upsilon}_{i,k}$ are the discrete state, input and disturbance transition matrix at time step k for the i-th uncertain parameter corner, which can be calculated by [ETP90, p. 8.196]

$$\begin{aligned}
\mathbf{\Phi}_{i,k} &= e^{\mathbf{A}_i\Delta t} \approx \mathbf{I} + \mathbf{A}_i\Delta t + \mathbf{A}_i^2\frac{\Delta t^2}{2} \\
\mathbf{\Gamma}_{i,k} &= \int_0^{\Delta t} e^{\mathbf{A}_i s}ds\mathbf{B}_i \approx \left(\mathbf{I}\Delta t + \mathbf{A}_i\frac{\Delta t^2}{2}\right)\mathbf{B}_i \\
\mathbf{\Upsilon}_{i,k} &= \int_0^{\Delta t} e^{\mathbf{A}_i s}ds\mathbf{G}_i \approx \left(\mathbf{I}\Delta t + \mathbf{A}_i\frac{\Delta t^2}{2}\right)\mathbf{G}_i
\end{aligned} \tag{6.10}$$

Note that equation (6.9) is only used for (linear) uncertainty propagation. For actual state propagation, the possibly nonlinear equations (6.7) can be directly used. The state covariance matrix $\mathbf{P}_i$, describing the uncertainties of the predicted state for the i-th parameter corner, is propagated using the equations for linear filtering and prediction of a Kalman filter [Kal60].

$$\mathbf{P}_{i,k+1} = \mathbf{\Phi}_{i,k}\mathbf{P}_{i,k}\mathbf{\Phi}_{i,k}^T + \mathbf{\Upsilon}_{i,k}\mathbf{Q}_k\mathbf{\Upsilon}_{i,k}^T \tag{6.11}$$

$\mathbf{Q}_k$ describes the covariance of stochastic uncertainties at the k-th time step and is usually referred to as process noise. After each measurement, the state covariance $\mathbf{P}_{i,0}$ is reset to the intersection between the predicted uncertain state space and the measurement covariance, see area E as intersection of C and D in figure 6.4. The measurement update is not done

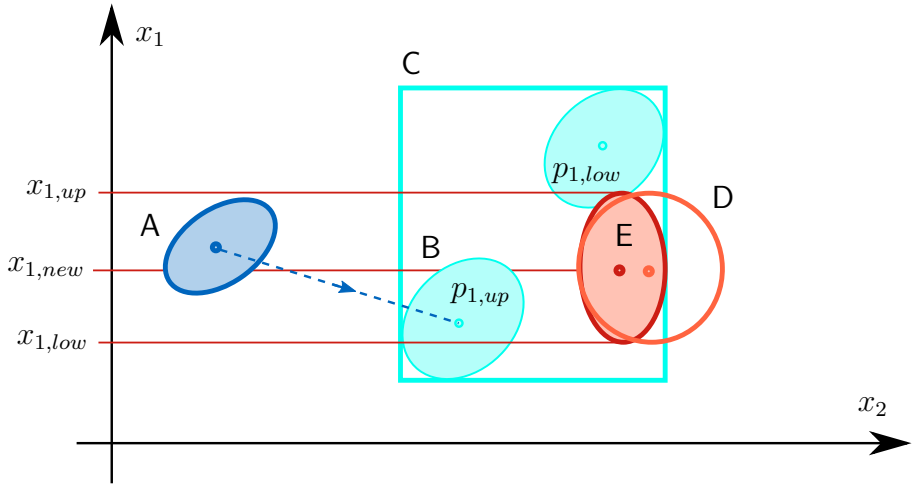


Figure 6.4: *Generating new initial state covariances*

in the fashion of a conventional Kalman filter, where the most likely intersection is selected based on the prediction and measurement states and uncertainties, since this would only give the most probable state, while for online monitoring, the objective is to identify the overall possible range of states where measurements and predictions coincide.
Figure 6.5 eventually gives a schematic flow chart of the monitoring algorithm. Although

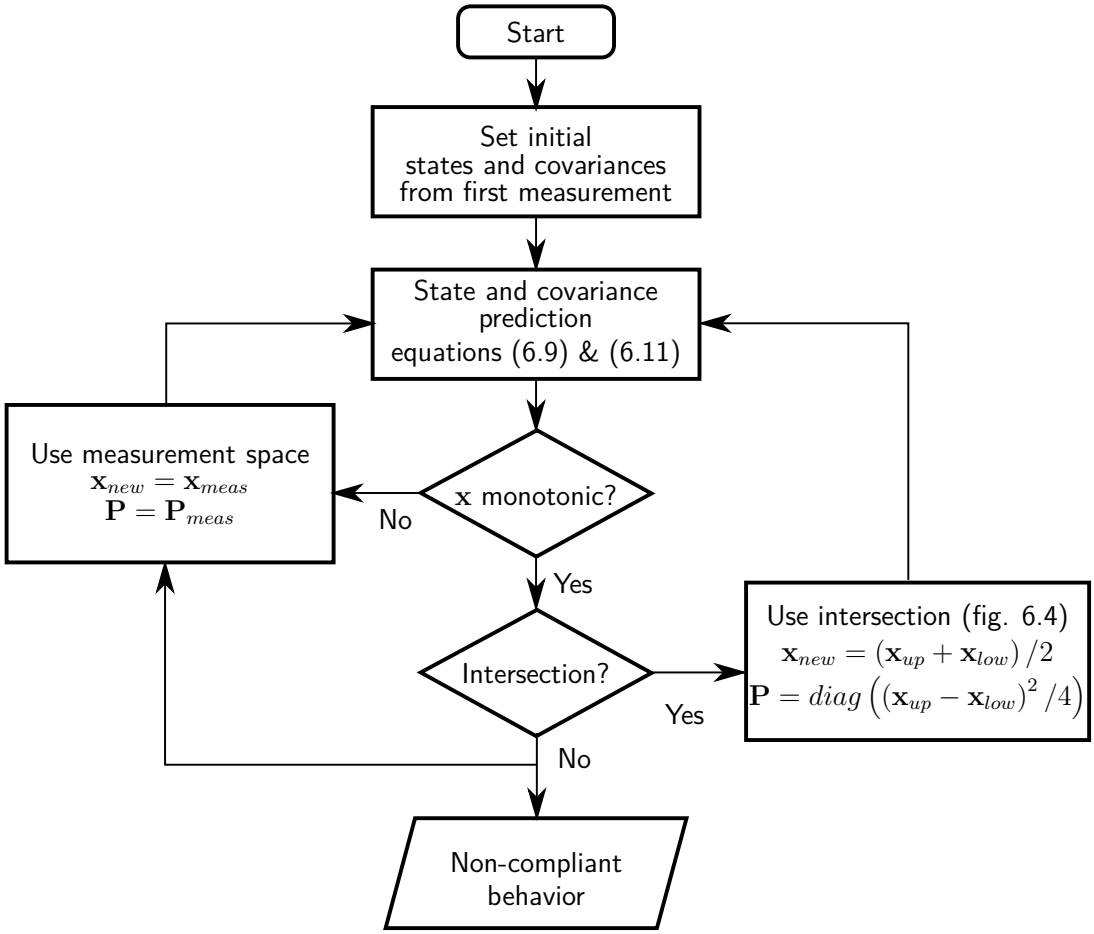


Figure 6.5: *Flowchart of monitoring algorithm*

the stated algorithm relies on linear optimal prediction, this is no severe limitation, since requirements on system dynamics are often formulated using linear models. Furthermore, uncertainties and stochastic processes with non-Gaussian distributions can be modeled by Gaussian mixture models [Rey09]. This is not further discussed in this thesis.

Explanation of Principle Using a Generic Example

The objective of the evaluation in this subsection is to explain the proposed monitoring algorithm and show the performance during stochastic excitation. A linear second-order system is used as generic example. System dynamics can be described by the following equations:

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -\omega_0^2 & -2\zeta\omega_0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \omega_0^2 \end{bmatrix} u + \begin{bmatrix} 0 \\ \omega_0^2 \end{bmatrix} v \\ y &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \end{aligned} \tag{6.12}$$

with the natural frequency ω_0 and relative damping ζ being uncertain parameters that are bounded by intervals. Furthermore, white noise v acts on $\dot{x}_2$. For analytical analysis of monotonicity, the differential equation (6.12) can be solved exemplarily for constant inputs u, resulting in an expression for the state error $\Delta x_1 = x_1 - u$ using $\omega = \omega_0\sqrt{1-\zeta^2}$ and $\sigma = -\zeta\omega_0$ for the frequency and absolute damping of the oscillation:

$$\Delta x_1 = e^{\sigma t}\left(c_1 \sin\omega t + c_2 \cos\omega t\right) \text{ for } \zeta < 1 \tag{6.13}$$

For $x_0 = 0$ and $\dot{x}_0 = 0$, the relative error of $\Delta x_1/u$ results:

$$\frac{\Delta x_1}{u} = e^{\sigma t}\left(\frac{\sigma}{\omega}\sin\omega t - \cos\omega t\right) \tag{6.14}$$

Monotonicity is exemplarily analyzed for the parameter ω_0. The derivative of Δx_1 with respect to ω_0 yields

$$\begin{aligned} \frac{d\Delta x_1/u}{d\omega_0} &= \zeta t e^{\sigma t}\left(\frac{\zeta}{\sqrt{1-\zeta^2}}\sin\omega t - \cos\omega t\right) \\ &\quad + e^{\sigma t}\left(\frac{\zeta}{\sqrt{1-\zeta^2}}\sqrt{1-\zeta^2}\, t\cos\omega t + \sqrt{1-\zeta^2}\, t\sin\omega t\right) \end{aligned} \tag{6.15}$$

Figure 6.6 depicts the derivative of Δx_1 over the parameter ω_0 for different integration times and for a constant damping $\zeta = 0.71$. It can be seen that for integration times up to $t = 0.89s$, $x_1(\omega_0, t)$ is monotonic for ω_0 between 0.5 and $2.5s^{-1}$, since the values of the gradients $d\Delta x_1/d\omega_0$ are all positive within this range. From a different perspective, for an update interval of $0.1s$ and for $\zeta = 0.71$, the natural frequency interval can range up to $22s^{-1}$ without a negative effect on the monotonicity criterion. A more intuitive interpretation of monotonicity in this context is shown in figure 6.7, where state histories of x_1 for different

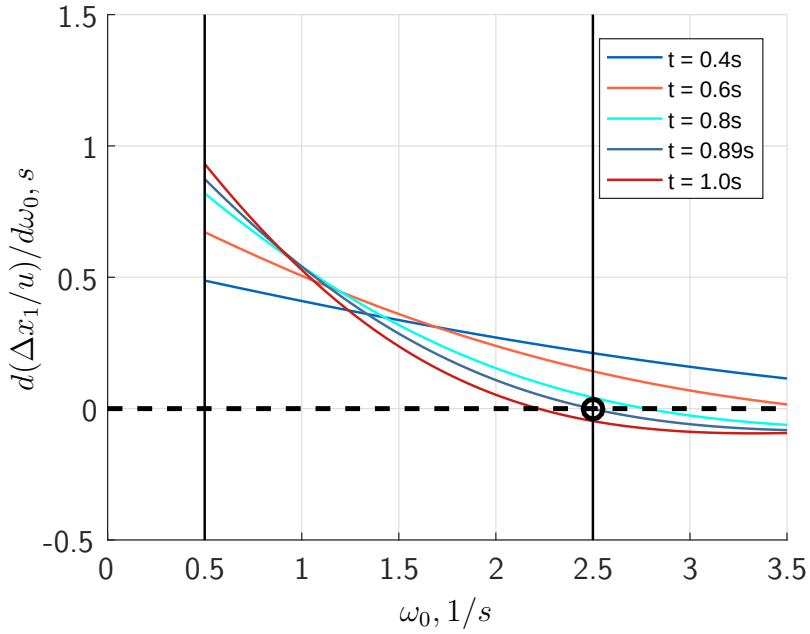


Figure 6.6: *Gradient of x_1 versus frequencies ω_0 for different integration times*

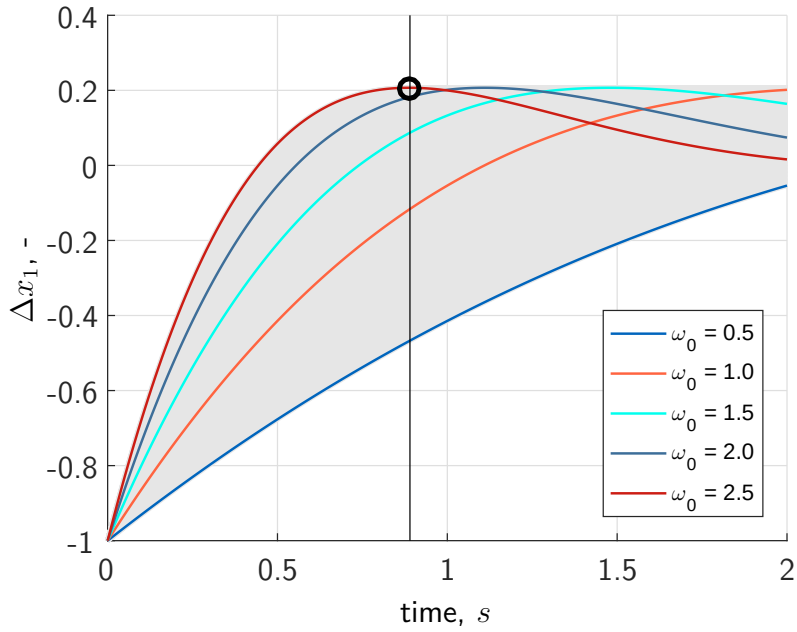


Figure 6.7: *Histories for x_1 versus time for different frequencies ω_0*

values of ω_0 are plotted.

The shaded area indicates the range of all possible trajectories for $\omega_0 = 0.5 \ldots 2.5 s^{-1}$. For integration times up to 0.89 seconds, it is guaranteed by the monotonicity property, that for any frequency value within the lower and upper parameter bound, the response lies within the trajectories associated to these limits, while for higher integration times this is no longer the case, which can be derived from the history of $\omega_0 = 2.5 s^{-1}$ that no longer forms the upper boundary of the shaded area for $t > 0.89$ seconds.

To demonstrate the principle of the proposed monitoring algorithm, several scenarios are simulated using the generic model (6.12) with two uncertain parameters ω_0 and ζ, limited

Table 6.1: *Simulation scenarios for generic example*

Case	Noise	Desired dynamics ω_0 range	Desired dynamics ζ range	Simulated ω_0	Simulated ζ
$1a$	No	$[1 \dots 2]$	$[0.3 \dots 1]$	1.5	0.7
$1b$	No	$[1 \dots 2]$	$[0.3 \dots 1]$	2.5	0.7
$1c$	No	$[1 \dots 2]$	$[0.3 \dots 1]$	0.2	0.7
$1d$	No	$[1 \dots 2]$	$[0.3 \dots 1]$	1.5	1.1
2	Yes	$[1 \dots 2]$	$[0.3 \dots 1]$	1.5	0.7

by discrete bounds together with white noise v representing stochastic disturbances. Table 6.1 shows the simulated scenarios.

Figure 6.8 shows the simulation results for cases $1a$ - $1d$. For each case, the states x_1 and x_2 are plotted for an arbitrary chosen control input (indicated by the black dashed line) and uniformly distributed, bounded measurement uncertainties $[-0.05, 0.05]$. Bounded measurement uncertainties are chosen to enable a separate analysis of the effects of non-compliant behavior due to exceedance of admissible plant parameters. These effects would be otherwise partly hidden by exceedance of single standard deviation bounds arising from measurement uncertainties. Plot a) corresponds to case $1a$, where the plant dynamics lie within the desired dynamics range. Plots b) to d) show the results for the cases where either the natural frequency of the plant exceeds desired limits or relative damping is too high. All non-compliant cases are successfully detected, where specific exceedances of the boundaries are marked by orange circles. Furthermore, figure 6.9 shows case 1a in combination with a steadily increasing measurement error for state x_1 triggered at 5 seconds, representing a faulty sensor. The closed-loop system follows the wrong measurement, which is why the measured response in x_1 still looks acceptable, while the actual trajectory diverges. However, since the measurements for x_2 do not correlate with the measured behavior of x_1, violation is detected quickly after error injection.

Figure 6.10 shows results for case 2, where the system is excited by white noise with zero mean and unit variance. To allow a separate consideration of the effects of stochastic uncertainties, perfect information, i.e. zero measurement errors are assumed. In contrast to online monitoring using only uncertainties with bounded intervals, there is always a certain probability of exceeding limits in the presence of stochastic uncertainties even if the plant dynamics complies with requirements. For plot a), single variance for the uncertainty propagation of the process noise, i.e. for the system uncertainty matrix $\mathbf{Q}$, is chosen. Since samples of white noise follow a Gaussian distribution, it is expected that approximately 32% of the monitored samples violate the single variance bounds. In the example, a slightly lower number of 27% of monitored samples exceed the estimated single variance limits, which can be explained by the additional buffer arising from the admissible range of plant parameters: Referring to figure 6.7, if $\omega_0 = 1.0$, there is a large buffer to the lower and upper boundary. Although disturbances force the trajectory towards the limits, the buffer leads to a delayed detection of exceedance,

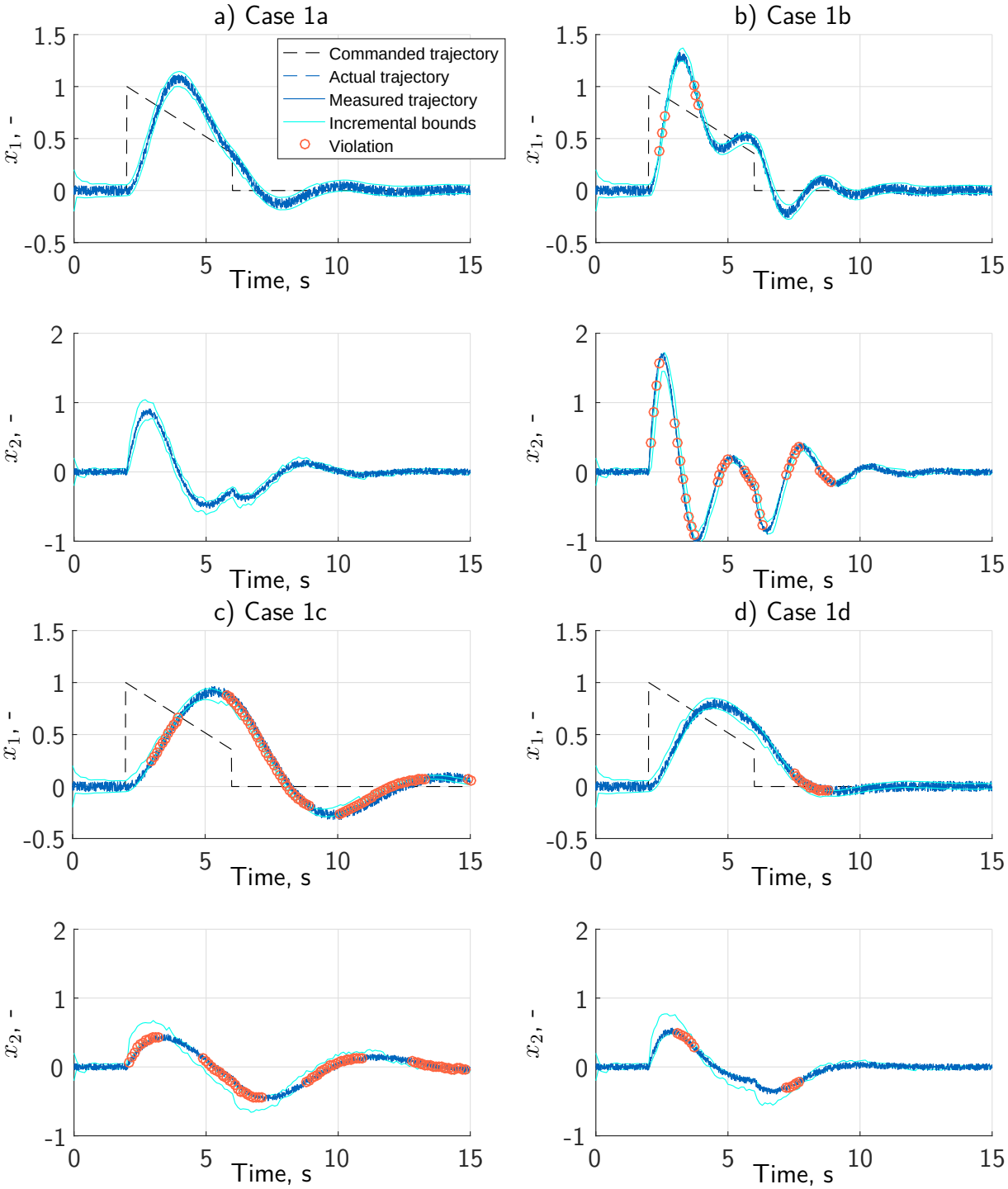


Figure 6.8: *Monitoring simulation results for cases* $1a$ - $1d$

especially for larger propagation times, when bounds become wider. Plots b) and c) show the same scenario but with double and triple standard deviation bounds used for the process noise $\mathbf{Q}$ during uncertainty propagation. This leads to detected exceedance rates of 4.5% and 0.4% respectively, which correlate well with the expected probability of exceeding 2σ and 3σ thresholds for a Gaussian distribution.

The proposed monitoring algorithm requires a responsive system, i.e. only in the case of control

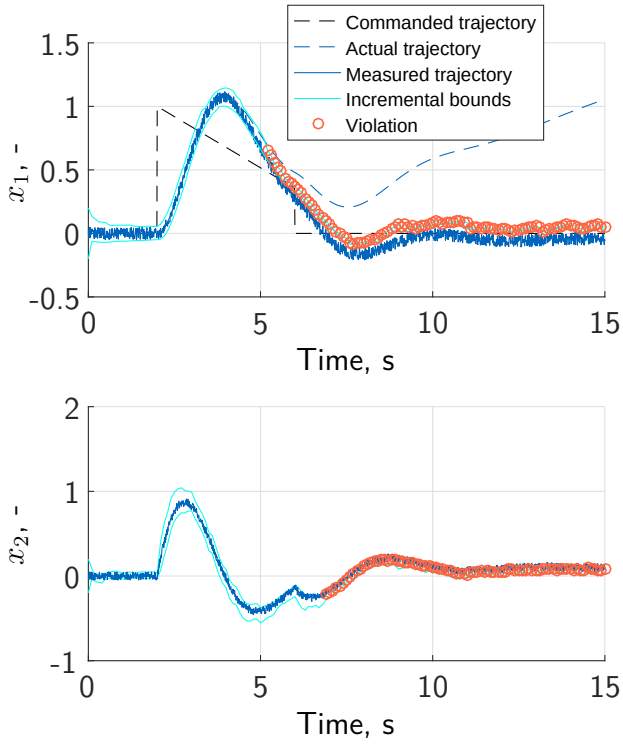


Figure 6.9: *Monitoring simulation results for case* 1*a with sensor failure*

inputs, disturbances, or failures that have an effect on measured aircraft states, non-compliant behavior can be detected. This becomes apparent in figure 6.8, where non-compliance is mainly detected during transient phases, while no violation is detected in case of constant control inputs. The higher the range of admissible design parameters and uncertainties are, the less accurate is the detection of non-compliant behavior. For example, if there are high measurement uncertainties, a temporary non-compliant behavior is hidden by uncertain measurements. However, since the propagation of uncertainties (6.11) also takes into account the dynamic relations between states and hence different measurements as well properties of the stochastic excitation, non-compliant behavior will still be detected, if it persists for a sufficiently long time.

Further Improvements of Monitoring and Prediction Performance

In the presented form, the monitoring algorithm can be used to ensure that a system complies with a range of admissible dynamics in the presence of uncertainties and disturbances, which is one common type of requirements. Certainly it is not possible to prove safety requirements with a very low acceptable violation probability online. However, the proposed algorithm provides measures for the performance of the plant at current uncertainties and disturbances. Comparing this performance to operation-dependent thresholds, which can be obtained offline to fulfill the probabilistic top-level safety requirements, it is possible to decide whether an operation is acceptable under the current conditions. For example, for an automatic landing, higher control performance is required close to ground. While off-line verification is used to evaluate the usually very low probabilities of critical events like runway excursions, related

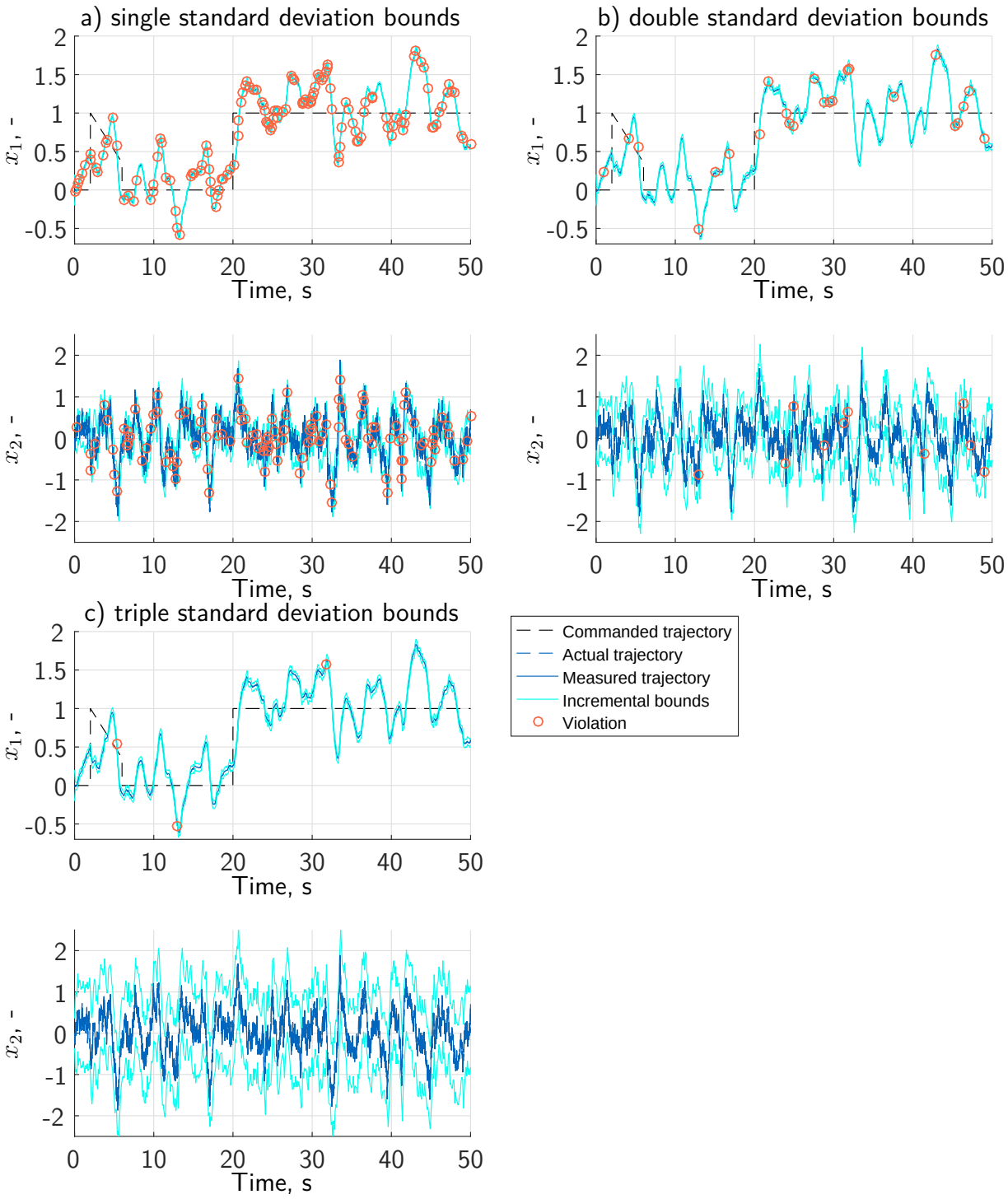


Figure 6.10: *Monitoring simulation results for case* 2

single or double standard deviation bounds can be extracted from these results, which then can be compared online with the monitored system performance. Furthermore, using the simulation environment with the implemented monitor, enhanced stochastic simulations can be conducted offline to prove that the actual probability for a critical event of the system with the monitor is lower than admissible.

One option to further reduce the level of conservatism of monitoring and prediction is the

detection of current levels of uncertainties. Instead of proving compliance for fixed acceptable ranges of uncertainties, the actual uncertainty range is identified during runtime. Using the proposed algorithm, this is possible by splitting up the ranges of individual uncertainties (e.g. the admissible plant parameter range) into smaller intervals. By only accepting those intervals during the update step, where the prediction intersects with the measurement, the level of conservatism of the prediction can be reduced. This is exemplarily shown in figure 6.11, where the single uncertain parameter $p_1 \in [p_{1,low} \dots p_{1,up}]$ is split up into two parameter intervals. Since the measurement only intersects with the box for $p_1 \in [p_{1,low} \dots p_{1,mid}]$, only this parameter range is used for the next prediction step. The same split up is possible for the intensity of stochastic disturbances. By that, the conservatism of the predicted bounds can be further reduced. Using the reduced range for uncertain parameters and the current level of disturbances, a prediction with longer time period (up to several seconds dependent on application and system dynamics) can be made to determine whether the current performance is sufficient to fulfill the desired operation and adequate measures can be triggered if emerging non-compliance is detected. Hence the proposed algorithm can be used to fulfill all objectives for online monitoring formulated in 6.3. A practical application of this monitor is shown in section 7.3.4.

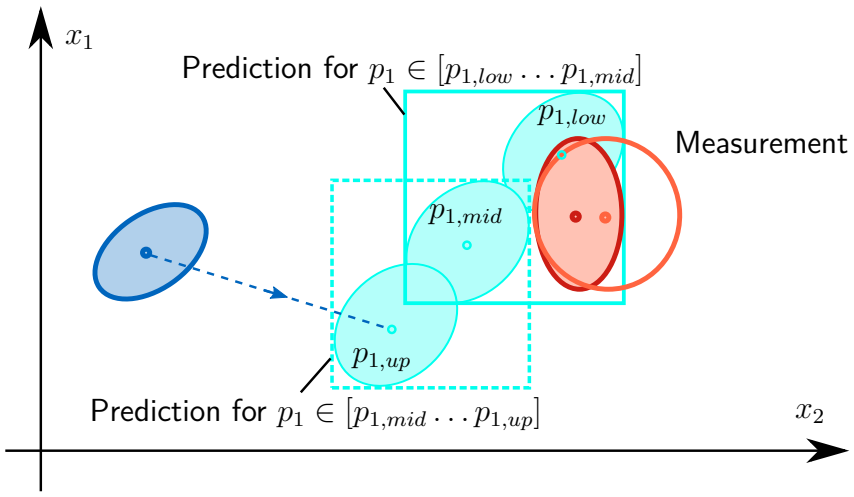


Figure 6.11: *Generating new initial state covariances with identification of reduced range of uncertain parameters*

7

Applications

In this chapter, the basic ideas of and algorithms for the development of safety-critical functions following the TCA are demonstrated for selected examples from the development of an autopilot for civil close formation flight. First, the operational concept is introduced, followed by descriptions of the models and simulation environment used during the different steps of the development process. Eventually, the algorithms described in this thesis are applied and results are evaluated to highlight the potential of the probabilistic top-down development process. Note that the used examples are only described to an extent that is required to understand the application of proposed algorithms and the evaluation of results. The objective is to provide easy understandable examples of a realistic application to facilitate clarity and comprehensibility. The application of the TCA to a real development project is subject of further research at the Institute of Flight System Dynamics [Mum18].
Some of the explanations and results given in this chapter were already presented, published and discussed in public by the author of this thesis in [LH13, LH14a, LH14b, LH15, LMH16a, LMH16b, LH17, Löb+17]. However, this chapter provides a cohesive evaluation of the different steps of the TCA for the first time, also taking into account latest findings and the obtained feedback.

7.1 Aircraft Formation Flight

Close formation flight is well established in military aviation. One important field of application is aerial refueling to extend range and endurance. With increasing fuel costs and tightening emission limits, there is a growing interest in close formation flight for civil aviation. It can be used to exploit the aerodynamic benefits of wing-tip vortices [Ban+06, Kle+13, ODB15] but also to enable aerial refueling of civil commercial aircraft, where studies promise a two-digit percentage fuel burn, and thus CO_2 emission reduction, even when taking fuel consumed by the tanker into account [Nan06, BV05, McR+15]. For civil aerial refueling, it has been found that the conventional military refueling operation, where the receiver aircraft approaches a tanker, is not desirable from an operational and also passenger-comfort point of view. Therefore, further research on novel cooperative guidance and control methods for a more desirable refueling

configuration was initiated where the tanker approaches the receiver. Another outcome of the afore mentioned research on the feasibility of civil aerial refueling is that such a refueling maneuver must be conducted fully automatically to ensure safety and consistent performance under various conditions without demanding piloting skills too much from civil airliner flight crews.
In the considered context, only the boom refueling method is reasonable due to much higher fuel flow rates and the lower control dynamics required for the tanker compared to the second very common refueling method using a probe and a drogue.

7.1.1 Problem Formulation

The task considered in this chapter is the development of an automatic flight control system (autopilot) for a tanker aircraft in close formation flight. According to the defined refueling operation, the taker is designated to conduct the relative maneuvering with respect to the receiver, while the receiver is considered to use its conventional autopilot for the sake of simplicity of the considered example, i.e. there is no cooperative control of the two aircraft. Figure 7.1 shows the arrangement of aircraft during close formation flight.

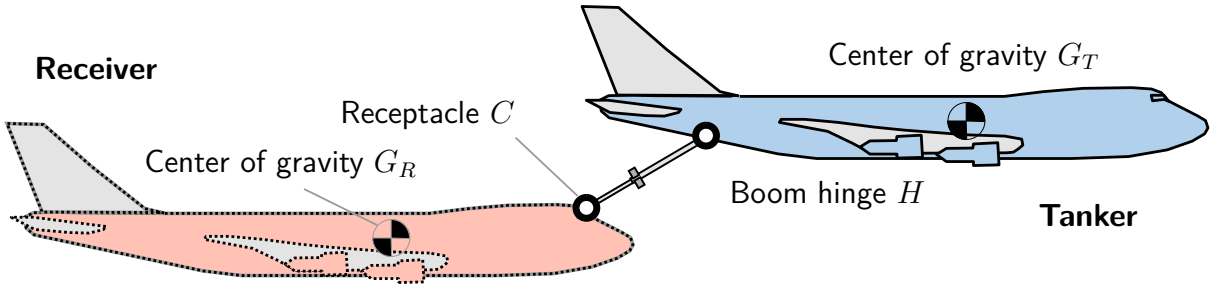


Figure 7.1: *Aircraft in close formation flight*

Since a conventional refueling boom is considered the tanker must be positioned ahead and above the receiver after an approach from in front and above. For this example, the aircraft designated for formation flight already exist, which also includes adequate models for aerodynamics, control surfaces, etc. The closed-loop receiver dynamics, using its baseline autopilot, with all its uncertainties must be assumed as part of the environment, since it cannot be changed during the development.
The tanker is designated to approach the receiver, hold the position during station-keeping to enable boom connect, fuel transfer, boom disconnect and to finally depart again. The example considered in this chapter is limited to the following aspects, which are sufficient for the demonstration of the different steps of the TCA:

- **Station keeping phase**: Only the control functions required for close formation flight are considered, which is also referred to as station-keeping phase. The approach and departure phases are not considered.

- **Longitudinal dynamics**: Only longitudinal control functions are considered. Decoupling of longitudinal and lateral dynamics is admissible for conventional fixed wing aircraft for straight flight and only small deviations from the trim point, which is the case here [BAL11, p. 261].

These limitations do not reduce the generality of the example and are not owed to the applicability of the TCA, but to comprehensibility, which can be best achieved by reduced example. The reduced set of functions is sufficient to explain the different algorithms presented in this thesis. Similar assumptions and splitting up of tasks is also common for conventional developments, hence the given descriptions are not on an academic level but still close to reality, which is the intention to also show the applicability and high potential of the TCA for real world problems.

7.1.2 Top-Level Requirements

There has been no civil commercial aerial refueling or formation flight up until now. Hence, there are also no requirements specifically for such operation. Requirements for military formation flight are usually only driven by the function and not by safety and therefore are not applicable for this case. However, top level requirements can be derived from current certification specifications, specifically from CS25.1309 [Eur17b], which specifies the level of confidence of equipment, systems and installations based on a safety assessment, where the safety impact of each function and related malfunctions added by the components are assessed. This has been conducted in the scope of the European research project RECREATE, where mainly the following top-level failure conditions and corresponding criticality were derived for close formation flight and aerial refueling of civil airliners [van15]:

- **Mid-air collision**: Collision between the tanker and the receiver is the most obvious safety risk. For the sake of simplicity, any contact between the aircraft, no matter which parts collide and with which intensity, is considered to be a catastrophic event, which is related to a failure probability of 10^{-9} per flight hour (see table 3.2). In other words, no failure of the automatic flight control system, be it hardware or software, or adverse behavior driven by uncertainties, disturbances and faults may cause a relative position change between the aircraft that leads to a mid-air collision with a probability higher than 10^{-9} per flight hour.

- **Hard boom impact**: If the boom would hit and break a cockpit window during a docking attempt, this could lead to fatal injuries to the pilots and in worst case full loss of the receiver aircraft, which is associated to an admissible probability of 10^{-9} per flight hour. However, using adequate mitigation means, e.g. reinforced airframe structure around the receptacle and additional protective measures for the cockpit windshield, the classification of criticality could be reduced to hazardous or even only major, which is related to a failure probability of 10^{-7} and 10^{-5} per flight hour for commercial civil operations.

- **Ignition of fuel spray**: Destruction of refueling equipment, e.g. due to excessive forces on the boom, could lead to fuel spray that is sucked into the engines leading to engine flame-outs. If there is a possibility that all engines are influenced, leading to a damage that prevent engine restart, this failure condition must be classified as catastrophic.

For the development of the automatic flight control system for close formation flight, mainly the first requirement is of relevance. Although too high maneuvering during station-keeping might cause mechanical damage and hence the destruction of the refueling boom, fuel spray could be reduced to a reasonable amount by emergency fuel cut equipment specifically designed as mitigation mean. Hence, the subsequent explanations focus on function specification, design, verification and monitoring, to prevent collision between the aircraft.

7.2 Simulation Environment

Models are used throughout the whole development process. In this section, models for the environment are introduced, which are valid for all development phases, although not the full level of detail is used for every phase. The environment models also include the receiver with its autopilot, since it is out of scope of this development task. In accordance with the simulation environment repeatedly described throughout this thesis, figure 7.2 shows the implementation-dependent portion of this environment, which are the system and environment models. The individual components and related uncertainties of the environment are discussed in this section, since they are independent of the specific development phase, while the models

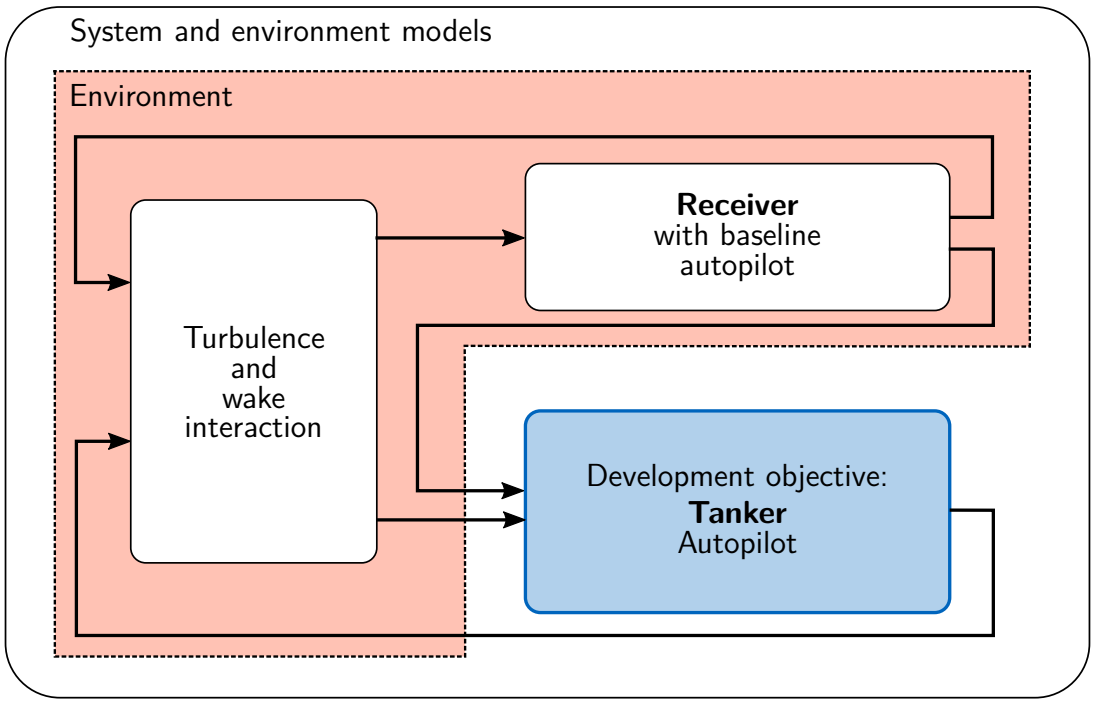


Figure 7.2: *System and environment models for the considered example*

for the development objective, i.e. the tanker with its autopilot, are enhanced throughout development and are discussed in 7.3, where the individual steps of the TCA are discussed.

7.2.1 Aircraft Models

For the research on close formation flight and aerial refueling, a simulation environment was developed and refined in Matlab Simulink. It consists of detailed nonlinear aircraft simulation models of a four-engined civil transport aircraft with a wingspan of almost $60m$ and a maximum take-off weight of $330t$, which are used for both tanker and the receiver. It includes engine and actuator dynamics and nonlinear aerodynamic models, based on exhaustive lookup tables that are scheduled over the angle of attack, angle of sideslip and the Mach number, just to mention the most significant dependencies [HN70]. Furthermore, the change of the center of gravity, the mass and the moment of inertia during refueling is simulated to allow a reasonable evaluation of the fuel transfer during station-keeping.

7.2.2 Wake Interaction

The simulation environment also includes a model for the wake interaction between the aircraft, which is essential for close formation flight [Löb+12]. Many available studies on wake interaction focus on the effect of larger leading on smaller trailing aircraft. To obtain reasonable numbers for the interaction between two similarly large aircraft, estimations by vortex lattice method and an extended lifting line theory by *Philips* and *Snyder* [PS00] are used. The latter constitutes an adaption of the classical lifting line theory applying a three-dimensional vortex lifting law instead of the two-dimensional Kutta-Joukowski law usually used in classical theory. This allowed for the estimation of force and moment increments due to wake interaction as a function of the relative position between the aircraft and the angle of attack of the leading aircraft. The interaction proved to be qualitatively correct and quantitatively reasonable when comparing to exemplary CFD analysis [Hep+12] and the results of previous modeling and flight test efforts [DVB05, DLB08], especially for close formation flight, where wake dissipation has only a minor effect. Although being relatively small, the aerodynamic influence of the trailing on the leading aircraft is also estimated using above described method, with the results being comparable to previous studies [DBH13]. A more detailed description of the wake model can be found in appendix B.1.

7.2.3 Uncertainties

During all offline steps of the TCA, enhanced stochastic algorithms (section 2.6) are extensively used. For the exemplary results shown in this chapter, only a few important uncertainties are considered, which are introduced in this subsection. The reduced number of uncertainties is not attributed to the limited capabilities of the applied stochastic algorithms – these are

insensitive to the number of uncertain parameters – but to the clarity of results and their discussion.

Plant Uncertainties

Plant uncertainties refer to the uncertainties of the aircraft models used for the receiver and the tanker. Two major sources of uncertainties can be identified: aerodynamics and inertia.

Aerodynamic forces and moments are described by sophisticated nonlinear models. However, the underlying data is usually obtained by simulations or identified from flight test results and is hence not exactly known. The same is true for the wake interaction between the aircraft. The uncertainties are usually described by distributions, often Gaussian or uniformly distributed. Table 7.1 provides the major aerodynamic derivatives relevant for the motion in the longitudinal plane and their uncertainties considered in this example. While the mean values are obtained from the detailed aerodynamic models, the variation is added on top, where the uncertainty is assumed to be Gaussian distributed with a constant variance σ^2 given in the table.

Table 7.1: *Aerodynamic uncertainties of tanker and receiver (left) and wake model (right)*

Parameter	Uncertainty σ
$C_{L\alpha}$	5%
$C_{M\alpha}$	5%
C_{Mq}	5%
$C_{M\eta}$	5%

Parameter	Uncertainty σ
$\Delta C_{L,wake}$	15%
$\Delta C_{D,wake}$	15%
$\Delta C_{m,wake}$	15%

The mass, the moment of inertia and the center of gravity can significantly change during station-keeping and refueling in the considered operation. During fuel transfer, the tanker becomes lighter, while the receiver becomes heavier. This also changes the reaction of the aircraft on control inputs and disturbances. Hence, the variation must be considered. Similar aircraft are used for tanker and receiver, with a mass variation between $m = 200 \ldots 250t$ and a center of gravity positions between $15 \ldots 35\%$ of the mean aerodynamic chord (MAC). The inertia tensor changes automatically based on the weight, assuming that fuel is first transfered from the center tanks, followed by the wing tanks.

The receiver uses a conventional fixed-gain PID autopilot to maintain altitude. Although this is not favorable for close formation flight, it is used for the sake of simplicity of the example.

Atmospheric Disturbances

Continuous atmospheric disturbances are considered, which are also referred to as wind turbulences. The Dryden turbulence model is well accepted for use in development of flight control systems and is specified in MIL-HDBK-1797 [US 97]. It is a linear filter that takes white noise as input and outputs wind velocities and rates in a body-fixed frame acting on an aircraft.

The power spectral densities of the vertical wind velocity and wind pitch rate are given by

$$\begin{aligned}\Phi_w(\omega) &= \sigma_w^2 \frac{L_w}{\pi} \frac{1+3\left(L_w\omega/V\right)^2}{\left(1+\left(L_w\omega/V\right)^2\right)^2} \\ \Phi_q(\omega) &= \frac{\left(\omega/V\right)^2}{1+\left(\frac{4b\omega}{\pi V}\right)^2}\Phi_w(\omega)\end{aligned} \tag{7.1}$$

with σ_w, L_w and V being the altitude and intensity-dependent root mean square turbulence amplitude according to MIL-HDBK-1797, the scale length with $L_w = 1750ft/2$ and the true airspeed, respectively. By spectral factorization, transfer functions can be obtained for implementation and simulation of the vertical velocity and pitch rate component of the turbulence field. The derivation of the related linear state space model, which will be required for monitoring, is given in appendix B.2.
Due to spacial correlation, turbulences acting on the tanker and the receiver are very similar. For evaluation, it is assumed that turbulence velocities and rates are the same for both aircraft, however, with a time delay similar to the longitudinal separation between the aircraft divided by the aircraft forward velocity. This assumption is legitimate especially for the lower frequent portion of turbulences, which is essential for aircraft flight dynamics.

Sensor Noise

For the evaluation of control performance especially during verification, adequate models of sensors are required. For the considered application, this especially includes measurements of aircraft rates, attitudes, the relative position and the velocity between the aircraft. Adequate sensor models are used for inertial measurements and to obtain relative state information, where for the latter an extensive sensor suite is used, where a sophisticated filter merges measurements from optical, ranging and *Global Positioning System* (GPS) measurements. Since the specific models and implementations are not essential for evaluation and understanding of the example in this chapter, sensor models are not further discussed. A detailed description can be found in [LH14b].

7.3 Application of the Total Capability Approach

7.3.1 Derivation of Requirements

Example Description

The objective of this section is to break down the top-level safety requirements to requirements on adequate tanker closed-loop dynamics. The top-level safety requirement, which is evaluated here, quotes that the probability of a collision between the aircraft must be smaller than 10^{-9} per flight hour. This requirement can be broken down to subfunctions, e.g. lateral control,

longitudinal control and thrust control. For the demonstration of the introduced requirements derivation approach, the top level requirement is quantified:

The probability for a vertical position error of more than $7m$ from station-keeping reference position towards the receiver aircraft must be less than 10^{-9} per flight hour.

Here, a position error from station-keeping reference position of more than $7m$ is defined as crash, see figure 7.3.

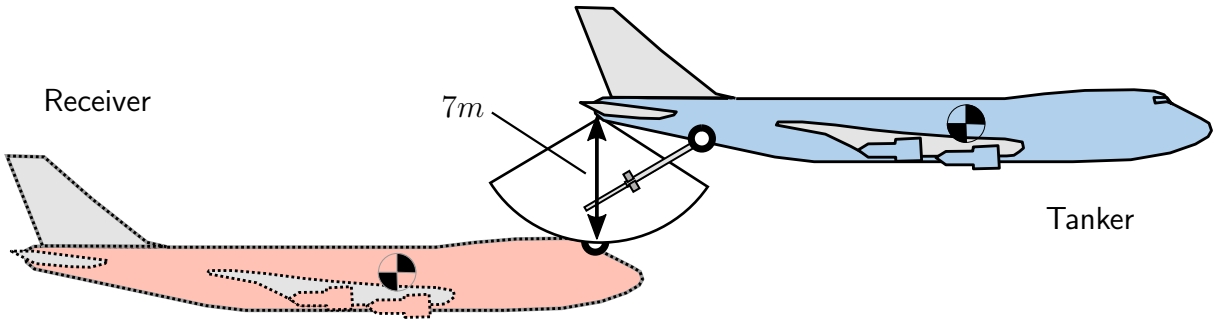


Figure 7.3: *Aircraft in close formation flight – definition of crash*

For this example, turbulences with moderate intensity according to the Dryden wind turbulence model are considered, which have an occurrence probability of 10^{-3} according to [US 97], leading to the requirement that the given threshold of $7m$ must not be exceeded with a conditional probability higher than 10^{-6} in the presence of moderate turbulences, using *Bayes*' theorem [Bro06, p. 773]. This requirement can be further broken down to a level that allows the direct use for system design, specifically to requirements for adequate closed loop dynamics of the tanker during formation flight. When only taking a look at the vertical motion and neglecting the actuator dynamics, the transfer function from control input (elevator) to vertical position change is of forth order. A physical interpretation of this is the second-order rotation dynamics from elevator input to angle of attack, causing a vertical acceleration, which then leads to vertical velocity and position after one and two integrations, respectively. Hence, the dynamics range of a fourth-order system is looked for, for which the above stated requirement is fulfilled. For that, the forth order dynamics is described by two second order dynamics for the rotational and translation motion respectively. For each, damping and frequency are varied independently. In accordance with the notation defined in 4.2.3, these are denoted as design parameters comprised in the design parameter vector $\boldsymbol{\lambda}$. Figure 7.4 shows the specification model used for the evaluation of the relative position response. It contains the known tanker open loop dynamics and an actuator model for the control input. To adjust the desired design parameter combination for the closed loop behavior of the tanker, a full state feedback with pole placement is used. For that, the tanker open-loop dynamics is linearized, which is acceptable for small deviations from the trim point. Sources of uncertainty in this setup are atmospheric disturbances using the Dryden turbulence model, which act on the tanker and are hence referred to as plant disturbances, as well as the receiver motion, which changes the relative position between the aircraft. Since the objective of the tanker closed loop is to

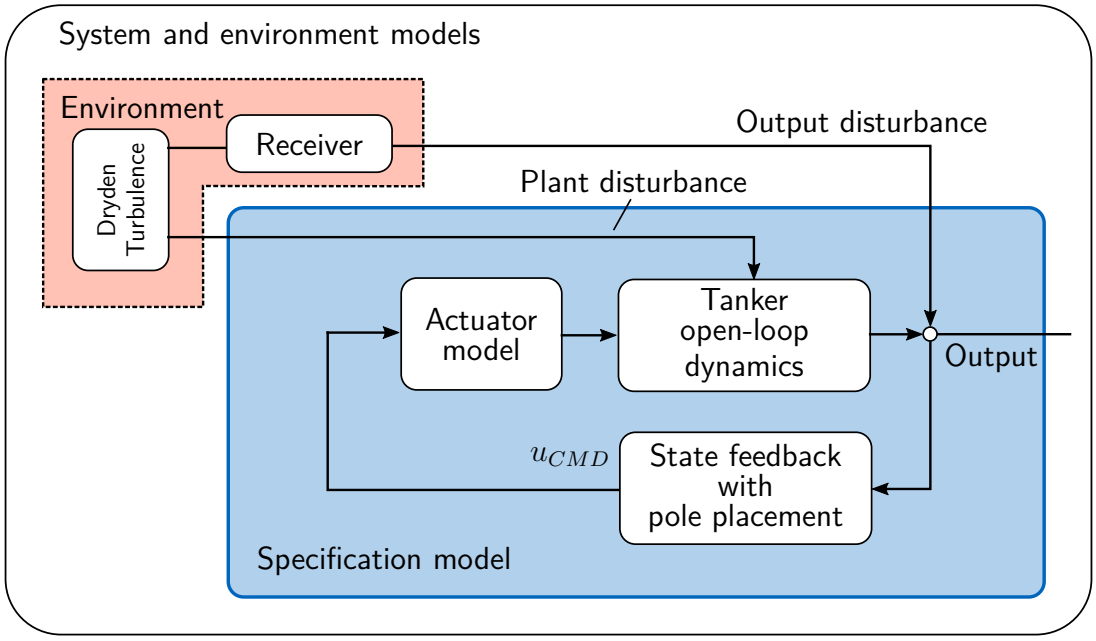


Figure 7.4: *Specification model for requirements derivation*

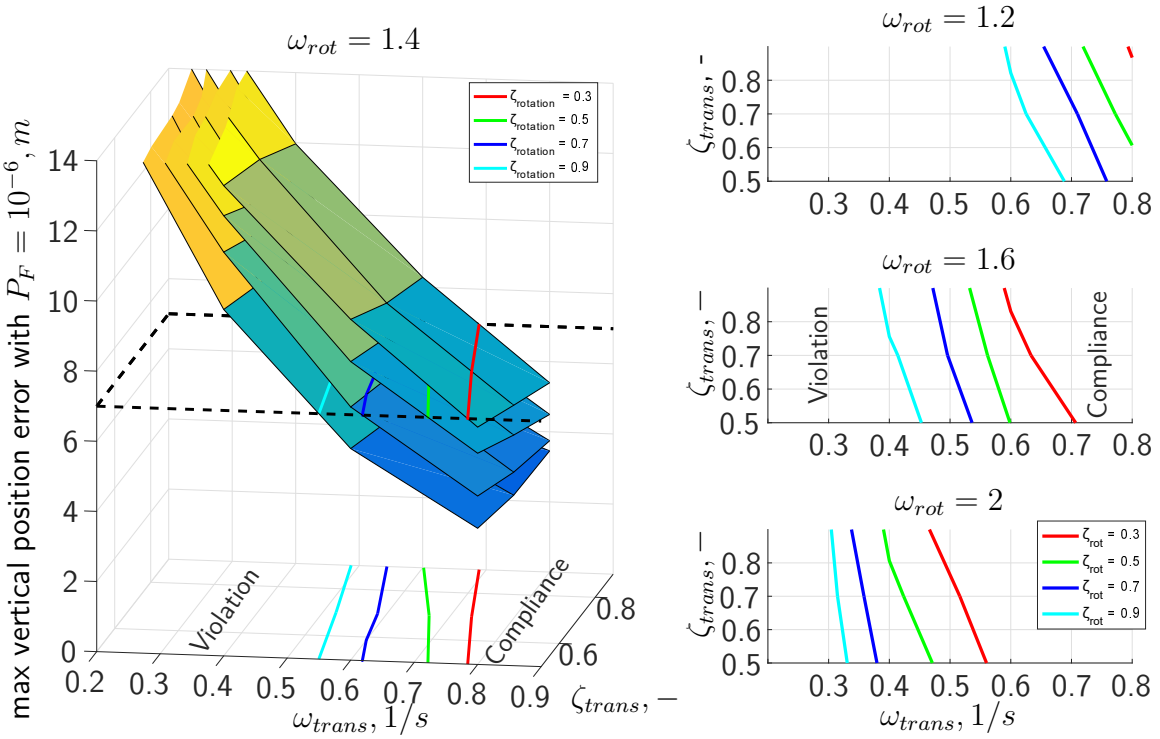
maintain a certain relative position with respect to the receiver, the receiver motion can be interpreted as output disturbance, since it influences the relative position and velocity between the aircraft. For assessment of the relative position control performance, the output is used to calculate the response variable r, which, in this case, is the relative approximation between the aircraft, which must not exceed $r = 7m$ with a probability higher than $P_F = 10^{-6}$.
The objective is to determine all acceptable combinations of the four design parameters. For that, the algorithm described in section 4.2.3 is used: First, for each contemplable parameter combination, the threshold is evaluated that is not exceeded with a probability higher than 10^{-6} in the presence of disturbances and uncertainties using Subset simulation described in 2.6. Second, the boundary parameter surface is looked for where the threshold of $7m$ is theoretically exceeded with a probability of exactly $1E-6$. For the purpose of comparison, the design parameter boundary is also estimated using an equal spaced grid for the four design parameters.

Determination of the Design Parameter Boundary using Grid Samples

First, grid sampling is considered to obtain a reference. For that, each of the four design parameters is discretized using the grid defined in table 7.2. Figure 7.5 presents the procedure

Table 7.2: *Limits and discretization of design parameters*

Parameter name	Symbol	Values	Unit
Frequency of rotational motion	ω_{rot}	$1, 1.2, 1.4, \ldots, 2.0$	s^{-1}
Damping of rotational motion	ζ_{rot}	$0.3, 0.5, 0.7, 0.9$	–
Frequency of translational motion	ω_{trans}	$0.2, 0.4, 0.6, \ldots 1.4$	s^{-1}
Damping of translational motion	ζ_{trans}	$0.5, 0.7, 0.9$	–



(a) *3D cut* **(b)** *Resulting 2D boundaries*

Figure 7.5: *Determination of design parameter bounds using grid method*

of determination of the design parameter boundary using grid samples. Each 3D surface displayed in a) gives the relative position error that is not exceeded with a probability higher than 10^{-6} for different combinations of frequency and damping of the translational motion. The individual surfaces stand for different damping ratios of the rotational motion. The plot is drawn for a single frequency of rotational motion. The black dashed line indicates the vertical position limit that must not be exceeded with the specified probability. The intersections between the 3D surfaces and the requirement limit $y = 7m$ lead to boundaries that separate the parameter combinations into acceptable and non-acceptable regions. Subfigure b) gives these limit lines for different frequencies of the rotational motion. Figure 7.6 visualizes the design parameter boundary in 3D. Note that the different lines and surfaces for different damping ratios of rotational motion are only a mean to visualize the higher-dimensional solution plane. Evaluation at 672 grid points are required to obtain the displayed design parameter boundary.

Determination of the Design Parameter Boundary using Gradient Method

This section shows the application of the stochastic gradient method described in section 4.2.3. The starting point of the evaluation of the design parameter boundary is $\omega_{rot} = 1.8s^{-1}$, $\zeta_{rot} = 0.5$, $\zeta_{trans} = 0.7$ and $\omega_{trans} = 0.495s^{-1}$. While the first three are chosen as freely modifiable parameter, the value for the forth parameter is determined so that it lies on the design parameter surface for $r = 7m$ with $P_F = 10^{-6}$. The step size for the gradient estimation is set to $\Delta\lambda = 0.1$, while the step size for generation of new samples along the hypersurface is

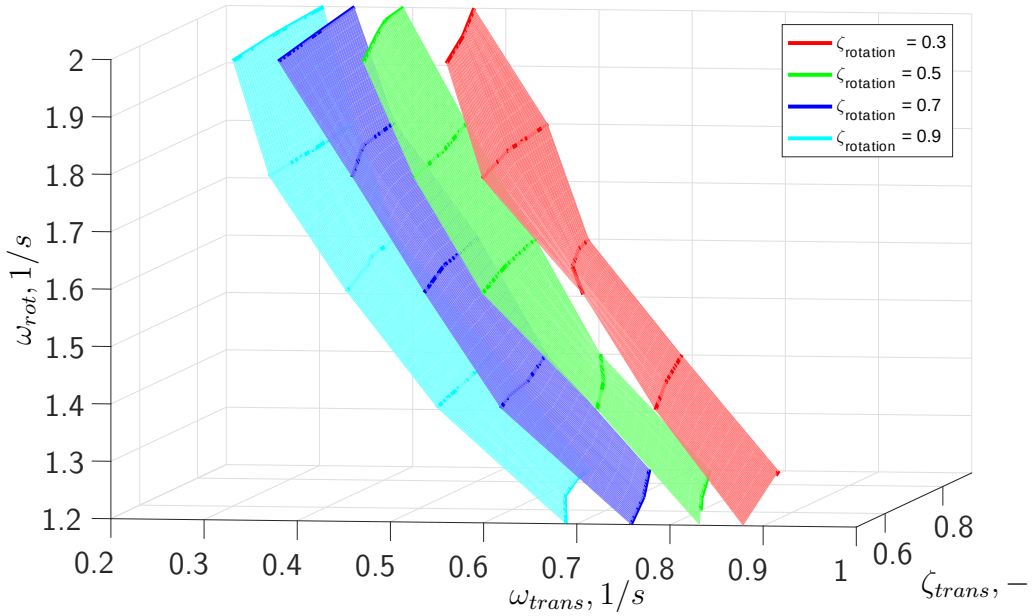


Figure 7.6: *Resulting design parameter boundary using a parameter grid and interpolation*

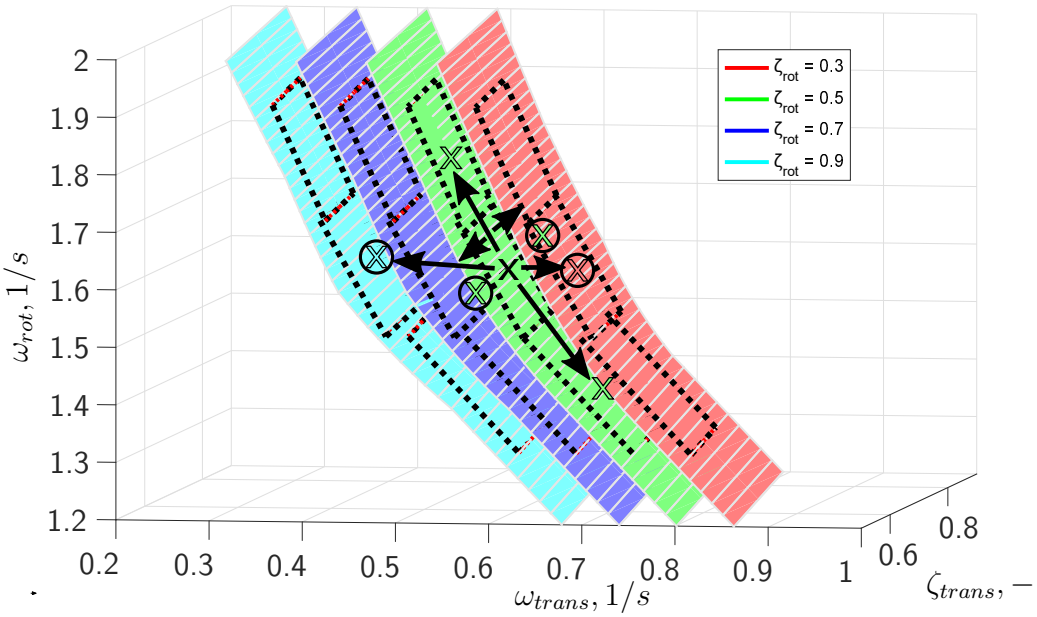


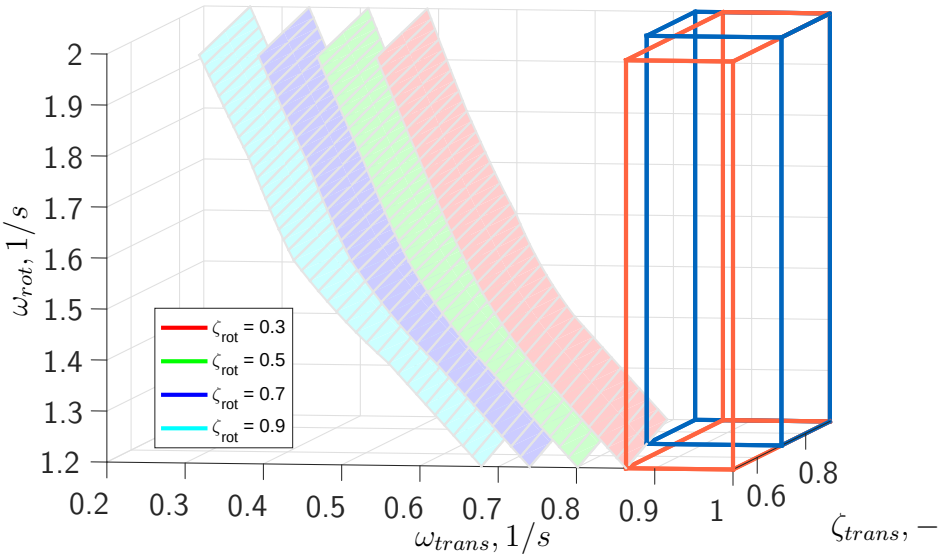
Figure 7.7: *Resulting design parameter boundary using the gradient method and interpolation*

set to $\Delta\lambda_S = 0.2$. Figure 7.7 shows the resulting design parameter boundary using the cubic interpolation given in (4.17). For this example, it was only required to calculate the gradient at seven points, marked by crosses in figure 7.7. The encircled crosses mark points where only a minor correlation is detected using equation (4.15). The black dashed lines represent the support planes for the design parameter boundaries which are connected using cubic spline interpolation. Evaluation of a total of 152 points with specific parameter combinations are required, composed by 56 evaluations for the estimation of the gradients at seven support points and 96 points for the verification of the approximation. For the verification points, a similar grid compared to the reference as defined in table 7.2 is used. This already gives a considerable reduction in computational time. The chosen step width for generation of new

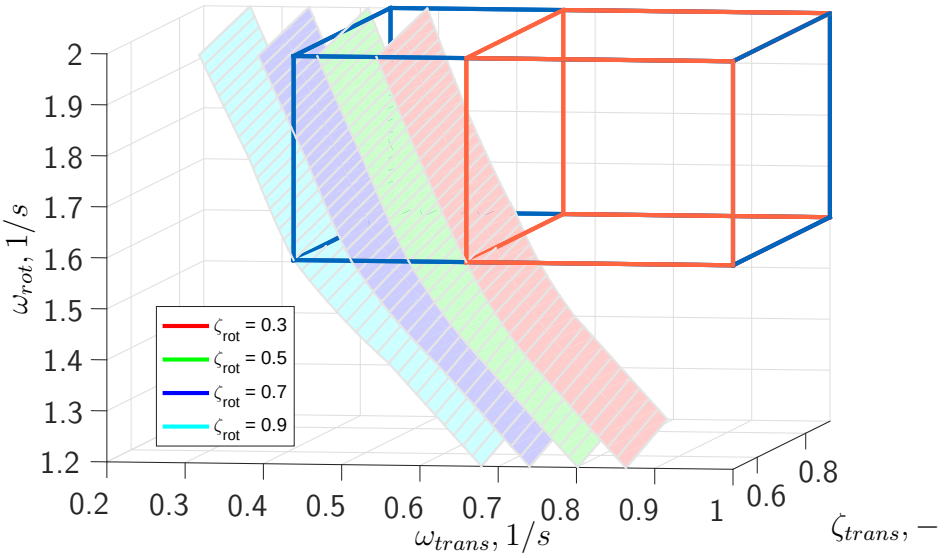
samples is chosen conservatively – increasing the step width would lead to additional reduction. Further reduction would only be possible by using knowledge about the possible shape of the hypersurface. Since by that the algorithm would lose its generality, this case is not further discussed here.
Comparing the parameter boundaries from simple grid interpolation and the results from the gradient based method, it can be recognized that the developed gradient method has an averaging effect. The unevenness recognizable in figure 7.6 mainly arises from the uncertainties during estimation of the system responses, i.e. are a non-desirable result inherent to the grid-based method for determination of uncertain boundaries. The developed algorithm based on uncertain gradients provides superior performance compared to the grid-based approach. Not only that the number of samples required can be reduced, but also smoother boundaries are obtained. However, although the presented algorithm can reduce the required number of samples by 1 to 2 orders, still the determination of uncertain hypersurfaces in higher dimensions is a challenging task.

Selection of Adequate Design Parameter Ranges

The description of the three-dimensional design parameter boundary derived previously might not be reasonable for further use due to the high correlation between different design parameters. For example, if admissible parameter intervals must be independent to allow for a separate design of different functions by different groups or also for online monitoring.
The ideas applied in this section refer to section 4.3. There, it is discussed that the best possible subspace for a simpler description of adequate design ranges is often not the one with the biggest overall volume, but it is also subject to the dependency structures between the parameters and the complexity and effort required to obtain certain parameter ranges during later design. Recall that the adequate performance range for the design parameters specifies the function to be designed and implemented during later development phases. For the autopilot example considered in this chapter, the flight controller designed later must result in a closed loop dynamics that lies within the derived rotational and translational dynamics range.
Figure 7.8 shows the most common way for a simplified description using a box-shaped solution space. An infinite number of solutions for such a box is available, with different choices coming along with different advantages and disadvantages. The task of identification of adequate design parameter ranges requires the analysis of solution properties to identify the most useful trade-off. Figure 7.8a shows the effect of only slightly correlated parameters ω_{trans} and ζ_{trans}, where the resulting performance does not much depend on the specific value of the damping of the translational motion. For this subfigure, $\zeta_{rot,min} = 0.3$ and $\omega_{rot,min} = 1.2s^{-1}$ are assumed. The blue box depicts the solution, for which a medium sized interval for ζ_{trans} is used. However, due to the low correlation, when accepting an only slightly increased value for $\omega_{trans,min}$, a much higher range for ζ_{trans} is obtained. Hence, the sensitivity of the resulting design volume with respect to the specific parameter selection



(a) *Varying range of* ζ_{trans}



(b) *Varying range of* ζ_{rot}

Figure 7.8: *Selection of adequate design parameter intervals*

should be considered during the selection of adequate design parameter ranges. This is a trivial task for lower dimensional parameter spaces (for up to four to five parameters), where a graphic representation of the parameter bounds is possible. Subfigure b adds another two dimensions to the discussion, by varying admissible ranges for ω_{rot} and ω_{trans}. A reduction of the admissible range of ω_{rot} leads to an increased range of ω_{trans}. A further increase of the admissible range of ω_{trans} can be achieved by increasing the minimal damping of the rotational motion ζ_{rot}. Both trade-offs can be directly attributed to aircraft flight dynamics: When increasing the rotational dynamics, the translational dynamics can be slower since the rotational dynamics required to follow translational commands is faster and hence better tracking of translational commands is possible. Furthermore, an increased minimum damping ζ_{rot} leads to lower overshoots of rotational motion, which results in lower required minimum

translational, but even more rotational frequencies, which can be seen by fixing ω_{trans} and varying ζ_{rot} in figure 7.8b. A correct answer for the best solution space cannot be given here, also since only one requirement is considered in this example, while other requirements can further limit the solution space. However, it can be derived that parameters, where a high correlation of design parameters exists with respect to the resulting design parameter boundary, these parameters should be considered in an integrated manner especially during later function design. This enables best possible exploitation of the available system performance.
One option to reduce conservativeness of the box-shaped solution spaces is the use of 2D solution spaces, as introduced in 4.3.3. This enables a better exploitation of the solution space by allowing correlation between two parameters. For figure 7.9, the correlated parameter couple is ω_{trans} and ω_{rot}. Alternatively, also ω_{rot} and ζ_{rot} could be chosen. While the orange box depicts the classical box-shaped solution, the blue shape represents the 2D solution space. By using this additional degree of freedom, the volume can be considerably increased by only requiring a slightly more complex formulation of the solution space. Unfortunately, even the latest algorithms in this field presented in this thesis assumes linear bounds, which is approximately the case in figure 7.9. However, if the bounds would be concave instead of convex, the 2D solution space would no longer result in a valid solution subspace, since it would violate the actual bounds.

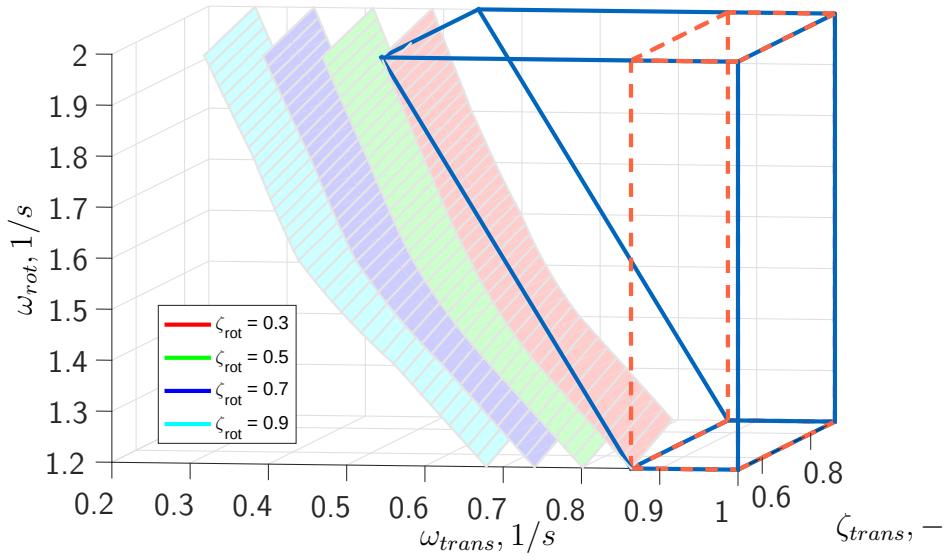


Figure 7.9: *Description of the adequate parameter combinations using 2D solution spaces*

Conclusions on Model-Based Requirements Derivation

The results obtained for requirements derivation and determination of adequate design parameter ranges for this application-related example allow to draw the following conclusion: The model-based derivation of requirements can lead to significantly increased design parameter ranges compared to conventional, conservative and experience-based intervals. However, due to the high computational effort and lack in availability of adequate methods for approximation of complex, high dimensional solution spaces, application is only considered useful for

a lower number of design parameters (four to five based on experience). Nevertheless, if the design problem can be split up into independent design sub-problems, this should be no showstopper for future applications of the model-based derivation of safety driven performance requirements.

7.3.2 Validation

Completeness and correctness is, by its nature, usually difficult to prove before design and implementation. As described in section 4.4, the validation process can be significantly improved using the specification models implemented for requirements derivation.
Specifically for the formation flight example, the formalized requirements and models describing the behavior of the closed loop can be readily used to support validation. The model-based approach can be exploit to different extends for validation, from checking compliance of conventionally selected lower level requirements with probabilistic top-level requirements to the validation of the whole operational concept. This subsection describes possible applications of model based validation in accordance with the findings in 4.4.3.

Correctness of Conventional Requirements

The implementation of all steps of the TCA leads to the highest benefits achievable using a model-based development approach. However, due to certain reasons, e.g. due to too high complexity or limited computational resources, this might not always be possible. In this case, the quantification of lower level safety-driven performance requirements is done in a conventional manner based on experience and rules of thumb. Furthermore, theoretically the lower level requirements derived using the model-based approach described in this thesis automatically comply with all top-level requirements. However, if independence of subproblems is assumed during requirements derivation, this independence must be proven.
For the example considered in this section, adequate dynamics for the translational and rotational motion could also be derived based on engineering judgment and available standards for similar (possibly military) applications. To prove compliance with the specific top-level safety requirement for civil formation flight, the system and environment models (figures 7.2 and 7.4) can be used. Instead of utilizing the simulation environment for the determination of all admissible designs, it is only used to prove that the selected range for the dynamic behavior complies with the probabilistic top-level requirement.
In a similar manner, the assumption of separated longitudinal and lateral dynamics for specification and design can be proven by integrated simulation. This proof does not suffer the curse of dimensionality, since for verification, the admissible design parameter intervals can be treated as uncertain parameters themselves. By stochastic simulation, the most probable critical scenario within the admissible design parameter space and related failure probabilities can be efficiently estimated. Certainly, the resulting failure probability should be less or equal to the admissible failure probability (within confidence bounds) – violation would imply invalidity

of the assumption on independence of longitudinal and lateral dynamics.

For both, validation of conventional requirements and proof of correctness of assumptions, validation by simulation is very similar to verification using enhanced stochastic simulations (see section 7.3.3). However, using the specification models instead of the actual implementation. Since this is discussed more in detail later for verification of the implemented function, an additional discussion does not provide any further information, which is why no specific results for validation are presented here.

Validation of Derived Requirements against Environment

The environment is defined as everything that cannot be directly influenced during the current development. The specified system might be integrated together with different other systems, which are developed independently and in parallel. The functions of different systems must not contradict each other. For the specific example, the flight control system is integrated into the overall aircraft and can, therefore, have adverse impacts on other systems like aircraft structure. If eigenmodes of the aircraft are not adequately addressed in the specification of the flight control functions (e.g. by using notch filters to prevent excitation with critical frequencies [NAT00, p. 4-1]), this can be detected by simulations taking into account models for the aircraft structure. Since such models are often not available at the development start or are too complex, the direct incorporation into the environment for requirements derivation is not possible or not recommended.

Completeness of Requirements

The set of specification models for functions of possibly different flight phases enables simulation of the whole operations to determine absence of relevant requirements in addition to correctness. For the example of civil aerial refueling, such an operation starts with the initial approach of the tanker towards the receiver, followed by station-keeping, boom connection, fuel transfer, boom disconnect and departure. In the research project RECREATE [Zaj15], where the author of this thesis participated in, the entire operation was specified and the resulting scenario simulated using (full) flight simulators for the tanker and the receiver [HLG15]. Several airliner pilots were asked to evaluate the overall refueling operation and the interaction with the specified system to determine if everything behaves as intended by user and customer. By that, missing or undesired functions could be identified and corrected, increasing the confidence in the completeness of the specified system and enabling continuous detailed function development. While some of the missing or unfavorable functions could have also been identified by thorough analysis, validation by simulation added additional insight into the specified system, which might have otherwise only be obtained during later stages of development, where corrections would have been more expensive.

7.3.3 Function Design and Verification

Control System Design

The application of the TCA for design and verification is discussed together for the same reasons as in chapter 5: The methods and their application are the same, only the use of results is different. While during design, simulations are conducted to identify weak spots of the designed functions and derive improvements, for verification, the compliance of the final system with the probabilistic top-level requirements is demonstrated.
For the considered example of close formation flight, nonlinear dynamic inversion is performed to design a relative position controller for the tanker aircraft. The idea has been originally presented for the control of the receiver aircraft and steady tanker in [Wan+15] with the author of this thesis as coauthor. The resulting control laws are only summarized since they are neither essential for the TCA nor for the understanding of the presented results. Figure 7.10 shows the outer loop control structure. $\left(\vec{\mathbf{r}}^{HC}\right)_O$, $\left(\vec{\mathbf{a}}_{des}^{HC}\right)_O^{OO}$ and $\left(\vec{\mathbf{a}}_{des}^{G_T}\right)_O^{OO}$ denote the relative position of the receptacle C with respect to the boom hinge H in North-East-Down frame (NED, denoted as O-frame), the according desired acceleration relative to the O-frame and the desired acceleration of the tanker center of gravity G_T, respectively.

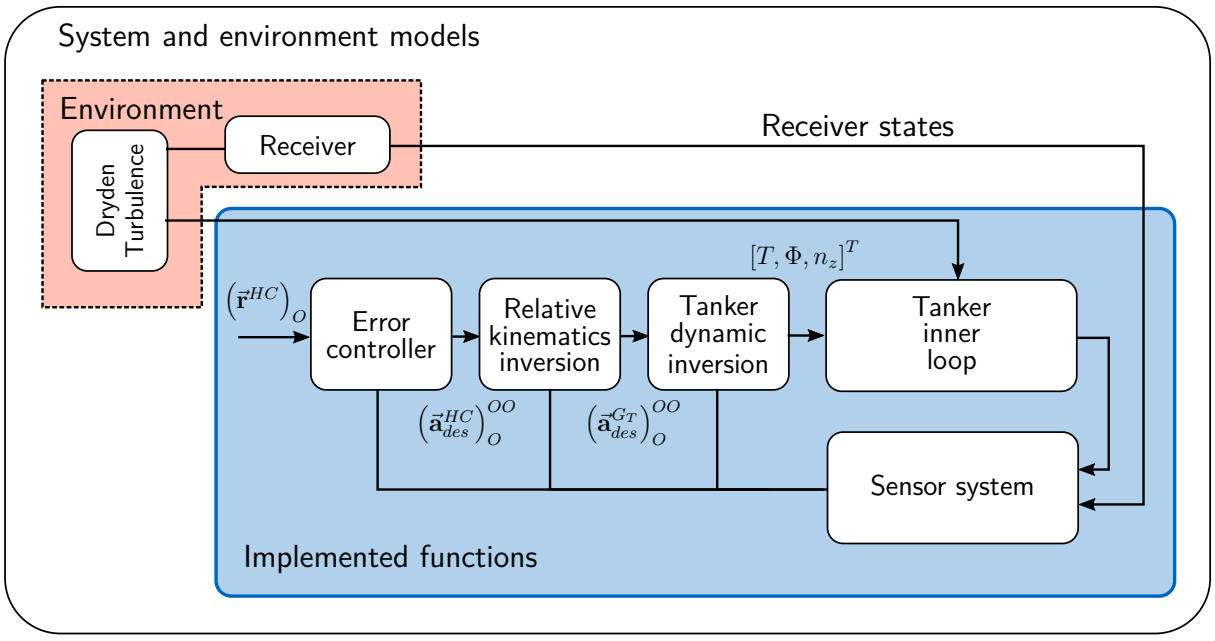


Figure 7.10: *Model for design and verification (simplified)*

First, the kinematic inversion controller is derived, followed by the description of the inner-loop and error controller. The comparison of the derivation of relative kinematics in different frames in [Wan+15] resulted in the decision for the O-frame, which results in the lowest complexity of the kinematic equations. Starting from the relative position between the boom hinge point H and receptacle C

$$\left(\vec{\mathbf{r}}^{HC}\right)_O = \left(\vec{\mathbf{r}}^{HG_T}\right)_O + \left(\vec{\mathbf{r}}^{G_TG_R}\right)_O + \left(\vec{\mathbf{r}}^{G_RC}\right)_O \tag{7.2}$$

a term for the acceleration is obtained by calculating the second time derivative in the NED-frame:

$$\begin{aligned}\left(\vec{\mathbf{a}}^{HC}\right)_O^{OO} = &\left(\vec{\mathbf{a}}^{G_R}\right)_O^{OO} - \left(\vec{\mathbf{a}}^{G_T}\right)_O^{OO} \\ &+ \left(\dot{\vec{\boldsymbol{\omega}}}^{OB_T}\right)_O^{B_T} \times \left(\vec{\mathbf{r}}^{HG_T}\right)_O \\ &+ \left(\dot{\vec{\boldsymbol{\omega}}}^{OB_R}\right)_O^{B_R} \times \left(\vec{\mathbf{r}}^{G_RC}\right)_O \\ &+ \left(\vec{\boldsymbol{\omega}}^{OB_T}\right)_O \times \left[\left(\vec{\boldsymbol{\omega}}^{OB_T}\right)_O \times \left(\vec{\mathbf{r}}^{HG_T}\right)_O\right] \\ &+ \left(\vec{\boldsymbol{\omega}}^{OB_R}\right)_O \times \left[\left(\vec{\boldsymbol{\omega}}^{OB_R}\right)_O \times \left(\vec{\mathbf{r}}^{G_TC}\right)_O\right]\end{aligned} \tag{7.3}$$

where B_T and B_R are the body-fixed frame of tanker and receiver. It is assumed that the NED frame is the same for both aircraft, which is approximately true for small relative distances. By inverting the relative dynamics given in (7.3), the absolute acceleration of the tanker center of gravity in the NED frame is obtained:

$$\begin{aligned}\left(\vec{\mathbf{a}}^{G_T}\right)_O^{OO} = &\left(\vec{\mathbf{a}}^{G_R}\right)_O^{OO} - \left(\vec{\mathbf{a}}^{HC}\right)_O^{OO} \\ &+ \left(\dot{\vec{\boldsymbol{\omega}}}^{OB_T}\right)_O^{B_T} \times \left(\vec{\mathbf{r}}^{HG_T}\right)_O \\ &+ \left(\dot{\vec{\boldsymbol{\omega}}}^{OB_R}\right)_O^{B_R} \times \left(\vec{\mathbf{r}}^{G_RC}\right)_O \\ &+ \left(\vec{\boldsymbol{\omega}}^{OB_T}\right)_O \times \left[\left(\vec{\boldsymbol{\omega}}^{OB_T}\right)_O \times \left(\vec{\mathbf{r}}^{HG_T}\right)_O\right] \\ &+ \left(\vec{\boldsymbol{\omega}}^{OB_R}\right)_O \times \left[\left(\vec{\boldsymbol{\omega}}^{OB_R}\right)_O \times \left(\vec{\mathbf{r}}^{G_TC}\right)_O\right]\end{aligned} \tag{7.4}$$

$\left(\vec{\mathbf{a}}^{G_T}\right)_O^{OO}$ can be expressed by the load vector in the kinematic frame of the tanker K_T, where $\mathbf{M}_{OK_T}$ is the transformation matrix from the K_T to the O-frame.

$$\left(\vec{\mathbf{a}}^{G_T}\right)_O^{OO} = \mathbf{M}_{OK_T}\left(\vec{\mathbf{a}}^{G_T}\right)_{K_T}^{OO} = \mathbf{M}_{OK_T}\left(\vec{\mathbf{f}}_T^{G}\right)_{K_T}^{OO} + \left(\vec{\mathbf{g}}^{G_T}\right)_O^{OO} \tag{7.5}$$

$\left(\vec{\mathbf{g}}^{G_T}\right)_O^{OO} = [0,0,g]^T$ is the gravitational vector with g being the gravitational acceleration. $\left(\vec{\mathbf{f}}_T^{G}\right)_{K_T}^{OO}$ is the specific force vector in the kinematic frame, i.e. the non-gravitational force normalized by mass. The load factor vector is defined by the specific force vector normalized by the gravitational acceleration g, which results in

$$\left(\vec{\mathbf{n}}^{G_T}\right)_{K_T} = \frac{1}{g}\mathbf{M}_{K_TO}\left[\left(\vec{\mathbf{a}}^{G_T}\right)_O^{OO} - \left(\vec{\mathbf{g}}^{G_T}\right)_O^{OO}\right] \tag{7.6}$$

The load factor in the kinematic frame is allocated to the inner loop commands

$$\begin{aligned} T_{cmd} &\approx mgn_x \\ \Phi_{cmd} &\approx \arctan n_y \\ n_{z,cmd} &\approx n_z \end{aligned} \tag{7.7}$$

where T_{cmd}, Φ_{cmd} and $n_{z,cmd}$ denote the commanded thrust, bank angle and load factor. Classical linear controllers are implemented for the inner loop.
A PD controller is used for error control, which results in the following control law (assuming a constant desired relative position):

$$\boldsymbol{\nu} = \left(\vec{\mathbf{a}}_{des}^{HC}\right)_O^{OO} = -K_d \left(\vec{\mathbf{v}}^{HC}\right)_O^O - K_p \left[\left(\vec{\mathbf{r}}^{HC}\right)_O - \left(\vec{\mathbf{r}}_{\mathbf{d}}\mathbf{es}^{HC}\right)_O\right] \tag{7.8}$$

The parameters of the inner loop as well as of the error controller are tuned to yield a closed-loop dynamics within the acceptable dynamics range for translational and rotational motion identified in 7.3.1.

Evaluation

Subset simulation is used to evaluate the performance of the implemented (longitudinal) control functions, to identify possible reasons for critical behavior and to prove compliance with the probabilistic top-level safety requirement in the presence of uncertainties and disturbances specified in 7.2.3. The evaluation is conducted for $30s$. This time range is sufficient for the evaluation of the failure probability, since after this time, there is no longer a correlation between the current and initial aircraft states. This can be proven, for example, by the evaluation of the autocorrelation function of the aircraft response during turbulence excitation.
If the flight controller is tuned according to the adequate limits obtained from requirements derivation, it should satisfy the top-level safety requirement. However, during requirements derivation, only limited knowledge is available about possible uncertainties and further sources of uncertainty usually arise during design and implementation. Due to the reduced level of conservatism, it is necessary to evaluate the top-level safety requirement also explicitly with the implemented system, since compliance is usually no longer guaranteed by pure compliance with lower level requirements.
For this evaluation, decreasing distance between the aircraft is defined more thoroughly by considering the relative motion in the vertical plane, i.e. also the longitudinal (thrust) control is taken into account. The response variable r is then defined as proximity of the tanker and the receiver, see figure 7.11. Note that the lines for increasing values of the response variable are fixed with respect to the tanker and the relative motion is defined as the movement of the receptacle with respect to the tanker, where it does not matter whether the motion is caused by the tanker or receiver movement.
The Subset Infinity algorithm described in 2.6.3 is used, with a conditional failure probability

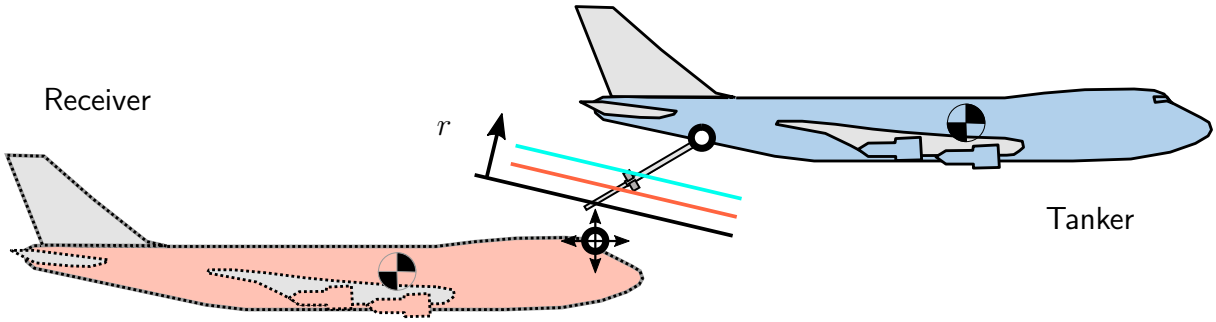


Figure 7.11: *Aircraft in close formation flight – definition of response variable r for verification*

of $P_0 = 10\%$ and $N = 1000$ samples per subset. To prove a failure probability of $P_F \leq 10^{-6}$, five subset levels are used, which yield a total of 6000 samples for estimation of the value of the response variable r that is not exceeded with a probability higher than admissible.
To show the big performance advantage of enhanced stochastic analysis, results from Subset simulation are compared to those of pure Monte Carlo simulation using the same total number of samples. First, figure 7.12 shows the histogram of the response variable, i.e. the proximity of the aircraft, for increasing subset levels. The height of each bar represents the number of samples within the interval of the bin. The vertical lines give the conditional failure thresholds. For comparison, figure 7.13 shows the same information using Monte Carlo simulation. It is obvious that for Monte Carlo simulation samples are mainly generated close to the most likely system response. Opposed to this, samples obtained by Subset simulation are generated at the tail of the distribution, which is, in this example, closer distance between the two aircraft.

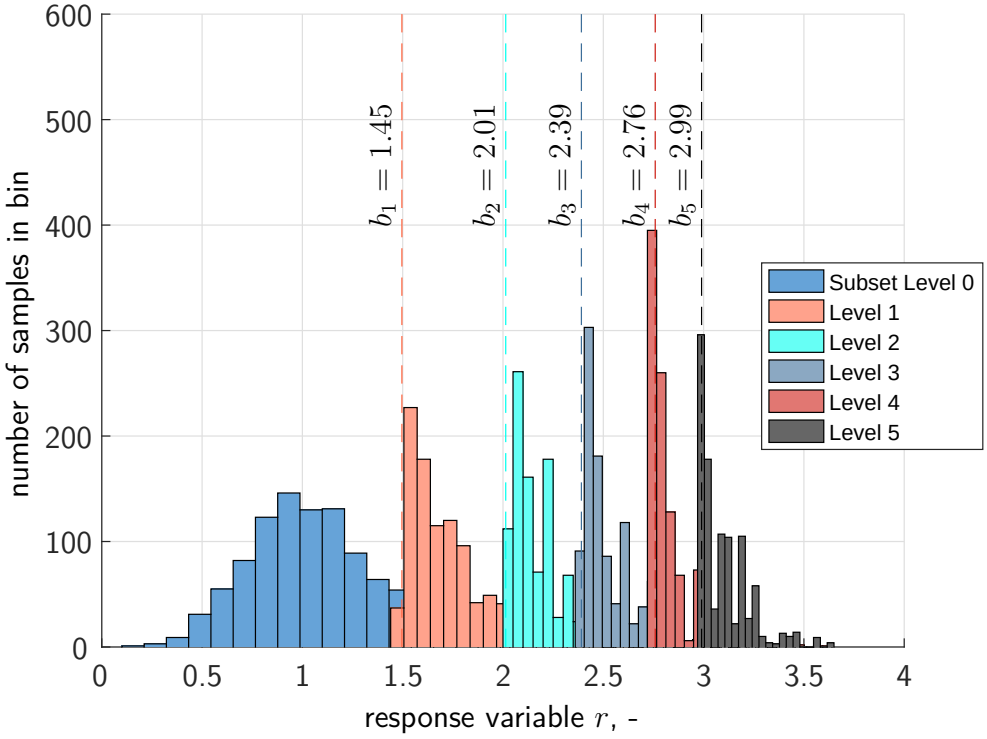


Figure 7.12: *Histogram for response variable using Subset simulation, $P_0 = 0.1$, $N_{total} = 6000$*

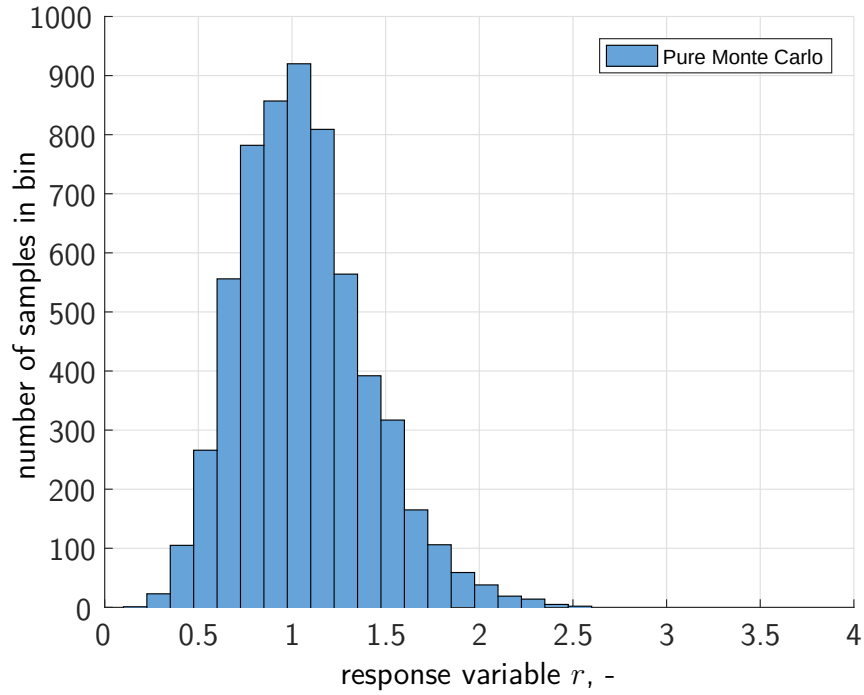


Figure 7.13: *Histogram for the response variable using Monte Carlo simulation,* $N_{total} = 6000$

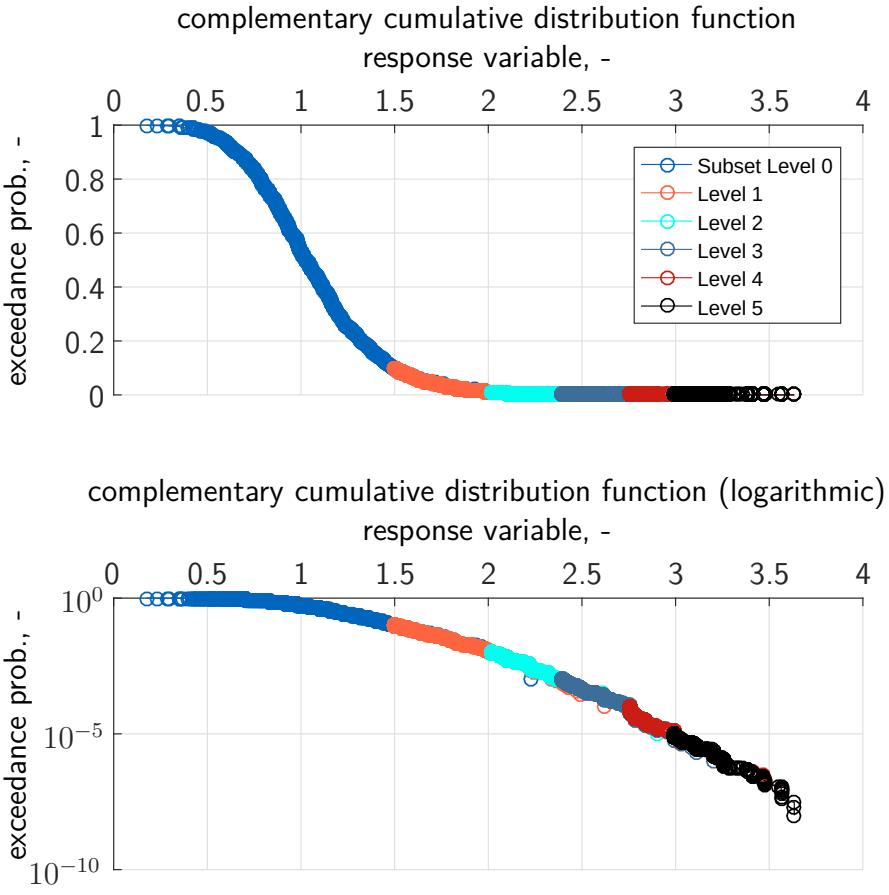


Figure 7.14: *CCDF of the response variable using Subset simulation,* $P_0 = 0.1$, $N_{total} = 6000$

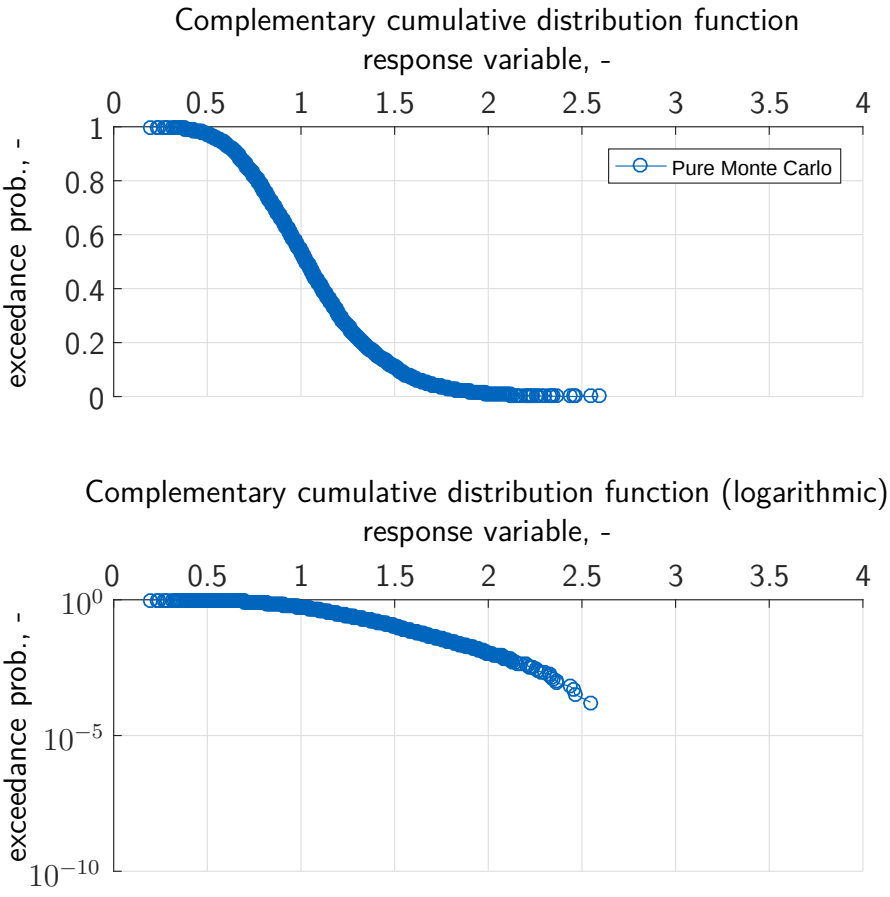


Figure 7.15: *CCDF of the response variable using Monte Carlo simulation,* $N_{total} = 6000$

Figures 7.14 and 7.15 show the *Complementary Cumulative Distribution Function* (CCDF) of the response variable, which gives the probability of exceedance (y-axis) of a certain response value (x-axis). Upper and lower plots show the same information, where the lower plots use a logarithmic scale for the failure probability. With the same number of samples, the estimable failure probability for Subset simulation is in the range of $\hat{P}_F = 10^{-6}$, while for Monte Carlo simulation it is $\hat{P}_F = 10^{-3}$. Subset simulation results in a response value of $3.25m$ with a probability of $\hat{P}_F = 7.7 \cdot 10^{-7}$. The according lower and upper CoV (see section 2.6.4) are 0.5 and 1.1. To obtain an equal estimation accuracy by conventional Monte Carlo simulation, this would require $1 - 5$ million samples (see equation (2.18)), which is 500 times more than for Subset simulation.

To further explain the functioning of Subset simulation especially for the case of stochastic excitation, figure 7.16 shows exemplary trajectories of the relative position in the vertical plane. The dashed lines indicate the conditional failure threshold for the related subset. Sample trajectories of a single Markov chain are used, i.e. the plotted lines relate to each other and the change between different subsets can be interpreted as random walks towards more critical regions. Figure 7.17 shows the according vertical wind component and response history. Note that the intensity of the excitation does not change – the model parameters as well as the intensity of the Gaussian noise, used as input for the Dryden filter, are constant. Only the shape of the excitation is changed, which can be interpreted as a change in sequence and timing

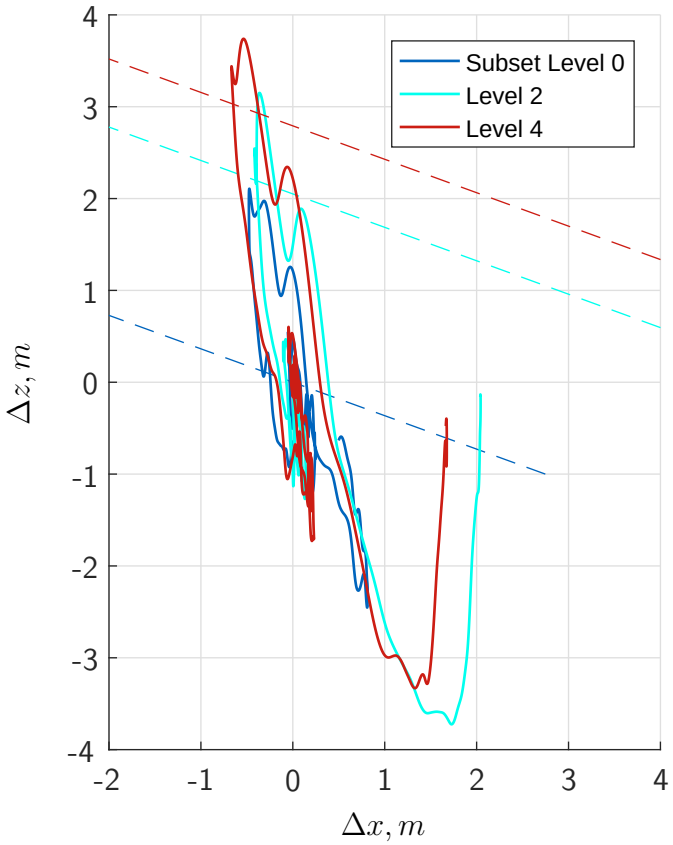


Figure 7.16: *Relative position trajectories in the vertical plane for increasing subset levels*

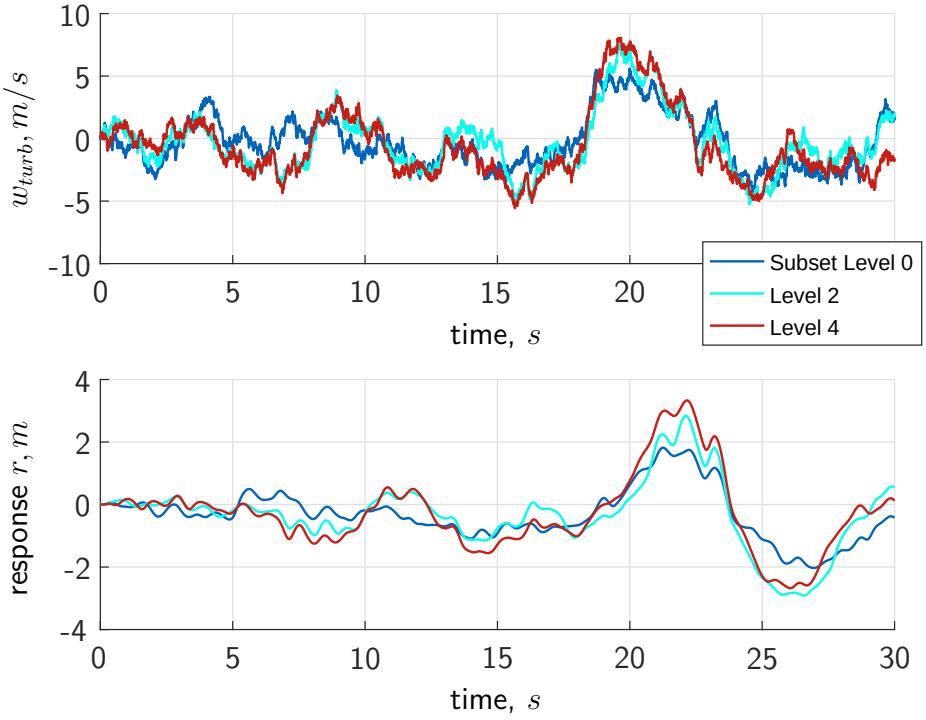


Figure 7.17: *Turbulence excitation and response histories in the vertical plane for increasing subset levels*

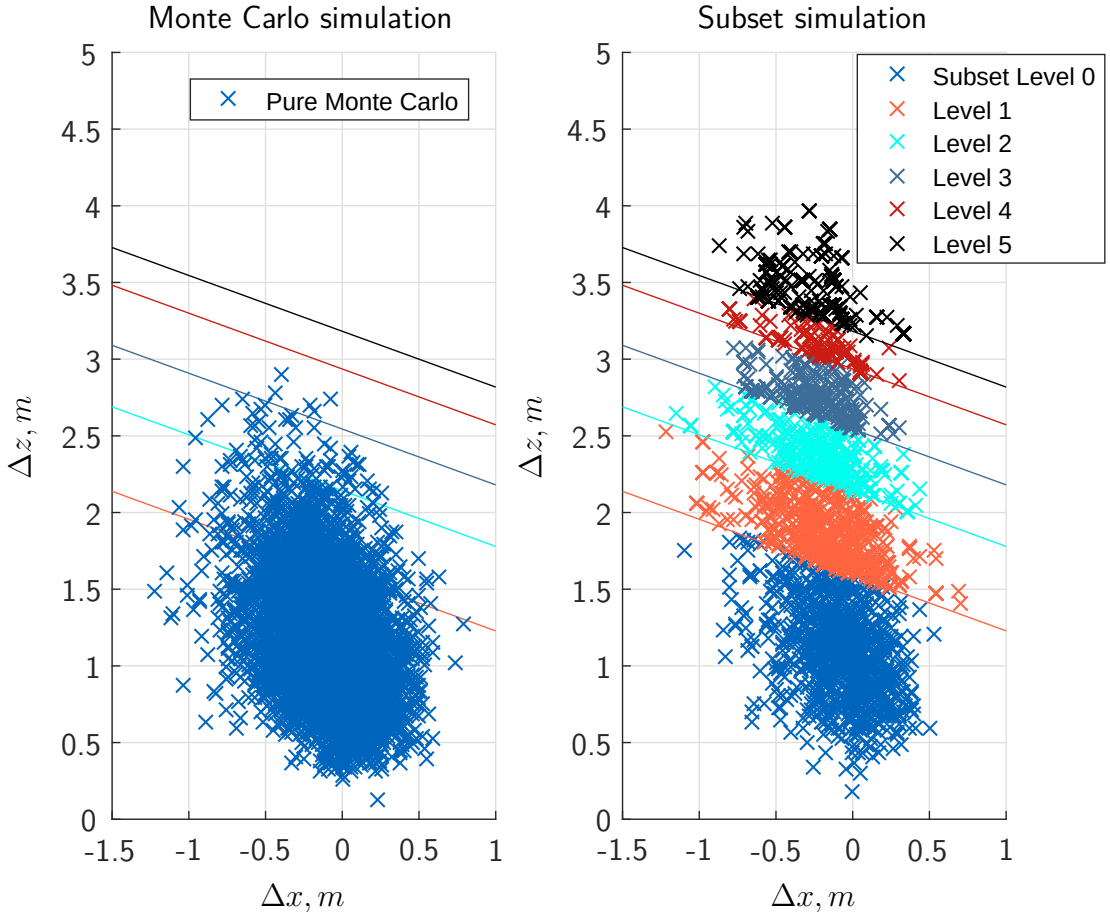


Figure 7.18: *Distribution of maximum response variable in vertical plane,* $N_{total} = 6000$ *for both methods*

of gusts in a way that the closed loop system is excited in the most critical manner. Despite this change, the stochastic uncertain parameters for higher subsets are still uncorrelated and Gaussian distributed, i.e. form a Gaussian (white) noise.

Eventually, figure 7.18 shows the point of maximum proximity for each generated trajectory. While for Monte Carlo simulation, the result is a dense cloud around the most probable maximum position error, samples from Subset simulation are gradually generated towards higher responses. Recall that the samples of the k-th subset does only occur with a probability smaller than 0.1^k, i.e. the samples of the 5-th subset have an estimated probability smaller than 10^{-5}.

The results prove that Subset simulation is very useful for the evaluation of critical system behavior, identification of main contributing uncertainties and for prove of compliance with probabilistic top-level safety requirement. It can be concluded that the proposed approach with enhanced stochastic methods is one powerful mean for optimization and evaluation during development and verification of safety-critical functions. However, note that this mainly aims at the evaluation of functions in the presence of uncertainties and disturbances, while still classical methods for safety assessment must be used to evaluate the effect of component failures (e.g. by Fault Tree Analysis), common failure modes (by Common Cause Analysis), etc.

7.3.4 Online Monitoring

Recall the objectives of online monitoring for runtime verification, which are especially the detection of non-compliant behavior with respect to the specified, admissible behavior, the justification of assumptions and the determination of the current level of safety. All objectives should contribute to the assurance of safety in case that no offline verification is possible or when the safety objective depends on operation and hence considerably higher availability could be achieved by runtime verification.

For the example considered in this chapter, the formation flight autopilot is monitored to justify the assumptions made during development and to identify non-compliant behavior. The application of monitoring to ensure continuous safety is possible here, since the formation flight function is not essential – in case that non-compliant behavior is detected, an abort maneuver can be immediately initiated that ensures a quick separation between the aircraft.

Monitoring of the automatic flight control system can be established on different levels, from individual components, e.g. actuators and flight control computers, to the overall system. In this section, the application of the model-based system monitoring approach introduced in 6.4.2 is shown. Specifically, the vertical relative position control performance is monitored. During requirements derivation, adequate intervals for the tanker translational and rotational motion are identified (7.3.1). For the basic version of the proposed monitoring algorithm, independence of the admissible parameter ranges is required, hence the admissible solution space is approximated by a box-shaped space.

The proposed monitoring algorithm uses linear models for the propagation of uncertainties and disturbances. Although these models can be obtained by (numerical) linearization during runtime, for this example, the linear models are derived manually to better explain the monitoring algorithm. Relative dynamics between the aircraft is compiled by rotational and translational dynamics of both aircraft, which can be considered as linear in proximity of the trim point:

$$\begin{bmatrix} \dot{\alpha}_R \\ \dot{q}_R \\ \dot{\theta}_R \\ \dot{h}_R \end{bmatrix} = \mathbf{A}_{R,closedloop} \begin{bmatrix} \alpha_R \\ q_R \\ \theta_R \\ h_R \end{bmatrix} + \mathbf{B}_{R,wind} \begin{bmatrix} w_{wind,R} \\ q_{wind,R} \end{bmatrix} \tag{7.9}$$

where the angle of attack, the pitch rate, the pitch angle and the altitude are used as receiver states. The system matrix $\mathbf{A}_{R,closedloop}$ reflects the dynamics of the closed loop system of the receiver with its autopilot for altitude hold. The disturbance input matrix $\mathbf{B}_{R,wind}$ maps the wind vertical velocity and pitch rate to the aircraft state derivatives,

$$\mathbf{B}_{R,wind} = \begin{bmatrix} Z_W & Z_Q \\ M_W & M_Q \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{7.10}$$

with Z_W, Z_Q, M_W and M_Q denoting the aerodynamic force and moment derivatives for vertical velocity and pitch rate disturbances. Similar dynamics can be formulated for the tanker

$$\begin{bmatrix} \dot{\alpha}_T \\ \dot{q}_T \\ \dot{\theta}_T \\ \dot{h}_T \end{bmatrix} = \mathbf{A}_{T,adequate}\left(\mathbf{p}\right) \begin{bmatrix} \alpha_T \\ q_T \\ \theta_T \\ h_T \end{bmatrix} + \mathbf{B}_{T,wind} \begin{bmatrix} w_{wind,T} \\ q_{wind,T} \end{bmatrix} \tag{7.11}$$

In contrast to the receiver dynamics, not the actual autopilot is linearized but instead models are used which describes the admissible behavior of the tanker using a full state feedback and pole placement (similar to 7.3.1). This system matrix is a function of the design parameters $\mathbf{p}$, which are modified by the monitoring algorithm to propagate the whole parameter space by using the corner points of the admissible design parameter space. Recall that this is admissible as long as the states are monotonous with respect to the design parameters, which is the case here for the high update rates considered in this example.

While the admissible aircraft closed loop dynamics are inherently linear, there is one major nonlinearity, which is the delayed action of the wind velocity on the receiver. The delay is similar to the longitudinal distance divided by the forward velocity of the aircraft. Due to the usually low resulting delays τ (approximately $\tau = 0.25s$), a first order *Padé* approximation can be made, which can be written in time domain as

$$\begin{aligned} \dot{w}^*_{wind,R} &= -\frac{2}{\tau} w^*_{wind,R} + w_{wind,T} \\ w_{wind,R} &= \frac{4}{\tau} w^*_{wind,R} - w_{wind,T} \end{aligned} \tag{7.12}$$

and likewise for the wind pitch rate. The derivation of the *Padé* approximation is given in appendix B.3. Using the state space representation of the Dryden turbulence model given in appendix B.2, all submodels can be compiled to one linear model for propagation of uncertainties during online monitoring:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{A} &= \begin{bmatrix} \mathbf{A}_{R,closedloop} & \mathbf{O}_{4x4} & -\mathbf{B}_{R,wind}, \mathbf{O}_{4x2} & \frac{4}{\tau}\mathbf{B}_{R,wind} \\ \mathbf{O}_{4x4} & \mathbf{A}_{T,adequate}\left(\mathbf{p}\right) & \mathbf{B}_{T,wind}, \mathbf{O}_{4x2} & \mathbf{O}_{4x2} \\ \mathbf{O}_{4x4} & \mathbf{O}_{4x4} & \mathbf{A}_{Wind} & \mathbf{O}_{2x2} \\ \mathbf{O}_{2x4} & \mathbf{O}_{2x4} & \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{matrix} & \begin{matrix} -\frac{2}{\tau} & 0 \\ 0 & -\frac{2}{\tau} \end{matrix} \end{bmatrix} \\ \mathbf{B} &= \begin{bmatrix} \mathbf{O}_{2x4} & \begin{matrix} \mathbf{B}^T_{CMD} \\ \mathbf{O}_{1x4} \end{matrix} & \begin{matrix} \mathbf{O}_{1x4} \\ \mathbf{B}^T_{wind} \end{matrix} & \mathbf{O}_{4x2} \end{bmatrix}^T \\ \mathbf{x} &= \begin{bmatrix} \mathbf{x}_{dyn} \\ \mathbf{x}_{dist} \end{bmatrix} \\ \mathbf{u} &= \begin{bmatrix} z^{HC}_{cmd} & \eta \end{bmatrix}^T \text{ö} \end{aligned} \tag{7.13}$$

$$\begin{aligned}\mathbf{x}_{dyn} &= \begin{bmatrix}\alpha_R & q_R & \theta_R & h_R & \alpha_T & q_T & \theta_T & h_T\end{bmatrix}^T \\ \mathbf{x}_{dist} &= \begin{bmatrix}w_{wind,T} & q_{wind,T} & w^*_{wind,T} & q^*_{wind,T} & w^*_{wind,R} & q^*_{wind,R}\end{bmatrix}^T\end{aligned} \tag{7.14}$$

The aircraft dynamics matrices are defined with respect to the boom hinge point H for the tanker and the receptacle C for the receiver, respectively. The inputs to the model are the relative vertical position command z^{HC}_{cmd} and the Gaussian noise input η for the turbulence model. Although the former is actually not required for station-keeping, where the commanded relative position is constant, this input is used to simulate the behavior during trajectory tracking required for approach and departure.

For this example, it is assumed that only the vertical relative displacement and velocity is measured. Hence, the output matrix can be obtained using the relations for straight flight $\gamma = \theta - \alpha$ and $\dot{h} \approx V\gamma$:

$$\begin{aligned}\mathbf{C} &= \begin{bmatrix}-\mathbf{C}_R & \mathbf{C}_T & \mathbf{O}_{2x4} & \mathbf{O}_{2x2}\end{bmatrix} \\ \mathbf{C}_{R/T} &= \begin{bmatrix}0 & 0 & 0 & 1 \\ -V & 0 & V & 0\end{bmatrix} \\ \mathbf{y} &= \mathbf{C}\mathbf{x} = \begin{bmatrix}\Delta h & \Delta\dot{h}\end{bmatrix}^T\end{aligned} \tag{7.15}$$

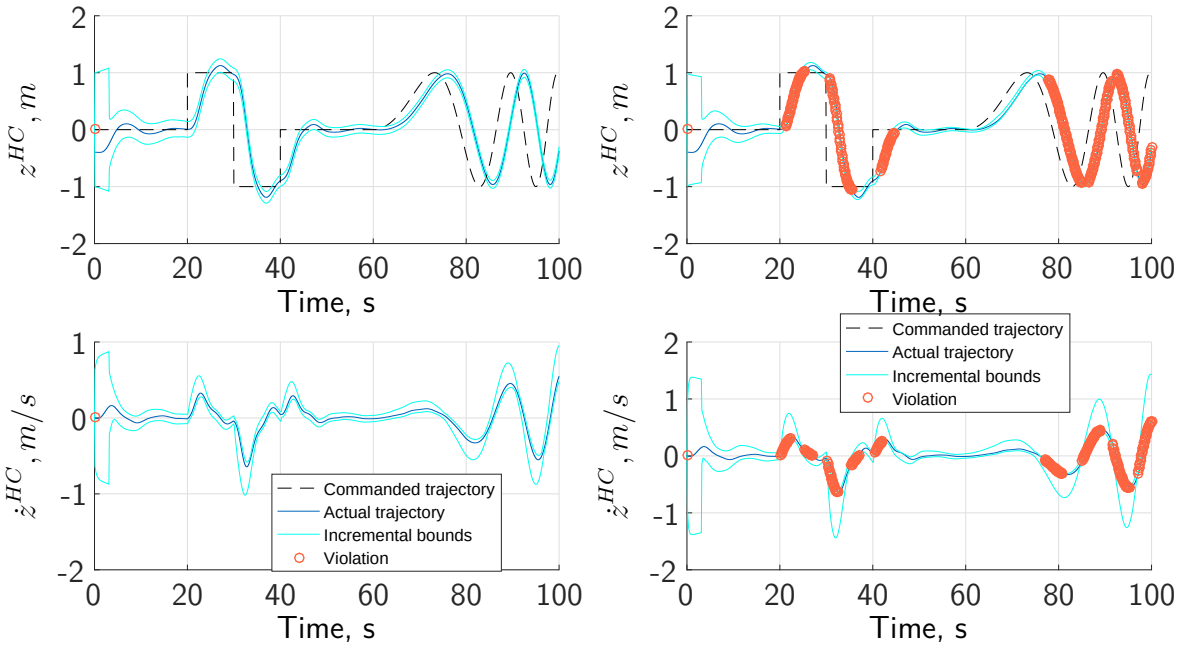
where V and $\Delta h = h_T - h_R$ denote the aircraft's velocity and the height difference, respectively.

The initial covariance matrix is set to zero except for the translational states for which the initial uncertainty is set to the measurement covariance of the sensor system. While the covariances of the translational states, i.e. relative position and velocity, are updated with each measurement according to the scheme presented in 6.4.2, the covariances of the other states are continuously propagated, leading to steady values after a certain settling time, provided that the time delay τ, the turbulence intensity and the airspeed do not change.

Five different scenarios are looked at, which are listed in table 7.3. To increase the comprehensibility of the results, only the tanker translational dynamics is considered as design parameter, while the rotational dynamics is considered as constant in this example. Figure 7.19 shows the resulting relative vertical position and velocity between the boom hinge H and the receptacle C from the sophisticated nonlinear simulation model, together with the

Table 7.3: *Limits and discretization of design parameters*

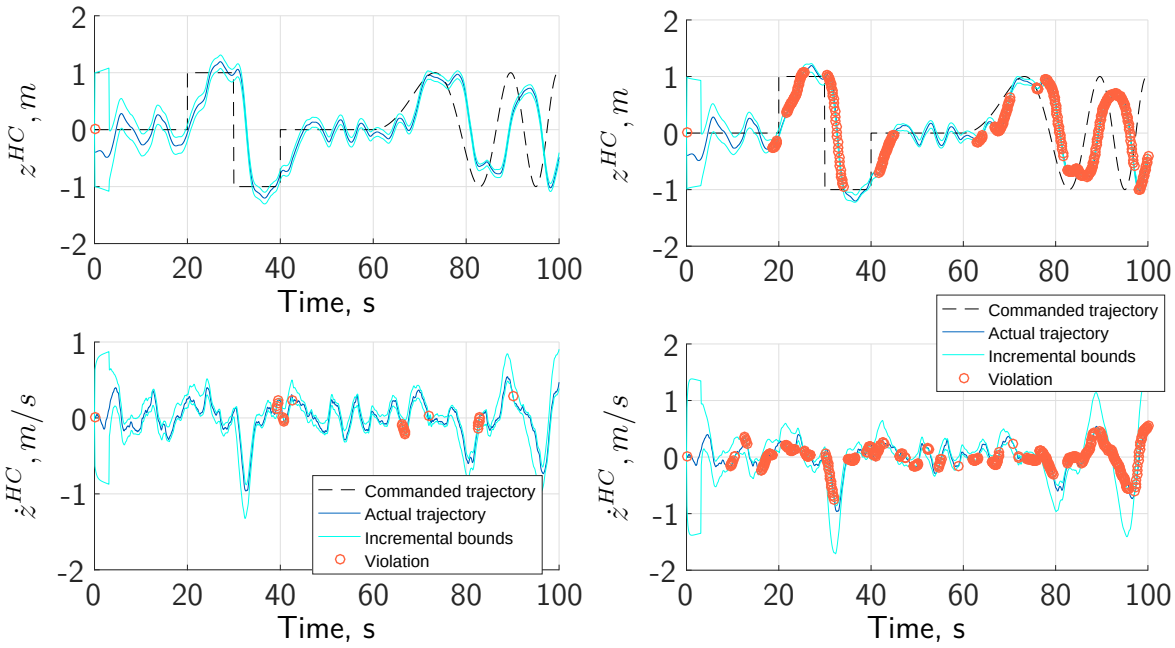
Case	Turbulence Intensity	Adequate dynamics ω_{trans} range	Adequate dynamics ζ_{trans} range	Remarks
$1a$	No	$[0.3 \dots 2]$	$[0.5 \dots 1.5]$	Normal operation
$1b$	No	$[1.5 \dots 2]$	$[0.5 \dots 1.5]$	Closed loop too slow
$2a$	Light	$[0.3 \dots 2]$	$[0.5 \dots 1.5]$	Turbulence excitation
$2b$	Light	$[1.5 \dots 2]$	$[0.5 \dots 1.5]$	Turbulence excitation with too slow closed loop
3	No	$[0.3 \dots 2]$	$[0.5 \dots 1.5]$	Autopilot failure at $25s$



(a) *case* $1a$ **(b)** *case* $1b$

Figure 7.19: *Monitoring results for case* 1

calculated single covariance monitoring bounds propagated using the linear dynamics (7.13). Arbitrary control inputs are chosen to show monitoring of trajectory tracking performance. The larger variance bounds during the first seconds are caused by the higher measurement uncertainty during sensor data fusion settling. In case $1a$, where the aircraft dynamics lie well within the adequate dynamics intervals, neither position nor velocity boundaries are exceeded, i.e. the actual system behaves according to the specifications (figure 7.19a). Figure 7.19b shows results for the case that the aircraft dynamics is too slow compared to the desired dynamics. For the sake of simplicity, the lower limit for the requirement is increased for this case instead of retuning the controller to result in too low dynamics. It can be seen that for both relative position and velocity, exceedances of the lower dynamic boundaries are detected multiple times, which is indicated by orange circles. Certainly, non-conformal behavior can only be detected as long as there are control or disturbance inputs that cause state variations. In cases $2a$ and $2b$, boundary propagation in the presence of stochastic turbulence excitation is examined. Similar to the generic example used in 6.4.2, there is a certain probability that boundaries are exceeded when stochastic disturbances are present. Figure 7.20 shows results for case 2, where double standard deviation bounds were chosen for the uncertainty propagation of the turbulence excitation, i.e. a maximum of 5% of samples is expected to violate the boundaries. In case $2a$, which represents an admissible system at acceptable turbulence levels, there are only a few boundary exceedances caused by the stochastic nature of the turbulence excitation. The results for case $2b$ in figure 7.20b show that the detection capabilities for non-admissible dynamic behavior also works well with stochastic disturbances acting on the plant. Finally, case 3 depicted in figure 7.21 shows the results when the autopilot is switched off after $25s$, simulating a non-detected autopilot failure. Although the plant behaves within



(a) *case* $2a$ **(b)** *case* $2b$

Figure 7.20: *Monitoring results for case* 2 *(with turbulences)*

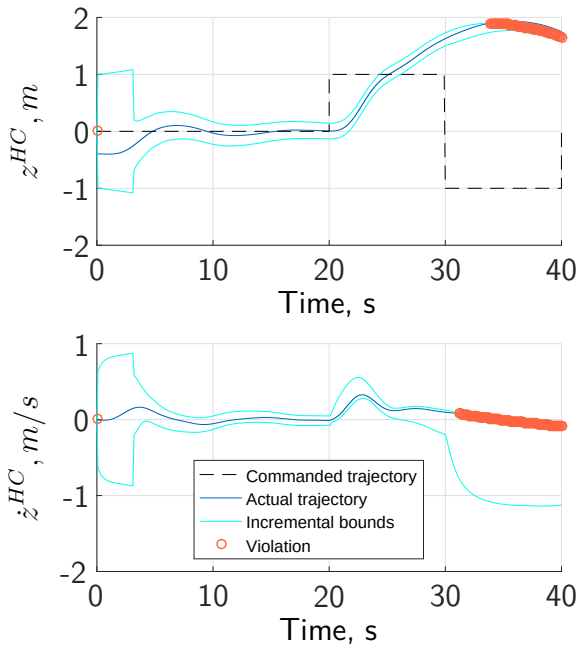


Figure 7.21: *Monitoring results for case* 3 *(autopilot failure)*

desired dynamic bounds for a couple of seconds, abnormal behavior is detected soon after both adequate velocity and position bounds are persistently exceeded.

Using the proposed algorithm, it is possible to monitor the behavior of a plant in the presence of disturbances and uncertainties. By using models for the admissible behavior, failures can be detected more effectively and fault detection rates are lower than for simple monitoring approaches using only comparisons of measurements with predefined limits. However, although models are used, the emphasis lies on using models that are only as complicated as necessary,

but as simple as possible. This is motivated by the demand for efficiency, which ensures real-time capabilities required for online application. If requirements are nonlinear or non-Gaussian stochastic processes act on the plant, piecewise linearization and Gaussian mixture models can be used. These aspects are investigated by another researcher at the Institute of Flight System Dynamics ([Müh18]). While there might be more sophisticated monitoring approaches, the proposed method captivates due to the simplicity and still high performance for monitoring of dynamic systems, especially if the acceptable dynamics is specified by linear models with admissible dynamic intervals.

8

Conclusions and Perspectives

8.1 Conclusions and Recommendations

The research presented in this work is driven by the desire to lay the foundation for a new development approach for safety-critical functions that replaces the current experience driven, conservative bottom-up process. This novel approach is named *Total Capability Approach* (TCA), since the total capabilities of a system in the presence of uncertainties, disturbances, and failures are utilized throughout the whole development process, from specification, design, implementation, to verification. This is enabled by the increased computational power of today's computers and the availability of adequate models for the simulation of the behavior of a function and system in its environment. In the following, the main contributions and related findings are recapitulated and extended by recommendations for application and enhancement of developed methods.

Efficient stochastic analysis is essential for the proposed model-based approach to estimate small failure probabilities. Chapter 2 introduces methods that have found to be useful for the intended applications during requirements derivation, function design and verification. Emphasis is laid on methods that are easy to apply and preferably follow the black box principle, which significantly reduces the effort and complexity during daily application. The methods are described in a vivid way, with the focus not on theoretic proofs but on understanding and implementation, to close the gap between theoretic text books and the intended application. This is a valuable contribution in this field. Especially *Subset simulation* based on *Markov Chain Monte Carlo* is discussed in detail, because this method has been found as versatile and powerful for efficient evaluation of small failure probabilities, which are inherent to the development of safety-critical functions. Subset simulation is considered to be one of the most efficient black box methods for such analyses. Although methods adapted to specific applications can be more efficient, they require considerably more awareness and expert knowledge to yield valid estimation results. This impedes the application in practice due to the additional effort and the loss of generality. For these reasons, it is recommended to stick to the black box methods, which allow engineers to efficiently evaluate small failure probabilities without the need for long training and adaption periods.

The core contribution of this research is the development of the overall **concept of the TCA**, which is introduced in chapter 3. After an introduction of the currently established development process, drawbacks are identified and a shift in paradigm is introduced, which constitutes a transition from the conventional bottom-up requirements derivation process towards a top-down approach, where high level safety objectives are broken down in a physically motivated manner. The motivation for that is the anticipated gain in design freedom due to the lower level of conservatism, but even more the lack in specific lower level design specifications for novel applications and operations. The latter inhibits application of the conventional certification approach, which highly depends on past experience. Although the TCA mainly aims at revolutionizing the way how requirements of safety-critical functions are derived, the proposed shift also has an influence on or provides additional useful means for the subsequent development steps, i.e. validation, design, implementation, verification and operation. Chapter 3 motivates the research presented in the remaining work, where later the different steps of the development process are discussed in detail to identify means to put the TCA into practice and overcome the top challenges identified in section 3.6. During beginning of the underlying research, certification authorities almost entirely relied on the conventional approach with prescriptive design requirements. Only during the recent months, they released novel certification specifications for normal, utility, aerobatic and commuter aeroplanes (CS-23/Amendment 5 [Eur17d]). This enables a shift to a probabilistic approach for certification of safety-critical functions, where it must only be proven that the implemented function ensures safety during operation. This strongly underlines the practical significance of the conducted research and hence the author highly recommends to continue research on this important topic.

Chapter 4 is dedicated to **requirements derivation and validation**. Efforts related to the model-based approach are discussed in section 4.1, which are especially the formalization of requirements, the definition of adequate specification models for simulation of the required behavior before the availability of actual implementations, as well as the modeling of the environment with all relevant uncertainties, disturbances and failures. The latter is essential for a probabilistic assessment of the specified functions and systems. Another major contribution is a standardized simulation framework, which is introduced in section 4.1. It enables the efficient utilization of formalized requirements, specification models and environment models throughout the whole development process. With development progress, individual components are enhanced (e.g. environment models) or replaced (e.g. specification by actual design models), while the interfaces required for evaluation and analysis are kept. The shift towards a top-down approach leads to the question on how to break down safety-related top level requirements with only very low acceptable violation probabilities to specific design requirements. The challenges related to this break down are discussed in 4.2 and, beyond the state-of-the-art, a method is developed and presented that can be used to identify high-dimensional admissible specification intervals of lower-level requirements that ensures compliance with the probabilistic top-level requirements. This proposed approach can lead to significantly increased design parameter intervals compared to conventional, conservative certification standards, which stip-

ulate experience-based intervals for the lower-level requirements. The results for an exemplary application of the developed method shown in 7.3.1 prove the usability of the algorithms. However, due to the high computational effort and the difficulty for the selection of adequate design parameter intervals based on the possibly complex and high-dimensional solution space, application is currently only considered useful for lower numbers of design parameters (four to five based on experience). Nevertheless, if the design problem can be split up into independent design sub-problems, this should be no showstopper for imminent applications of the proposed ideas and methods for model-based requirements derivation. Still, future research in this area should focus on increased efficiency and automation, since this is the key for acceptance outside of academia.

Validation of specified functions can benefit from the model-based approach. This is discussed in 4.4. Due to the early availability of models, which describe the specified system behavior, it is possible to easily evaluate a set of specifications. This is a useful means for proving completeness and correctness of requirements. This additional usage is not essential for the TCA, but can be seen as very helpful by-product of the model-based derivation process. Since it comes with almost no additional effort, the author recommends to utilize this additional means for validation. The high potential of model-based support of validation is also discussed for the exemplary application in 7.3.2.

The design specifications resulting from the model-based derivation still uses conventional metrics, with only the range of these metrics being determined in a significantly less conservative, physically driven manner. This is important to ensure ease of application, since design engineers do not have to get used to unfamiliar metrics. Hence, the TCA does not require changes to possibly well established design processes. However, similar to validation, the design process can benefit from the formalized requirements and enhanced stochastic methods described in this work, since they allow an easy evaluation and optimization of preliminary designs to obtain the function with the highest availability at any given level of safety. This is discussed in section 5.1.

The proposed development approach requires a change of thinking for **verification**, which is discussed in 5.2 and demonstrated in 7.3.3. Verification by analysis using models is an accepted means of compliance already today. However, it becomes even more important if requirements are derived using the model-based approach proposed in this work. During requirements break down, it is inevitable that assumptions are made on certain aspects that are unknown during specification. Furthermore, additional sources of uncertainties usually arise during design and implementation, which are not considered during the probabilistic break down of requirements. Therefore, it is no longer sufficient to prove compliance with probabilistic top-level requirements by just fulfilling lower-level requirements. Although assumptions are also required for today's development approaches, compliance with highly conservative prescriptive design specifications ensures safety of the overall system. For the TCA, compliance of the implemented functions must therefore also be shown for the probabilistic top-level requirements. Here, the application of enhanced stochastic algorithms described in chapter 2, and especially Subset

simulation, has proven as very powerful for evaluation of small failure probabilities, even for complex simulation models, which require high computational effort. The main contribution for verification is the simulation environment used throughout this work that allows easy application of enhanced stochastic methods thanks to the fixed interfaces between implemented functions and evaluation routines. Furthermore, for the first time the application of enhanced stochastic methods and the implications on verification or safety-critical functions are captured.
The development of the TCA is motivated by emerging novel (e.g. unmanned) aircraft operations. For conventional aircraft, safety is mainly driven by the effects on crew, passengers and airframe. For unmanned operation, safety is especially driven by the actual mission, e.g. flight above a crowd of people versus over an uninhabited desert. Hence, it is desirable to adapt the safety goals to the current application. **Runtime verification** is considered as one major mean to allow time-varying safety goals. Following a detailed motivation, evaluation and derivation of related challenges and objectives in sections 6.1-6.3, a beyond state-of-the-art online monitoring algorithm is derived in 6.4.2 and exemplarily applied in 7.3.4. The proposed algorithm allows for monitoring of one important type of system requirements in the presence of uncertainties and stochastic disturbances. Good violation detection capabilities could be demonstrated by simulation. To further enhance the algorithm and its robustness, the author proposes the evaluation during real flight tests.
The application of the TCA for selected aspects of development of a formation flight autopilot in chapter 7 proves the potential of the principle idea as well as of the methods developed and described in this work. This proof of principle of the TCA is important, since it underlines the validity of the contributed ideas, methods and especially the established path towards implementation of a performance-based certification approach.

8.2 Outlook

Especially during the recent months, the European and US certification authorities started the transition from exhaustive requirements catalogs towards a much more flexible performance based approach that is perfectly in line with the conducted research. Although the results presented in this thesis, and especially the concept of the TCA, constitute a promising approach to tackle the challenges related to this shift, it can only lay the foundation. Ongoing research is conducted by another researcher at the Institute of Flight System Dynamics of TUM, who currently works on a proof of concept by applying the ideas of the TCA to the development of an automatic landing system for a twin engine, optionally piloted vehicle [Mum18].
During the conduction of research, several further interesting aspects came up, which could not all be considered in this research. Besides the short term recommendations already mentioned in the previous section to further enhance the proposed methods, the following challenges should be addressed in future:

- **Formalization of requirements**: The availability of formalized requirements and related specification models is essential for model-based requirements derivation. While

currently formalization and development of models is mainly done manually, a high level of automation is the key to acceptance of the TCA and its future implementation in industrial projects.

- **Adequate modeling of uncertainties**: The accuracy of the results heavily depends on the availability of adequate models for uncertainties and disturbances. While their availability is assumed in this thesis, determination of these models is usually related to high efforts. To account for poor uncertainty models and assumptions, a certain level of conservatism should be kept in the TCA. Therefore, further research should evaluate the influence of poor models, their implications on safety of the resulting functions and find ways to ensure an adequate level of robustness.

- **Increased efficiency of model-based requirements derivation**: The computational complexity of the proposed approach explodes with the size of the design problem. This research presented in this dissertation approaches this challenge from two sides: enhanced stochastic algorithms and more efficient methods for determination of design bounds. The combination leads to a reasonable performance for up to $4 - 6$ design parameters. This is an already usable number, especially if design problems can be divided into several subproblems. Still, an integrated consideration of complex functions would further utilize the potential of the TCA.

- **Operation-dependent adaption of safety goals**: The potential of time-variant safety goals dependent on the current operation is already introduced in this work and the proposed online monitoring algorithm can be utilized to monitor the system performance in the presence of uncertainties and disturbances. Still, the practical implementation raises issues, which must be addressed in future, especially to ensure a high level of confidence in the monitor (low rate of missed detection) as well as high availability (low rate of false alarms). Furthermore, the legal basis for such an adaption must be developed or identified, which cannot be foreseen at the current time.

In summary, it can be said that the model-based development approach introduced in this work is very promising for tackling future challenges arising from the development of safety-critical functions of unconventional aircraft topologies and novel operations. Therefore, the author highly recommends and promotes continuation of research in this field, to enable a quick transition from research to industry.

Appendix A

Stochastics

A.1 Gauss Standardization

A.1.1 Generation of Samples with Arbitrary Distributions

Many non-Gaussian random variables can be constructed from standard Gaussian variables by transformation. The method originally used for inversion sampling can be used to transform a standard Gaussian variable to any distribution where the CDF is strictly monotonically increasing [Dev86, p. 27ff.].
Assume that a sample θ_T must be generated from the target probability density function $\phi_T(\cdot)$ and according CDF $\Phi_T(\cdot)$ based on standard Gaussian random variables θ_N, i.e. with zero mean and unit standard deviation. Using the CDF of the standard Gaussian distribution $\Phi_N(\cdot)$, samples distributed according to the target distribution are obtained by

$$\theta_T = \Phi_T^{-1}\left(\Phi_N\left(\theta_N\right)\right) \tag{A.1}$$

Note that the result of $\Phi_N\left(\theta_N\right)$ is a random variable with a uniform distribution. The second step, i.e. the transformation of a uniformly distributed random variable using the inverse of the target CDF is usually referred to as inverse principle. In many cases, no analytical expression of the inverse CDF is available. Although this is a common exclusion reason for the efficient generation of random variables distributed according to ϕ_T, for the application of Gauss standardization, performance losses due to less efficient inversion algorithms to obtain $\Phi_T^{-1}\left(\cdot\right)$ (e.g. by look-up tables or gradient methods) are usually outweighed by the performance gains when only working with standard Gaussian random variables. This is for example the case when using Infinity Sampling (see section 2.6.2).
Figure A.1 gives an example of transforming a standard Gaussian random variable to a Weibull distribution. The left upper plot shows the PDF of the standard Gaussian distribution $\phi_N\left(\cdot\right)$ from which a sample θ_N is generated. The lower left figure shows the CDF $\Phi_N\left(\cdot\right)$. The mapping of θ_N using $\Phi_N\left(\theta_N\right)$ leads to a uniformly distributed random number. In the next step, this number is mapped using the inverse of the target CDF $\Phi_T\left(\cdot\right)$ (arrow to the right)

which directly results in a random number that is distributed according to the target density $\phi_T(\cdot)$.

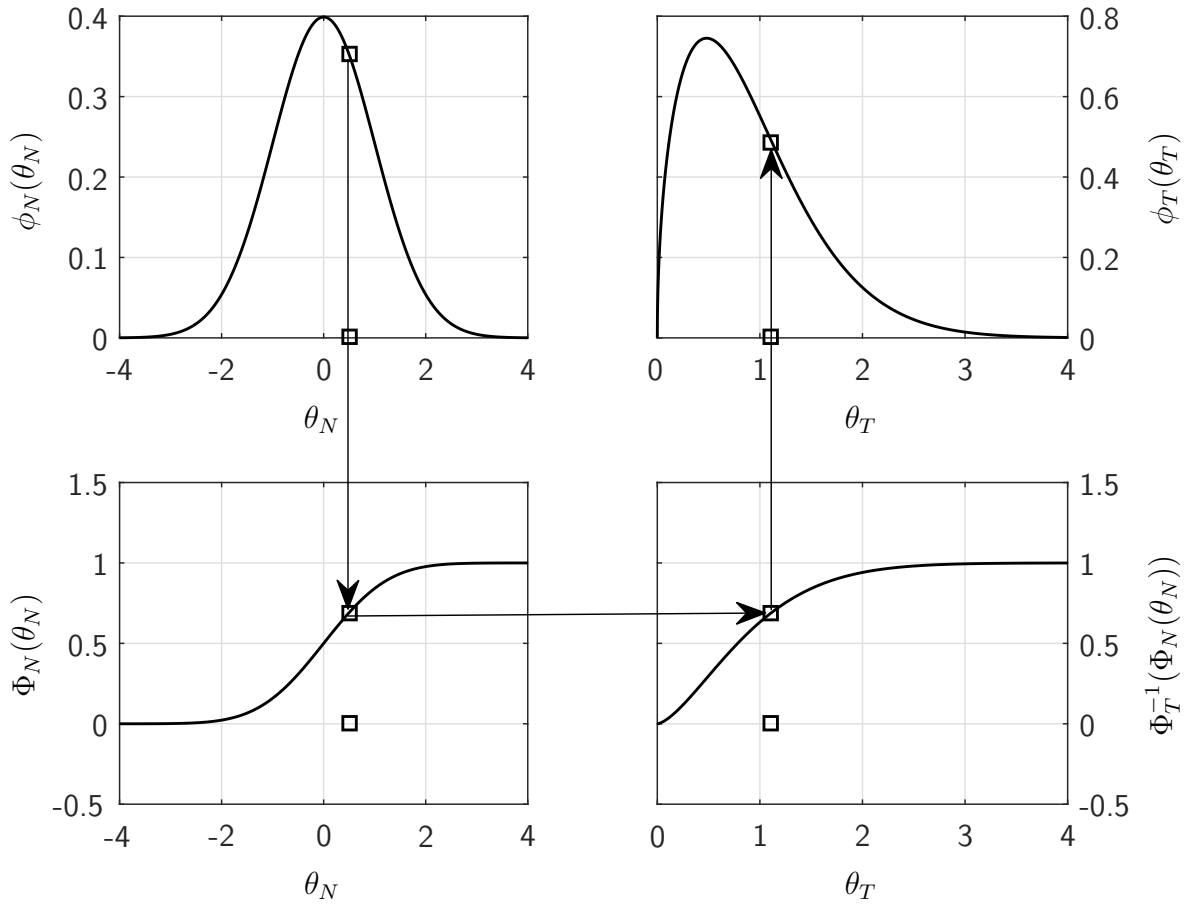


Figure A.1: *Exemplary transformation from a standard Gaussian random variable to a Weibull distribution*

A.1.2 Generation of Correlated Samples

Correlated samples can be obtained from uncorrelated samples. The most common description of correlation is using a covariance matrix $\mathbf{\Sigma}$ which is defined by [Dev86, p. 564]:

$$\Sigma_{ij} = cov(y_i, y_j) = E\left(y_i - \mu_i, y_j - \mu_j\right) \tag{A.2}$$

The matrix $\mathbf{\Sigma}$ is symmetric and usually positive definite, which means that there exist a matrix $\mathbf{G}$ so that

$$\mathbf{G}\mathbf{G}^{\mathbf{T}} = \mathbf{\Sigma} \tag{A.3}$$

For the given conditions, a matrix $\mathbf{G}$ can always be found that has a lower triangular form. This can be done by *Cholesky* decomposition [HJ13, chapter 7.2]. If $\mathbf{G}$ is a lower triangular matrix, a vector $\mathbf{y}$ of Gaussian distributed random variables with covariance matrix $\mathbf{\Sigma}$ and mean $\boldsymbol{\mu}$ can be obtained by

$$\mathbf{y} = \mathbf{G}\mathbf{x} + \boldsymbol{\mu} \tag{A.4}$$

where $\mathbf{x}$ is a vector of independent zero mean unit variance random numbers.

A.2 Inverse Gaussian Distribution

$$\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{-t^2/2} \mathrm{d}t \tag{A.5}$$

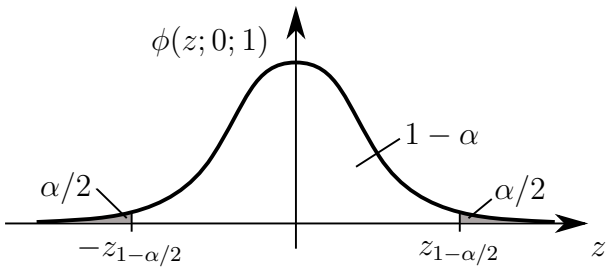


Figure A.2: $1-\alpha/2$ *quantile for standard Gaussian distribution*

Table A.1: $1-\alpha$ *and* $1-\alpha/2$ *quantiles for the standard Gaussian distribution*

α	$z_{1-\alpha/2}$	$z_{1-\alpha}$	α	$z_{1-\alpha/2}$	$z_{1-\alpha}$
0.0010	3.2905	3.0902	0.0600	1.8808	1.5548
0.0020	3.0902	2.8782	0.0700	1.8119	1.4758
0.0027	3.0000	2.7822	0.0800	1.7507	1.4051
0.0030	2.9677	2.7478	0.0900	1.6954	1.3408
0.0040	2.8782	2.6521	0.1000	1.6449	1.2816
0.0050	2.8070	2.5758	0.1100	1.5982	1.2265
0.0060	2.7478	2.5121	0.1200	1.5548	1.1750
0.0070	2.6968	2.4573	0.1300	1.5141	1.1264
0.0080	2.6521	2.4089	0.1400	1.4758	1.0803
0.0090	2.6121	2.3656	0.1500	1.4395	1.0364
0.0100	2.5758	2.3263	0.1600	1.4051	0.9945
0.0200	2.3263	2.0537	0.1700	1.3722	0.9542
0.0300	2.1701	1.8808	0.1800	1.3408	0.9154
0.0400	2.0537	1.7507	0.1900	1.3106	0.8779
0.0455	2.0000	1.6901	0.2000	1.2816	0.8416

Appendix B

Simulation Environment

B.1 Model of Wake Interaction

This section introduces the method used for calculation of the wake interaction between the aircraft. It is based on a project report written by the author of this thesis [Löb+12]. Only main points are outlined here, while a detailed evaluation of the wake interaction model can be found in the referred report.

The most important influence between the aircraft in close formation flight are by far the aerodynamic interactions. The receiver flies close behind the tanker and is therefore directly within the downwash of the tanker. This influences on the one hand the performance of the receiver, since it will have to fly with a higher angle of attack, causing more drag and thus requiring more thrust. Thrust is limited, especially at higher altitudes where the maximum available power of the engines decreases. On the other hand, the non-uniform downwash causes additional moments (i.e. roll, pitch and yaw moments) on the receiver, which complicates the control task during the formation flight.

The provided simulation model does certainly not comprise aerodynamic effects due to formation flight. Thus, the aerodynamic influences have to be modeled. To do this, a preliminary design tool for propeller-wing aerodynamics (by *Hans-Jörg Steiner*, [Ste10, Ste11]) is utilized which is generally used for generation of aerodynamic data for conceptual aircraft designs. This tool is based on lifting line method for calculation of steady state aerodynamics. According to the manual of the tool [Ste10, p. 21], it "*is based on the method described by Phillips and Snyder and constitutes an adaption of the classical lifting line theory applying a three-dimensional vortex lifting law instead of the two-dimensional Kutta-Joukowski law used in classical theory. This enables the method to be used for systems of lifting surfaces with arbitrary camber, sweep and dihedral*".

For calculation, all lifting surfaces are divided into (an arbitrary count of) segments. The tool first calculates the induced velocity at each segment. Afterwards the aerodynamic forces and moments arising from the resulting air stream – i.e. the sum of free stream and induced velocities – are calculated. These reactions are then totalized and converted to aerodynamic coefficients according to the following definition:

$$
\begin{aligned}
C_D &= [-1,0,0]^T \frac{\left(\vec{F}_A^R\right)_A}{\bar{q}_{Ref} S_{Ref}} \\
C_Y &= [0,1,0]^T \frac{\left(\vec{F}_A^R\right)_A}{\bar{q}_{Ref} S_{Ref}} \\
C_L &= [0,0,-1]^T \frac{\left(\vec{F}_A^R\right)_A}{\bar{q}_{Ref} S_{Ref}} \\
C_l &= [1,0,0]^T \frac{\left(\vec{M}_A^R\right)_B}{\bar{q}_{Ref} S_{Ref} b_{ref}/2} \\
C_l &= [0,1,0]^T \frac{\left(\vec{M}_A^R\right)_B}{\bar{q}_{Ref} S_{Ref} c_{ref}} \\
C_l &= [0,0,1]^T \frac{\left(\vec{M}_A^R\right)_B}{\bar{q}_{Ref} S_{Ref} b_{ref}/2}
\end{aligned}
\tag{B.1}
$$

Force and moments are calculated relative to a user defined reference point R and with respect to the aerodynamic A and body-fixed B frame for aerodynamic forces $\left(\vec{F}_A^R\right)_A$ and moments $\left(\vec{M}_A^R\right)_B$ respectively. $\bar{q}_{Ref}$, S_{Ref}, b_{ref}, and c_{ref} are dynamic pressure, reference wing area, wing span, and chord length respectively.

To be able to calculate the aerodynamic interactions between two aircraft, first a single model of the respective aircraft is implemented in the aforementioned aero-tool (see figure B.1). The geometry of the aircraft is used as input. An average center of gravity is used as reference point (i.e. 25% mean aerodynamic chord). The fuselage is modeled as a rotationally symmetric tube using slender body theory for calculation. Since the calculated interactions shall be implemented into the underlying simulation model, the aim is to match the aerodynamic properties of the calculated data of a single aircraft to these of the simulation model. Thus, the aerodynamic model used for calculation is validated against the aerodynamic data of the aircraft simulation model. To achieve this, a lift coefficient C_L versus angle of attack α curve is extracted from the provided aircraft simulation model. The same curve is calculated for the given configuration with the mentioned aero-tool. By modifying adjustable parameters (e.g. incidence angle of wings and horizontal stabilizer), a good match between these two curves can be achieved (see figure B.2). Being able to calculate the aerodynamics of a single aircraft, a second aircraft is modeled behind the first aircraft to calculate the influence of wake of the leader aircraft on the follower aircraft, see figure B.3. Since the resulting, absolute aerodynamic coefficients would be too inaccurate for an implementation in the existing simulation model, differences between the influenced coefficients and the ones resulting from free flow are calculated. These differences can be added as increment to the already existing,

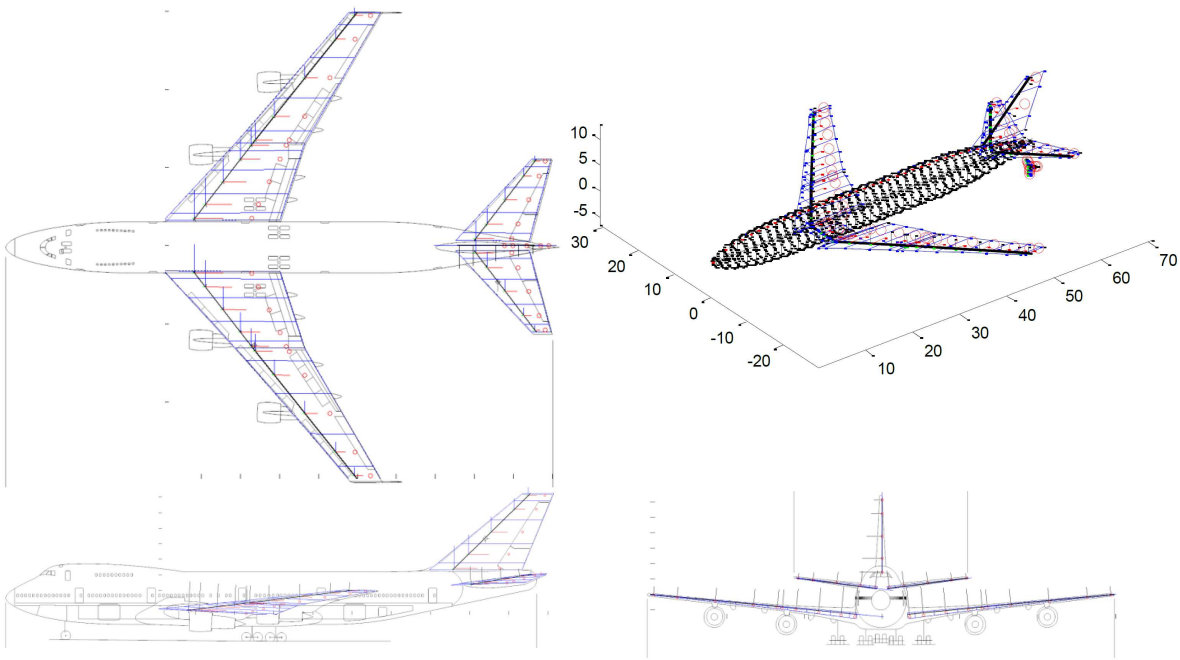


Figure B.1: *3-side view of the aerodynamic model of the tanker*

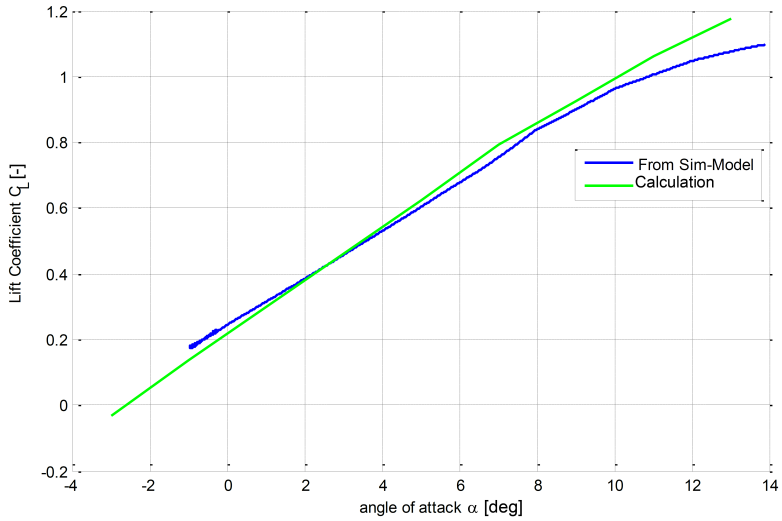


Figure B.2: C_L *versus* α *curve for a single aircraft*

sophisticated aerodynamic model of the provided simulation model.

$$\begin{bmatrix} \Delta C_D \\ \Delta C_Y \\ \Delta C_L \end{bmatrix} = \begin{bmatrix} \Delta C_D \\ \Delta C_Y \\ \Delta C_L \end{bmatrix}_{wake} - \begin{bmatrix} \Delta C_D \\ \Delta C_Y \\ \Delta C_L \end{bmatrix}_{Freeflow}$$
$$\begin{bmatrix} \Delta C_l \\ \Delta C_m \\ \Delta C_n \end{bmatrix} = \begin{bmatrix} \Delta C_l \\ \Delta C_m \\ \Delta C_n \end{bmatrix}_{wake} - \begin{bmatrix} \Delta C_l \\ \Delta C_m \\ \Delta C_n \end{bmatrix}_{Freeflow} \tag{B.2}$$

The incremental derivatives are calculated dependent on the relative position between the

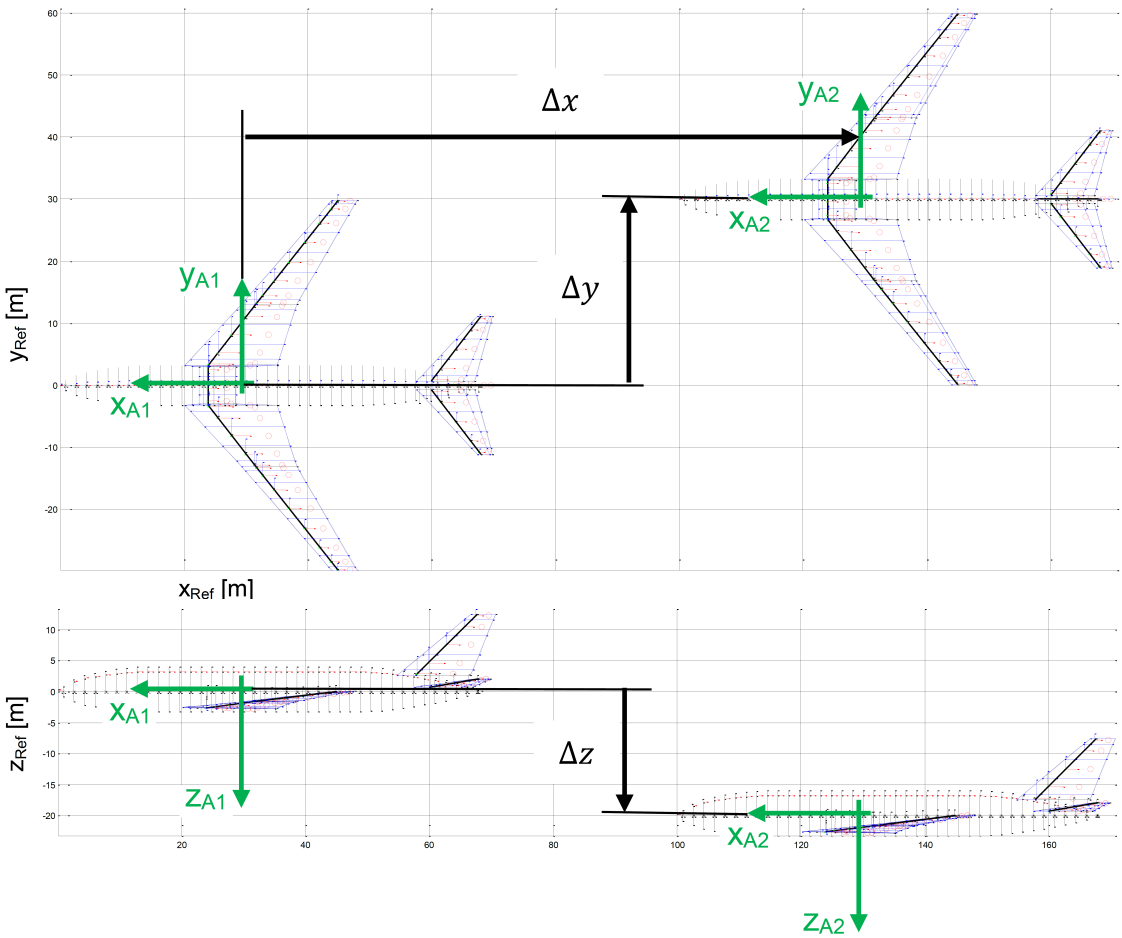


Figure B.3: *Definition of relative position for determination of aerodynamic interference*

aircraft and on the angle of attack of the forward aircraft. The relative position is defined as the distance between the reference points of the involved aircraft, given in the aerodynamic frame of the Tanker. The aerodynamic frame is used, since vortices approximately leave in direction of the free flow. This gives the best approximation of downwash for close distances. Eventually, figure B.4 shows the downwash velocity right behind the tanker aircraft.

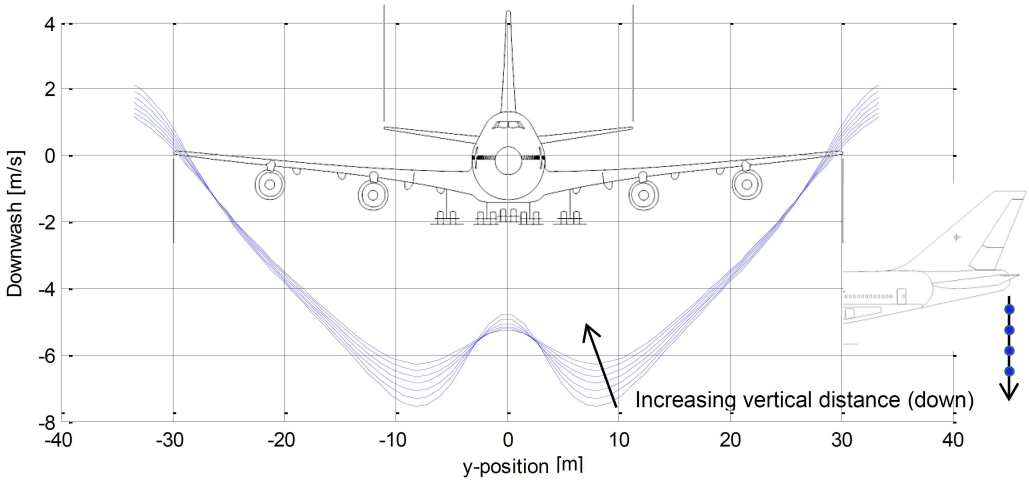


Figure B.4: *Downwash velocity of the tanker*

B.2 State Space Representation of the Dryden Turbulence Model

According to MIL-HDBK 1797, the spectrum of the vertical wind velocity is given by [US 97]:

$$\Phi_w(\omega) = \sigma_w^2 \frac{L_w}{\pi} \frac{1 + 3\left(L_w\omega/V\right)^2}{\left(1 + \left(L_w\omega/V\right)^2\right)^2} \tag{B.3}$$

with σ_w, L_w, and V being the altitude- and turbulence-dependent root mean square turbulence amplitude according to MIL-HDBK-1797, the scale length with $L_w = 1750ft/2$, and the aircraft velocity respectively. By spectral factorization, the transfer function G_w is obtained, which gives the vertical gust velocity for a white noise input:

$$G_w = \sigma_w \sqrt{\frac{L_w}{V}} \frac{1 + \sqrt{3}\frac{L_w}{V}s}{\left(1 + \frac{L_w}{V}s\right)^2} \tag{B.4}$$

Conversion to time domain gives the following equation:

$$\left(\frac{L_w}{V}\right)^2 \ddot{w} + \frac{2L_w}{V}\dot{w} + w = \sqrt{\frac{L_w}{V}}\eta + \sqrt{3\left(\frac{L_w}{V}\right)^3}\dot{\eta} \tag{B.5}$$

with η being the white noise input. To eliminate the time derivative of η, a substitution is made for $\dot{w}$:

$$w^* = \dot{w} - \sqrt{\frac{3L_w}{V}}\eta \rightarrow \dot{w^*} = \ddot{w} - \sqrt{\frac{3L_w}{V}}\dot{\eta} \tag{B.6}$$

Equation (B.5) can be rearranged and plugged into (B.6):

$$\ddot{w} = -\frac{2L_w}{V}\dot{w} - \frac{V^2}{L_w^2}w + \sqrt{\left(\frac{V}{L_w}\right)^3}\eta + \sqrt{\frac{3V}{L_w}}\dot{\eta} \tag{B.7}$$

$$\rightarrow \dot{w}^* = -\frac{2V}{L_w}\dot{w} - \frac{V^2}{L_w^2}w + \sqrt{\left(\frac{V}{L_w}\right)^3}\eta + \cancel{\sqrt{\frac{3V}{L_w}}\dot{\eta}} - \cancel{\sqrt{\frac{3V}{L_w}}\dot{\eta}} \tag{B.8}$$

Finally, by substituting $\dot{w}$ by a function of $\dot{w}^*$ and η, the state space model can be written as:

$$\begin{aligned}\begin{bmatrix}\dot{w}\\ \dot{w}^*\end{bmatrix} &= \begin{bmatrix}0 & 1\\ -\frac{V^2}{L_w^2} & -\frac{2V}{L_w}\end{bmatrix}\begin{bmatrix}w\\ w^*\end{bmatrix} + \begin{bmatrix}\sigma_w\sqrt{\frac{3V}{L_w}}\\ \sigma\left(1-2\sqrt{3}\right)\sqrt{\left(\frac{V}{L_w}\right)^2}\end{bmatrix}\eta\\ &= \mathbf{A}_{wind}\begin{bmatrix}w\\ w^*\end{bmatrix} + \mathbf{B}_{wind}\eta\end{aligned} \tag{B.9}$$

The same derivation can be done for the wind pitch rate component, which has the same structure as (B.3).

B.3 State Space Representation of a First Order Padé Approximation

The transfer function for a first order Padé approximation is given by (B.10), where τ denotes the time delay [Pad92].

$$G_{Pade} = \frac{1-\frac{\tau}{2}s}{1+\frac{\tau}{2}s} \tag{B.10}$$

When v_1 and v_2 denote the original and delayed signal respectively, the transfer function can be converted to time domain:

$$v_2 + \frac{\tau}{2}\dot{v}_2 = v_1 - \frac{\tau}{2}\dot{v}_1 \tag{B.11}$$

The derivative of v_1 is eliminated by substitution of v_2 by v_2^*, which is defined by

$$\begin{aligned}v_2^* &= \frac{\tau}{4}\left(v_2 + v_1\right) \rightarrow v_2 = \frac{4}{\tau}v_2^* - v_1\\ \dot{v}_2^* &= \frac{\tau}{4}\left(\dot{v}_2 + \dot{v}_1\right)\end{aligned} \tag{B.12}$$

Equation (B.11) can be rearranged and plugged into (B.12):

$$\dot{v}_2^* = \frac{\tau}{4}\left(\frac{2}{\tau}\left(v_1 - v_2\right) - \cancel{\dot{v}_1} + \cancel{\dot{v}_1}\right) \tag{B.13}$$

Finally, substituting v_2 by a function of v_2^* and v_1, the state space model can be written as

$$\begin{aligned}\dot{v}_2^* &= -\frac{2}{\tau}v_2^* + v_1\\ v_2 &= \frac{4}{\tau}v_2^* - v_1\end{aligned} \tag{B.14}$$

Scientific Publications by the Author

This bibliography includes all scientific publications, where the author of this thesis is either first author or significantly contributed to.

[HLG15] B. Heesbeen, D. Löbl, and J. Goreneweg. "Human Factors Aspects of Air-to-Air Refuelling of Civil Aircraft - a Human-in-the-Loop Study". In: *Applied Aerodynamics Conference 2014*. Red Hook, NY: Curran, 2015, pp. 479–487. ISBN: 978-1-5108-0269-8.

[LH13] D. Löbl and F. Holzapfel. "High Fidelity Simulation Model of an Aerial Refuelling Boom and Receptacle". In: *Deutscher Luft- und Raumfahrtkongress*. Stuttgart, Germany, 2013.

[LH14a] D. Löbl and F. Holzapfel. "Closed-Loop Simulation Analysis of Automated Control of Aircraft in Formation Flight". In: *Deutscher Luft- und Raumfahrtkongress*. Augsburg, Germany, 2014.

[LH14b] D. Löbl and F. Holzapfel. "Simulation Analysis of a Sensor Data Fusion for Close Formation Flight". In: *AIAA Guidance, Navigation, and Control Conference*. AIAA SciTech Forum. National Harbor, MD, USA: American Institute of Aeronautics and Astronautics, 2014.

[LH15] D. Löbl and F. Holzapfel. "Subset Simulation for Estimating Small Failure Probabilities of an Aerial System Subject to Atmospheric Turbulences". In: *AIAA Atmospheric Flight Mechanics Conference*. AIAA SciTech Forum. Kissimee, FL, USA: American Institute of Aeronautics and Astronautics, 2015.

[LH17] D. Löbl and F. Holzapfel. "Pseudo-Inverse Determination of Design Parameter Boundaries for Systems Subject to Stochastic Disturbances and Uncertainties". In: *Euro GNC 2017 - 4th CEAS Specialist Conference on Guidance, Navigation & Control*. 2017.

[LMH16a] D. Löbl, N. C. Mumm, and F. Holzapfel. "Model Based Online Monitoring of Uncertain Plants Subject to Stochastic Disturbances". In: *AIAA Atmospheric Flight Mechanics Conference*. AIAA AVIATION Forum. Washington, D.C., USA: American Institute of Aeronautics and Astronautics, 2016.

[LMH16b] D. Löbl, N. Mumm, and F. Holzapfel. "A Total Capability Approach for Development of Safety-Critical Functions". In: *13th International Conference on Probabilistic Safety Assessment and Management (PSAM13)*. Seoul, Korea, 2016.

[Löb+17] D. Löbl, M. Weiss, F. Holzapfel, and T. Y. Shima. "Cooperative Docking Guidance and Control with Application to Civil Autonomous Aerial Refueling". In: *IACAS 2017 - 57th Israel Annual Conference on Aerospace Sciences*. Tel Aviv and Haifa, Israel, 2017.

[Löb+18] D. Löbl, M. Weiss, F. Holzapfel, and T. Y. Shima. "Cooperative Docking Guidance and Control with Application to Civil Autonomous Aerial Refueling". In: *AIAA Guidance, Navigation, and Control Conference*. AIAA SciTech Forum. Kissimee, Florida, USA: American Institute of Aeronautics and Astronautics, 2018.

[MLH16] N. Mumm, D. Löbl, and F. Holzapfel. "Failure Probability Analysis of an Automatic Landing System for a General Aviation Aircraft Using "Subset Simulation"". In: *13th International Conference on Probabilistic Safety Assessment and Management (PSAM13)*. Seoul, Korea, 2016.

[Tsu+16] A. Tsukerman, D. Löbl, T. Y. Shima, M. Weiss, and F. Holzapfel. "Integrated guidance-autopilot approach for civil autonomous aerial refueling". In: *IACAS 2016 - 56th Israel Annual Conference on Aerospace Sciences*. Tel Aviv and Haifa, Israel, 2016.

[Tsu+17] A. Tsukerman, D. Löbl, M. Weiss, T. Y. Shima, and F. Holzapfel. "Trajectory Shaping Autopilot-Guidance Design for Civil Autonomous Aerial Refueling". In: *AIAA Guidance, Navigation, and Control Conference*. AIAA SciTech Forum. Grapevine, TX, USA: American Institute of Aeronautics and Astronautics, 2017.

[Wan+15] J. Wang, D. Löbl, T. Raffler, and F. Holzapfel. "Kinematic Modeling and Control Design for an Aerial Refueling Task". In: *Applied Aerodynamics Conference 2014*. Red Hook, NY: Curran, 2015, pp. 488–501. ISBN: 978-1-5108-0269-8.

Bibliography

Citations are referred to and sorted by the initial letters of the surname of the authors, followed by the year of publication. For the case with more than three authors, only the first three letters of the surname of the first author are given, followed by a "+" to indicate additional authors.

[AB01] S.-K. Au and J. L. Beck. "Estimation of small failure probabilities in high dimensions by subset simulation". In: *Probabilistic Engineering Mechanics* Vol. 16. No. 4 (2001), pp. 263–277. ISSN: 0266-8920.

[AB03] S. K. Au and J. L. Beck. "Important sampling in high dimensions". In: *Structural Safety* Vol. 25. No. 2 (2003), pp. 139–163. ISSN: 01674730.

[AlA+09] A. Al-Ashaab, S. Howell, K. Usowicz, P. Hernando Anta, and A. Gorka. "Set-Based Concurrent Engineering Model for Automotive Electronic/Software Systems Development". In: *19th CIRP Design Conference – Competitive Design*. 2009, pp. 464.

[AP16] S.-K. Au and E. Patelli. "Rare event simulation in finite-infinite dimensional space". In: *Reliability Engineering & System Safety* Vol. 148 (2016), pp. 67–77. ISSN: 09518320.

[ASA17] Axel Busboom, Simone Schuler, and Alexander Walsch. "formalSpec - Semi-Automatic Formalization of System Requirements for Formal Verification". In: *ARCH16. 3rd International Workshop on Applied Verification for Continuous and Hybrid Systems*. Ed. by Goran Frehse and Matthias Althoff. Vol. 43. EPiC Series in Computing. EasyChair, 2017, pp. 106–114.

[AW14] S.-K. Au and Y. Wang. *Engineering risk assessment with subset simulation*. 2014. ISBN: 9781118398043.

[Bak+14] S. Bak, T. T. Johnson, M. Caccamo, and L. Sha. "Real-Time Reachability for Verified Simplex Design". In: *IEEE Real-Time Systems Symposium (RTSS)*. 2014, pp. 138–148.

[BAL11] R. Brockhaus, W. Alles, and R. Luckner. *Flugregelung*. Dordrecht: Springer, 2011. ISBN: 978-3-642-01443-7.

[Ban+06] Z. A. Bangash, R. P. Sanchez, A. Ahmed, and M. J. Khan. "Aerodynamics of Formation Flight". In: *Journal of Aircraft* Vol. 43. No. 4 (2006), pp. 907–912. ISSN: 0021-8669.

[Bar94] B. R. Barmish. *New tools for robustness of linear systems*. New York: Macmillan, Toronto : Maxwell Macmillan Canada, and New York : Maxwell Macmillan International, 1994. ISBN: 978-0023060557.

[Bat+05] A. Bateman, C. Elks, D. Ward, and J. Schierman. "New Verification and Validation Methods for Guidance/Control of Advanced Autonomous Systems". In: *Infotech@Aerospace*. Arlington, VA, USA: American Institute of Aeronautics and Astronautics, 2005.

[BB94] A. Bonarini and G. Bontempi. "A qualitative simulation approach for fuzzy dynamical models". In: *ACM Transactions on Modeling and Computer Simulation* Vol. 4 (1994), pp. 285–313.

[BDF98] P. Bickel, P. Diggle, and S. Fienberg. *Random and Quasi-Random Point Sets*. Springer New York, 1998. ISBN: 978-1-4612-1702-2.

[Bec13] E. F. Beckenbach, ed. *Modern mathematics for the engineer*. Dover books on engineering. Mineola New York: Dover Publications Inc, 2013. ISBN: 978-0486497464.

[Beh00] E. Behrends. *Introduction to Markov chains: With special emphasis on rapid mixing*. Advanced lectures in mathematics. Braunschweig/Wiesbaden: Vieweg, 2000. ISBN: 978-3-528-06986-5.

[Beh13] E. Behrends. *Elementare Stochastik: Ein Lernbuch - von Studierenden mitentwickelt*. Lehrbuch. Wiesbaden: Springer Spektrum, 2013. ISBN: 978-3-8348-1939-0.

[BN87] K. Behnen and G. Neuhaus. *Grundkurs Stochastik: Eine integrierte Einführung in Wahrscheinlichkeitstheorie und mathematische Statistik ; mit 253 Aufgaben und zahlreichen Beispielen*. 2., durchges. Aufl. Teubner-Studienbücher. Mathematik. Stuttgart: Teubner, 1987. ISBN: 978-3-519-12069-8.

[Bro06] I. N. Bronštejn. *Taschenbuch der Mathematik*. Nachdruck der 6., vollständig überarbeiteten und ergänzten Aufl. Frankfurt am Main: Harri Deutsch, 2006. ISBN: 3-8171-2016-8.

[BS07] H.-G. Beyer and B. Sendhoff. "Robust optimization – A comprehensive survey". In: *Computer Methods in Applied Mechanics and Engineering* Vol. 196. No. 33-34 (2007), pp. 3190–3218. ISSN: 00457825.

[But08] R. W. Butler. *A Primer on Architectural Level Fault Tolerance: NASA/TM-2008-215108*. Ed. by NASA Langley Research Center. Hampton, Virginia, 2008.

[BV05] M. A. Bennington and K. D. Visser. "Aerial Refueling Implications for Commercial Aviation". In: *Journal of Aircraft* Vol. 42. No. 2 (2005), pp. 366–375. ISSN: 0021-8669.

[CD06] G. Calafiore and F. Dabbene, eds. *Probabilistic and Randomized Methods for Design under Uncertainty*. London: Springer-Verlag, 2006. ISBN: 1-84628-094-X.

[CDT11] G. C. Calafiore, F. Dabbene, and R. Tempo. "Research on probabilistic methods for control system design". In: *Automatica* Vol. 47. No. 7 (2011), pp. 1279–1293. ISSN: 0005-1098.

[CH13a] G. Chowdhary and J. P. How. "Concurrent Learning Adaptive Control in Presence of Uncertain Control Allocation Matrix". In: *AIAA Guidance, Navigation, and Control (GNC) Conference*. Boston, MA, USA: American Institute of Aeronautics and Astronautics, 2013.

[CH13b] G. Christoph and H. Hackel. *Starthilfe Stochastik: Studium*. Vieweg+Teubner Verlag, 2013. ISBN: 978-3-322-84799-7.

[Che+13] B. Chen, Y. Zhu, J. Hu, and J. C. Príncipe. *System parameter identification: Information criteria and algorithms*. First edition. London UK and Waltham MA USA: Elsevier, 2013. ISBN: 978-0-12-404574-3.

[DBH13] A. Dogan, W. Blake, and C. Haag. "Bow Wave Effect in Aerial Refueling: Computational Analysis and Modeling". In: *Journal of Aircraft* Vol. 50. No. 6 (2013), pp. 1856–1868. ISSN: 0021-8669.

[DC00] X. Du and W. Chen. "Towards a Better Understanding of Modeling Feasibility Robustness in Engineering Design". In: *Journal of Mechanical Design* Vol. 122. No. 4 (2000), p. 385. ISSN: 10500472.

[Dep16] Department of Transportation, Federal Aviation Administration. *Revision of Airworthiness Standards for Normal, Utility, Acrobatic, and Commuter Category Airplanes: RIN 2120–AK65*. 12/12/2016.

[Dev86] L. Devroye. *Non-uniform random variate generation*. New York: Springer, 1986. ISBN: 3-540-96305-7.

[DK89] D. Dvorak and B. Kuipers. "Model-Based Monitoring of Dynamic Systems". In: *11th International Joint Conference on Artificial Intelligence - Volume 2*. San Francisco,CA, USA, 1989.

[DLB08] A. Dogan, T. A. Lewis, and W. Blake. "Flight Data Analysis and Simulation of Wind Effects During Aerial Refueling". In: *Journal of Aircraft* Vol. 45. No. 6 (2008), pp. 2036–2048. ISSN: 0021-8669.

[DM98] T. Dang and O. Maler. "Reachability analysis via face lifting". In: *Hybrid Systems: Computation and Control: First International Workshop, HSCC'98 Berkeley, California, USA, April 13 – 15, 1998 Proceedings*. Ed. by T. A. Henzinger and S. Sastry. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 96–109. ISBN: 978-3-540-69754-1.

[Dre17] L. Drees. *Predictive Analysis: Quantifying Operational Airline Risks*. München: Dr. Hut, 2017. ISBN: 978-3-8439-2987-5.

[DS12] W. L. Dunn and J. K. Shultis. *Exploring Monte Carlo methods*. Amsterdam and Boston: Elsevier/Academic Press, 2012. ISBN: 978-0-444-51575-9.

[DVB05] A. Dogan, S. Venkataramanan, and W. Blake. "Modeling of Aerodynamic Coupling Between Aircraft in Close Proximity". In: *Journal of Aircraft* Vol. 42. No. 4 (2005), pp. 941–955. ISSN: 0021-8669.

[Ers+17] S. Erschen, F. Duddeck, M. Gerdts, and M. Zimmermann. "On the optimal decomposition of high-dimensional solution spaces of complex systems". In: *ASCE-ASME J. Risk and Uncert. in Engrg. Sys., Part B: Mech. Engrg.* (2017). ISSN: 2332-9017.

[Ers18] S. Erschen. "Optimal Decomposition of High-Dimensional Solution Spaces for Chassis Design". PhD thesis (submitted 14.11.2017). Munich: Technical University of Munich, 2018.

[ETP90] O. W. Eshbach, B. D. Tapley, and T. R. Poston. *Eshbach's handbook of engineering fundamentals*. 4th ed. / editor, Byron D. Tapley, managing editor, Thurman R. Poston. New York: Wiley, 1990. ISBN: 0471890847.

[Eur07] European Aviation Safety Agency. *Decision of the Management Board amending Decision of the Management Board No 07-2004 concerning the general principles related to the certification procedures to be applied by the Agency for the issuing of certificates for products, parts and applliances*. 20/01/2007.

[Eur13] European Aviation Safety Agency. *Technical Implementation Procedures for Airworthiness and Environmental Certification*. April 23, 2013.

[Eur17a] European Aviation Safety Agency. *Amending Certification Specifications for normal, utility, aerobatic and commuter category aeroplanes: ED 2017/013/R*. 29/3/2017.

[Eur17b] European Aviation Safety Agency. *Certification Specifications for Large Aeroplanes: CS-25 Amendment 19*. 5/12/2017.

[Eur17c] European Aviation Safety Agency. *NPA 2017-05 (A) Introduction of a regulatory framework for the operation of drones: Unmanned aircraft system operations in the open and specific category*. 4/05/2017.

[Eur17d] European Aviation Safety Agency. *Certification Specifications for Normal-Category Aeroplanes: CS-23 Amendment 5*. 29/03/2017.

[Fed08] Federal Aviation Administration. *The Federal Aviation Administration: A Historical Perspective, 1903-2008*. Ed. by Federal Aviation Administration. Washington, DC, 2008.

[Fed64] Federal Aviation Administration. *14 CFR 25 - Airworthiness Standards: Transport Category Airplanes*. 12/24/1964.

[Fed67] Federal Aviation Administration. *14 CFR 23 - Airworthiness Standards: Normal, Utility, Acrobatic, and Commuter Category Airplanes*. 3/30/1967.

[Fen+14] J. Fender, L. Graff, H. Harbrecht, and M. Zimmermann. "Identifying Key Parameters for Design Improvement in High-Dimensional Systems With Uncertainty". In: *Journal of Mechanical Design* Vol. 136. No. 4 (2014), p. 041007. ISSN: 10500472.

[FYN17] M. L. Fravolini, T. Yucelen, and M. Napolitano. "Probabilistic Analysis and Verification Framework for Adaptive Flight Control". In: *Journal of Guidance, Control, and Dynamics* Vol. 40. No. 3 (2017), pp. 537–547. ISSN: 0731-5090.

[Gen03] J. E. Gentle. *Random number generation and Monte Carlo methods*. 2nd ed. Statistics and computing. New York and London: Springer, 2003. ISBN: 978-0-387-00178-4.

[Gey05] C. J. Geyer. *Markov Chain Monte Carlo Lecture Notes*. University of Minnesota, 11/21/2005.

[GHZ16] L. Graff, H. Harbrecht, and M. Zimmermann. "On the computation of solution spaces in high dimensions". In: *Structural and Multidisciplinary Optimization* Vol. 54. No. 4 (2016), pp. 811–829. ISSN: 1615-1488.

[GM01] P. J. Green and A. Mira. "Delayed Rejection in Reversible Jump Metropolis-Hastings". In: *Biometrika* Vol. 88. No. 4 (2001), pp. 1035–1053. ISSN: 00063444.

[Goo16] A. Goodloe. *Challenges in High-Assurance Runtime Verification*. Vol. 9952. Cham: Springer International Publishing, 2016. ISBN: 978-3-319-47165-5.

[GPC10] A. E. Goodloe, L. Pike, and L. R. Center. *Monitoring Distributed Real-time Systems: A Survey and Future Directions*. NASA contractor report. National Aeronautics and Space Administration, Langley Research Center, 2010.

[Gra13] L. Graff. "Stochastic algorithm for the identification of solution spaces in high-dimensional design spaces". PhD thesis. Basel, Switzerland: University of Basel, Faculty of Science, 2013.

[Gra15] G. Gratton. *Initial airworthiness: Determining the acceptability of new airborne systems*. Switzerland: Springer, 2015. ISBN: 978-3-319-11409-5.

[Gro+17] K. H. Gross, M. A. Clark, J. A. Hoffman, E. D. Swenson, and A. W. Fifarek. "Run-Time Assurance and Formal Methods Analysis Nonlinear System Applied to Nonlinear System Control". In: *Journal of Aerospace Information Systems* Vol. 14. No. 4 (2017), pp. 232–246. ISSN: 2327-3097.

[GT13] C. Graham and D. Talay. *Stochastic simulation and Monte Carlo methods: Mathematical foundations of stochastic simulation / Carl Graham, Denis Talay*. Vol. 68. Stochastic modelling and applied probability, 0172-4568. 2013. ISBN: 978-3-642-39362-4.

[Has70] W. K. Hastings. "Monte Carlo Sampling Methods Using Markov Chains and Their Applications". In: *Biometrika* Vol. 57. No. 1 (1970), p. 97. ISSN: 00063444.

[HCK92] W. Hamscher, L. Console, and J. de Kleer. *Readings in model-based diagnosis*. San Mateo, CA: Morgan Kaufmann Publishers, 1992. ISBN: 1-55860-249-6.

[Hep+12] M. Hepperle, G. La Rocca, L. Manfriani, and R. K. Nangia. *Initial Design of Cruiser Aircraft and Feeder Aircraft: Deliverable 4.1 of WP4 of the EU FP7 RECREATE project*. 2012.

[Hes05] R. A. Hess. "From Health and Usage Monitoring to Integrated Fleet Management - Evolving Directions for Rotorcraft". In: *IEEE Aerospace Conference*. Big Sky, MT, USA, 2005, pp. 1–6.

[HJ13] R. A. Horn and C. R. Johnson. *Matrix analysis*. 2nd ed. Cambridge: Cambridge University Press, 2013. ISBN: 978-0-521-83940-2.

[HLG15] B. Heesbeen, D. Löbl, and J. Goreneweg. "Human Factors Aspects of Air-to-Air Refuelling of Civil Aircraft - a Human-in-the-Loop Study". In: *Applied Aerodynamics Conference 2014*. Red Hook, NY: Curran, 2015, pp. 479–487. ISBN: 978-1-5108-0269-8.

[HN70] C. R. Hanke and D. R. Nordwall. *The Simulation of a Jumbo Jet Transport Aircraft. Volume 2: Modeling Data*. Wichita, KS, United States, 1970.

[Hol13] M. Holicky. *Introduction to probability and statistics for engineers*. New York: Springer, 2013. ISBN: 978-3-642-38299-4.

[IBM17] IBM. *Rational DOORS*. 2017. URL: `http://www-03.ibm.com/software/products/en/ratidoor` (visited on 9-4-2017).

[IBM56] IBM. *Fortran - Programmer's Reference Manual*. New York, NY, 1956.

[Int06] International Civil Aviation Organization. *Convention on International Civil Aviation*. 9th ed. Montreal, 2006. ISBN: 92-9194-754-7.

[Int10] International Civil Aviation Organization. *Annex 8: Airworthiness of aircraft*. 11th ed. International standards and recommended practices. Montréal, Québec, 2010. ISBN: 978-92-9231-518-4.

[Int13] International Civil Aviation Organization. *Performance-based Navigation (PBN) Manual*. 4th edition. 2013.

[Jac08] S. Jacklin. "Closing the Certification Gaps in Adaptive Flight Control Software". In: *AIAA Guidance, Navigation and Control Conference and Exhibit*. Honolulu, HI, USA, 2008.

[Kal60] R. E. Kalman. "A New Approach to Linear Filtering and Prediction Problems". In: *Transactions of the ASME–Journal of Basic Engineering* Vol. 82. No. Series D (1960), pp. 35–45.

[Kar+93] G. Karsai, S. Padalkar, H. Franke, and J. Sztipanovits. "Model-based programming tools for integrated monitoring, simulation, diagnosis and control". In: *9th Computing in Aerospace Conference*. 1993.

[KC06] D. Kurowicka and R. Cooke. *Uncertainty analysis with high dimensional dependence modelling*. Wiley series in probability and statistics. Chichester: John Wiley, 2006. ISBN: 978-0-470-86306-0.

[Kel03] C. T. Kelley. *Solving nonlinear equations with Newton's method*. Fundamentals of algorithms. Philadelphia, Pa.: Society for Industrial and Applied Mathematics, 2003. ISBN: 978-0-89871-889-8.

[Kle+13] J. E. Kless, M. J. Aftosmis, S. A. Ning, and M. Nemec. "Inviscid Analysis of Extended-Formation Flight". In: *AIAA Journal* Vol. 51. No. 7 (2013), pp. 1703–1715. ISSN: 0001-1452.

[Kou16] P. S. Koutsourelakis. "Variational Bayesian strategies for high-dimensional, stochastic design problems". In: *Journal of Computational Physics* Vol. 308 (2016), pp. 124–152. ISSN: 00219991.

[LCK10] G. Larchev, S. Campbell, and J. Kaneshige. "Projection Operator: A Step Toward Certification of Adaptive Controllers". In: *AIAA Infotech@Aerospace*. Atlanta, GA, USA, 2010.

[Leu08] M. Leucker. *Runtime verification: 8th international workshop, RV 2008, Budapest, Hungary, March 30, 2008. selected papers / Martin Leucker (ed.)* 1st ed. Vol. 5289. Lecture notes in computer science, 0302-9743. Berlin: Springer, 2008. ISBN: 978-3-540-89246-5.

[LH13] D. Löbl and F. Holzapfel. "High Fidelity Simulation Model of an Aerial Refuelling Boom and Receptacle". In: *Deutscher Luft- und Raumfahrtkongress*. Stuttgart, Germany, 2013.

[LH14a] D. Löbl and F. Holzapfel. "Closed-Loop Simulation Analysis of Automated Control of Aircraft in Formation Flight". In: *Deutscher Luft- und Raumfahrtkongress*. Augsburg, Germany, 2014.

[LH14b] D. Löbl and F. Holzapfel. "Simulation Analysis of a Sensor Data Fusion for Close Formation Flight". In: *AIAA Guidance, Navigation, and Control Conference*. AIAA SciTech Forum. National Harbor, MD, USA: American Institute of Aeronautics and Astronautics, 2014.

[LH15] D. Löbl and F. Holzapfel. "Subset Simulation for Estimating Small Failure Probabilities of an Aerial System Subject to Atmospheric Turbulences". In: *AIAA Atmospheric Flight Mechanics Conference*. AIAA SciTech Forum. Kissimee, FL, USA: American Institute of Aeronautics and Astronautics, 2015.

[LH17] D. Löbl and F. Holzapfel. "Pseudo-Inverse Determination of Design Parameter Boundaries for Systems Subject to Stochastic Disturbances and Uncertainties". In: *Euro GNC 2017 - 4th CEAS Specialist Conference on Guidance, Navigation & Control*. 2017.

[LMH16a] D. Löbl, N. C. Mumm, and F. Holzapfel. "Model Based Online Monitoring of Uncertain Plants Subject to Stochastic Disturbances". In: *AIAA Atmospheric Flight Mechanics Conference*. AIAA AVIATION Forum. Washington, D.C., USA: American Institute of Aeronautics and Astronautics, 2016.

[LMH16b] D. Löbl, N. Mumm, and F. Holzapfel. "A Total Capability Approach for Development of Safety-Critical Functions". In: *13th International Conference on Probabilistic Safety Assessment and Management (PSAM13)*. Seoul, Korea, 2016.

[Löb+12] D. Löbl, F. Holzapfel, F. Berefelt, M. Hagström, and J. Robinson. *Initial Automatic Flight Control of Cruiser Aircraft and Feeder Aircraft: Deliverable 5.1 of WP5 of the EU FP7 RECREATE project*. 2012.

[Löb+17] D. Löbl, M. Weiss, F. Holzapfel, and T. Y. Shima. "Cooperative Docking Guidance and Control with Application to Civil Autonomous Aerial Refueling". In: *IACAS 2017 - 57th Israel Annual Conference on Aerospace Sciences*. Tel Aviv and Haifa, Israel, 2017.

[Lop15] M. L. Loper. *Modeling and simulation in the systems engineering life cycle: Core concepts and accompanying lectures / Margaret L. Loper, editor*. Simulation foundations, methods and applications, 2195-2817. London: Springer, 2015. ISBN: 978-1-4471-5634-5.

[LS09] M. Leucker and C. Schallhart. "A brief account of runtime verification". In: *The Journal of Logic and Algebraic Programming* Vol. 78. No. 5 (2009), pp. 293–303. ISSN: 15678326.

[Mac04] U. Mackenroth. *Robust Control Systems: Theory and Case Studies*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004. ISBN: 978-3-662-09775-5.

[MCJ12] M. Mühlegg, G. Chowdhary, and E. Johnson. "Concurrent Learning Adaptive Control of Linear Systems with Noisy Measurements". In: *AIAA Guidance, Navigation, and Control Conference*. Minneapolis, MN, USA: American Institute of Aeronautics and Astronautics, 2012.

[McR+15] R. McRoberts, J. M. Early, F. Morscheck, M. Price, and B. Korn. "Tanker Mission Implication on a Civil Aerial Refuelling Transport System's Benefit Evaluation". In: *Journal of Aircraft* Vol. 52. No. 1 (2015), pp. 320–328. ISSN: 0021-8669.

[Met+53] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. "Equation of State Calculations by Fast Computing Machines". In: *The Journal of Chemical Physics* Vol. 21. No. 6 (1953), p. 1087. ISSN: 00219606.

[Müh18] M. Mühlegg. "Run-Time Monitoring of Model Reference Adaptive Controllers". PhD thesis. Munich: Technical University of Munich, Forthcoming 2018.

[Mum18] N. Mumm. "Total Capability Approach - Development of a Certifiable Automatic Landing System for a General Aviation Aircraft Using Stochastic Methods". PhD thesis. Munich: Technical University of Munich, Forthcoming 2018.

[Nan06] R. K. Nangia. "Operations and aircraft design towards greener civil aviation using air-to-air refuelling". In: *The Aeronautical Journal* Vol. 110. No. 1113 (2006), pp. 705–721. ISSN: 0001-9240.

[NAT00] NATO. *Structural aspects of flexible aircraft control*. 2000. ISBN: 92-837-0014-7.

[Nee93] T. Needham. "A Visual Explanation of Jensen's Inequality". In: *The American Mathematical Monthly* Vol. 100. No. 8 (1993), p. 768. ISSN: 00029890.

[ODB15] W. Okolo, A. Dogan, and W. Blake. "Effect of Trail Aircraft Trim on Optimum Location in Formation Flight". In: *Journal of Aircraft* Vol. 52. No. 4 (2015), pp. 1201–1213. ISSN: 0021-8669.

[OMG07] OMG. *Systems Modeling Language - OMG SysML(TM)*. 9/1/2007. URL: `http://www.omg.org/spec/SysML/1.0/PDF` (visited on).

[PA15] E. Patelli and S.-K. Au. "Efficient Monte Carlo Algorithm For Rare Failure Event Simulation". In: *12th International Conference on Applications of Statistics and Probability in Civil Engineering, ICASP12*. Vancouver, Canada, 2015.

[Pad92] H. Padé. "Sur la représentation approchée d'une fonction par des fractions rationnelles". In: *Annales scientifiques de l'École normale supérieure* Vol. 9 (1892), pp. 3–93. ISSN: 0012-9593.

[Pap+15] I. Papaioannou, W. Betz, K. Zwirglmaier, and D. Straub. "MCMC algorithms for Subset Simulation". In: *Probabilistic Engineering Mechanics* Vol. 41 (2015), pp. 89–103. ISSN: 0266-8920.

[PCN11] S. D. Pinder, T. G. Crowe, and P. N. Nikiforuk. "Experimental Testing of a Parametric-Model-Based Takeoff Performance Monitoring Strategy". In: *Journal of Aircraft* Vol. 48. No. 1 (2011), pp. 56–65. ISSN: 0021-8669.

[Pel09] M. F. Pellissetti. "Parallel processing in structural reliability". In: *Structural Engineering and Mechanics* Vol. 32. No. 1 (2009), pp. 95–126.

[PH12] A. Post and J. Hoenicke. "Formalization and Analysis of Real-Time Requirements: A Feasibility Study at BOSCH". In: *Verified Software: Theories, Tools, Experiments: 4th International Conference, VSTTE 2012, Philadelphia, PA, USA, January 28-29, 2012. Proceedings*. Ed. by R. Joshi, P. Müller, and A. Podelski. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 225–240. ISBN: 978-3-642-27705-4.

[PS00] W. F. Phillips and D. O. Snyder. "Modern Adaptation of Prandtl's Classic Lifting-Line Theory". In: *Journal of Aircraft* Vol. 37. No. 4 (2000), pp. 662–670. ISSN: 0021-8669.

[Rey09] D. A. Reynolds. "Gaussian Mixture Models". In: *Encyclopedia of Biometrics*. 2009.

[RGG97] G. O. Roberts, A. Gelman, and W. R. Gilks. "Weak convergence and optimal scaling of random walk Metropolis algorithms". In: *The Annals of Applied Probability* Vol. 7. No. 1 (1997), pp. 110–120.

[Ris96] H. Risken. *The Fokker-Planck Equation: Methods of Solution and Applications*. Second Edition. Vol. 18. Springer Series in Synergetics. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996. ISBN: 978-3-642-61544-3.

[RT09] G. Rubino and B. Tuffin. *Rare event simulation using Monte Carlo methods*. Chichester: Wiley, 2009. ISBN: 9780470772690.

[RTC00] RTCA. *DO-254 Design Assurance Guidance for Airborne Electronic Hardware*. Washington, DC, 4/19/2000.

[RTC05] RTCA. *DO-297 Integrated Modular Avionics Development Guidance and Certification Consderations*. Washington, DC, 11/08/2005.

[RTC11a] RTCA. *DO-178C Software Considerations in Airborne Systems and Equipment Certification*. Washington, DC, 12/13/2011.

[RTC11b] RTCA. *DO-331 Model-Based Development and Verification Supplement to Do-178C and DO-278A*. Washington, DC, 12/13/2011.

[Rus08] J. Rushby. "How Do We Certify For The Unexpected?" In: *AIAA Guidance, Navigation and Control Conference and Exhibit*. 2008.

[RW02] B. Rinner and U. Weiss. "The model-based online monitoring system MOSES". In: *OeGAI Journal* Vol. 21(3):5-12 (2002).

[RW04] B. Rinner and U. Weiss. "Online Monitoring by Dynamically Refining Imprecise Models". In: *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)* Vol. 34. No. 4 (2004), pp. 1811–1822. ISSN: 1083-4419.

[SAE07] SAE. *AS 94900 Aerospace - Flight Control Systems - General Specification for Design, Installation and Test of Piloted Military Aircraft*. 6/07/2007.

[SAE10] SAE. *SAE ARP 4754A Guidelines for Development of Civil Aircraft and Systems*. 12/21/2010.

[SAE96] SAE. *SAE ARP 4761 Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment*. 12/01/1996.

[SC08] J. L. Speyer and W. H. Chung. *Stochastic processes, estimation, and control*. 1st ed. Vol. 17. Advances in design and control. Philadelphia: Society for Industrial and Applied Mathematics, 2008. ISBN: 978-0-89871-655-9.

[SES16] SESAR. *European Drones Outlook Study: Unlocking the value for Europe*. November 2016.

[SG14] S. Sankararaman and K. Goebel. "An Uncertainty Quantification Framework for Prognostics and Condition-Based Monitoring". In: *16th AIAA Non-Deterministic Approaches Conference*. AIAA SciTech Forum. National Harbor, MD, USA: American Institute of Aeronautics and Astronautics, 2014.

[She68] D. Shepard. "A two-dimensional interpolation function for irregularly-spaced data". In: *the 1968 23rd ACM national conference*. Ed. by R. B. Blue and A. M. Rosenberg. 1968, pp. 517–524.

[Sie17] Siemens. *Polarion (R) Requirements (TM)*. 2017. URL: `https://polarion.plm.automation.siemens.com/products/polarion-requirements` (visited on 9-4-2017).

[SS99] D. Seto and L. Sha. *A Case Study on Analytical Analysis of the Inverted Pendulum Real-Time Control System*. Ed. by Software Engineering Institute, Carnegie Mellon University. Pittsburgh, PA, 1999.

[Ste10] H.-J. Steiner. *Preliminary Design Tool for Propeller-Wing Aerodynamics, Part II: Theory*. Munich, 2010.

[Ste11] H.-J. Steiner. *Preliminary Design Tool for Propeller-Wing Aerodynamics, part I: Implementation and Reference Manual*. Munich, 2011.

[Ste12] M. O. Steinhauser. *Computer simulation in physics and engineering*. Berlin: De Gruyter, 2012. ISBN: 978-3-11-025606-2.

[SWL99] D. Sobek, A. C. Ward, and J. Liker. *Toyota's Principles of Set-Based Concurrent Engineering*. Ed. by ABI/INFORM Global. 1999.

[TCD13] R. Tempo, G. Calafiore, and F. Dabbene. *Randomized algorithms for analysis and control of uncertain systems: With applications / Roberto Tempo, Giuseppe Calafiore, Fabrizio Dabbene*. 2nd ed. Communications and control engineering. London: Springer, 2013. ISBN: 978-1-4471-4609-4.

[US 97] U.S. Military Handbook. *MIL-HDBK-1797 - Flying Qualities of Piloted Aircraft*. 1997.

[van15] T. van Birgelen. "Air-to-Air Refuelling of Civil Air Transport Aircraft - a Certification Approach". In: *Applied Aerodynamics Conference 2014*. Red Hook, NY: Curran, 2015. ISBN: 978-1-5108-0269-8.

[Wan+14] C. Wang, L. Drees, N. Gissibl, L. Höhndorf, J. Sembiring, and F. Holzapfel. "Quantification of Incident Probabilities Using Physical and Statistical Approaches". In: *6th International Conference on Research in Air Transportation*. 2014.

[Wan+15] J. Wang, D. Löbl, T. Raffler, and F. Holzapfel. "Kinematic Modeling and Control Design for an Aerial Refueling Task". In: *Applied Aerodynamics Conference 2014*. Red Hook, NY: Curran, 2015, pp. 488–501. ISBN: 978-1-5108-0269-8.

[Wes15] T. Westermann. *Mathematik für Ingenieure*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015. ISBN: 978-3-642-54289-3.

[Wil06] William Scheck. *Lawrence Sperry: Autopilot Inventor and Aviation Innovator*. 2006. URL: `http://www.historynet.com/lawrence-sperry-autopilot-inventor-and-aviation-innovator.htm` (visited on 9/30/2017).

[WL13] X. Wang and S. Liu. "Guided Requirements Clarification for Automatic Formalization". In: *2013 14th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*. 2013, pp. 348–355.

[Wym93] A. W. Wymore. *Model-based systems engineering: An introduction to the mathematical theory of discrete systems and to the tricotyledon theory of system design / A. Wayne Wymore*. Systems engineering series. Boca Raton and London: CRC Press, 1993. ISBN: 0-8493-8012-X.

[Zaj15] S. Zajac. *European Union FP7 - RECREATE Project Final Report*. Amsterdam, Netherlands, 2015.

[ZO99] Y.-G. Zhao and T. Ono. "A general procedure for first/second-order reliabilitymethod (FORM/SORM)". In: *Structural Safety* Vol. 21. No. 2 (1999), pp. 95–112. ISSN: 01674730.

[Zue+12] K. M. Zuev, J. L. Beck, S.-K. Au, and L. S. Katafygiotis. "Bayesian post-processor and other enhancements of Subset Simulation for estimating failure probabilities in high dimensions". In: *Computers & Structures* Vol. 92-93 (2012), pp. 283–296. ISSN: 00457949.