Prognosis based decision support for improved system availability

Eine Prognose basierte Entscheidungsunterstützung zur Erhöhung der Systemverfügbarkeit

Zur Erlangung des akademischen Grades Doktor-Ingenieur (Dr.-Ing.)
Genehmigte Dissertation von Henrik Heier aus Frankfurt am Main
Tag der Einreichung: 29.06.2023, Tag der Prüfung: 05.12.2023

1. Gutachten: Prof. Dr.-Ing. Uwe Klingauf
2. Gutachten: Prof. Dr.-Ing. Tobias Melz
Darmstadt – D 17

TECHNISCHE UNIVERSITÄT DARMSTADT

Mechanical Engineering

Institute of Flight Systems and Automatic Control

Prognosis based decision support for improved system availability
Eine Prognose basierte Entscheidungsunterstützung zur Erhöhung der Systemverfügbarkeit

Genehmigte Dissertation von Henrik Heier

1. Gutachten: Prof. Dr.-Ing. Uwe Klingauf
2. Gutachten: Prof. Dr.-Ing. Tobias Melz

Tag der Einreichung: 29.06.2023
Tag der Prüfung: 05.12.2023

Darmstadt – D 17

Bitte zitieren Sie dieses Dokument als:
URN: urn:nbn:de:tuda-tuprints-266612
URL: http://tuprints.ulb.tu-darmstadt.de/26661

Dieses Dokument wird bereitgestellt von tuprints,
E-Publishing-Service der TU Darmstadt
http://tuprints.ulb.tu-darmstadt.de
tuprints@ulb.tu-darmstadt.de

Erklärungen laut Promotionsordnung

§8 Abs. 1 lit. c PromO

Ich versichere hiermit, dass die elektronische Version meiner Dissertation mit der schriftlichen Version übereinstimmt.

§8 Abs. 1 lit. d PromO

Ich versichere hiermit, dass zu einem vorherigen Zeitpunkt noch keine Promotion versucht wurde. In diesem Fall sind nähere Angaben über Zeitpunkt, Hochschule, Dissertationsthema und Ergebnis dieses Versuchs mitzuteilen.

§9 Abs. 1 PromO

Ich versichere hiermit, dass die vorliegende Dissertation selbstständig und nur unter Verwendung der angegebenen Quellen verfasst wurde.

§9 Abs. 2 PromO

Die Arbeit hat bisher noch nicht zu Prüfungszwecken gedient.

Darmstadt, 29.06.2023

H. Heier

Danksagung

Die vorliegende Dissertation habe ich während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Flugsysteme und Regelungstechnik an der Technischen Universität Darmstadt verfasst. Mein herzlicher Dank gilt daher zuallererst Herrn *Prof. Dr.-Ing. Uwe Klingauf*, der mir als Institutsleiter und Doktorvater meine Promotion in einem bestmöglichen Forschungsumfeld ermöglicht hat. Die fachliche Unterstützung, das entgegengebrachte Vertrauen sowie die Möglichkeit, mich vielseitig am Institutsleben einbringen und so persönliche Erfahrungen sammeln zu können, bildeten zu jeder Zeit sehr gute Voraussetzungen für mein Vorhaben. Darüber hinaus danke ich auch Herrn *Prof. Dr.-Ing. Tobias Melz* für das Interesse an meiner Arbeit sowie für die Übernahme des Korreferats.

Mein Dank gilt natürlich auch allen meinen Kolleginnen und Kollegen für die erfolgreiche gemeinsame Zeit am Institut sowie die durchweg tolle Zusammenarbeit. Insbesondere dem PHM-Team, in deren Umfeld ein häufiger Fachaustausch stattfand, der das Entstehen dieser Arbeit maßgeblich mitgeprägt hat. Mein spezieller Dank gilt dabei vor allem Herrn *Dr.-Ing. Simon Mehringskötter*, der mir als Team- und Bürokollege stets für Fachgespräche und Sparrings zur Seite stand.

Herrn *Lorenz Dingeldein* danke ich für die kritische Auseinandersetzung mit dieser Arbeit und den daraus resultierenden Anregungen. Bei Herrn *Sebastian Stern* möchte ich mich für die Unterstützung beim Aufbau und der sicheren Inbetriebnahme der vielen unterschiedlichen Versuchsaufbauten bedanken.

Abschließend bedanke ich mich herzlich bei meiner Familie, die mich in jeglicher Art und Weise immer unterstützt hat und mir so letztendlich diesen Weg ermöglicht hat. Meiner wundervollen Frau *Lisa* danke ich für ihre unermüdliche Geduld sowie liebevolle Unterstützung während der gesamten Zeit der Erstellung meiner Dissertation.

Henrik Heier, Juni 2023

Abstract

Complex technical systems such as aircraft, power plants or manufacturing systems play a vital role in our modern world as large portions of society rely on their correct functioning every day. The achievement of high system availabilities is therefore an important goal for system operators. Still, an accurate estimation of a system's future capability is not trivial as modern systems are often composed of many individual parts and subsystems and are often operated in varying modes or changing environments. Traditional statistics-based methods from Reliability Engineering reach their limits here as they struggle to capture the true current system state. An alternative approach is given with the discipline of Prognostics and Health Management (PHM) that uses onboard sensors to estimate the current and future health status based on real measurements. A still open research question in this context is how to aggregate PHM results of a multi-component system and integrate them into the exiting reliability frameworks in order to obtain a practical and useful decision support that provides accurate system state predictions.

With this thesis, the described problem is addressed and a *Dynamic Hybrid Reliability Model* (DHRM) is developed. Therefore, the DHRM combines traditional methods from Reliability Engineering with the novel PHM approach. Based on the V-Model methodology, it is described how the DHRM is constructed, implemented, verified and evaluated. The overall concept is described on three different aggregation levels, namely the *part-*, *component-* and *system-level*. On the part-level the integration of PHM results in a reliability context is highlighted, while on the component-level a fault-tree model is used for further aggregation. For the system-level, a state-space model is then used to represent the individual states of the considered system. The conception of the DHRM method concludes by showing how uncertainties are considered throughout the model and by the implementation of a software prototype for the calculation of the DHRM.

The DHRM is then applied to the Control Surface Actuation System (CSAS) of a hybrid drone to examine its applicability. The degradation of the CSAS is simulated with generated failure data based on a stochastic differential equation framework and predicted

with a PHM algorithm based on a Gaussian Process regression model. To assess the capabilities of the novel method, the CSAS is calculated by the DHRM for different parameter combinations and compared to a reference case without any dynamic PHM input data. During an evaluation phase, the results of the DHRM are then quantified based on a set of predefined metrics and compared to the reference case. It is shown that the DHRM outperforms the reference method in accuracy and precision.

The thesis concludes with a discussion of the results. The main advantages of the novel DHRM are seen in its capability of not only accurately predicting upcoming state transitions, but also providing a precise estimation of the actual new state the system will enter. This is seen as a major advantage of this method as it allows operators to plan ahead with a precise knowledge about the future system capabilities. However, the approach also has its merits and limits as the quality of the DHRM results largely depend on the accuracy and availability of the required underlying PHM data.

Table of Contents

Nomenclature

Symbols

Latin letters:

A	-	Availability
B	-	Accomplishment level (in comparison to Y)
$\mathbf{B}$	-	Binary combination matrix
$\overline{\mathbf{B}}$	-	Negated binary combination matrix
C	-	Costs
CI	-	Confidence interval
D	-	Damage fraction (Wöhler curve)
D_t	-	Stochastic process (general)
dX_t	-	Process variable / degradation rate (SDE damage model)
E	-	Expectation value
$\overline{E}$	-	Mean expectation value
E_a	J	Activation energy (Erying Model)
$\mathbf{f}$	-	Vector of latent function values (Gaussian Process)
$\mathbf{f}_*$	-	Gaussian Process posterior prediction (vector of latent function values)
F	-	Unreliability
F_l	-	Failure probability of part l
F_m	-	Top node failure probability (Fault Tree Model)
ΔG	J	(Gibbs) free energy (Free Energy Model)
ΔG^*	J	(Gibbs) activation energy (Free Energy Model)

h	-	Step-size (numerical simulation)
i	-	Index variable (generic)
I	-	Investment
j	-	Index variable (generic)
$\mathbf{I}$	-	Identity matrix
k	-	Index variable for time instance (discrete)
l	-	Index variable (current unit/part)
L	-	Number of units
K_B	$\frac{J}{K}$	Boltzmann constant
$\mathbf{K}$	-	Covariance matrix (Gaussian Process)
$\mathbf{M}$	-	System state-space probability matrix
N, n	-	Number of elements (generic)
R	-	Reliability
R_l	-	Reliability of part l
r	-	Remaining useful lifetime
r_l	-	Remaining useful lifetime of unit l
r_l^*	-	True remaining useful lifetime of unit l
S_i	-	System state i
$\mathbf{S}$	-	State vector / matrix
T	$^\circ C$	Temperature
t	-	Time (continuous)
$\hat{t}$	-	Accumulated time
t_{EoL}	-	Time instance for the true end of life
t_{EoP}	-	Time instance for the end of life as prognosed
t_P	-	Time instance when the prognosis was started
w	-	Weighting factor
x	$\%$	Degradation level
$\dot{x}$	-	Degradation rate (Free Energy Model)
X_t	-	Process variable / degradation level at time t (SDE damage model)
$\mathbf{x}$	-	General input vector (Gaussian Process learning)
$\mathbf{x}_*$	-	Input vector for new observations (Gaussian Process)

$\mathbf{X}$	-	Training input for Gaussian Process ($n \times n$ matrix)
$\mathbf{X}_*$	-	Observation input for Gaussian Process ($n \times n$ matrix)
x_k	%	Degradation level at time instance k
x_{th}	%	Degradation level threshold
$\mathbf{y}$	-	General output vector (Gaussian Process)
Y	-	Performance state

Greek letters:

β	-	Shape parameter (Weibull distribution)
Δ_l	-	Remaining useful lifetime error of current unit l
ϵ	-	Noise or measurement error
η	-	Scale parameter (Weibull distribution)
$\mathbf{\Theta}$	-	Parameter vector (hyper-parameters for Gaussian Process)
$\mathbf{\Theta_i}$	-	Model variances (general path model)
λ	$\frac{1}{10^6 h}$	Failure rate
λ_b	$\frac{1}{10^6 h}$	Base failure rate
λ_l	$\frac{1}{10^6 h}$	Part failure rate
Λ	-	Lambda value for predictions ($\Lambda = t_P/t_{EOL}$)
μ	-	Mean value
π_T	-	Temperature factor
π_S	-	Stress factor
π_C	-	Construction factor
π_Q	-	Quality factor
π_E	-	Environment factor
σ	-	Variance
υ	-	Scaling factor (degradation data generation)
$\mathbf{\Phi}$	-	Model parameter (general path model)
ψ	-	Offset term (uncertainty handling)

Subscripts:

$(.)_k$	-	Value at a specific time instance
$(.)_l$	-	Value for an individual unit l
$(.)_{obs}$	-	Observed / measured value
$(.)_{pred}$	-	Predicted value
$(.)_{true}$	-	True value

Functions & mathematical operations:

$\mathbf{0}$	-	Zero vector (vector only containing zeros)
$a(.)$	-	Damage accumulation function (SDE damage model)
$b(.)$	-	General degradation from (SDE damage model)
$cov(.)$	-	Covariance function (GP)
$Dist(.)$	-	Distribution function (placeholder)
$GP(.)$	-	Gaussian Process
$H(.)$	-	System architecture function (S-RUL)
$k(.)$	-	Covariance (kernel) function of the Gaussian Process
$m(.)$	-	Mean function
$M(.)$	-	Median function
$O(.)$	-	Big-O notation
$P(.)$	-	Probability function (cumulative distribution function)
$p(.)$	-	Probability density function
$\Gamma(.)$	-	Gamma distribution function
$\eta(.)$	-	General path function (general path degradation model)
$N(.)$	-	Normal distribution function
$\sigma(.)$	-	Standard deviation

Abbreviations

AACC	Aggregated Accuracy
ACC	Accuracy

AELR	Aggregated Early/Late-Ratio
AG	Advisory Generation
AGREE	Advisory Group on Reliability of Electronic Equipment
ALT	Accelerated Life Tests
ANN	Artificial Neural Network
ATPR	Aggregated True Positive Rate
AUC	Area Under Curve
B	Average Bias
BLDC	Brushless Direct Current
BMVg	German Ministry of Defence (Bundesministerium der Verteidigung)
BVLOS	Beyond Visual Line of Sight
CBFTA	Condition-based Fault Tree Analysis
CbM	Condition-based Maintenance
CDF	Cumulative Distribution Function
CLT	Central-Limit-Theorem
CM	Corrective Maintenance
CMOS	Complementary Metal-oxide-semiconductor
COTS	Commercially of the Shelf (Product)
CSAS	Control Surface Actuation System
DA	Data Acquisition
DB	Database
DFT	Dynamic Fault Tree
DHRM	Dynamic Hybrid Reliability Model
DM	Data Manipulation
EASA	European Union Aviation Safety Agency
ECU	Electronic Control Unit
EF	Error Factor
ELR	Early/Late-Ratio
EMA	Electromechanical Actuator
EoL	End of Life
EoP	End of Prediction

ER	Exploitation Ratio
FAA	Federal Aviation Administration
FFT	Fast Fourier Transformation
FMECA	Failure Modes and Effects Analysis
FN	False Negative
FNR	False Negative Rate
FORM	First Order Reliability Model
FP	False Positive
FPR	False Positive Rate
FPGA	Field Programmable Gate Array
FSR	Institute of Flight Systems and Automatic Control
FT	Fault Tree
FTA	Fault Tree Analysis
GBM	Geometric Brownian Motion
GGP	Geometric Gamma Process
GP	Gaussian Process
GPR	Gaussian Process Regression
HA	Health Assessment
HCI	Hot Carrier Injection
IC	Integrated Circuit
IIM	Inoperability Input-Output Model
INF	Infrastructure Costs
IoT	Internet of Things
KNN	k-Nearest-Neighbor
LHS	Latin Hypercube Sampling
LRU	Line Replaceable Unit
MAD	Mean Absolute Deviation from Sample Median
MAPE	Mean Absolute Percentage Error
MCS	Monte Carlo Simulation
MCSA	Motor Current Signal Analysis
MIL-HDBK	Military Handbook
MMA	Multiple Model Approach

MOSFET	Metal-Oxide-Semiconductor Field-Effect Transistor
MP	Markov Process
MSE	Mean Squared Error
MTBF	Mean Time Between Failure
MTBUR	Mean Time Between Unit Replacement
MTTF	Mean Time To Failure
NRE	Non-recurring Costs
OEM	Original Equipment Manufacturer
OOP	Object-Oriented Programming
OSA-CBM	Open System Architecture for Condition-Based Maintenance
PA	Prognostics Assessment
PBC	Performance-based Contracting
PDF	Probability Density Function
PdM	Predictive Maintenance
PDMP	Piece-wise Deterministic Markov Process
PHM	Prognostics and Health Management
PM	Preventive Maintenance
PoF	Physics of Failure
QC	Quadcruiser
RAMS	Reliability, Availability and Maintainability System
RBD	Reliability Block Diagram
REC	Recurring Costs
Rel	Reliability
ROI	Return of Investment
RPAS	Remotely Piloted Aircraft System
RUL	Remaining Useful Lifetime
S	Standard Deviation
SD	State Detection
SDE	Stochastic Differential Equation
SIR	Sampling Importance Resampling
S-RUL	System Remaining Useful Lifetime
SVM	Support Vector Machine

T&MC	Time and Material Contracts
TDDB	Time-Dependent Dielectric Breakdown
TN	True Negative
TNR	True Negative Rate
TTF	Time to Failure
TP	True Positive
TPR	True Positive Rate
TUDA	Technische Universität Darmstadt (Technical University of Darmstadt)
UAV	Unmanned Aerial Vehicle
UD	Uncertainty Distribution
UUT	Unit Under Test
VLOS	Visual Line of Sight
VTOL	Vertical Take-off and Landing

1. Introduction

Technical systems become continually more sophisticated, as they are intended to solve complex problems in our modern sociotechnical world. These systems often consist of many different components and subsystems, which interact and interfere with each other to accomplish a common goal. Used in varying usage-scenarios and environments, they are often referred to as a system of systems or complex systems and can be found among different high-tech industries such as the energy, defence or aerospace sector. For operators of these costly assets it is important to achieve high availabilities and meet desired performance targets at all times. This is not only due to economic considerations, but also includes safety and social factors, as these systems are often part of critical infrastructure or required to maintain our daily safeness and wellbeing. [Jam08]

To highlight the relevance and associated effects of unavailability, the following examples are given: The accumulated costs of unexpected downtimes for the Fortune Global 500 companies are estimated to be 1.5 billion USD in 2021. The costs of a one-hour loss of production, e.g. due to a breakdown of machinery, reaches from 39 thousand USD for the consumer goods industry up to 2 million USD for the automotive industry. [Sen22] In the aviation branch the breakdown and delay costs have been studied and evaluated in [And15]. The average costs of an *at-gate breakdown* are reported here to be 540 EUR per each 15 min of delay for a Boeing 737 passenger aircraft. Further, reports from the German ministry of defence (BMVg) state that systems at the beginning and near the end of their intended lifetime are especially affected from low operational availabilities. For example, it is reported that the recently introduced new transport aircraft A400M as well as the helicopter NH 90 reached an unsatisfactory availability of below 40 % during the year 2019. Old systems such as the Tornado or P-3C Orion, both of which are near the end of their lifetime, were reported to have availabilities below 50% [Bun19]. As a result of those numbers, the improvement on the operational availability, planning and logistics have been defined as the main strategic goals for the BMVg [Bun21].

From an operator's point of view, it is therefore crucial to accurately assess the current and future availabilities and performance levels of such systems as this helps to quantify whether

the system's future capabilities will be sufficient to meet defined performance targets or not. With sufficient time to prepare, the operator thus can incorporate this information into his decision-making process and decide on mission (re-)planning, maintenance actions or even strategic decisions such as leasing or outsourcing of performance and thus improve the overall availability while constraining costs [Tia+12; TAC99]. However, this task becomes more difficult as systems increase in complexity with multiple interacting components and are being used in progressively more versatile scenarios [Zio09].

The de facto standard to address the stated issues is found in the discipline of *Reliability Engineering*. The tools and concepts provided herein are mainly statistical based approaches, where reliabilities and safety margins are estimated with the help of known failure distributions from historical data. Standards such as the *ARP4763* [SAE96] or the *Military Handbook of failure rates* [Dep91] have evolved as first choice tools to estimate and predict the reliability of components and systems. Although this concept has worked well during previous decades, it becomes more questionable as the complexity of systems rises and their individual usage increases in variance, which makes it difficult to find common failure patterns or correlations between influencing factors [Zio09]. This is further stressed, when systems are used over a long period of time and the initial usage scope changes, the system is modified, or its performance expectations are increased over time. The initial stated estimates will then have to be reassessed and adapted [SL06]. In addition to this, most quantitative methods from Reliability Engineering were originally developed to capture binary events (functioning or faulty). However, modern complex systems are often multi-state systems that can settle on different performance levels (e.g. 100%, 90%, 50% of nominal capacity) depending on the current operating conditions [Zio09].

To challenge the above stated difficulties, industries and researchers are keen to develop novel concepts to improve the health assessment and decision-making process for complex technical systems. A very promising development is the emerging of the *Prognostics and Health Management* (PHM) discipline during the previous two decades. By monitoring and evaluating on-board system data, PHM intends to estimate the current (*diagnosis*) as well as the future (*prognosis*) health state of a considered system [Hes02]. This approach is considered as a game-changer for the industry as maintenance actions and other operational decisions can be made based on the individual state of a given system and the estimated remaining useful lifetime (RUL) of its components. The expected benefits from this approach include increased system availability, reduced risk of a loss, reduction of unplanned maintenance and downtimes, improved logistics as well as an increased mission success rate [Voh+08]. Further, it is considered as a key-enabler for new business models such as performance-based contracting (PBC) and maintenance strategies such as predictive maintenance (PdM), where precise information about the current health state is

required. [Sun+12] Still, PHM is an emerging discipline, with an active field of research. During recent years, different approaches and concepts have been developed including the utilization of data-driven and physical-based models, the consideration of economic aspects as well as the definition of evaluation metrics and monitoring algorithms. However, most of this research is still focused on the assessment of subcomponents and single items such as motors, gears or pumps. The assessment of the full system state, considering the dependencies of interconnected components forming a larger system, as it is relevant for larger complex systems, is still an emerging development [Rod17].

In order to structure the above-described challenges and to derive the aim of this thesis, the terms of *performability*, *complex systems*, *performance-based contracting* and *predictive maintenance* are introduced and defined in the following section.

1.1. Performability of complex systems

Many modern technical systems are considered as being complex. These systems are often found in high technology industries such as the aerospace, defense and energy sectors. Although there exists no distinct definition in literature of what a complex system is, they can still be characterized and described by a set of attributes, which are often held by these systems [GSB08]. These attributes include:

- Complex system structure with inter-dependencies and interconnections
- Multiple system states (operating states, performance levels)
- Different domains (mechanical-, electrical-, hydraulic-subsystems, software, etc.)
- Embedded uncertainties (operational, failure and degradation)
- High level of automation with decision support
- Human interaction and organizational aspects

For operators, it is relevant to continuously monitor and track the performance and effectiveness of such systems in order to meet desired system outputs. Therefore, historically different metrics have been introduced, which are suited to describe a given system in its performance and dependability. In this context, performance describes the general ability to accomplish intended services with given hardware assuming no changes in its structure. Dependability instead, is the ability of a system to provide its services in a justifiable way and considers structural changes [BDN11]. As systems start to degrade or

are allowed to have faults, which is the case with fault-tolerant designs, a clear differentiation between performance and dependability becomes difficult. To overcome this and describe the effectiveness of a system under varying conditions and performance levels the term *performability* was introduced by Meyer [Mey80]. Performability thus connects performance metrics such as capacity or accuracy with dependability metrics such as availability and reliability and thus is suited to describe a system's performance under consideration of changes and uncertainties. The performability of a system is thus defined as the probability P that a system delivers a performance Y, which is specified through a set of measurable accomplishment levels B [Mey92]:

$$Perf(B) = P(Y \in B) \tag{1.1}$$

An overview of performability and its associated properties is given in Figure 1.1. The used taxonomy is based on [BDN11] and will be used throughout this thesis.

From these definitions it can be concluded that somebody who wants to optimize a complex system e.g. in an economic and ecological way, needs to improve on the properties of its performability. This can be achieved by improving the performance itself (metrics such as capacity and throughput) as well as by improving the associated dependability metrics (such as availability and reliability). Accordingly, the design-thinking of performability is considered as one of the foremost needs of the 21st century for modern technical systems [Mis08] as the ability to quantify the performance and effectiveness of systems builds the foundation for new business models such as PBC.

1.1.1. Performance-based contracting

For many industries, after-sales product support has become a crucial role, as maintenance and service actions can be cumbersome and expensive. This is especially the case for products with long life cycles such as aircraft, power plants or complex systems in general, where the consequences of any downtimes are severe. Traditionally, original equipment manufacturers (OEM) have addressed this through the use of time and material contracts (T&MC), where the customer is supposed to pay for the resources required to service the product. With increasing system complexity and competition, operators are tending towards a more service-based approach, where performance is procured according to product usage time or defined system outputs instead of the system itself. This is considered as performance-based contracting or sometimes also performance-based logistics or power

Performability: Ability to accomplish a service in the presence of faults over a specified period	
Performance: Ability of a system to accomplish its intended services within given non-functional constrains	**Dependability:** Ability of a system to provide its intended services in a justifiable way
Timeliness: Ability of the system to provide a service according to given time requirements **Precision:** Ability of the system to provide the same results under unchanged conditions **Accuracy:** Ability of the system to provide exact results **Capacity**: Ability of the system to hold a certain amount of data **Throughput**: Ability to handle a certain amount of operations	**Availability**: Readiness for correct service **Reliability**: Continuity of correct service **Safety**: Absence of catastrophic consequences **Integrity**: Absence of improper system state alterations **Maintainability**: Ability to undergo modifications and repairs

Figure 1.1.: Taxonomy of performability properties, based on [BDN11]

by the hour[1] [Gua+12]. As given in [Bun18], PBC presents with new opportunities and risks at the same time for OEMs. On the one hand manufacturers gain more control of (service) processes and freedom regarding technical realization, on the other hand there is much more responsibility for providing the performance agreed upon in the contract. Thus, a central aspect of PBC is a high availability of the technical product. In this regard, the term availability describes the readiness of a system for a correct service, which is generally defined as $A = uptime/(downtime + uptime)$. It is tightly coupled to the reliability of the used components and its overall maintainability, as time-consuming maintenance actions or frequent maintenance intervals directly affect the availability of a system. Thus, predictive maintenance, a novel maintenance strategy, as well as PHM, a discipline for an accurate prediction of failure times, are considered as technological cornerstones for the

[1] Commercial business model by Rolls-Royce

facilitation of PBC [JS11].

1.1.2. Predictive maintenance

Maintenance is an important factor in the life-cycle of every system as it is intended to retain its performability. As set out in [Swa01], maintenance can be broadly split into a corrective (CM) and a preventive maintenance (PM) strategy. A corrective strategy is the simplest and cheapest form of maintenance. Here, components are replaced after they have failed, which can be done immediately or deferred. This strategy is applicable for those systems, where failures and downtimes are tolerable as breakdowns can occur unexpectedly. For safety critical systems and those where downtimes need to be kept to a minimum, e.g. in production lines, a preventive strategy is chosen instead. PM includes any mechanisms utilized to avoid unexpected downtimes and can be further divided into predetermined and condition-based maintenance (CbM) schemes. Scheduled maintenance is considered the state of the art in PM. Herein maintenance actions are based on a fixed time schedule or number of operating cycles. This can include inspections, repairs and general overhauls. In contrast to scheduled maintenance, CbM schedules actions according to the true condition of components and is therefore considered improving on maintenance costs and efficiency. A special case of CbM is predictive maintenance (PdM), where not only the current condition is assessed, but also future states are forecasted in order to derive an optimal replacement time. Being highly automated, CbM and PdM can be undertaken on scheduled, continuous or on-request basis. In Figure 1.2, an overview of available maintenance strategies, together with their timing concepts are given based on [NYP10].

Although PM is at present the standard maintenance strategy to achieve high reliability and safety levels, it has two known drawbacks: 1) A (larger) portion of lifetime of a component usually remains unutilized as most components are replaced long before they fail. 2) A scheduled maintenance still cannot achieve a 100% failure free operation. Thus, there always remains a minor percentage of failures occurring, which will result in unplanned and expensive downtimes. Overall, the PM approach is a trade-off between an acceptable risk level, described by a tolerable number of failures versus the failure prevention costs resulting from any maintenance actions. The economic side behind this is illustrated in Figure 1.3. Systems governed by a PM strategy will have overall less failure times as components are replaced before they fail. This leads to less repair activities (and costs) but comes at high prevention costs as frequent maintenance actions are required. Systems with a more corrective strategy instead will result in more failure events, increasing the repair costs. However, the costs for prevention are much lower. As CBM is used to identify the best time for replacement before a breakdown occurs, it defines the optimum total costs,

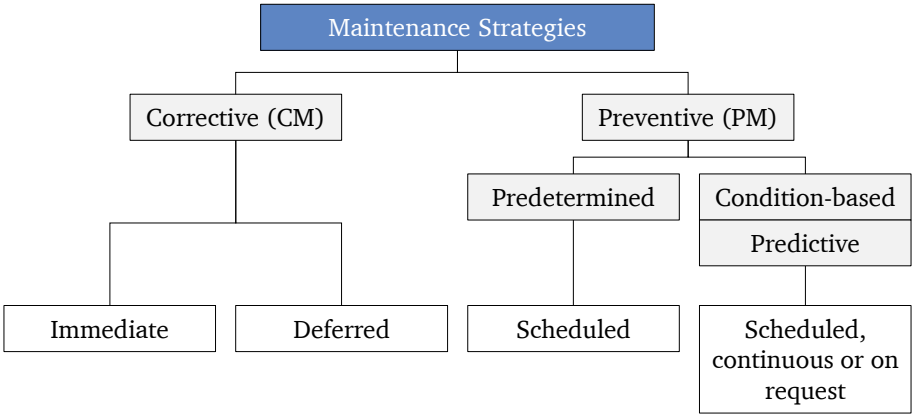


Figure 1.2.: Comparison of different maintenance strategies, based on [NYP10]

as repair and prevention costs are kept to a minimum. However, it has to be mentioned that CBM usually requires an initial implementation effort, which is not considered in this figure. Further, a too aggressive PM or too weak CM strategy can also lower the overall availability as systems will not be available due to preventive respectively corrective maintenance actions. [NP09]

1.2. Aim of this thesis

With the above stated insights and challenges, the need for novel decision support methods is derived that overcome the limitation of traditional static assessment methods from Reliability Engineering and help to accurately estimate the current and future system capabilities. The need for such a method is justified by the increased complexity of modern systems together with the demand for higher performability at reasonable costs.

Hence, the aim of this thesis is to develop a decision support method that uses PHM to improve the performability of a complex system by estimating the availability of individual future performance levels. Based on prognostic results from the system's underlying components, the output shall be dynamically (re-)calculated to capture the individual degradation process of the monitored system. A main research question in this thesis is

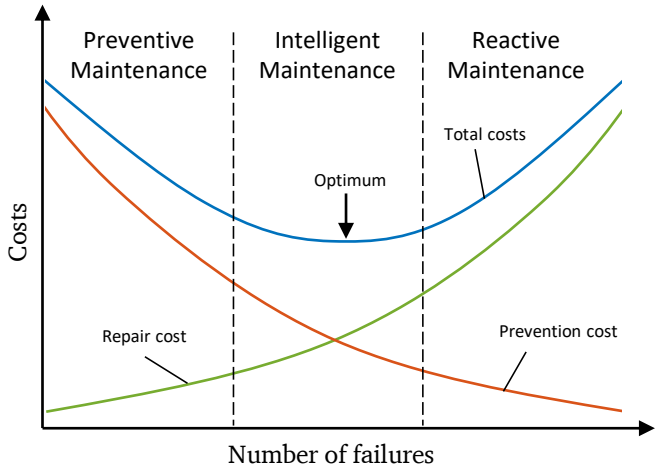


Figure 1.3.: Maintenance costs based on the chosen strategy, based on [Tch+14]

how to aggregate different PHM results in a multi-component system up to system-level to give a precise and meaningful assertion. The overall method is considered as a proof of concept and therefore shall be generic and system independent.

From the foreseen method it is expected that a system's overall availability will improve as operators are supported in their mission and maintenance planning based on actual feedback from the considered system(s). Further, such a method will also provide benefits to contractors that have agreed to deliver predefined performance levels in a performance-based contracting scheme. Finally, it is hoped that this work provides further application areas to foster the practical use of PHM.

1.3. Structure of this thesis

This thesis is structured into six consecutive chapters: In the first chapter the introduction and motivation to the topic is given. Chapter 2 is then dedicated to the fundamentals and state of the art relevant for this thesis. It covers the general concepts, the terminology as well as the required basics of the reliability engineering and PHM discipline and concludes

with the indicated research gap, based on an extensive literature research. Afterwards, the development of the proposed new method is described in Chapter 3. Based on the V-Model the method is designed, implemented as a prototype and verified in a first step. In Chapter 4, the method is then applied to a practical use-case from the aerospace domain. Therefore, all required input parameters and values are generated and an exemplary data-driven prognosis algorithm is developed. Within Chapter 5, the created method is then evaluated and tested against a predefined reference case. To quantify and interpret the results, appropriate metrics are introduced. Chapter 6 finally concludes and summarizes all results before additional recommendations for future research in this topic are given.

2. State of the art of availability assessments for complex systems

In the following chapter, the state of the art relevant for this thesis is outlined. At first, the general physical mechanisms of degradation (physics of failure) are described in Section 2.1, before the conventional reliability engineering, the de facto standard discipline related to problems such as availability and reliability forecasting, is presented in Section 2.2. Afterwards, PHM as an evolving alternative is presented in Section 2.3 with a special emphasis given to the aggregation on system level. Finally, the insights on this topic and the identified research gap are summarized and outlined in Section 2.4.

2.1. Degradation basics (Physics of Failure)

Nothing lasts forever. This proverb also applies to technical systems. All systems from small micro-electromechanical systems to large mechanical structures suffer from wearout and degradation. If not replaced or repaired in time, the degradation will continue until a representative system parameter will exceed a predefined threshold and the system is considered to be failed. The period from the original intact state of a system until reaching its specification limits, also known as system/device failure, is defined as *time-to-failure* (TF). It is important to highlight that the TF for any given device not only depends on the chosen material, but also on the affecting stressors. Although often associated with mechanical stress, these stressors can have many origins. For electronics for example, current density, electric-field or voltage stress are often the dominant root causes for failures. For mechanical systems instead, cyclic, frictional or thermomechanical stress often causes degradation. Nevertheless, in practical applications these stressors usually coexist. Thus, the TF for any device is highly individual as it depends on the usage, experiencing loads and the surrounding environment [McP10]. This makes life-time predictions and reliability estimations difficult as the degradation process is always subject to variations.

The physical description of materials degradation is the subject of *Reliability Physics* also known as *Physics of Failure* (PoF). A fundamental way to argue the existence of material degradation as a physical process can be found in the second law of Thermodynamics and its definition of a system's disorder based on Entropy. As it is argued in [BKL08] and [Fei18] every system interacting with its environment (e.g. not ideally isolated) will run a spontaneous and irreversible degradation process in order to get towards the thermodynamic equilibrium. As the system ages, its disorder naturally increases. This process only runs in one direction, as there is no system which repairs itself. Based on the fundamental laws of thermodynamics, it can be explained why every system suffers from degradation. The interesting question for an engineer at this point is to understand, how this degradation evolves and how fast this process runs or in other words, when a given system will fail. To understand and describe the kinetics of the degradation process the *Transition State Theory* is used [McP10], in which any material or system is considered to exist in metastable states. Metastable in this context means, that these materials are only apparently stable and will change their state with time to reach a more stable state at a lower free energy level. To leave the current state, some sort of activation energy ΔG^* is required to move the system to a lower, more stable, degraded stated. During the degradation the free energy ΔG is released. This coherence is illustrated in Figure 2.1:

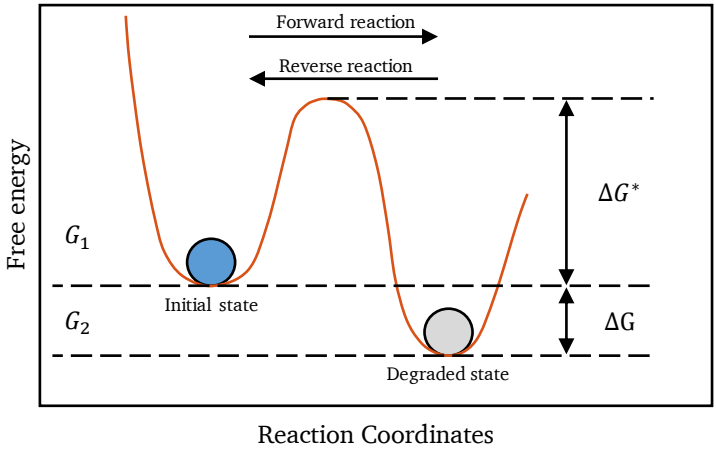


Figure 2.1.: Illustration of the degradation process based on the free energy model, based on [McP10]

Although ΔG being the main driver for degradation, the overall degradation speed is thus

governed by the available activation energy ΔG^*. In analogy to chemical reaction rates, this speed of the overall process is described by the net degradation rate $\dot{x}_{net}$ resulting from the forward and backward reaction as given in [McP10, p. 98]

$$\dot{x}_{net} = \underbrace{\dot{x}_0 \exp\left[-\frac{\Delta G^*}{K_B T}\right]}_{\text{forward}} - \underbrace{\dot{x}_0 \exp\left[-\frac{\Delta G + \Delta G^*}{K_B T}\right]}_{\text{reverse}}. \tag{2.2}$$

Here T is the temperature and K_B is the Boltzmann constant. In the general case where, $\Delta G \gg K_B T$ the above formula simplifies to

$$\dot{x}_{net} = \dot{x}_0 \exp\left[-\frac{\Delta G^*}{K_B T}\right]. \tag{2.3}$$

The above formulation can be further extended to the Erying model, a popular equation used in chemical kinetics to describe the rate of a process based on a given temperature. However, under the assumption of a constant activation entropy and enthalpy the Erying model becomes similar to the empirical Arrhenius equation, which is given as:

$$\dot{x} = \dot{x}_1 \exp\left[-\frac{E_a}{K_B T}\right]. \tag{2.4}$$

This formula is one of the fundamental equations within physics of failures as it directly correlates the given activation energy E_a to the degradation rate $\dot{x}$ of a material. Further, this model explains, why the speed of degradation increases if more energy, e.g. through high temperatures, is available. The above formulation is often used in the physics of failure discipline to model the degradation process and estimate the time to failure. Several degradation phenomenons for electronics as well as mechanical parts, such as the hot carrier injection (HCI), corrosion, the time-dependent dielectric breakdown (TDDB) or the estimation of the creep rate, exists in literature.

With the Transient State Theory and the metastable system states, the degradation process itself can be explained. Further, this theory provides an argument, why the speed of a degradation process varies from system to system and is influenced by its usage, the experienced loads as well as the surrounding environment, or in general, the acting (activation) energy.

The here described physics of failure discipline is often used for lifetime estimations, especially during system design phase, when no failure data is available as well as for

electronic parts, where a direct measurement of the degradation process is usually not possible. However, this method requires a solid understanding of the underlying degradation process and thus becomes impractical and expensive for large and complex systems.

2.2. Reliability Engineering

In the following subsection, *Reliability Engineering* as the de facto standard discipline for reliability analyses is introduced. Emerged in the early fifties[1], Reliability Engineering is nowadays the standard across many industries to assess the reliability of technical systems. Reliability Engineering is mainly based on the statistical processing of historical failure data, which makes it very practical as failure data is easy to obtain and does not require any additional sensors or a profound knowledge of the underlying degradation processes. The mathematical models and tools combined in this discipline are further complemented by standards and guidelines from the corresponding industries, which allow a common framework to define a system's reliability and give failure time predictions. In this context and for the remainder of this work, *reliability* and *failure* are defined as follows:

- *Reliability* is the probability that the required function will be provided under given conditions for a given time interval. [Bir10, p. 375]
- *Failure* is defined as the termination of the ability to perform the required function. [Bir10, p. 369]

Reliability Engineering combines different models and tools to assess, predict and maintain the reliability of technical systems along the complete life-cycle. To provide the reader a general guideline, for the remainder of this work it will be distinguished between those methods and models which focus on the reliability assessment of single items and those which are used to aggregate the overall system's reliability. The following subsections are thus structured in that way.

2.2.1. Basic reliability concepts (item description)

In this section, the basics of reliability calculations are outlined, with a special emphasis on the modelling of single items. As presented above, the reliability $R(t)$ in this context is

[1] The AGREE (Advisory Group on Reliability of Electronic Equipment, Secretary of Defense of the United States) Report from 1954 is often used to mark the birth of Reliability Engineering [SM06].

understood as the probability that a given item (or system) is still functional at a given time instance t, which refers to the *cumulative distribution function* (CDF). In contrast, the failure probability $F(t)$ describes the probability that an item (or system) has failed until time instance t. Both functions are correlated with $R\,(t) = 1 - F\,(t)$ as depicted in Figure 2.2.

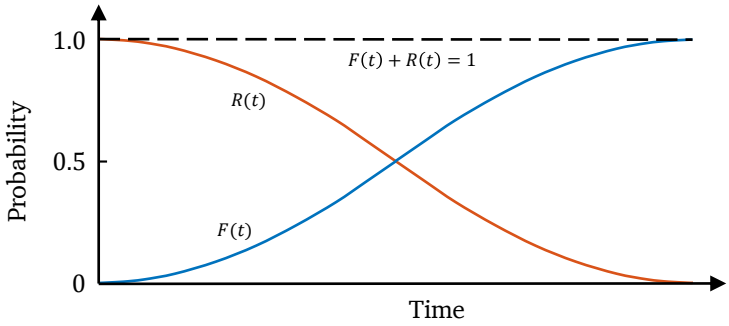


Figure 2.2.: Correlation between $R(t)$ and $F(t)$ [KP14]

An often used measure to describe the failure characteristic of a given item is the *failure rate*. The failure rate describes the number of failures for a given time interval. It is a very practical measure as it is easy to determine from historic failure data. The failure rate is defined as

$$\lambda\,(t) = \frac{f\,(t)}{R\,(t)} = -\frac{dR\,(t)}{dt}\frac{1}{R\,(t)} \quad \left[\frac{1}{h^*}\right] \tag{2.5}$$

and usually given in failures per hours or a million hours. In formula 2.5, $f\,(t)$ is the *probability density function* (PDF) of F with $f\,(t) = dF\,(t)\,/dt$. For many systems it can be shown, that the failure rate varies over time due to infant failures and late wear out effects. This characteristic is described with the bathtub curve, shown in Figure 2.3. The curve begins with a high decreasing failure rate at the beginning of an item's lifetime, which is referred to as *infant mortality* and describes early failures, e.g. due to manufacturing or installation problems. After this, usually short period, a longer period of nearly constant low failure rates is reached. That period is considered to capture and describe the appearance of random failures. Processing further in time, the failure rate often increases again as

items begin to suffer from degradation and mechanical wear out as described in [KP14, p. 26].

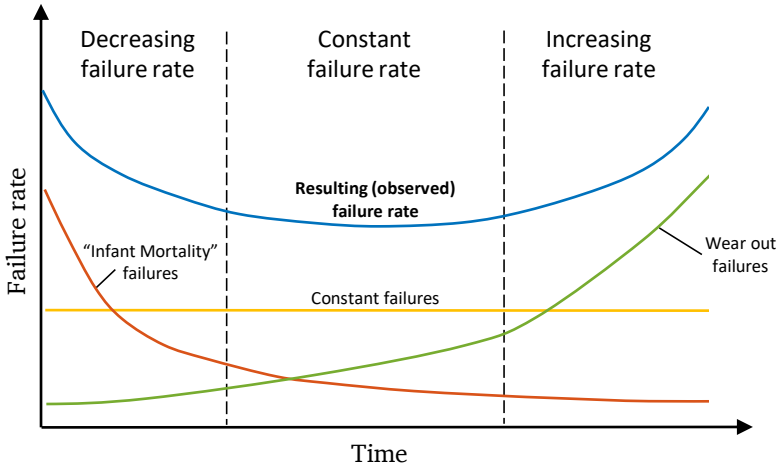


Figure 2.3.: Phases of different failure rates among an item's life-time, known as bathtub curve, based on [KP14]

As Reliability Engineering depends on the statistical evaluation of failure data, probability distributions play an important role as they are used to model the failure distribution of items. Common used distributions include the *exponential*, the *Weibull*, the *normal* and the *lognormal distribution* [KP14, p. 55-75]. The Weibull as well as the exponential distribution are briefly outlined in the following, as they will be of further interest for this thesis.

Exponential Distribution
The Exponential failure distribution considers a negative exponential, which leads to an increasing failure probability over time. The cumulative distribution function of the exponential failure distribution is defined as

$$F(t) = 1 - e^{-\lambda t}. \tag{2.6}$$

In this equation, F denotes the failure probability over time t and λ is known as the failure rate, which is assumed to be constant. The exponential failure distribution is often described as being memoryless, as it is only a function of the current mission time t and does

not consider any previously accumulated operation hours $\hat{t}$. This can be mathematically shown by evaluating the conditional exponential probability given as:

$$R\left(t|\hat{t}\right) = \frac{R\left(\hat{t}+t\right)}{R\left(\hat{t}\right)} = \frac{e^{-\lambda\left(\hat{t}+t-\gamma\right)}}{e^{-\lambda\left(\hat{t}-\gamma\right)}} = e^{-\lambda t} \tag{2.7}$$

Due to its memoryless property and constant failure rate, the exponential failure distribution is generally used to model random failures. It is often found as failure model for electronic items. [Bir10, p. 40]

Weibull Distribution
The Weibull distribution is another popular lifetime distribution. Its CDF is defined as follows:

$$F\left(t\right) = 1 - e^{-\left(\frac{t}{\eta}\right)^{\beta}}. \tag{2.8}$$

The above formula is known as two parameter Weibull distribution in which η is the scale and β the shape parameter. The Weibull distribution is thus of high importance in Reliability Engineering as the shape parameter can be used to define different failure rate types: For $\beta = 1$ the failure rate becomes constant, and the Weibull distribution becomes equal to the exponential distribution. For $\beta < 1$ the failure rate decreases over time (infant mortality) and for $\beta > 1$ it increases respectively. Thus, the Weibull distribution is often chosen as it can be used to model all parts of the bath tub failure curve, which makes it a powerful tool in reliability modelling. [KP14, p. 55]

2.2.2. Failure rate estimation

A crucial part in the reliability assessment based on a statistical approach is the thorough identification of suitable failure rates and models. A general approach to accomplish this task is the statistical evaluation of historical failure data, which includes a model selection, e.g. Weibull or exponential model, followed by a parameter fitting. However, this process can become cumbersome and expensive for complex systems or new designs, when no historic data is available or the system is foreseen to operate in a different environment. As given in [SV02, p. 85] failure rate data should be ideally obtained from actual mission data on the considered component or one of similar kind. If no mission data is available, *accelerated life tests* (ALT) can be used, which simulate the usage of an item. If no mission or ALT failure data is available often existing databases or so-called *failure rate handbooks*

are considered. Although being heavily criticized in literature for being inaccurate and outdated [MR93; FFP17; PK88], failure rate handbooks are nowadays still a common approach to conduct a first reliability assessment. Based on the evaluation of failure data for various items, these standards list empirical formulas to derive and estimate failure rates. By including additional parameters describing the considered installation environment, the quality of the item or its expected usage, failure rates can be fine-tuned and adapted to the foreseen application. Important handbooks as well as their application area are summarized in Table 2.1. Especially in the aerospace industry the *Military Handbook 217* [Dep91] is often consulted to obtain failure rate estimations with a strong focus on electronic equipment.

Table 2.1.: Overview of popular failure rate handbooks, their application and release date, based on [FFP17]

Name	Application	Last Issue
Mil-HDBK-217	Mil/Commercial	1995
Bellcore / Telcordia	Telecom	2006
RDF 2000	Telecom	2000
Siemens SN29500	Siemens Prop.	1999
PRISM	Mil/Commercial	2000

2.2.3. Reliability models (system description)

In the previous section the basics of the statistical assessment of single items was outlined. However, for most operators the analysis of the complete system, consisting of multiple, interconnected items, is relevant for a comprehensive safety and reliability analyses. Thus, it is necessary to model the dependencies and interactions among all contained items in a quantitative way to estimate the system's overall availability and reliability. This is the aim of so-called *dependability models* [JM88].

Over the past decades many of these models have been developed and used for safety and reliability analyses. Popular dependability models include the *Parts Count Method*, *Fault Trees* (FT), *Reliability Block Diagrams* (RBD) and *Markov Processes* (MP), which are widely used in many industries such as the aerospace or nuclear power industry. Broadly, these models can be categorized into three types, namely the *parts-count*, *combinatorial*, and *state-space models* [RV91]. Choosing the right model is not trivial, as they differ in their

modeling power and analysis complexity [Rv02]. To define the relative modeling power, the authors of [MT94] proposed a concept that transforms a given dependability model into another based on a given algorithm. As this transformation is not always possible in both directions, e.g. every parts-count model can be transformed into an equal fault tree but not vice-versa, a hierarchy among these models is derived. The resulting classification of dependency models together with the obtained *power hierarchy* described by Malhotra [MT94] and Everdij [EB03] is show in Figure 2.4.

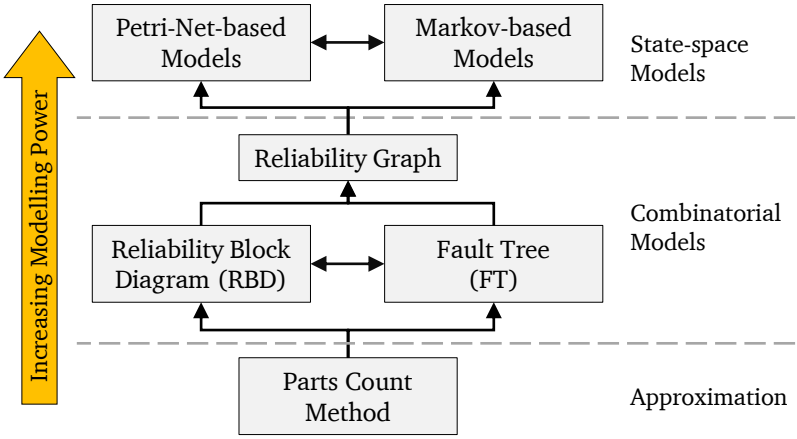


Figure 2.4.: Modelling power hierarchy of dependability models, based on [MT94] and [EB03]

However, it has to kept in mind that not only the modelling power has to be considered when selecting a suitable dependability model, as also other characteristics such as the visualization, their complexity or required computational effort play a role. To give further insights, in the following the three different model classes are briefly introduced, and their main characteristics outlined.

Parts count method

As given in Figure 2.4, the parts count method is among the simplest models with the least modelling power among the dependency models. The system failure rate of these models is calculated as the sum of all containing item failure rates [RV91]. Thus, the model neglects any redundancies and assumes a system to fail as soon as any of the contained items fails. This is a strong simplification and does not hold for most systems. However, this method

still is used to approximate a system's failure probability, especially during early design phases, as it is easy to apply [Bir10, p. 51].

Combinatorial models
In contrast to the parts count method, combinatorial models are able to model serial and parallel item configurations and thus are able to consider system redundancies. The mathematical basis of the combinatorial models is found in Boolean algebra, as those models can be represented with logical AND and OR connections. Important combinatorial models are the Reliability Block Diagram and the Fault Tree model. Although both models have the same modelling power (see Figure 2.4) and can be transformed into each other, both follow a different modelling approach, which also emphasizes their intended usage. While the Fault Tree analysis is a top-down approach, in which the causes of system failure are recursively searched, the RBD is a bottom up method, investigating how each item of a system contributes to the system's overall reliability. However, they can be used interchangeably depending on the analyst favor. [MT94]
Combinatorial models are often chosen, as they are intuitive, provide a visual representation and can be used in a quantitative and qualitative way. However, these models also suffer a few drawbacks. Being limited to a combinatorial decomposition of a system, they cannot be natively used to model recurring events or complex interactions between parts. Further, these models can only be used to represent two distinguishable states, *failed* and *not failed*, which limits their application to two-state problems. [JK95]

State-space models
State-space models can be broadly split in *Markov-based* and *Petri-Net-based* models. These dependability models are constructed on a finite set of states, in which each state represents an individual status of the system under study. Based on certain events, such as repairs or failures, the system's internal state is changed. This is also referred to as state transition. By assigning a transition probability, these models can be used to model the dynamics of a given system. Thus, these models are naturally well suited to model complex and dynamic systems with multiple inherent states. These can describe certain degradation levels for fault-tolerant systems, individual system configurations e.g. during defined mission phases or model available resources such as the availability of maintenance personal or spare parts. As state transitions can run in both directions, repair processes can also be modelled. This makes these models more sophisticated compared to the simpler combinatorial models as not only failure probabilities but also availability estimates can be generated [SAE96; MT94]. Representatives of state-space models include *Markov chains*, *stochastic Petri-Nets*, *Markov reward models* and *stochastic reward nets*.

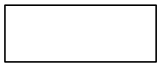
Although this model class has the highest modeling power among the dependability models, this class also has its disadvantage, as it is computational intense, which can be easily

shown: For a system composed of n components, which can be either operational or failed, there exists 2^n states in total. As this model complexity raises exponentially with the number of components, these models become computational heavy for large systems. This is also referred to as *state-space explosion*, a common problem of these models. Therefore, over the past many tools have been developed such as HARP, SHARPE or CARE-III, which accelerate the calculation speed of these models by either compressing and truncating states or by decomposing the model into multiple simpler subsets [JM88]. In addition to this and compared to combinatorial models such as Fault Trees or Reliability Block Diagrams, these models are difficult to visualize, which makes their development difficult and error-prone. Thus, state-space models are often chosen, when combinatorial models reach their modelling limits or additional considerations such as availability estimates are of interest to the analyst.

Fault Tree Analysis

Being relevant for this thesis, the fault tree model is introduced in more detail at this point. A Fault Tree Analysis (FTA) can be used as quantitative or qualitative analysis tool and has been well established over the last decades for safety and reliability analysis of complex systems. Used industries include nuclear power systems, chemical process plants as well as the aerospace industry. The fault tree analysis is a top-down approach: Starting from the unwanted top event, e.g. the system failure, all connected events (component failures), which lead to this top event, are analyzed in a recursive manner. Depending on the system structure the contributing events are modelled as logical AND or logical OR connections. Events, which cannot be further subdivided are known as basic events and are typically modelled with known probability distributions. This leads to a tree structure ranging from the top event to several basic events. The tree itself is visualized with defined symbols, making the fault tree model very intuitive and easy to read. In Figure 2.5, the basic fault tree symbols and their usage are presented. It shall be noted that there exist much more fault tree symbols, such as additional gates (e.g. inhibit, priority or exclusive OR gate) as well as specific event types (e.g. conditioning event or external events). However, these are not further outlined in this thesis and the interested reader is referred to [SAE96, p50] and [SV02].

For a quantitative analysis, the tree must be finally evaluated. This can be done by means of analytical equations. However, this approach can become very inefficient for larger trees with many gates. Alternatives suggest calculating the solution by an approximation of the result. Among this approach *Monte Carlo simulations* (MCS) are used frequently to solve larger trees. Finally, also decomposition methods can be used, which simplify a given tree by finding the minimal cut-set [SV02]. A common drawback of fault trees is that they are only able to distinguish between two states, usually describing the failed



Description Box: Description of a node's output

Basic Event: Internal event that cannot be further developed

AND-Gate: Logical AND connection to model parts that are in parallel configuration

OR-Gate: Logical OR connection to model parts that are in series configuration

Figure 2.5.: Basic symbols of the fault tree analysis as given in [SAE96, p50]

or functional mode of a system or component. Modelling multiple states, e.g. describing different degradation levels, requires multiple trees, making the overall model complex and error-prone. Further, fault trees are at their basis very static models, which cannot model any dynamic system aspects such as repairs or system changes by their nature. However, in the past there has been a lot of effort to overcome these limitations. Examples consider the use of dynamically updated failure rates [ST07] or the use of dynamic gates to model system changes as shown in [ČM02] and [Dur+09].

2.2.4. Uncertainty propagation

As reliability assessment is an inherent uncertain task, a thorough uncertainty propagation within these models becomes necessary. For conventional reliability models such as Fault Trees or Markov models, uncertainties emerge from used parameters, such as failure rates. As failure rates are obtained from field data and reliability handbooks they suffer from an epistemic lack of knowledge due to limited samples, as well as from a tolerance uncertainty as the considered items might be operated under different circumstances [JHY82; AL77]. To account for these uncertainties, different methods such as *fuzzy set* methods or *boundary approaches* have been developed over the years to propagate uncertainties throughout the model and determine lower and upper bounds for each failure event [SBR96]. A common

approach is to use an MCS for this task, in which each model parameter is assigned with an additional parameter, describing the assumed distribution of the model parameter. Common distributions to describe the uncertainty of such parameters are the beta, the gamma and the log normal distribution. Finally, the overall variation and upper and lower bounds of the uncertainty analysis can be determined in that way [SV02, p. 91-94]. In contrast to this, tolerance uncertainties only can be treated within a sensitivity analysis. In this form of analysis parameters are changed to investigate their influence on the overall model solution. This method can be used during design phase of a system to investigate the behavior on operational changes.

2.3. Prognostics and Health Management (PHM)

Prognostics and Health Management is a novel engineering discipline dedicated to an automated health assessment of technical systems. In contrast to conventional methods, such as PoF or classical Reliability Engineering, PHM estimates the current health state of an individual system from onboard sensor data. Based on that concept the current as well as the future system's health state is predicted. In close analogy to medicine, this is also referred to as *diagnosis* and *prognosis*. While diagnosis is carried out to identify the current level of degradation, it is followed by the prognosis, in which the further evolution of the degradation is predicted until a predefined failure threshold is exceeded. In this way, the *remaining useful lifetime* (RUL) of the monitored component is predicted. PHM itself is a multifaceted approach including elements from various domains such as machine learning, failure modelling as well as statistics and has been fostered by new developments in sensor technology, computational models and increasing processing power over the last years [KAC17, p. 1]. PHM is considered as one of the key-enablers for condition based maintenance as well as performance-based contracting and can be used as a decision support for operators and maintainers [JS11; Hes02]. It is challenged in various industries, especially in those working with costly assets and those where high availability and safety requirements exist. Further, this technique plays a crucial role for autonomous systems, such as drones, which operate over long periods in the absence of any personnel and thus need to self-asses their current health state and remaining capabilities in order to provide reliable services [Glo+10]. Accordingly, PHM is an emerging discipline and industry driver which targets at many aspects of performability. However, it shall be noted that PHM is still at an early stage and in the focus of active research among various fields.

2.3.1. Scope and benefits

The scope and benefits of PHM are based on the accurate prediction of the remaining useful lifetime of monitored systems and components. Via processing of sensor data acquired from the monitored system, PHM is sensitive to account for any life-time affecting factors such as individual loads, usage patterns or material and production variances. This makes the PHM approach superior to conventional ones, which are not capable to account for individual usage and degradation as being sensorless. This advantage is illustrated in Figure 2.6, where a traditional failure distribution, given as PDF, is compared to the predicted failure distribution by a PHM system. Conventional failure distributions are derived from historic failure data and tend to cover a broader timespan, especially if different usage patterns are present and failure events are of great variance. Statistical measures such as the mean

(in case of a normal distribution) or quantiles are then used to specify a threshold of operation time after which the component needs to be replaced. Depending on the severity and risk acceptance this threshold is selected more conservatively or not, leading to early or late replacement times. While an early replacement prevents unexpected failure and thus improves the safety and overall reliability of a system, it also is more expensive as components needs to be replaced more often, leading to increased downtimes and material effort. By estimating RULs based on the current and past health states, PHM in contrast is able to generate more accurate failure distributions as the historical usage and loads are considered indirectly. Further, these predictions gain in accuracy as more and more data becomes available for later predictions. While the conventional reliability measures remain static. PHM is thus able to adjust its estimations and therefore has the ability to cover large variability in the usage of the monitored system [Eng+00].

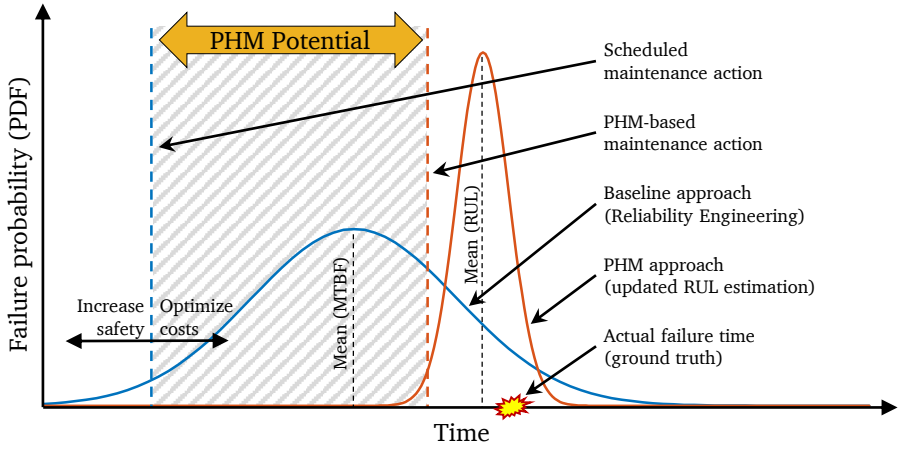


Figure 2.6.: Expected PHM potential compared to conventional reliability methods, based on [Eng+00]

PHM offers a lot of optimization potential by aligning a required maintenance action closer to the true failure event, which allows a better planning and can thus be used to increase the safety and availability of the system. Compared to the traditional statistics based reliability approach, this potential is therefore especially interesting for systems where historical failure distributions are wide-spread or where high safety and reliability requirements exists.

From the above it can be concluded that the central aspect of PHM lies in the accurate prediction of failure times. With this information at hand, operators are supported in their decision-making. Thus, PHM is also a decision support tool at a larger scope, which also explains its importance to the maintenance process and the business models such as PBC.

2.3.2. General methodology (OSA-CBM process)

The general methodology of PHM is often described with the Open System Architecture for Condition-Based Maintenance (OSA-CBM) process, as given in [NP09]. This standard was developed by industry experts to align and harmonize the implementation of condition-based maintenance systems across various industries. The process describes six[2] consecutive steps, which are run through in order to transform initial sensor data into valuable information and describes all activities, which are carried out by a PHM application. The process reaches from the raw data acquisition to the advisory generation and is shown in Figure 2.7. However, it shall be noted that the process itself does not specify any hardware bindings and thus might be implemented in various forms and distributed over different soft- and hardware solutions.

Figure 2.7.: The six steps of the Open System Architecture for Condition-Based Maintenance (OSA-CBM) process

The OSA-CBM process starts with the *Data Acquisition* (DA) step, in which raw sensor data from the monitored system is captured. This can include general sensor data such as vibrations or electrical current signals as well as environmental parameters e.g. ambient temperatures or the humidity. For existing systems usually already installed sensors are used for this, while for new developments an initial failure modes and effects and criticality analysis (FMECA) can be utilized to identify degradation mechanisms and select appropriate sensors for an additional installation. Once data is available, this data is processed during the *Data Manipulation* (DM) step. The aim of this step is to largely reduce the amount of available data and to extract meaningful features, which can be used an indicator of the current health state. Examples of such features include (vibration)

[2] Sometimes the OSA-CBM process is given with seven steps, as the last step, the advisory generation, can be further split into a decision support and presentation step. However, in this thesis the six-step process is used.

frequency signatures, statistical moments of the time-signal or quantifiable measures such a size and amount of debris. In the following *State Detection* (SD) step these features are then evaluated to detect if the system is in a normal operating state or has already started to degrade. In case of a present degradation, the *Health Assessment* (HA) step is carried out, in which the degradation is quantified, and the problem is identified. Based on the current and past health states, a RUL will be estimated during the *Prognostics Assessment* (PA) step, before the final *Advisory Generation* (AG) will be generated in the last step.

2.3.3. Prognosis algorithms and modelling approaches

The following subsection is intended to give a brief overview of commonly used algorithms and modelling approaches, with a special focus on the prognosis step as this is especially relevant for the remainder of this thesis.

Over the past two decades PHM has been in the focus of research, with a special interest dedicated to the development and evaluation of methods and algorithms for the health and prognostics assessment. Examples of this development can be found in [Lee+14], where a detailed overview of health monitoring methods for rotary machinery is given, or in [SHM11], where the current state of algorithm development is presented for different industries. In addition to this, lot of research has been conducted for predicting the lifetimes of specific components. This includes the prediction of bearing failures [Ang18], motors and fans [OSP12; Bro+09], batteries [DK13] or pneumatic components [DG11]. To accomplish a precise degradation and RUL prediction, different approaches have been developed so far, which can be categorized into three different groups, namely *physics-based*, *data-driven* and *hybrid models* [KAC17, p. 10].

Physics-based prediction models are based on the physical description of the given system. This usually involves differential and physical equations describing the system's behavior and is often implemented in a state-space representation. By adjusting and propagating model parameters such as friction or efficiency coefficients the further degradation path and system state can be described. Model-based prognostics have been successfully applied to well-understood systems such as batteries or hydraulics. These approaches are, if well implemented, very accurate and come with the benefit of describing the future behavior of the system, which can be used to extract additional information about future performance or weakness of the system. However, these models are usually very system specific and less generalizable. They must be developed by system experts and require a profound a-priory understanding of the system's degradation mechanisms and ways to model them. This

often makes it more difficult to apply physics based models to larger or less understood new systems [DG11].

In contrast to this, *data-driven* algorithms are developed using available historic degradation data from the system of interest. Algorithms from the machine learning domain such as *support vector machines* (SVMs), *Gaussian Processes* (GP) or *artificial neural networks* (ANN), are used to learn degradation patterns and identify correlations between different features, which are trained into prediction models. In this way, a mathematical degradation model of the system is learned from the given training data. This makes the approach very generic and adaptable as no deep system understanding is required. It is also suited for large and complex systems, where no detailed information about the degradation mechanisms are known and can be continuously improved by retraining the model. On the other hand, this approach also has its drawback. As it largely depends on available degradation data, this approach is only suited for systems, where a large amount of data is available. Thus, for new designs or systems where only few quantities are build, the data-driven approach is often impractical [Sch05].

Finally, there exist *hybrid models*, which combine a physics-based model with data-driven approaches. This strategy is often found for systems where physical models are only partially known, and few data is available. Here the disadvantages of both approaches are compensated by combining them [LK14].

2.3.4. Diagnostics and prognostics fundamentals (notation)

A crucial step for every PHM application is the diagnosis and prognosis step, which are briefly described here. The aim of the diagnosis is a quantification of the current degradation level of the monitored system. The degradation level, denoted as x, is defined based on one or more features constructed from the available sensor data. Thus, x remains an estimation to the true, unknown, health state x of the system. Due to the sampling and processing of x, it becomes as a discrete time series, where x_k is the estimated degradation level for time instance k. As x_k is not given as general physical unit it is usually given in percentage of a predefined failure threshold x_{th}. As soon as multiple observations $x_{k,obs}$ become available, a prognosis can be performed. The beginning of the prognosis is denoted with time instance t_P. From this point on, the observed degradation trajectory is extrapolated until a predefined failure threshold is reached. This task is accomplished with trained or expert models of various kinds as presented in the previous section. The time instance when the prediction $x_{k,pred}$ hits the failure threshold is then defined as the *end of prediction* t_{EoP}. The remaining useful lifetime can then be estimated as $r\,(k) = t_{EoP} - t_P$.

Analogue to this estimation, also the true remaining useful lifetime can be stated, based on the true degradation trajectory x_k^* and the true failure time, denoted as *end of life* (EoL), t_{EoL}. The true RUL is then defined as $r^*(k) = t_{EoL} - t_P$. Although the true health state and remaining useful lifetime are obviously unknown during the prognosis itself, it makes sense to define them, as they build the basis for a later performance evaluation of a given prognosis algorithm. More details on such performance metrics will be given in Section 2.3.5.

As with any predictions, also RUL estimations are subject to uncertainties, which need to be considered thoroughly. These uncertainties result from various sources such as sensor and process noise and need to be propagated during the prediction of the degradation trajectory to allow a reliable interpretation of the prognosis results. A well-defined prognosis algorithm is expected to provide some information about the trust given into the result. This is usually done by providing confidence intervals CI for each $x_{k,pred}$, stating the probability that the true degradation falls into this range. As these intervals are also propagated till the failure threshold, an overall estimated failure distribution F can be derived. Based on this distribution and the allowable risk tolerance, suitable end of use times can be defined by the operator (see Figure 2.6). The remainder of this work will use the above presented notation and termini, which is largely based on [KAC17; SCS09]. To provide an example of the introduced definitions, an exemplary prognosis based on artificial degradation data is shown in Figure 2.8.

2.3.5. Algorithm performance metrics

For a proper evaluation and to further allow a comparison between algorithms, performance metrics play a crucial role during the development and design phase of any prognosis algorithm as these metrics build the foundation to specify requirements and run benchmarks. Over the years a common set of standardized metrics have evolved. As given in [Sax+08], these metrics are categorized into the three following groups:

Cost-benefit
Although often neglected in research, the economical consideration of any PHM algorithms is of high importance for the industry. Thus, metrics of this group are dedicated to assess the overall cost-benefits associated with the installation of a prognostic algorithm. Metrics of this group include a specification of the return of investment (ROI), the $MTBF/MTBUR$[3] ratio or the overall life-cycle costs (acquisition plus operating costs).

[3]MTBUR – Mean time between unit replacement; MTBF Mean time between failure

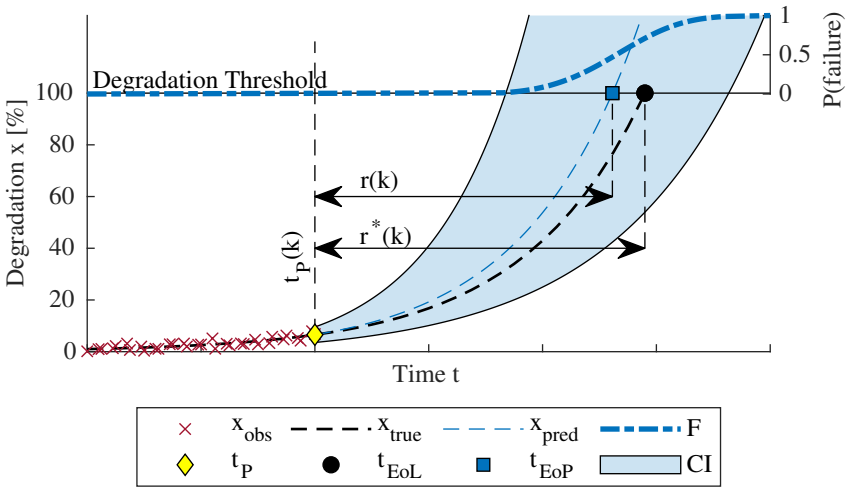


Figure 2.8.: Exemplary prognosis output

Computational performance
As PHM algorithms have to work with data, their technical implementation can become resource intense especially when the number of sensors or the sampling rate is increased. Algorithms such as the Fast Fourier Transformation (FFT), used for the extraction of features from the frequency domain, particle filters, used for filtering, or common machine learning algorithms such as neural networks or genetic programming, used to learn degradation trajectories, become computational intense when the problem size (amount of data) is increased. Methods to quantify on the computational performance and demands are usually found within the discipline of computer science. A common approach to state the computational complexity of algorithms is the *Big-O* notation [RRH00], in which the number of calculation steps is correlated to the input size of the problem. Other methods include the assessment of the overall used CPU time or required memory occupied by the algorithm. These metrics play especially a crucial role when the algorithm is to be deployed to a target hardware.

Algorithm performance
These metrics are dedicated to state and evaluate the performance of a given algorithm. This includes an assessment of prediction accuracy and precision as well as statements

about the overall robustness of a given algorithm as well as a measure of the overall trajectory performance. Being relevant for the remainder of this work, these metrics will be further outlined in the following with a focus on accuracy and precision metrics. The performance evaluation of PHM algorithms is based on the RUL error for a given prediction at time instance $t_P = k$. In case that the true RUL $r_l^*(k)$ is given, the RUL error $\Delta_l(k)$ of a given prediction of component l is given as $\Delta_l(k) = r_l^*(k) - r_l(k)$. Based on this definition, several metrics are derived. To allow a better comparison independent of time, the time instance k can be normalized to the true end of life by using the following substitution $\Lambda = \frac{k}{EoL}$. Common accuracy performance metrics include the false positive (FP) and false negative (FN) metric, the mean absolute percentage error (MAPE) as well as the mean squared error (MSE) and the mean absolute error (MAE). The FP and FN metric are both logical performance metrics, which state whether a prediction was unacceptably early (FP) or late (FN). As a reference two acceptance thresholds t_{FP} and t_{FN} are used which must not be surpassed by the prediction error. While the FP measure is more related to a cost-effective maintenance (prevention of unnecessary early replacements), the FN measure aims at avoiding unexpected failures. The MAPE, MSE and MAE instead are used to quantify the prediction error and build mean values of the relative, squared and absolute error. It is important to highlight, that the mean values can consider multiple components of same type ($l = 1 \dots n$) at the same $\Lambda = const$ or can be used unit-wise ($l = const$) over all available prediction times $\Lambda \in [0, 1]$. Common precision related performance metrics include the sample standard deviation (S) and the mean absolute deviation from the sample median (MAD) metric. Both measures are used to quantify the overall deviation of the error with respect to the mean of the error. While the S metric is limited to a normal distribution of RUL errors, the MAD can be applied to an arbitrary error distribution. A comprehensive overview of the algorithm performance metrics is given in the appendix of this work in A.1 together with the value ranges.

2.3.6. PHM for systems

The insights given in sections 2.3.1 to 2.3.5 give a general overview of the PHM methodology, the used approaches and models. Being still at an early phase, PHM is in focus of active research. Over the last two decades many papers and results have been published by the research community, fostering the evolution of PHM. However, most of this research is dedicated to the monitoring of distinct components, such as motors, batteries, pumps or bearings. On the other hand as stated in [Rod17], the application of PHM methods on a system-wide scale is of much more relevance for the industry, especially when it comes to costly complex systems, where the interaction of multiple subsystems leads to differing

performance states. As this is one of the main goals of this thesis, this section is dedicated to the current state of research on this special sub-topic.

Considering PHM for multi-component systems, one has to distinguish between two fundamental but different problems: A common challenge for operators of complex systems is to find a suitable maintenance policy, which allows an efficient execution of maintenance actions at an optimum instance of time. Thus, authors such as [RPS15] or [vP13], have investigated methods that consider the current health state of monitored components to generate a dynamic maintenance planning support. While this obviously addresses safety and economic goals, it does not aggregate the health and performance capabilities of an individual system. Thus, the second problem one could be interested to solve is the assessment of the overall system's capabilities under consideration of the current component's health states. This is especially interesting for those systems with built-in redundancies. In this regard multiple papers have been published in the last years, which are discussed further in the following.

In [ST07] for example a condition-based fault tree analysis is proposed. The authors suggest utilizing condition-monitoring data to update the failure rates of a fault tree according to the detected deterioration stage. As soon as a new deterioration stage is detected the fault tree is recalculated leading to a dynamic update of the model. In the following work [Sha+16] this concept was further extended to support reliability, availability, maintainability and safety (RAMS) predictions. By using a Monte Carlo Simulation this extension is also able to account for various failure distributions. Another approach is presented by [DBR12], who utilize a model-based system state-space representation in which the authors use a structural model decomposition technique to decompose a larger system model into a set of smaller independent sub-models. After the decomposition, each subsystem can be predicted on its own. However, this concept assumes that the component's degradation is not interfered by other components. In the papers [LZ16] and [LLZ15], a Piece-wise Deterministic Markov Process (PDMP) is utilized to model and predict the degradation of a monitored system. Therefore, it is assumed that the system consists of components, which can be modelled by physical models (deterministic part) as well as components modeled as multistate models with an underlying stochastic process (random state transitions). Both model types together form the PDMP. As the transition rate of the multistate models depend on the physical degradation level and the parameters of the latter depend on the current states of the multistate models, the different component's degradation becomes strongly coupled. Thus, the model is capable of considered degradation interactions between components. To make the model dynamic it is continuously updated with monitored data, which is included via Bayesian filtering. In the end, existing uncertainties are propagated via a Monte Carlo Simulation or a Fuzzy

Set scheme to derive the distribution of a system RUL (S-RUL). Another state-space model based method is presented in [KBS16], where the authors use a regular state-space system model with included degradation models. These degradation models are a function of the internal system states as well as of other existing degradation parameters in the system. Thus, the internal degradation progress is strongly coupled over the system's components. A structural decomposition of the system model is not foreseen. The RUL assessment process is two-fold: First the current system state as well as the degradation model's parameters are estimated via Bayesian filtering. Afterwards the degradation is estimated by the propagation of the system's state equations until a predefined performance threshold is reached. For this propagation two algorithms are proposed: a) a particle filter with sampling importance resampling (SIR) and b) an inverse first order reliability model (FORM) algorithm, which is more efficient than a) and therefore recommended. In this way the S-RUL is derived and uncertainties about the used parameters can be considered. Another form of S-RUL generation is described in [Rod17], where the author proposes a system architecture function, which aggregates the individual component's failure probabilities into one system-wide S-RUL. Finally, in [TNM18] another model is presented, where the prediction of an S-RUL is based on an inoperability input-output model (IIM). This model assumes that degradation models are given for each component and that the linear degradation interactions are a-priori known. Based on a recursive formula the inoperability[4] of the system is thus propagated. Further, by introducing an influence factor also external factors such as environmental changes can be accounted for. However, both factors (influence factors and degradation interaction parameters) remain constant during the simulation.

In Table 2.2, the relevant papers for this thesis are summarized and the main characteristics of the proposed methods are outlined. Empty cells in this table state that this feature is not explicit stated in the papers.

2.4. Conclusions and identified research gap

Within the previous sections, the general state of the art of availability assessments for complex systems has been outlined. It has been shown, how the performability of a complex system can be described and why every technical system suffers from degradation and wearout. Further, the relevance of an accurate health and availability estimation for

[4]The inoperability is defined as the ratio between the performance loss compared to the initial flawless performance state.

Table 2.2.: Reviewed literature dedicated to the aggregation of PHM data on system level for assessing the overall system's capabilities

Approach	Output	Inter-dependencies	External factors	Uncertainty propagation	Publication year
Condition-based Fault Tree Analysis (CBFTA) [ST07]	MTBF	×			2007
Condition-based RAMS [Sha+16]	MTBF	×			2016
Structural model decomposition [DBR12]	S-RUL	×		×	2012
Piecewise-deterministic Markov Process (PDMP) [LZ16] and [LLZ15]	S-RUL (PDF)	✓	✓	✓	2015/16
State-space model + inverse-FORM/stochastic simulation [KBS16]	S-RUL (CDF)			✓	2016
Aggregation via system architecture function [Rod17]	S-RUL			×	2017
Inoperability input-output model (IIM) [TNM18]	S-RUL	✓	✓	×	2018

new business models such as performance based contracting as well as its importance in maintenance and the decision-making process for operators was presented.

As the state of the art to conduct reliability and availability estimations Reliability Engineering was identified and presented. As a main disadvantage of this well established discipline, its approach based on the statistical evaluation of failure events was examined, which easily becomes inaccurate especially for complex systems, as these systems are often used in very versatile usage scenarios with varying lifespan-impacting external influences. Although the models being applied have become more sophisticated over time to cope with these challenges, they still lack direct feedback from the system itself. Accordingly, PHM as a novel alternative has been introduced. By making predictions based on sensor feedback,

PHM is intended to overcome the limitations of the rather static Reliability Engineering approach. PHM as well as its models and methodology have matured during recent decades, while still being an active field of research. However, it was also shown, that most research and solutions are still focussing on the monitoring and prediction of single items, rather than systems combined of multiple components. Thus, as a remaining research gap in the field of prognostics, the lack of an aggregation methodology that can be used to assess the overall capabilities of multi-component system was identified. The reviewed literature on this topic mainly expands the concept of an RUL prediction to that of a system RUL, where multiple component's prognoses are aggregated by a user-defined system function, which states the overall health status of the entire system. This concept however is considered as unpractical as it cannot distinguish between multiple system states, which allow the operator to gain a detailed preview on the future expected capabilities of the monitored system. Further, it is often not clear how to include external factors such as environmental stressors or how to consider parts that are not monitored.

With this said, the overall problem to be solved within this thesis is defined as follows: The objective of this work is to develop a **decision support method to increase the system availability based on prognostics**. The idea behind this is to aggregate multiple prognostics results obtained from different monitored components to estimate the current and future performance level of a given system. The resulting method is intended as a decision support for operators of complex systems. In contrast to the existing aggregation methods from Table 2.2, an emphasis is given in this work to the capability of differentiating various performance levels and in this way supporting the concept of performability. Overall, the scope of this work can be subdivided into the following three research objectives.

O1 – Feasibility: Show and proof the technical feasibility to aggregate multiple PHM results on system level to allow a system-wide performance assessment with distinguishable performance levels.

O2 – Integration: Present a general methodology which is aligned to existing reliability methods.

O3 – Performance: Identify the advantages that can be achieved with the proposed method. What are the merits and limitations, and what are the future challenges in this regard?

3. Conception of the proposed aggregation method

Based on the previously outlined insights and identified research gap, this chapter is dedicated to the conception and development of a method, which allows the aggregation of dynamic PHM results from multiple monitored components up to system-level. The envisaged method is aligned at the very end of the OSA-CBM process as introduced in Section 2.3.2 and intended as a decision support tool for operators of complex systems with the overall objective of generating an accurate and precise system state estimation under the consideration of prognostic results. The development process of the method is based on the *V-model methodology* [VDI21] and includes the following steps:

Problem statement: Definition of the problem to solve.

Requirements definition: Based on the identified problem in Section 2.4, general requirements as well as constraints of the envisaged method are defined. These requirements build the foundation for the later validation of the method.

Method development: This step describes the conceptual phase in this work, in which the method itself is developed. In this part, a general solution is defined, which is independent of any implementation (software) and free from any specific use-case.

Implementation: During the implementation phase, a general framework in form of a (prototyped) software is created. This solution implements the method as defined before and is used to conduct the required experiments to test and evaluate the proposed method.

Verification: The verification step describes a formal check, that the software solution is correctly implemented and works as intended.

Validation: In this step, a cross-check against the initially specified requirements is carried out. The goal of this phase is to ensure that the defined requirements are met and that the overall solution solves the addressed problem.

Evaluation: The evolution is the final phase of the development process. This part will be described in Chapter 5 based on the defined use-case from Chapter 4. The goal of the evaluation is to measure the performance of the developed solution and identify its merits and possible limitations.

The overall development process of the foreseen method is illustrated in Figure 3.1.

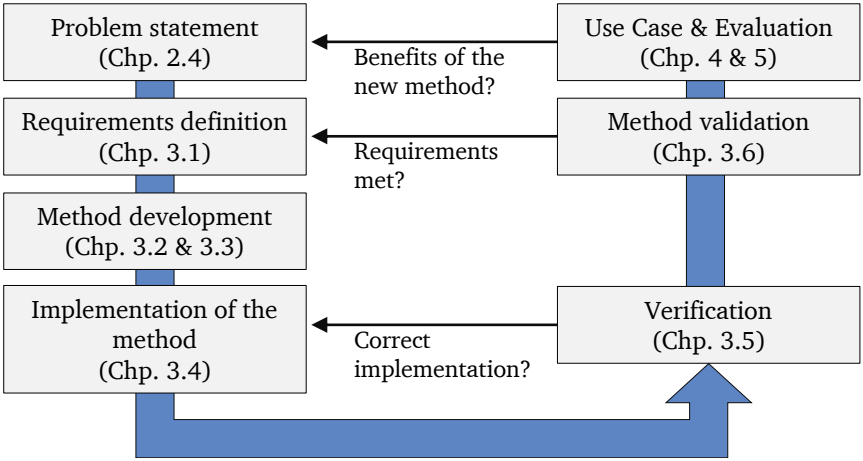


Figure 3.1.: Development process of the proposed method based on the V-model, based on [VDI21]

3.1. Definition of requirements and constraints

The overall objective of this method is to aggregate prognostics information from different monitored components on system-level. The method is intended to support operators of complex systems in their decision-making, based on an accurate and up-to-date estimation of the current and future system health state. The method itself shall be generic and not system specific. In contrast to existing methods, as introduced in Section 2.3.6, the method shall be able to distinguish between different performance levels, which can arise due to the complexity and redundancy of a given system. As outlined in Section 2.2.4, the proper treatment of uncertainties is important when making assumptions about future events. Thus, such a decision support is only viable when associated risks as well as uncertainties

of the derived estimations can be stated. Therefore, the proposed method shall be able to handle uncertainties derived from the used algorithms and sensors thoroughly. As given in Section 2.2, the discipline of Reliability Engineering has been around for many decades now. Many industry standards as well as safety and reliability processes are established and based on this existing framework. In this context, it is reasonable to require that the developed method shall rather extend the existing methods than replace them. Therefore, a further requirement is that the considered method shall align and connect with this exiting framework and the established reliability tools.

Besides the above stated requirements, also some assumptions and simplifications are made at this point. As the method is intended for the application with complex degrading systems, it is assumed that the monitored system consists of at least one component which suffers from a continuous and predictable degradation. It is further assumed that a PHM-system is installed, which implements the first four steps of the OSA-CBM process as described in Section 2.3.2 and is capable of producing RUL predictions on a frequent basis. Ideally the installed prognostic algorithms provide their estimations together with uncertainty bounds. For those components not monitored by the PHM-system, it is assumed that the correlating failure rates are known or can be estimated by means of conventional reliability measures. As a further constraint, it is presumed that the degradation of a component does not affect the degradation rate of another component. This is of course a hard constraint as there exists many systems where exactly this is the case. However, as the focus of this work is on the aggregation method this is neglected at this point.

Finally, the developed method shall be limited to a decision support functionality only. It is intended to visually inform an operator about the current and future system state of the considered system. Thus, it does not implement an automatization logic, which acts on the given information. This is argued as a fully autonomous system would require specific and additional information about the system and its mission goals, which stands in direct contrast to the requirement for a generic solution. The derived requirements, constraints and limitations are summarized in Table 3.1.

3.2. Development of the decision support method

In the following section, a prognostics-based decision support method is developed, which is tailored down to the specified requirements given in Table 3.1. The key concept of the proposed method is to extend existing reliability methods to integrate recurring and up-to-date PHM data. As outlined in Chapter 2, Reliability Engineering is well established

Table 3.1.: Derived requirements, constraints and limitations for the proposed method

Category	ID	Definition
General Requirements	R1	The method must be able to aggregate prognostic data on system-level
	R2	The method shall provide a decision support for operators with a focus on the future performance availability of a given system
	R3	The method shall be generic and not system specific
	R4	The method shall be able to consider different performance levels (due to redundancy)
	R5	The method shall be able to consider any uncertainties which arise from the used PHM algorithms
	R6	The method shall align to existing reliability assessment methods and should (ideally) extend them rather than to formulate a complete different approach
Constrains & Assumptions	C1	The considered system (or parts of it) degrades in a continuous manner and thus is monitorable by PHM
	C2	The monitored system is considered being complex having multiple components and distinguishable performances levels
	C3	The RULs of selected components are estimated by means of a PHM algorithm (see OSA-CBM steps 1 to 4) and (ideally) provided with a measure of uncertainty
	C4	The degradation rate of a component is not interfered by the degradation of other components of the same system
Limitations / non-requirements	L1	The method does not implement an automation logic as it is intended as a decision support only

across many industries and covers methods for the reliability assessment of single components (failure rates and failure models) as well as methods to conduct analyses based on dependency models (fault trees, Markov models) on system-level. However, these models are largely based on static parameters, which are defined with data from historic failure events or derived from experiences with similar systems. They provide good estimates for systems used in similar environments and under constant loads, but easily become

inaccurate if systems are used in versatile scenarios, experiencing different loads and stresses, as these statistic based methods become inaccurate as the individual degradation cannot be captured. PHM on the other hand is capable of assessing and predicting the health state of an individual system based on sensor signals. This makes any life-time predictions much more accurate as variations in the degradation process are captured. On the other hand, the PHM discipline is still in an early development stage with only few commercialized solutions. Most monitoring solutions and algorithms focus on the assessment of single parts and do not consider complex systems. Based on this and the identified research gap (see Section 2.4), the proposed method is intended to combine both approaches into an integrated solution, which is capable to overcome the limitations of both disciplines. This key concept is illustrated in Figure 3.2.

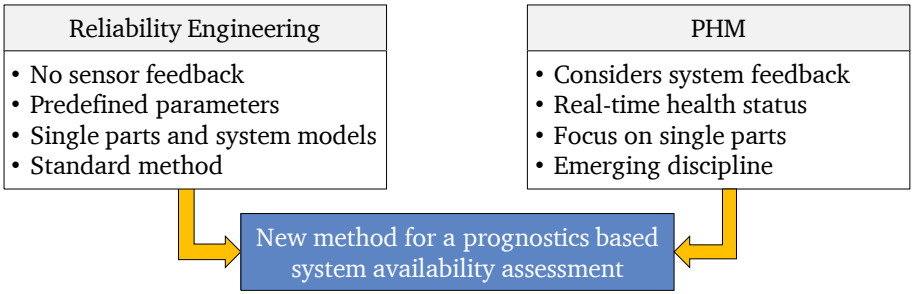


Figure 3.2.: Development of a new method based on the combination of Reliability Engineering and PHM

3.2.1. Used taxonomy

Throughout this work different levels of hardware integration are described and modelled. In order to define a common definition, the following taxonomy is used, which is in close analogy to industry standards SAE ARP4761 and MIL-HDBK-217F and distinguishes between three levels of integration, namely *parts*, *components* and *systems* [Dep91; SAE96].

Parts, in literature sometimes also denoted as items, are defined as the smallest physical entities for which a reliability can be stated. For many parts, empirical failure rates have been identified over the years, which are summarized in handbooks and databases such as the MIL-HDBK-217F (see Table 2.1). Examples are small motors, gearboxes or electronic parts such as capacitors, ICs or MOSFETs. Depending on their complexity, size

and degradation mechanism, parts might be qualified to be monitored by PHM or not. *Components* define the next higher level of integration and form a physical assembly of one or more parts. Components perform a distinct function and are usually self-contained in a way that they can be replaced in the advent of a failure. This is sometimes also referred to as Line Replaceable Unit (LRU). Further, it is assumed in this work that components can be described by two states, either working or failed. Examples of components include actuators, pumps or motor-assemblies. The highest level of integration is found in a *(sub-) system*, where multiple components are interconnected to accomplish a greater task. Depending on the arrangement of components, e.g. parallel or in line, a system can have multiple states. These states depend on the failure combinations and will result in different system performance levels. An example for a system could be a hydraulic actuation system, which is composed of components such as a pump, a tank and multiple valves and actuators used to deliver hydraulic energy.

The resulting taxonomy is summarized in Table 3.2 and will be used throughout this thesis. For the sake of completeness, it should be noted at this point that the term *system* is often further subdivided within systems engineering. Usually, a distinction is made between *sub-systems* (e.g. a flight control system), *system* (e.g. an aircraft) and *system-of-systems* (e.g. an entire aircraft fleet including required infrastructure). However, for better readability the general term *system* is used throughout this paper, even if most examples discussed in the following are strictly speaking *sub-systems*.

Table 3.2.: Used taxonomy in this thesis

Term	Definition
Part	Smallest physical entity for which a reliability can be stated.
Component	Any self-contained part or combination of parts, which performs a distinct function necessary to operate the system. A component has two states, either working or failed.
(Sub-)System	A combination of n inter-related components arranged to perform a specific function. A system can have up to 2^n failure states, depending on the combination of component failures.

3.2.2. Aggregation concept

A central aspect of the proposed decision support method is the aggregation of prognostics data from various monitored parts up to system-level in order to obtain the future

performance and availability levels. Therefore, it is assumed that recurring and up-to-date PHM data are available from an installed PHM-system. Further, it is assumed that additional environmental data are available from on-board sensors, which can be used to specify the current usage scenario and assess external factors such as temperatures or the humidity. Besides these time-dependent values it is presumed that there is some additional information about the system itself and possible failure relevant parameters, such as failure distributions and rates for distinct parts. This information is related to as *expert knowledge* as it reflects a sufficient deep system understanding. The overall aggregation concept and its in- and outputs are illustrated in Figure 3.3. The different inputs are highlighted and divided into dynamic and static (expert knowledge) data. The output of the proposed method is a precise system state prediction, which forms the basis for the later decision support. The aggregation of the information of all the three levels part, component and system is described in the following subsections in more detail, before the overall aggregation method is outlined at the end of this section.

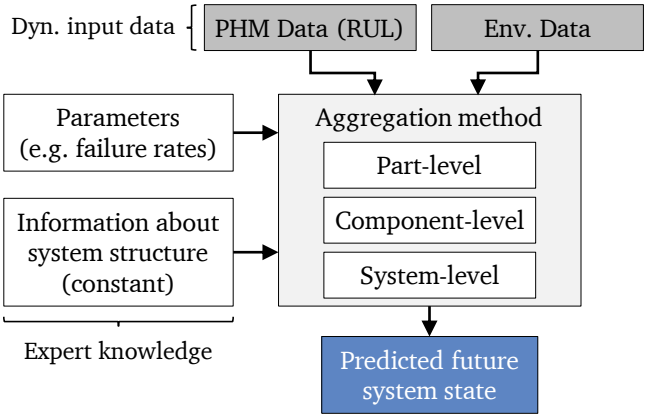


Figure 3.3.: Overall aggregation concept

3.2.3. Part-level description

As given in Section 3.2.1 parts are considered as the smallest entities in a system that can fail. Thus, parts form the first level to be considered when aggregating failure and prognostics data. The failure behavior of parts can be broadly split into two different

groups. The first group includes parts, which fail in an abrupt and spontaneous manner (constant part of the bathtub curve, see Figure 2.3). This failure behavior is often found with electronic parts, which suddenly fail without any measurable precursors. As it is difficult to assess any feature-rich sensor data from these parts, a statistical evaluation of failure events is usually used to give life-time predictions. The other group of parts includes those, which degrade in a rather slow and continuous way with measurable failure indicators (increasing part of the bathtub curve, see Figure 2.3). Examples of this group include motors or gearboxes, which start to heat up, lose their efficiency or start to vibrate with an increased degradation. For those parts, the PHM approach is interesting as indicators can be measured to develop precise lifetime prediction models based on sensor data. It is proposed in this work to use conventional reliability methods for those components, where monitoring is technically or economically not feasible. However, to be still able to consider dynamic effects like changing temperatures or humidity it is proposed to integrate environmental factors that can be measured into this calculation and update them periodically. All other components, which are qualified to be monitored by means of a PHM-system, shall be considered by incorporating lifetime predictions based on the available sensor data. The mathematical representation within the proposed aggregation method is outlined in the following for the spontaneous failure type and the PHM approach. It shall be noted that both representation forms lead to a defined failure distribution in form of a CDF.

Spontaneous failure type
Parts characterized by a spontaneous failure behavior, or where no prognostics data are available, are modelled with conventional reliability methods. The general approach here is to use failure distributions such as the exponential or Weibull distribution as introduced in Section 2.2. For a given part l, for which failure events are considered to be exponentially distributed, the cumulative distribution function is thus defined as:

$$F_l\left(t\right) = 1 - e^{-\lambda_l t} \tag{3.9}$$

To account for changing environments and usage conditions it is proposed to further adjust the failure rate λ_l. As given in [Dep91] failure rates, especially those for electronic equipment, are predicted with part failure rate models. These models usually use a part specific base failure rate λ_b, which is then adjusted with factors π to meet the expected usage conditions, also referred to as part stresses [FFP17]. The base failure rate is adjusted with various factors, which will affect the reliability of the given part. These factors include the stress factor π_S, the construction factor π_C, the quality factor π_Q, a temperature factor π_T as well as an environment factor π_E, which will differentiate between different usage

environments such as on ground fixed (GF), airborne inhabited cargo (AIC) or airborne with rotary wings (ARW). The part failure rate models therefore account for various different usage conditions, which must be precisely stated during the design phase of a system. Some enumerated factors like the quality or construction factor can be stated precisely in the design phase of a system and remain constant as long as parts are not redesigned and replaced. Other factors like the stress or temperature level however might vary along individual parts and are largely affected by the true usage scenario of the part. To illustrate this, in Figure 3.4 the failure rate model for a CMOS digital gate array is estimated based on an exemplary calculation given in [Dep91]. The predicted failure rate λ_l for this part is calculated for different case temperatures as well as for three different environments (ground fixed, airborne and airborne rotary wing). It can be seen that these two factors have a significant influence on the estimated part reliability.

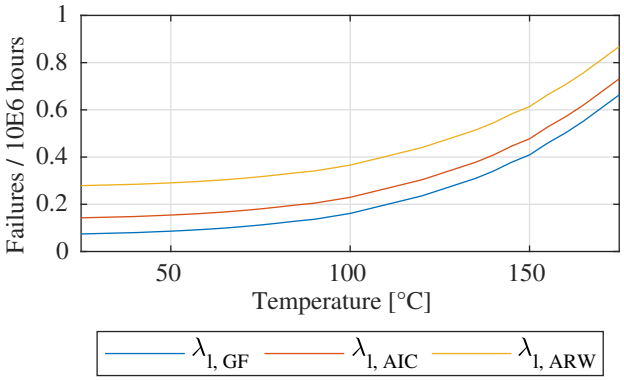


Figure 3.4.: Failure rate variations for different temperature settings and environments, based on [Dep91]

Based on the above insights, it is proposed to consider additional information such as external temperatures or the environment to adjust the failure rates according to the expected mission profile. The part failure rate λ_l as used in Equation 3.9 is thus modified by additional factors as given below:

$$\lambda_l = \lambda_b \pi_T \pi_S \pi_C \pi_Q \pi_E \tag{3.10}$$

Degrading failure type (RUL-CDF)
For those parts that will degrade over time and can be monitored by means of a PHM-system, it is instead proposed to use a dynamically generated failure time CDFs, which are adjusted as soon as new prognostic data are available. This will be referred to as *remaining useful lifetime CDF* (RUL-CDF) in the remainder of this work.

The RUL-CDF is obtained from the prognosis step of the OSA-CBM process as introduced in Section 2.3.2. As a fundamental concept of prognostics, the estimation of the RUL (prognosis) is based on the observed degradation trajectory and the assessment of the current health level (diagnosis). No matter if a physical model or a data-driven approach is used to model the degradation behavior, the underlying model is then propagated forward in time to predict the future evolution of the degradation trajectory. To consider any uncertainties within the prognosis, it is widely accepted to work with distributions and perturbated parameters as initial input values. As a result, the distribution of the estimated degradation level x is given for each time step k as $p(x|k)$, instead of a single estimate. This is often achieved by filtering approaches such as Kalman or particle filtering as described in [OV09], by using statistical approaches such as the Gaussian Process or the Relevance Vector Machine [PAK14] or by simply using the training error, which can be used for neural networks [WTC13]. During the prognosis, the degradation model is then recursively calculated until all trajectories exceed a predefined failure threshold x_{th}.

Based on the above considerations, the RUL-CDF describes the probability that the true degradation level x is above the defined failure threshold x_{th} for time instance k considering the predicted failure level distribution $p_x\left(x|t\right)$ from the algorithm. The RUL-CDF is thus defined as:

$$F_l\left(t\right) = \int_{x_{th}}^{\infty} p_x\left(x|t\right) dx. \tag{3.11}$$

The probability density function for the end of life is then obtained by deriving Equation 3.11

$$p_{EoP}\left(t|x_{th}\right) = \frac{dF_l\left(t\right)}{dt} \tag{3.12}$$

or by applying the law of total probabilities as given in [OV09] for discrete time steps

$$p_{EoP}\left(k|x_{th}\right) = \sum_{i=1}^{N} F_l\left(k\right) \cdot w_i, \tag{3.13}$$

where w_i is a weighting factor chosen to normalize the PDF. The above formulation of the RUL-CDF can also be used if the end of life distribution $p\left(t|x_{th}\right)$ is given directly by the algorithm (in cases where the trajectory is not propagated). In this case the RUL-CDF is formed as

$$F_l\left(t\right) = \int_{t_P}^{t} p_{EoP}\left(t|x\right) dt. \tag{3.14}$$

In cases where the failure threshold is not distinct but specified as a distribution $p_{x_{th}}\left(x\right)$ itself, the definitions 3.11 and 3.14 can be extended to

$$F_l\left(t\right) = \int_{x_{th}}^{\infty} p_x\left(x|t\right) \cdot p_{x_{th}}\left(x\right) dx \tag{3.15}$$

and

$$F_l\left(t\right) = p_{x_{th}}\left(x\right) \cdot \int_{t_P}^{t} p_{EoP}\left(t|x\right) dt. \tag{3.16}$$

The calculation of the RUL-CDF is illustrated in Figure 3.5 for an exemplary prognosis starting at $t_P = 40$ h.

With the above definitions the RUL-CDF can be constructed from various prognosis algorithms as long as the output includes the degradation distributions (for all predictions steps) or the expected EoP distribution. Either way it is important to highlight that the RUL-CDF $F_l\left(t\right)$ is not constant and will change for updated prognoses at later t_P. This is due to the effect that most prognoses become more accurate for later predictions, as more data are available to process and the prediction horizon (time till failure) is shorter. It is therefore necessary to periodically recalculate and aggregate the RUL-CDFs for the monitored parts. In Figure 3.6, the evolution for different prognoses times, expressed with $\Lambda = t_P/(t_{EoL})$, of an RUL-CDF for exemplary data are shown. It can be seen how the distribution becomes more accurate for later predictions and converges to the true failure time.

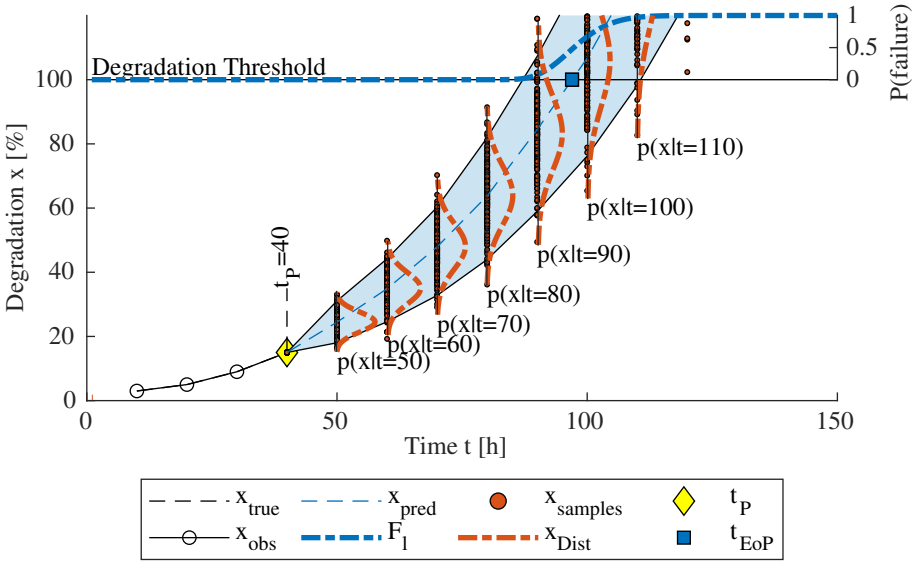


Figure 3.5.: Illustration of the RUL-CDF generation based on a prognosis

3.2.4. Component-level description

The next level of aggregation takes place on the component-level, where the failure probabilities of multiple parts are aggregated to derive the overall failure probability of the component of interest. In the PHM domain this has been tackled by introducing a system-wide RUL function, short S-RUL, in which the component performances are aggregated by a system architecture function H, which models the interactions among containing parts, as shown in Section 2.3.6. In traditional Reliability Engineering in contrast, this has been achieved by utilizing dependability models, which propagate the failure probabilities of distinct parts $F_{1:L}(t)$ through the system to derive the overall component's failure probability $F_m(t)$. Typical dependability models include Reliability Block Diagrams, Fault Trees as well as Markov Models, which were briefly outlined in Section 2.2.3.

Selection of a dependability model

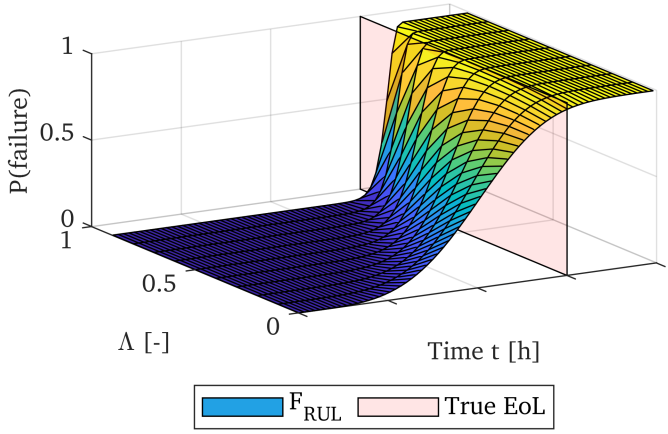


Figure 3.6.: Exemplary evolution of the RUL-CDF for increasing Λ, based on [Eng+00]

In this work it is proposed to aggregate the individual part's failure probabilities on component-level by means of a dependability model rather than an H-function. This is argued as the required input of these models will be failure probabilities of the containing parts, as defined in the previous section. In that way, the conventional failure distributions as well as the PHM provided RUL-CDFs can be considered by the model at the same time. As a further side effect the underlying dependability model can be chosen from a variety of well known and established models from the traditional Reliability Engineering domain, which directly addresses requirement R6, demanding a solution compatible to existing tools and frameworks from that discipline. In addition to this, commercial tools and software are available and for most safety critical systems the corresponding models do already exists. Nevertheless, it shall be noted at this point that speaking mathematically, dependability models can also be considered as a single aggregation function as required by the S-RUL approach introduced in Section 2.3.6 for PHM monitored systems. However, by using a dependability model, instead of an arbitrary H-function, a structured and formal way to develop the required aggregation function is given. Thus, the herein proposed method can be seen as a special case of H, which is developed by means of conventional reliability methods.

To choose an appropriate dependability model, the differences of combinatorial and state-space models are considered: Most combinatorial models only consider two states, e.g. failed or non-failed states. There are some exceptions such as dynamic fault trees (DFT), which allow an adaptive change of the single tree branches on certain events to model effects such as spare parts or cascading failures, but in the end these models quickly become very complex especially when being solved [Chi+11]. Thus, often multiple models are required to model different failure combinations. State-space models on the other hand are not limited to two states. As the name suggests, they are capable to represent many system configurations and failure combinations. However, this capability comes at a cost, as their computation easily becomes intense for an increasing number of states. As an example, for a system with n parts, in which each can fail individually, there exists 2^n failure combinations and thus states that have to be calculated. This phenomenon is often referred to as *state-space explosion*, as the number of states rapidly increases for larger systems. Although there exist many approaches to reduce the number of required states, this problem is inherent for this kind of aggregation models and cannot be neglected [JM88]. Further merits and drawbacks of both dependency model types are summarized and compared in Table 3.3.

Table 3.3.: Comparison of combinatorial and state-space models

	Combinatorial models	State-space models
Pros	• Easy implementation / development + intuitive • Graphical representation • Fast solving	• Higher modelling power (compared to combinatorial models) • Dynamic and recurring events possible (e.g. repairs) • Multiple system states (failure modes) can be modelled
Cons	• Only two states per node • No dynamics • Limited modelling power (with respect to model hierarchy)	• Computational intense (2^n for a system with n components) • Can become complex and difficult to interpret for large systems/models

Fault tree construction
Based on the above insights, the Fault Tree model is chosen to model the dependencies among the parts of a component. This dependency model is chosen as it is widely established in the industry, is simple to interpret and is less computational expensive compared to

the more complex State-space models. As introduced in Section 2.2.3 a fault tree is constructed by a set of logical AND and OR connections, which are used to model the failure dependencies among the considered parts of the component. In this way, failure probabilities are propagated throughout the model and the top-level failure probability F_m is calculated. Given as a tree structure, the fault tree can be solved in a recursive manner, by applying the following rule set starting at the top node [Bit+86]:

$$F_m(t) = F_l^n(t) \tag{3.17}$$

where $F_l^n(t)$ represents the failure probability at time t of the n-th subtree calculated as

$$F_l^n(t) = \prod F_l^{n-1}(t) \qquad \text{, for series sub-branches} \tag{3.18}$$

$$F_l^n(t) = 1 - \prod R_l^{n-1}(t) \qquad \text{, for parallel sub-branches} \tag{3.19}$$

$$F_l^n(t) = F_l(t) \qquad \text{, for basic events without further branches.} \tag{3.20}$$

In this regard $R_l^{n-1}(t)$ is the complement of $F_l^{n-1}(t)$ and thus defined as:

$$R_l^{n-1}(t) = 1 - F_l^{n-1}(t). \tag{3.21}$$

With the above formulation the individual parts' failure probabilities are aggregated for a given component m. The required input probabilities of the containing parts $F_l(t)$ are selected according to the definitions from Section 3.2.3, so that RUL-CDFs as well as common statistics based CDFs can be used. The output of the model thus is the aggregated failure probability $F_m(t)$ of the component.

3.2.5. System-level description

Based on the previous definitions of the part and component-level, the remaining system-level is modelled in this subsection. Therefore, the system and its belonging performance levels are considered with respect to the general performability of a system (see Section 1.1).

The degradation of a complex system can be considered as a stochastic process. These processes play a vital role in conventional reliability engineering where they are used to model and analyze failure free and repair times [Bir10, p. 452]. However, these processes can also be used to model the degradation itself [PP06]. A stochastic process is defined by a set of states $\boldsymbol{S}$, also known as state-space, a random variable $\zeta\,(t)$ that evolves over time, and a probability measure $\boldsymbol{P}$, which denotes the probability of a state change. With this definition the degradation of a complex system can be described as a state change from a higher performance level to one of equal or less performance (compare Section 2.1). State changes occur to a certain degree randomly as the system's components fail at individual times. This relation is illustrated in Figure 3.7, where multiple degradation trajectories of a given system are shown[1] and compared to the required performance demand. In this context, it is assumed that the performance demand also varies over time, which is often the case for complex systems e.g. due to different mission profiles etc. as it was shown in Chapter 1. With these considerations in mind, the resulting method from this work is supposed to provide predictions about the expected future system state changes, in which the exact new state, in terms of an associated performance level, as well as the expected time of transition is predicted. Thus, the stochastic process becomes in a way predictable, which allows operators more flexibility to act and optimize missions or maintenance actions as the performance mismatch between required system output and demand becomes quantifiable.

Based on these considerations, the theoretical concept for the component's failure probability aggregation on system-level is defined as follows: The performance level of each system is described with a set of finite states $\boldsymbol{S} = \{S_1, S_2, \ldots, S_m\}$, where each state represents an individual combination of failed components and thus describes a distinct performance level. Assumed that each component has exactly two states (failed or functional), a system with n components can have up to $m = 2^n$ system states. To model the existing failure combinations a binary combination matrix $\boldsymbol{B}$ is used, in which $\boldsymbol{B} = \{0, 1\}$. In $\boldsymbol{B}$, each row represents an individual system state, respectively failure combination, while the columns mark the considered components. The size of $\boldsymbol{B}$ therefore is always determined by the number of components in the system with $size(\boldsymbol{B}) = 2^n \times n$. Finally, a failed

[1] In the illustration possible repairs of the system are neglected.

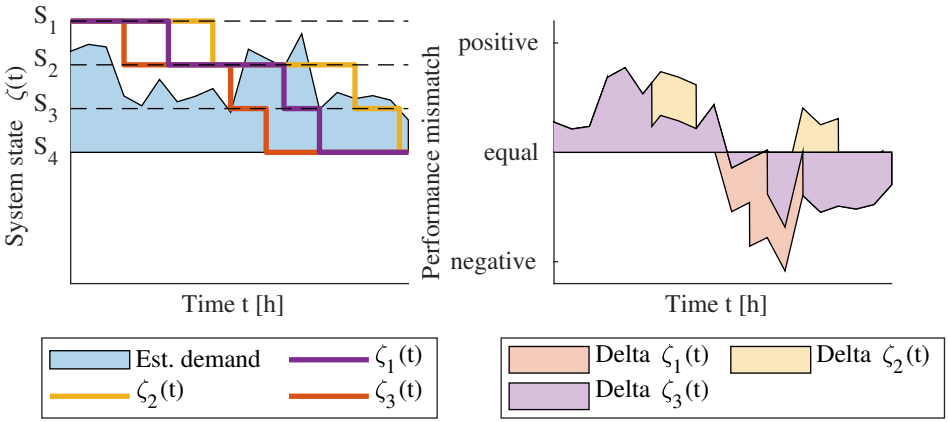


Figure 3.7.: Performance degradation of complex system described as stochastic process without repairs (states are sorted according to their performance)

component is characterized by a 1, while a functional component is modelled by a 0. Thus, the combination matrix for an exemplary system comprised of two components would have the following form

$$\boldsymbol{B} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}. \tag{3.22}$$

The associated state table for this example is shown in Table 3.4.

In order to derive the probability measure P that describes the likelihood to be in a certain state at a specified point in time t, the aggregated probability of the combinations has to be calculated. Therefore, the component reliabilities respectively unreliabilities, as defined in Section 3.2.5, are considered again. Provided that the failure probabilities for each component are given for equidistant points in a time, a vector $\boldsymbol{F}$ can be defined as follows:

Table 3.4.: Exemplary state table for a system composed of two components A and B

State	Comp A	Comp B
S_1	0	0
S_2	0	1
S_3	1	0
S_4	1	1

$$\boldsymbol{F}(t) = [F_1(t), F_2(t), \ldots, F_n(t)] \tag{3.23}$$

In Equation 3.23 each element represents the failure probability of individual components at time instance t. Based on $\boldsymbol{F}$, also the complementary $\boldsymbol{R}$ vector can be defined, which denotes the reliability of each component for the same time instance:

$$\boldsymbol{R}(t) = [R_1(t), R_2(t), \ldots, R_n(t)] = 1 - \boldsymbol{F} \tag{3.24}$$

Based on Equations 3.22, 3.23 and 3.24 the system's state-space probability matrix $\boldsymbol{M}$ is derived for time instance t as follows:

$$\boldsymbol{M}\,(t) = \boldsymbol{B} * \boldsymbol{F}\,(t) + \overline{\boldsymbol{B}} * \boldsymbol{R}\,(t)\,. \tag{3.25}$$

In $\boldsymbol{M}$, the binary matrix $\boldsymbol{B}$ is multiplied with the unreliabilities $\boldsymbol{F}$ of the system, while its negation $\overline{\boldsymbol{B}}$ is multiplied with the $\boldsymbol{R}$ vector. In this way, each cell of $\boldsymbol{M}$ represents the probability that the component is in the according state as required by $\boldsymbol{B}$.

It shall be noted, that $\boldsymbol{M}(t)$ is time depended, whereas $\boldsymbol{B}$ remains constant for the system under study, as long as it's structure is not changed, which is usually the case for most systems. Based on $\boldsymbol{M}(t)$, the probability that the system will be in state S_i at time instance t is then defined via the row product

$$P(S_i, t) = \prod_{1}^{n} M_{i,j}\,(t)\,. \tag{3.26}$$

For the previously introduced example with two components, the following state-space probabilities are derived:

State	Comp A	Comp B	State Probability
S_1	$R_1(t)$	$R_2(t)$	$P(S_1) = R_1(t)R_2(t)$
S_2	$R_1(t)$	$F_2(t)$	$P(S_2) = R_1(t)F_2(t)$
S_3	$F_1(t)$	$R_2(t)$	$P(S_3) = F_1(t)R_2(t)$
S_4	$F_1(t)$	$F_2(t)$	$P(S_4) = F_1(t)F_2(t)$

The above introduced state-space model has the characteristic that the sum of all state probabilities is equal to one so that $\sum P_{S_i} = 1$. Further, the probability of similar states can be grouped by calculating the row-wise sum. This is often the case for systems with built-in hardware redundancies. E.g. if the system from the above example would represent two identical pumps (A and B) that run in parallel, it would not make a difference from a reliability point of view, whether pump A or B fails. Thus, state probabilities of $P(S_2)$ and $P(S_3)$ could be combined to an artificial state $P(S_{2,3}) = P(S_2) + P(S_3)$ stating the probability that one of the two pumps has failed. As it will be revealed in Chapter 5 this has a significant impact on the overall calculation time of the model.

3.2.6. Resulting Dynamic Hybrid Reliability Model (DHRM)

With the formulations given throughout Subsections 3.1 to 3.2.5, a generalized aggregation method is defined, which can be used to integrate prognostics data from multiple monitored parts up to system-level. Further, the method is capable to consider conventional statistically based failure probabilities for those parts, which cannot be monitored via PHM. The resulting final *Dynamic Hybrid Reliability Model* (DHRM) is illustrated in Figure 3.8. The inputs of the DHRM are the periodically updated prognoses of the monitored parts as well as environmental data such as temperatures. Each part in turn is either modelled with statistics based functions such as the exponential or Weibull distribution or based on prognostics data in form of a derived RUL-CDF as defined in Subsection 3.2.3. On component-level, each component is modelled with an individual fault tree, which considers the dependency of the containing parts. Once the failure probability of each component is assessed, they are passed to the state-space model as defined in Subsection 3.2.5 to calculate the resulting state probabilities. Working with time-variant data, the DHRM must be periodically recalculated as soon as updated data becomes available. Thus, the model will generate updated state predictions for each assessment.

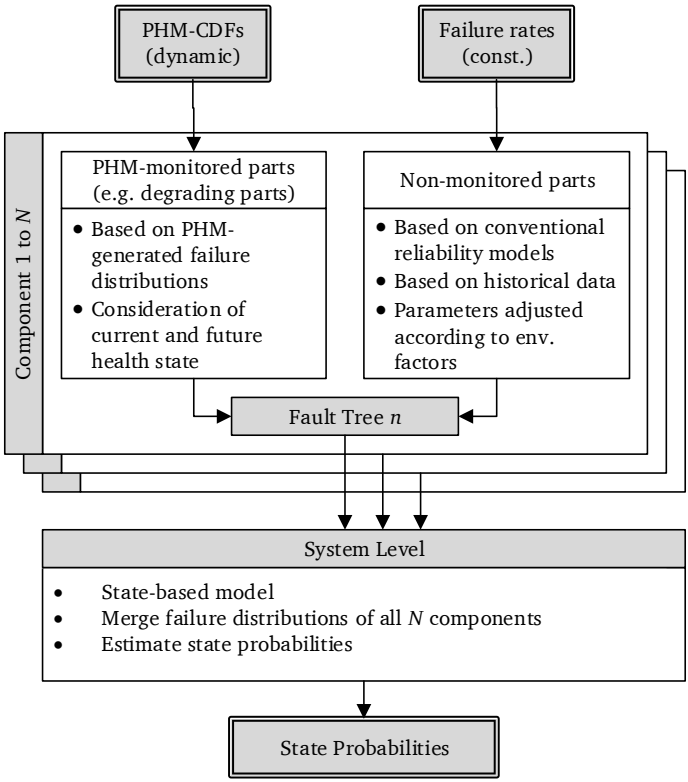


Figure 3.8.: Overall composition of the aggregation method

3.3. Uncertainty propagation

Uncertainty propagation plays a crucial role in every model, as the propagation and interpretation can have a significant impact on the model's results. Therefore, the following section is dedicated to the extension of the previously introduced DHRM concept by describing how to manage the uncertainties associated with the used models and inputs.

Uncertainties especially arise when estimating future events or states as predicting the future is an inherent uncertain task. They result from a variety of sources such as used input data, estimated parameters or assumptions made about future events. A common way to classify uncertainties is to distinguish between *aleatory*, *epistemic* and *ontological* uncertainties as given in [DDH19; ELB18]:

- **Aleatory uncertainty**, as the "known knowns", describe those uncertainties, which are known and can be quantified. A popular example is sensor noise, which is known to exist and can be specified to lie in a certain band or follow a given distribution for a given sensor.
- **Epistemic uncertainty**, also the "known unknowns", describes uncertainties resulting from a lack of knowledge. Nobody can precisely predict the future and some model parameters might simply not be known. Thus, still being aware of their existence, those uncertainties remain and are difficult to specify.
- **Ontological uncertainties** are finally the "unknown unknowns". This term refers to those uncertainties, which are not considered at all as neither their existence, nor their extent are initially known.

Depending on the type of uncertainty, different counter measures can be taken to contain and reduce their impact on the model's results. These approaches range from theoretical methods such as an uncertainty propagation (aleatory), consideration of different scenarios with parameter bounds (epistemic) to technical solutions in form of robust and adaptive design rules (ontological uncertainties).

3.3.1. Uncertainty propagation within the reliability engineering domain

Within the traditional reliability domain uncertainties often arise from a reliability model's parameters, e.g. the used failure rates, which are often not exactly known. To address this and to account for uncertainty, distributions describing possible parameter values are used in practice. Distributions to name in this context are the beta, gamma and log-normal distribution. The amount of uncertainty is thereby specified by the spread of the chosen distribution, also known as dispersion. Common measures for the dispersion include the standard deviation (gamma and beta distribution) as well as the error factor (EF) for the log-normal distribution, which is defined as the ratio between the 5th and the 50th (median) percentile: $EF = (50^{th}\text{percentile})/(5^{th}\text{percentile})$. Together with a mean value, all three distributions can be defined. Accordingly, to account for uncertainty in a reliability model, all parameters are described by one of the distributions mentioned. Subsequently,

the entire model is then calculated with the help of a Monte Carlo simulation, whereby the model parameters used are drawn from the distributions specified. In this way, the uncertainties are propagated through the model and considered in the final reliability estimates. However, it shall be highlighted, that this approach considers only (aleatory) uncertainties resulting from the sampling process and does not reflect any application variations [SV02].

In order to avoid confusion, the distributions and their parameters described here which are used to model the uncertainty of traditional reliability models, will be denoted as uncertainty distributions (UD_{Rel}) in the remainder of the work.

3.3.2. Uncertainties associated with prognostics

Making estimations about the future is an inherently uncertain task, which makes it important to consider and treat uncertainties properly in any PHM application. According to [SG13] uncertainties with prognoses arise from the following sources:

- **Present uncertainty:** Describes any uncertainties associated with data used prior to the prognosis, e.g. to assess the current health state of the system.
- **Future uncertainty:** Describes any uncertainties that result from a lack of knowledge about the future usage of a system, such as expected loads or operating conditions.
- **Modelling uncertainty:** Modelling uncertainties arise from model simplifications, estimated parameters, the chosen model form or inherent process noise and are independent of the selected modelling approach (physics-based or data-driven).

All together, the above sources of uncertainty are referred to as prediction method uncertainty, as they will have a combined effect on the RUL estimation. Despite the different sources of uncertainty also the interpretation can be distinguished. In the context of PHM, it can be differentiated between a prognosis-specific and an algorithm-specific uncertainty in a broader sense [SG13]. To further illustrate this, Figure 3.9 is presented, in which two schematic RUL prognoses $P1$ and $P2$ of the same unit under test, at the same time of prediction t_P, resulting from two different algorithms are shown. Both algorithms provide an RUL-CDF (CDF I and CDF II), which reveals the likelihood of a failure over time according to the algorithm. However, both algorithms vary in their overall accuracy and precision. While algorithm $P2$ has a very narrow uncertainty band with a very close t_{EoP} compared to the true t_{EoL}, the bounds of $P1$ are much broader and also the deviation from the true end of life t_{EoL} is wider. This example shows that there is no information

about the (algorithm-specific) trustworthiness or performance of the used algorithm in the RUL-CDF itself.

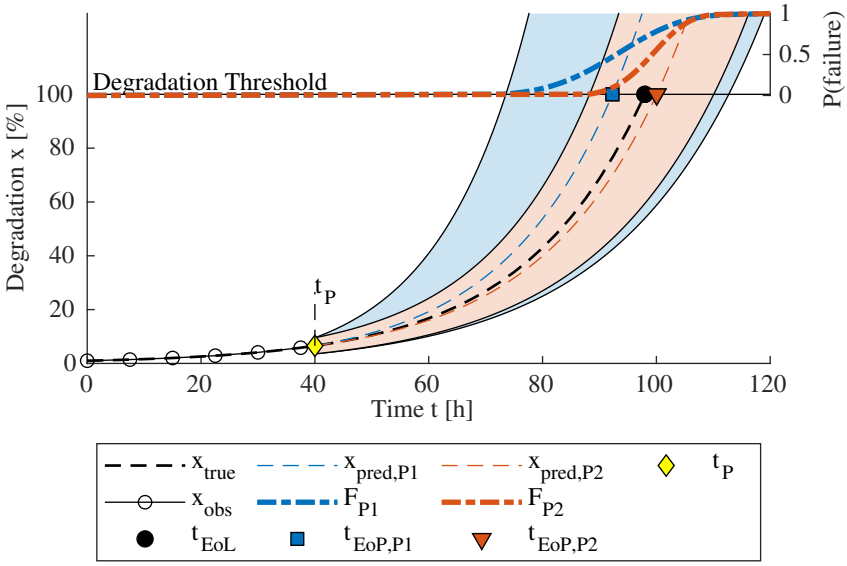


Figure 3.9.: Exemplary prognoses with different algorithm-specific uncertainty

Comparisons as the one in Figure 3.9 are often conducted to analyze and state the performance of a given prognosis algorithm. The general framework for the performance assessment is based on defined performance metrics, as introduced in Section 2.3.5. These metrics are used to specify and quantify quality measures such as the accuracy or precision of used or new prognoses algorithms. It is important to note, that these metrics are usually specified from a training or validation dataset, where the true failure times and thus the true RULs are known. Most of these metrics are then defined based on the error Δ between a given RUL prediction r and the true RUL r^* for a given time index k and a specific unit under test (UUT) l:

$$\Delta_l(k) = r_l^*(k) - r_l(k) \tag{3.27}$$

3.3.3. Uncertainty handling concept for the DHRM

With these insights given, it becomes clear that every prognosis is subject to uncertainty and that the performance of a prognosis algorithm varies as well. To account for these uncertainties, it is proposed in this work to handle uncertainties in a two-tiered way, in analogy to the framework presented in [ELB18]. This concept foresees to distinguish between prognoses-specific and algorithm-specific uncertainties and to propagate both individually.

In this context, prognoses-specific uncertainties are considered to be mainly of aleatory nature as they result from (known) sensor noise and parameter variations. Typical measures to handle these uncertainties include filtering approaches like Bayesian updating e.g. Kalman or particle filtering, which is applied during the prediction. In this way, uncertainties are propagated throughout the model and the estimated failure probability is derived.

The algorithm-specific uncertainties instead are considered to be mainly of epistemic nature. By definition these uncertainties result from a lack of knowledge and include modelling uncertainties as well as simplifications. These uncertainties are difficult to quantify in advance, but can be assessed retrospectively during algorithm validation. To account for these uncertainties it is therefore proposed to feedback performance metrics, which have been derived during the development phase of the algorithm, as a quality measure into the DHRM. In this way, the aggregation method is not only supplied with the likelihood of failure (RUL-CDF), but also has an information about the trustworthiness of the used prediction algorithms. The foreseen workflow is illustrated in Figure 3.10, where it is differentiated between the development (offline) and usage (online) phase of an algorithm.

The proposed method is based on the following assumptions:

- Multiple algorithms are used in collaborative fashion as it is proposed in the DHRM to predict the reliability of a system composed of multiple PHM-monitored components.
- Each PHM algorithm/solution has an individual performance, which can be expressed by predefined performance metrics.
- Algorithm's performance metrics are available before deployment of the algorithm (e.g. the performance metric is provided by the supplier of the component) and thus can be fed into the aggregation model as additional parameter.

Degradation-based performance metrics
As outlined in Section 2.3.5, most PHM metrics are defined for distinct prediction times

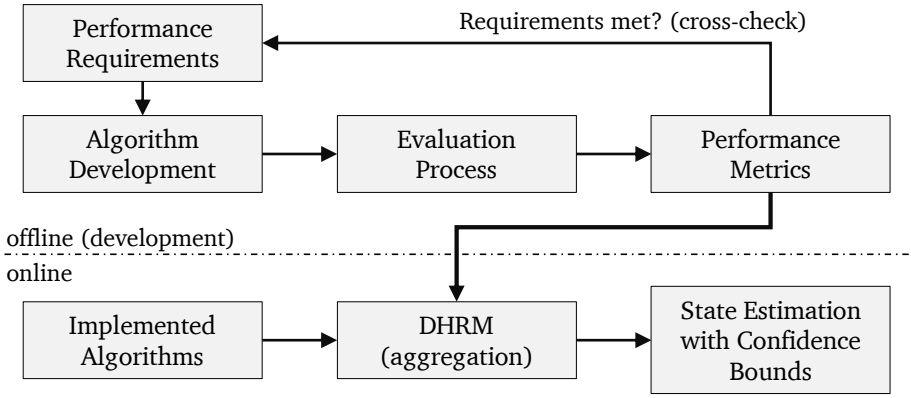


Figure 3.10.: Proposed workflow in which performance metrics from the development phase are fed back into the online prognosis

t_P, denoted with time index k. To be able to further compare different algorithms, the time index k is further normalized by the overall end of life, which leads to $\Lambda = k/t_{EoL}$. Obviously this definition is not suited as a real-time measure during the prognosis itself, as the end of life (t_{EoL}) is not known yet. To overcome this limitation it is proposed in this work to convert the defined time-based metrics from Section 2.3.5 into *degradation-based* performance metrics, in which the performance metric of a prognosis is normalized by the current degradation level instead of time instance. Formula 3.27 is thus rewritten as

$$\Delta_l(x) = r_l^*(x) - r_l(x) \tag{3.28}$$

with x being the current level of degradation. Based on this, also the B, S and MAD metric can be rewritten to depend on x and the following metrics are derived as presented in Table 3.5.

In the following, it is proposed to use these degradation-based metrics as a measure of dispersion, analogue to the described uncertainty distributions as presented in Section 3.3.1. In this way, the metrics from Table 3.5 can be used to quantify the uncertainty and to define an uncertainty distribution, which will be referred to as $UD_{PHM}(x)$ in the following.

Table 3.5.: Degradation-based performance metrics

Metric	Degradation-based notation
RUL Error	$\Delta_l = r_l^* (x) - r_l (x)$
Average Bias (B)	$B (x) = \frac{1}{L} \sum_{l=1}^{L} \Delta_l (x)$
Sample standard deviation (S)	$S (x) = \sqrt{\frac{\sum_{l=1}^{L} (\Delta_l (x) - M)^2}{L-1}}$ M is the sample mean of the error
Mean absolute deviation from the sample median (MAD)	$MAD (x) = \frac{1}{L} \sum_{l=1}^{L} \lvert \Delta_l (x) - M \rvert$ M is the sample median of the error

Based on the definition from Section 3.2.3 the probability that a part l, monitored by a PHM-system, has failed at time instance t is given as

$$F_{l,PHM} (t) = P (t_{EoP} \leq t) . \tag{3.29}$$

To account for the uncertainty given through the used prognosis algorithm, the correction offset $\psi(x)$ is added to F, so that Equation 3.29 can be rewritten to

$$F^*_{PHM}(t, x) = P(t_{EoP} + \psi(x) \leq t) \tag{3.30}$$

The correction offset itself is drawn from the uncertainty distribution as previously defined

$$\psi(x) \sim UD_{PHM}(x). \tag{3.31}$$

In this way, the uncertainty induced by the used PHM algorithm can be accounted for. In this context it is important to point out that the $UD_{PHM}(x)$ is used to draw an offset $\psi(x)$ by which the actual function is shifted, while the UD_{REL} is used to vary individual parameters of the function. Also, $UD_{PHM}(x)$ depends on the current level of degradation x, while UD_{REL} is assumed to be constant.

For a very modest problem, where the error metric Δ_l is Gaussian distributed, $UD_{PHM}(x)$ becomes the normal distribution with a sample standard deviation of $S(x)$ and the average bias of $B(x)$ as defined in Table 3.5. The correction offset $\psi(x)$ is then obtained via:

$$\psi(x) \sim UD_{PHM}(x) = \mathcal{N}\left(B(x), S(x)\right) \tag{3.32}$$

For more complex RUL deviations the error distribution can have an arbitrary shape and can be described by a best-fit approximation (Minimum-likelyhood-estimation).

Uncertainty propagation on component-level (MCS-I)
Based on the previously stated definition a common framework is developed, which is suited to account for different types of uncertainty during the aggregation of failure predictions. This two-tiered concept therefore separates the inherent variability (aleatory effects, tier 1) from the modeling uncertainty (epistemic effects, tier 2).

In case of the non-monitored parts, which are modelled by means of traditional reliability methods, inherent time of failure variability is covered by the used distributions itself e.g. the exp. or Weibull distribution. The modeling uncertainty is then specified by the UD_{Rel}, which is used to sample parameters from. In this way the epistemic uncertainty is addressed.

For those parts being monitored by a prognostic algorithm it is assumed that the general variability such as sensor noise or variations in the degradation process are governed by the prognosis algorithm itself and the uncertainties will be propagated resulting in the RUL-CDF, as argued in Section 3.3.2. The remaining modeling uncertainty (epistemic) resulting from simplifications and a lack of knowledge is then derived from specified performance metrics of the used algorithms in form of the UD_{PHM} as defined in Section 3.3.3. The two-tiered concept is summarized in Table 3.6.

During the calculation of each component, each part's failure probability is accounted for using the fault tree model as described in Section 3.2.4. The required failure probabilities are obtained in form of conventional reliability distributions as well as in form of the RUL-CDFs for those parts being monitored by a PHM system. To further allow the propagation of the uncertainties (second tier), a Monte Carlo simulation (MCS) is then used in which the fault tree model of each component is calculated multiple times. For each individual calculation (sample), parameters are drawn based on the specified uncertainty distributions to modify the CDFs as well as the RUL-CDFs. The resulting failure probability of each component is then described by a family of probabilities, which reflects the associated uncertainty with the overall failure prediction.

Table 3.6.: Two-tiered concept for uncertainty handling within the DHRM

Domain	Tier 1: Known variations (aleatory effects)	Tier 2: Unknown variations (epistemic effects)
Reliability Engineering	Common failure distributions (CDFs) e.g. Weibull or exp. distribution	Uncertainty distributions describing parameter variations (UD_{REL})
PHM	RUL-CDF obtained from prognosis algorithm	Algorithm-specific uncertainty distributions based on degradation-based metrics (UD_{PHM})
Implementation	CDF aggregation (Fault Tree formulas)	MCS propagation

Uncertainty aggregation on system-level (MCS-II)
In contrast to the normal data aggregation on system-level as described in Section 3.2.5, the extended DHRM with uncertainty propagation has to deal with a family of CDFs per each component. To aggregate those CDF families two options are considered:

1) The first option is to select one representative CDF for each CDF family. This CDF is selected based on a statistical measure such as mean or a predefined confidence bound. Afterwards, for each component the selected representative CDF is used to calculate the state probabilities. The final output of this approach is a single state probability given as a CDF again.

2) The second option is sample based. Therefore, a CDF is randomly chosen for each component from the corresponding CDF family, describing one sample. By defining n-samples, the state model is then calculated n-times with an MCS, leading to a family of CDFs for each state probability.

For this work, it is chosen to use the sample based option. Although being computationally more expensive, it is expected to generate more accurate predictions as uncertainties are thoroughly propagated until the very end of the aggregation. It is then still possible to reduce the result space by selecting a representative CDF for each state based on statistical measures. The overall uncertainty propagation concept is illustrated in Figure 3.11.

MCS performance considerations
One noteworthy drawback of the chosen concept is required high computational effort,

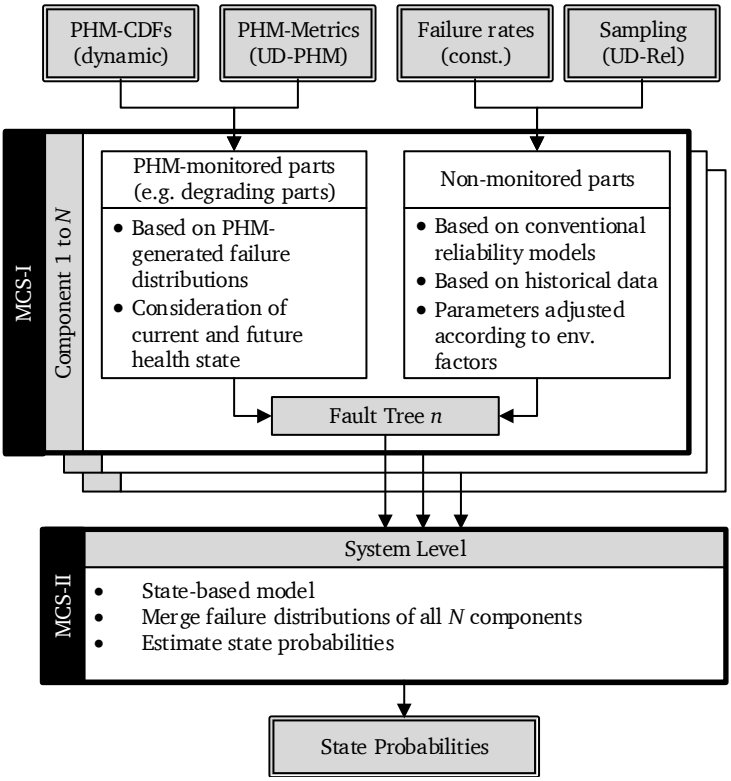


Figure 3.11.: Extension of the DHRM to consider uncertainties from the PHM (UD_{PHM}) and the traditional reliability domain (UD_{REL})

due to the sampling and calculation of multiple MCS trials. In general, the accuracy and resolution of the MCS is defined by the number of samples used. Thus, there is always a trade-off between efficiency and accuracy when applying an MCS. To optimize the runtime of an MCS different approaches have been discussed in literature. Despite increasing hardware resources or parallelizing the required calculations, e.g. with graphic cards or Field Programmable Gate Arrays (FPGA), the MCS can also be tuned by optimizing the sample size and space. Examples of these techniques include the Latin hypercube sampling

(LHS) as well as the bootstrapping method, a method for resampling [IH88; JHY82]. Being a simple and generic approach, for this work the bootstrapping method is chosen to assess the quality of the MCS and thus be able to optimize (reduce) the number of required samples to an acceptable level.

The bootstrapping method is based on the idea to artificially resample data based on a given sample $x = (x_1 \dots x_n)$ from an unknown underlying distribution F. Being the only information of F, the data can be resampled, by drawing new samples from the empirical CDF F_{emp}. In this way, the original experiment can be artificially rerun, generating multiple independent and ideally distributed replications X, without having to rerun the complete MCS.

The performance of the MCS can then be assessed by building n replicas of the results based on the bootstrap method, leading to n expectation values $E_1, ..., E_n$ for each solution. As these samples are independent and ideally distributed they will vary due to the stochastic nature of the MCS. According to the Central-Limit-Theorem (CLT), statistic measures such as the mean will follow a normal distribution, as long as the variance of E_i is finite and N is sufficiently large. Thus, the quality of the MCS can be derived by evaluating the distribution of E_i with mean and variance defined as

$$\bar{E} = 1/N \sum_{i=1}^{N} E_i \tag{3.33}$$

and

$$S^2 = \frac{1}{N} \sum_{i=1}^{N} (E_i - \bar{E})^2 \tag{3.34}$$

Finally, the confidence interval of the solutions is described as:

$$(\bar{E} - z_{(1-\frac{\alpha}{2})} \frac{A}{\sqrt{N}}, \bar{E} + z_{(1-\frac{\alpha}{2})} \frac{A}{\sqrt{N}}) \tag{3.35}$$

In this formula $1 - \alpha$ represents the confidence level, while $z(1 - \alpha/2)$ is the corresponding standard's deviation quantile [KTB11]. As shown in [Yan11] the confidence interval as well as the variance of the MCS-results can be used as a direct performance indicator of the MCS as both indicators will converge as the number of samples is increased.

It shall be noted, that the explanations given about uncertainty handling in this chapter are of theoretical nature and will not be pursued further in the following course of this thesis as no concrete algorithms are given that could be evaluated. The interested reader is referred to [Hei19].

3.4. Method implementation

In the following section and subsections the implementation of the proposed method as introduced in Sections 3.1 to 3.3 is outlined. As programming environment Mathwork's *MATLAB* is chosen. Matlab is a rapid prototyping tool, which allows a quick algorithm and software development. It is originally intended to solve mathematical problems and is extendable by various add-on products and toolboxes, such as *Simulink*, a graphical modeling tool, which will be also used in this work. Being very flexible due to extensive scripting possibilities, the Matlab environment is well suited to realize the proposed DHRM method.

The overall implementation is split into two distinct software modules: The first module is a custom fault tree library implemented in Simulink that is used to create and calculate the required component fault tree models. The second module is the DHRM aggregation tool, which is used to define the system model and its states, to ingest any data from various sources and to calculate the system model for a given time instance to derive the estimated state probabilities. An overview of both modules, their functions and the output is given in Figure 3.12. They are also further outlined in the following. The implementation of the (later) required simulation environment and application to a specific system is not shown in this figure and will be discussed in detail in Chapter 4.

3.4.1. Custom fault tree library in Simulink

The custom fault tree library is intended to provide any means to create a custom fault tree model and calculate it. As it is required to ingest time-varying input data to the model and frequently changing parameters, the resulting models must provide the necessary interfaces to easily modify input data and perturbate parameters. Further, the calculation performance of the found solution must be sufficiently fast in order to run the required MCS of the model.

For this work, Simulink is chosen as an implementation framework for a generic custom fault tree software library. Although there are different commercially and free to use fault tree software tools, such as *Reliability Workbench*[2] from Isograph, *Blocksim*[3] from ReliaSoft or *ITEM ToolKit*[4] by item Software, Simulink was chosen as it is tightly coupled with

[2]https://www.isograph.com/software/reliability-workbench/
[3]https://www.reliasoft.com/products/blocksim-system-reliability-availability-maintainability-ram-analysis-software
[4]https://www.itemsoft.com/item_toolkit.html

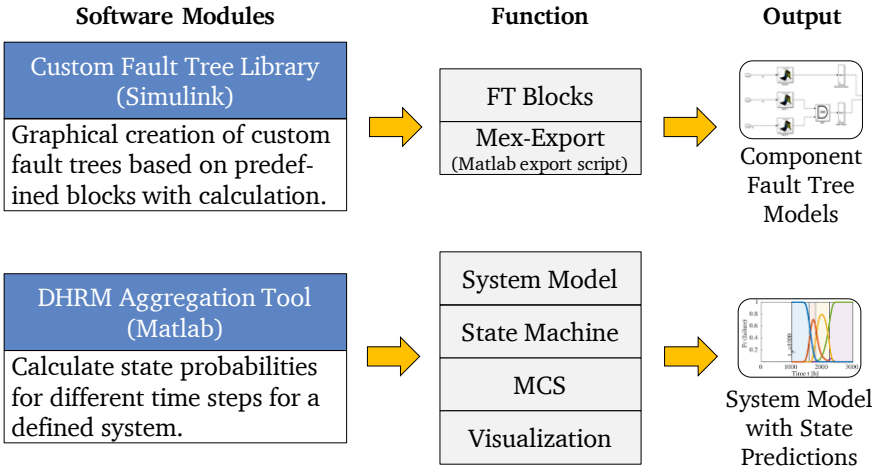


Figure 3.12.: Overview of the implemented software modules, their function and the used environment for implementation

MATLAB (which is also used for the DHRM aggregation tool) and offers many ways to ingest and export data via custom interfaces as required, which is difficult to achieve with closed source proprietary software. Further, Simulink offers a graphical user interface and allows graphical programming by placing reusable code snippets, called blocks. As the fault tree model itself is also given as a graphical representation the Simulink environment is an ideal candidate for a prototype implementation.

Within Simulink, the fault tree implementation is realized in form of a library, which contains reusable software modules, denoted as blocks. Each block has an arbitrary number of in- and output ports, a set of parameters and encapsulates a custom mathematical function $f(x_k)$, which transforms the inputs x_k to the output values y_k for a given time step k. In this way, the basic elements of a fault tree (AND, OR, XOR, NOT) are implemented based on the functions given in Section 3.2.4 so that at each simulation time step k, the corresponding Boolean operation is performed on the inputs of each block. Further, the library contains a set of additional input elements, which extends the standard basic events of a fault tree and allow the ingestion of external dynamic input. These inputs

can include changing probabilities such as the RUL-CDF or time-varying parameters like environmental parameters. A screenshot of the custom library is shown in Figure 3.13, a complete definition of the implemented blocks is given in the Appendix B.1.

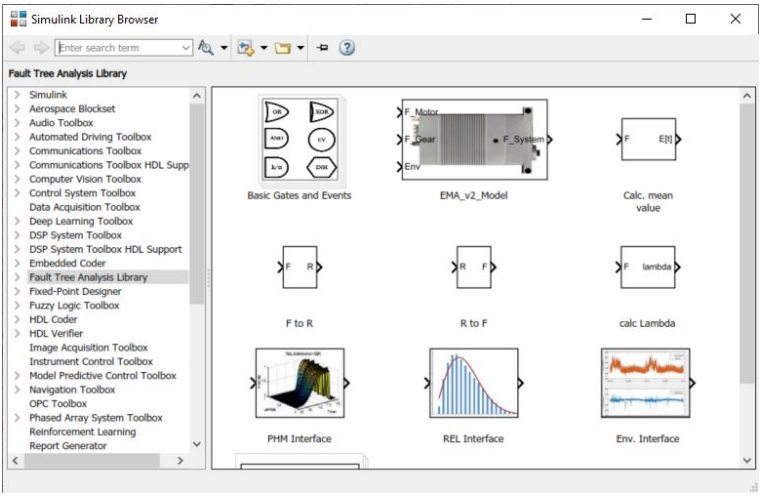


Figure 3.13.: Custom fault tree library basic blocks implemented in Simulink (excerpt shown)

With the developed custom fault tree library it is possible to create general fault tree models based on the implemented blocks. Therefore, the required blocks are copied to a new Simulink model, where they are interconnected according to the component's structure. By running the simulation, Simulink's internal simulation time variable k is used to derive the current probabilities in the basic events or to select an input value from connected RUL-CDF data respectively. The simulation start and end time t_0 and t_{end} can be arbitrary chosen by the user. However, it shall be noted that the complete calculation is discrete and returns the fault tree's top-node probability for each time step k. By changing the step-size of the simulation, the fault tree's output resolution can be adjusted.

In Figure 3.14, an exemplary fault tree created with the custom Simulink fault tree library is shown. The modelled system describes an electromechanical actuator (EMA) as given in [Yuy+15] and will be used later for the verification of the implementation.

Mex-Export
As initially stated, it is crucial for the final fault tree model to be calculated fast. Therefore,

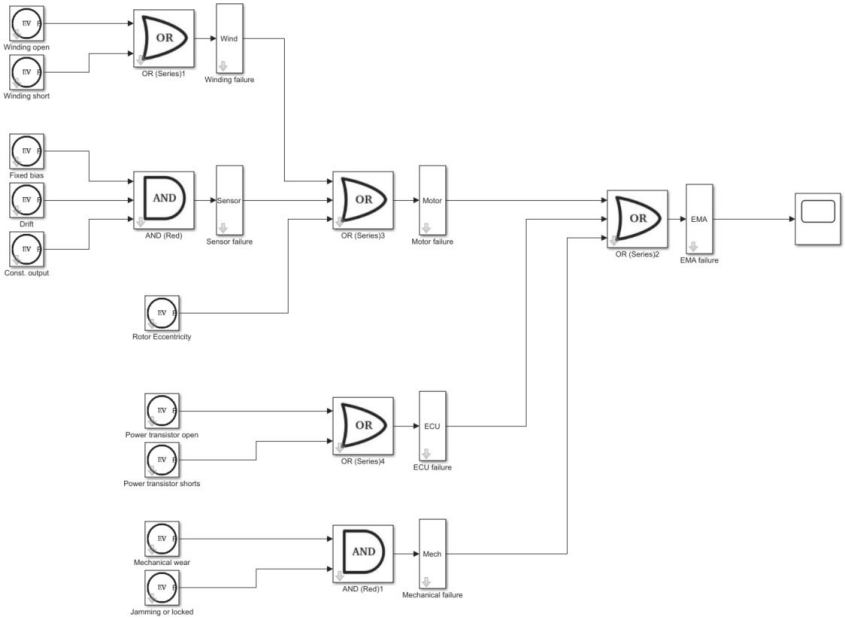


Figure 3.14.: Exemplary fault tree model of a generic EMA based on [Yuy+15] created in Simulink with the custom fault tree library

the fault tree library was extended with a custom mex-export script. *Mex* is a built-in Simulink function that allows the conversion from a Simulink model to executable c-code. The compiled c-code can then be used again by any Matlab application, which makes it much faster to run. The custom mex-export script provides all means to automatically export and compile a given fault tree model.

The output of the mex-export is a fully functional fault tree model that can be calculated for different time intervals and fed with dynamic input data, depending on the existing input nodes of the model. The output of the model is always the top-node's failure probability for the current time-step.

3.4.2. DHRM aggregation tool

While the fault tree library is used the implement the part- and component-level calculations, the DHRM aggregation tool is used to implement the system-level logic as defined in Section 3.2.5. The tool itself is implemented in Matlab in an object-oriented programming (OOP) style. Accordingly, each logical entity within the aggregation tool is considered as an object. Each object is then derived from a class, which is used as a blue-print and defines the data structure, the parameters of the object, as well as the methods used to interact, change or visualize this data. The OOP approach is used in this work, as the problem itself, the aggregation of the failure probabilities of a complex system, can be divided in a hierarchical way into several sub-problems, which makes it ideally suited for an object-oriented representation.

In Figure 3.15, the structure of the program is shown as a Unified Modeling Language (UML) class diagram to lay out the defined classes and their connections. The diagram is read as follows: Each system is composed of one or more components, while each component contains exactly one fault tree model and one Monte Carlo simulation object. The general functionality of each class is briefly outlined in the following.

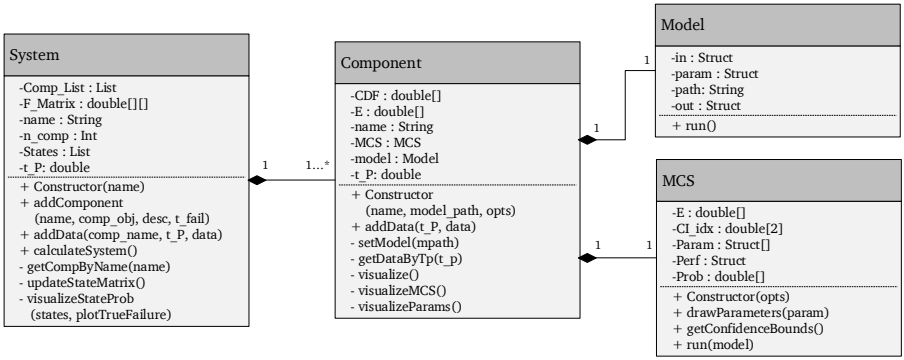


Figure 3.15.: UML class diagram of the DHRM aggregation tool

System class
The system class is used to define the system itself. Therefore, it implements all methods required to add component objects to the system and to define the relevant system states and their severities. Once the system model is defined, data can be added as input streams

for each component and the system model can be calculated. Finally, this class provides all necessary means to visualize the calculated state probabilities.

Component class
The component class is used to store and bundle all information associated with a specific component of the system. It contains an underlying model object, which embeds the fault tree model and further contains an MCS object, which is used to run the MCS on the fault tree model. It shall be noted that the underlying fault tree model (as mex-function) can be reused for multiple components, although each component object always marks an independent and individual instance. Therefore, the component class is equipped with all means to calculate the failure probability of the considered component. Finally, it also implements the required function to visualize the calculation results as well as the MCS samples.

Model class
The model class is used to store all relevant information of the underlying fault tree model. This includes the number of in- and outputs, the definition of the model's parameters, including their distributions, as well as the location of the model's mex-executable[5]. From the *run* method the executable mex-file is run and results are returned.

MCS class
The MCS class is the implementation of the Monte Carlo simulation. This class holds a set of samples, which are initialized at the beginning. According to the definition in Section 3.3.1, each sample contains a set of parameters e.g. failure rates, which are sampled according to the specified distributions in order to account for any parameter uncertainty. All samples are then propagated using the underlying model for increasing points in time. The result contains a set of CDFs, according to the sampling size of the MCS, from which the final CDF is chosen based on a predefined confidence interval.

3.5. Model verification

The following subsection of this chapter is dedicated to the verification of the proposed method. To verify the correct implementation, the individual parts of the solution are tested against reference values from literature or proprietary software. Those parts implementing novel concepts are tested for plausible results based on simplified input data. In the following section, the overall solution is then checked against the initially stated requirements to validate that it complies to the defined objectives.

[5]All models are stored in a dedicated folder from which they can be added to the DHRM method on the fly.

3.5.1. Verification of the fault tree implementation

The correct implementation of the custom fault tree library in Matlab/Simulink is verified against a reference fault tree model presented in the literature [Yuy+15]. This reference model describes a generic electromechanical actuator (EMA) as it is often found in the aerospace industry, where such motors are used to actuate the control surfaces of smaller aircraft (e.g. drones). The considered EMA consists of an electrical motor, an electronic control unit as well as a transmission gear. The model is characterized by multiple logical AND and OR connections and has ten different basic events, representing failures on the lowest level. Event $E6$, representing a rotor eccentricity failure, will be used either with a constant failure rate, which represents the default case or with dynamic PHM data, which is considered as the innovation of the DHRM approach. The overall model together with the failure probabilities of the basic events is shown in Figure 3.16.

For the verification of the custom fault tree library, the reference model from [Yuy+15] is recreated and calculated in *Reliability Workbench*, a professional software for reliability analyses from Isograph. The resulting failure probabilities of the model's cutsets are then compared to results obtained from the same model calculated in the custom fault tree implementation used in this thesis. The results of the EMA model are presented in Figure 3.17, with the failure probability over time as calculated by the custom fault tree solution. Selected reference values from the commercial software are overlaid as reference. In addition, the resulting failure rates of each cutset are presented in Table 3.7 for the reference and custom fault tree implementation.

Table 3.7.: Resulting failure rates as calculated by the reference software and the custom implementation with a step-size of $h = 1$ given in failures/hour.

cutset	λ_{Ref}	$\lambda_{cust.FT}$	cutset	λ_{Ref}	$\lambda_{cust.FT}$
EMA	5.4E-06	5.4E-06	Sensor	4.3E-18	4.3E-18
Motor	2.1E-06	2.1E-06	ECU	3.3E-06	3.3E-06
Wind.	1.7E-05	1.7E-05	Mech.	5.6E-13	5.6E-13

The overall calculated failure rate for the EMA top event is $\lambda = 5.4E-6$ failures/hour which is exactly as specified in the paper from [Yuy+15]. The individual cutsets of the model are also calculated with same results as given by Reliability Workbench, at least for a small step-size ($h = 1$). For larger step-sizes the results between the proprietary and

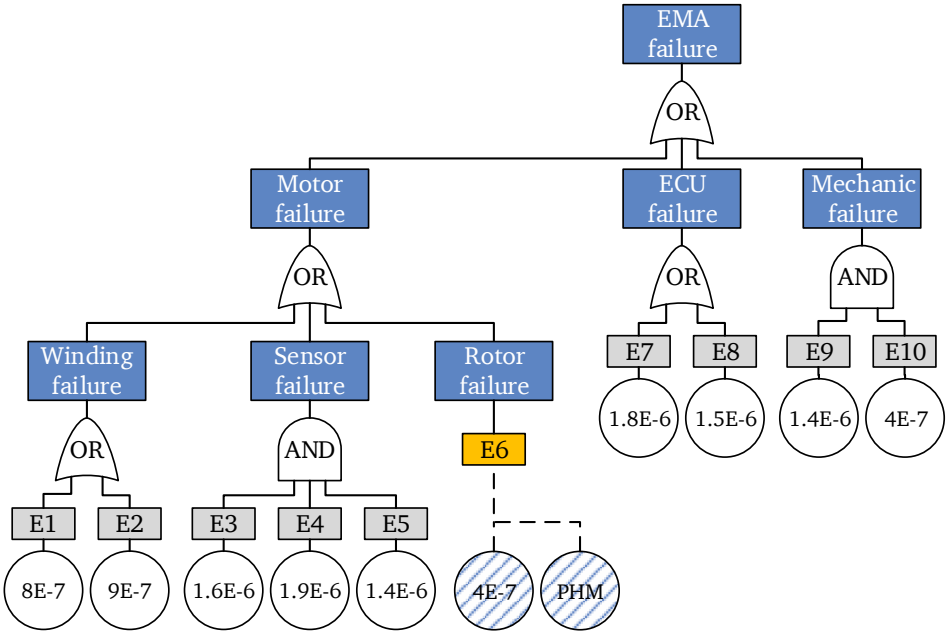


Figure 3.16.: Used Fault Tree model to verify the FT implementation with E6 modified to optionally ingest dynamic PHM data. Inputs given in failures/hour.

the custom software start to drift away. The detailed comparison of all input and output values of both models can be found in the Appendix B.2.1 of this thesis.

From the above test it is concluded that the custom fault tree solution is implemented correctly and gives accurate results.

3.5.2. Verification of the aggregation function

In a next step, the implementation of the aggregation function is tested and verified. Therefore, a simplified multi-component system is considered, which consists of two identical EMAs, where each EMA has the same model structure as the one from Figure

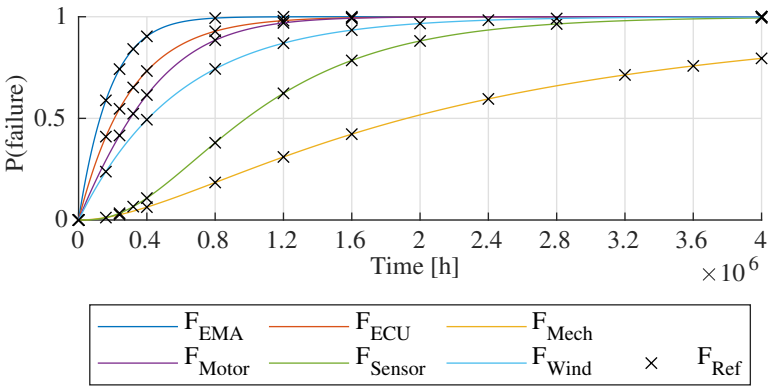


Figure 3.17.: Graphical comparison of cutset unreliabilities calculated from the commercial (Ref) and custom fault tree implementation

3.16. However, to allow the investigation of dynamic failure data provided by a PHM algorithm, one of the model basic events, event $E6$, is replaced by a PHM input node allowing time-varying input data. The overall system setup is shown in Figure 3.18. The outputs at this stage are the predicted state probabilities of that system.

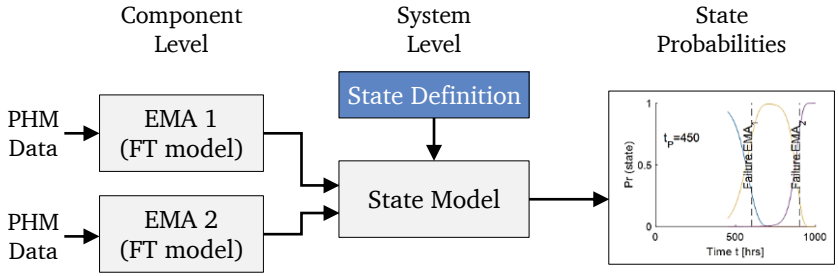


Figure 3.18.: Schematic system configuration used to verify the aggregation function of the DHRM

Generated PHM input data

To test the setup, initially a set of predefined RUL-CDFs is artificially generated, intended

Table 3.8.: Parameters used for the generation of the generic RUL-CDFs

	α_0	β_0	α	β	$t_{EoL}[h]$
Dataset A	500	2.4	1.02	1.2	600
Dataset B	500	4	1.02	1.2	900
Dataset C	500	3	1	1	$n.a.$

to mimic the behavior of a prognosis algorithm. Therefore, a Weibull distribution is chosen, in which the shape and scale parameters evolve over time in a way that the distribution gets narrower and converges to a true assumed failure time. The recursive scheme used to generate this artificial PHM data is defined as follows:

$$F_{RUL,k}\left(t\right) = 1 - e^{-\left(\frac{t-t_0}{a_k}\right)^{b_k}}. \tag{3.36}$$

In Equation 3.36, the scale and shape parameter, defining the distribution for the current time step k, are constructed based on the values from the preceding time instance $k-1$:

$$\begin{aligned} a_k &= a_{k-1} \cdot \alpha \\ b_k &= b_{k-1} \cdot \beta. \end{aligned} \tag{3.37}$$

With the above definitions, three different datasets A, B and C are generated. The chosen parameter combination is given in Table 3.8.

While dataset A and B are used to mimic the behavior of a prognosis output, dataset C defines a static failure distribution, which will be used in the following as a general distribution as long as no predictions are available. The true end of life times are defined to be $t_{EoL} = 600$ h for component A and $t_{EoL} = 900$ h for component B. The RUL-CDF evolution of datasets A and B is shown in Figure 3.19 with respect to the point of prediction normalized to the true failure time with $\Lambda = t_p/t_{EoL}$. Also, the generated default CDF is shown and highlighted in each plot. From both figures, it can be seen that both RUL-CDFs converge to the true failure times for later predictions as it would be assumed for a well working PHM algorithm, while the default CDF is a more rough estimate at the very beginning. This can be seen by the increased steepness of RUL-CDFs for later predictions, which align more and more with the vertical true EoL pane.

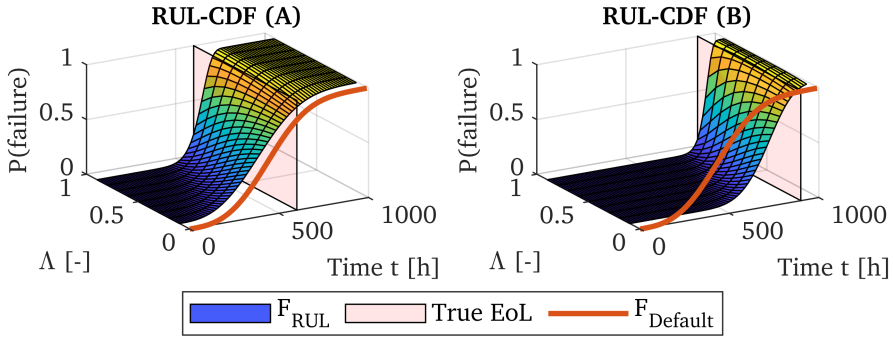


Figure 3.19.: RUL-CDFs of dataset A and B at different prediction times

Component-level verification

The scope of the component-level verification has two objectives: At first it shall be verified that the dynamic ingestion of PHM-like data is working correctly. Secondly, it shall be assured that the model generates reasonable outputs for dynamic input data. Therefore, the EMA model from Figure 3.16 is considered again, with node $E6$ being replaced to include dynamic PHM data. With this setup, the model is calculated at distinct prognosis times between $t_P = 100$ h and $t_P = 500$ h, with the corresponding input data from dataset A for each t_P. All other model parameters are kept constant as presented in Section 3.5.1. To further verify the calculation, the setup is duplicated within Reliability Workbench and calculated for $t_P = 100$ h and $t_P = 500$ h to generate reference results. As input node for basic event $E6$ a Weibull node with constant parameter setting is used, for which the parameters are manually changed for both calculation times. The model's output is given in 3.20. The two solid lines represent the calculated failure probability of the EMA estimated at $t_P = 100$ h respectively $t_P = 500$ h. The reference outputs from Reliability Workbench for the same parameters are overlaid. In between, the dashed lines visualize the calculated probabilities for intermediate prognosis times, while the vertical line indicate the true point of failure, as defined for the considered component.

With the generated results, it is shown that the created aggregation function allows the integration of time-varying RUL-CDFs from a PHM algorithm. The calculation, at least for two RUL-CDFs, could be recreated with the reference software Reliability Workbench. Further, the model shows the expected behavior in a way, that the overall EMA's failure

probability will get more precise for later predictions, as the input data evolves as well.

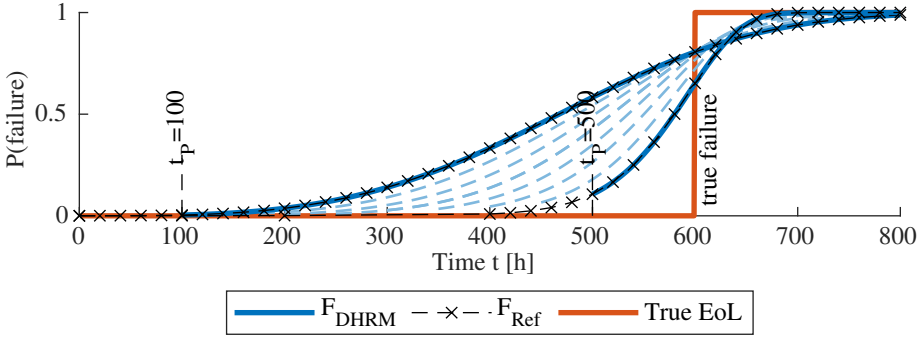


Figure 3.20.: Output of the EMA model's top node for different calculation times based on dynamic input data

System-level verification (State-space model)
In a next step, the overall aggregation function on system-level is verified. Therefore, the simplified system from Figure 3.18 consisting of two identical EMAs is considered and fed with the generated input data from Table 3.8. According to the definitions from Section 3.2.5 this system has in total four distinguishable states $S = \{S_1, S_2, S_3, S_4\}$, where S_1 and S_4 represent the failure free, respectively the failed state, while S_2 and S_3 indicate that one of the two EMAs has failed. From a practical point of view, the described actuation system could represent a subset of a larger system. An example could be the control surface actuation system of an aircraft, where one steering-axis is controlled by two identical actuators acting in parallel with the assumption that both actuators are independent of each other (e.g. no force fighting).

In this configuration, EMA 1 is fed with dataset A, while EMA 2 is fed with dataset B as dynamic input data for the input node $E6$. Each component is then calculated for different prediction times t_P and the resulting state probabilities are derived. With the described setup, the given states and the defined failure times at $t_{EoL,EMA1} = 600$ h and $t_{EoL,EMA2} = 900$ h, it is expected that the resulting state probabilities converge towards the true state transitions for later predictions, which is the scope of this verification test.

The output of the state model is presented in Figure 3.21 for selected prediction times at $t_P = \{1\,\text{h}, 250\,\text{h}, 450\,\text{h}, 700\,\text{h}\}$. For $t_P = 1$ h, where no predictions are available, the default failure curve from dataset C is considered again. The probability of state S_1 remains high for a while until it starts to be reduced as the probability for a first fault, states S_2 and S_3,

rises. For the following period, the probability of S_2 and S_3 reduces again as it becomes more likely that both EMAs have failed and thus state S_4 is reached. At this point it is not clear, which EMA will fail first as the probabilities of S_2 and S_3 are the same. Nor it is precisely known when exactly the state transition will occur. From $t_P = 100\ \mathrm{h}$ on, first PHM results (RUL-CDFs) are available and considered for the calculation of the state probabilities. In the subfigures for $t_P = \{250\,\mathrm{h}, 450\,\mathrm{h}, 700\,\mathrm{h}\}$, it can be seen how the state probabilities start to change for later prediction times and move towards the true state transitions, indicated by the vertical lines. The gradient of the state probability curves becomes steeper for later prediction points, providing a more accurate indication when the state transition will occur. At $t_P = 450\ \mathrm{h}$ for example, it can be seen that the probability of S_3 already has risen (meaning that only EMA 2 is available). Thus, the model expects EMA 1 to fail first, which is correct with the according true EoL-times. It also can be seen that state S_4 becomes more probable from $t = 800\ \mathrm{h}$ on. With a true EoL of $t = 900\ \mathrm{h}$ for the second EMA this is plausible, as the system will enter the failed state S_4 on the failure event of EMA 2. The state S_3 in turn is never reached, as this would mean that EMA 2 has failed while EMA 1 is still working, which is not the case.

The given example complies with the expected output from the aggregation function. The state probabilities converge towards the true values and the overall model gives reasonable results. The sum of all states remains one at all times $\sum P(S_i) = 1$ and is therefore also mathematically correct. The implementation of the system-level aggregation is therefore considered verified.

3.5.3. Verification of the uncertainty propagation

Based on the previous insights, the uncertainty propagation concept of the DHRM is verified in the following. Therefore, it is distinguished between component and system-level and the introduced EMA model as well as the simplified actuation system from the previous sections are considered again.

Component uncertainties
On the component-level, the induced uncertainties result from the UD_{REL} as well as the UD_{PHM} as outlined previously. The verification at this point thus is therefore two-fold and will first focus on the effects of parameter uncertainty (UD_{REL}) as introduced in Section 3.3.1 before considering any PHM related uncertainties. At first, the EMA model from the previous subsections is considered again (without PHM input). However, this time each failure rate is described by a lognormal distribution instead of using constant failure rates. Thus, the failure rate itself is randomly drawn for each calculation. The

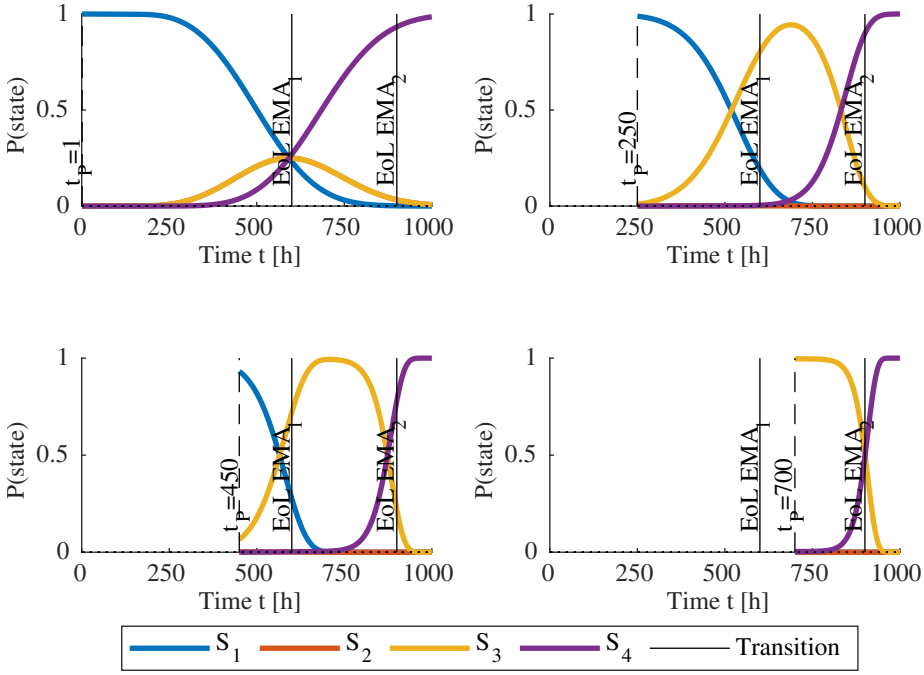


Figure 3.21.: DHRM output (probability of future system states) for different prediction times t_P

associated uncertainty of the failure rate is then expressed by the standard deviation of the lognormal distribution σ. To make this effect visible, two experiments are conducted with standard deviations chosen to be $\sigma = 0.1$ and $\sigma = 0.3$. In each experiment the EMA model is calculated by means of an MCS with a sampling size of $n = 500$. From the variation of the model's parameters it is expected that the overall failure probability for the EMA as well as the probability over time is affected and will vary for the individual samples.

In Figure 3.22, the outputs of both experiments of this verification test are given. It can be seen that by providing an uncertainty measure for each parameter, the output of the model results in a family of CDFs with a distribution of the expectancy value E. Further, the effect of uncertainty factor, here the standard deviation of the lognormal distribution, can be seen. While the results for the first experiment with $\sigma = 0.1$ are much closer to

each other, those with $\sigma = 0.3$ are spread more widely. The detailed configuration of this experiment together with an overview of the drawn parameters is given in the Appendix B.2.3 of this work.

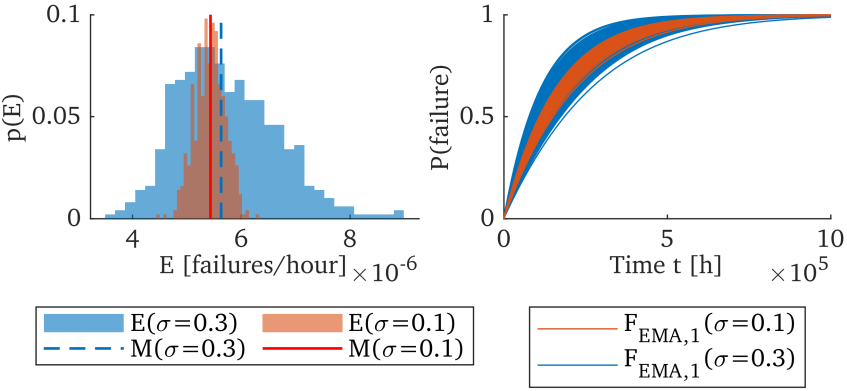


Figure 3.22.: Distribution of expectancy value E and associated median value M (left) together with the resulting failure distributions for both experiments with considered uncertainties of $\sigma = 0.1$, respectively $\sigma = 0.3$ (right)

The second part of the component-level uncertainty verification is dedicated to the correct consideration of the UD_{PHM} as introduced in Section 3.3.2. Therefore, the EMA model is considered again, but with enabled PHM data input node. As input data, the generic RUL-CDF dataset A, as specified in Table 3.8, is used. To account for any PHM algorithm specific uncertainties, the dataset is enriched with an uncertainty distribution. The artificially created correction offset psi is based on a normal distribution that evolves over time in a way, that the overall error becomes smaller for later predictions, as it would be expected from a well-working PHM algorithm.

The resulting failure probability calculated by the method with dynamic PHM input data and an associated correction offset is given in Figure 3.23. For each prediction time, a family of CDFs is created based on the MCS samples. It can be seen that the shape of the sample distribution gets narrower for later prediction times as expected. As a reference, the true failure time as well as the default CDF $F_{Default}$, used if no PHM data are available,

are plotted as well.

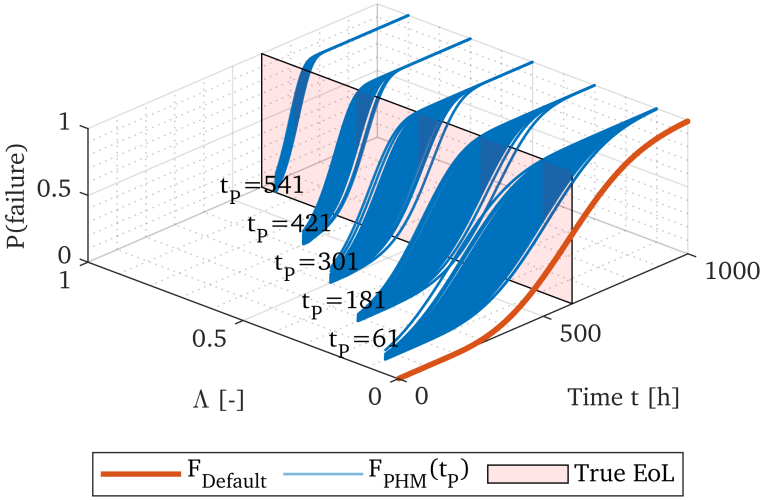


Figure 3.23.: Estimated failure distribution for the EMA model with dynamic PHM input data and evolving UD_{PHM}

Both tests meet the expectations put into the aggregation function with respect to the uncertainty handling on component-level. The implementation is therefore considered as verified at this point.

System-level uncertainties

The system-level uncertainty test is similar to the one conducted in Section 3.5.2, but with additional UD_{PHM} being considered during the prediction times. Therefore, the UD_{PHM} is again artificially set to decrease with later prognoses times t_P similar to the experiment shown in Figure 3.23. From the DHRM it is expected that the uncertainties from component-level are propagated through to system-level, as specified in Section 3.3.3.

In Figure 3.24, the results of this experiment are shown. It can be seen, that in contrast to the results shown in Figure 3.21, each state prediction is represented by a family of CDFs, according to the MCS sampling size. Similar to the results from the component-

level verification test, the uncertainty bounds become narrower for later predictions. The described characteristic is reasonable and complies to the expectations.

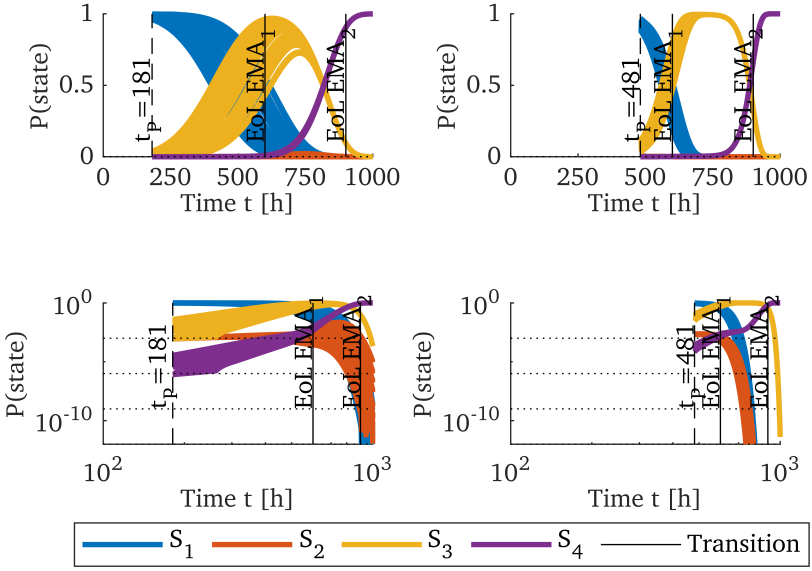


Figure 3.24.: Predicted system states for different prediction times t_P with uncertainties (linear and log scale visualization)

3.6. Results and conclusions

Within this section the implementation of the proposed method was verified against generic and reference data.

With respect to the initially stated requirements from Table 3.1 it can be said that the proposed method is able to include dynamic changing prognostics data. This is realized by incorporating the RUL-CDF for distinct prediction times into the fault tree model and recalculating it. The outcomes of this procedure are adapted component failure probabilities

as presented in Figure 3.20. This directly addresses requirement R1. Further, the method is able to account for distinguishable system states in terms of performance or safety based on the individual failure combinations of the system's components. Realized by a state-space model, the probabilities of all (possible) system states are predicted for different prediction times. From Figure 3.21, it can already be concluded that this has a significant effect as future system states as well as their transition times become predictable. This feature addresses R4 and builds the basis for a decision support as required by R2. Further, the method implements a scheme that allows to define uncertainties that are associated with model parameters (UD_{REL}) as well as with prognosis algorithms UD_{PHM}), which complies to R5. Finally, the overall method is constructed in a generic way, allowing different systems to be modelled. By using the fault tree model on component-level and a state-space model on system-level, it aligns to existing reliability modelling frameworks. Further, conventional reliability measures such as failure rates can be integrated into the model to account for those components, where a monitoring by means of a PHM system is not available. This corresponds to the requirements R3 and R6.

With the end of this chapter, the DHRM approach has been conceptually defined, all required software modules have been implemented and the resulting DHRM has been tested and verified with simplified input data. The initially stated requirements are considered to be fulfilled, and the approach is ready to be applied to a real system for further evaluation.

4. Application of the method to an UAV actuation system

Within this chapter, the developed aggregation method from Chapter 3 is applied to a current use-case from the aerospace discipline to outline and prove its applicability. The use-case and system under study will also build the basis for the further evaluation of the method in the remainder of this work.

The structure of this chapter and the application of the aggregation method is as follows: At first the chosen use-case and the considered technical system are introduced in Section 4.1. Afterwards, the system is further analyzed based on a predefined process regarding its failure modes and behavior. Based on this, two system models are derived in Section 4.2. The first is functioning as a reference model without considering any PHM data at all, while the second is defined to allow time-variant input data, as suggested for the DHRM approach. In Section 4.3 the required data for both models is created, before first results are obtained and discussed in Section 4.4.

4.1. Use-case description and relevance

According to two current outlook studies from the European Union Aviation Safety Agency (EASA) and the American Federal Aviation Administration (FAA), drone usage and operation is expected to rapidly increase within the next decades. As an example, the EASA predicts a cumulated fleet of over 400 thousand unmanned aerial vehicles (UAVs), used commercially and by the government[1] by the year 2050. The envisaged drones reach from small surveillance drones to mid-sized and large cargo drones, which conduct missions in visual line of sight (VLOS) as well as beyond visual line of sight (BVLOS). Further, the EASA estimates the European drone market to become a multi-billion euro market, with

[1] Not considered in this number are military drones and those for leisure (private / hobbyist drones)

an annual growth between 10 and 15 billion Euro by 2035, respectively 2050, creating over hundred thousand new jobs. A great demand for drones is seen in the agriculture, energy, delivery as well as public safety and security sector, where drones could be used already in near future to enable new business models, increase efficiency or replace more expensive solutions such as helicopters [Fed16; SES16]. Accordingly, UAVs are expected to bring a major impact to our society and economics in the upcoming years. However, to unlock the full potential their safety and effectiveness has to be thoroughly assured in order to have this technology accepted by society and to be also economically beneficial.

4.1.1. Description of the Quad-Cruiser, a hybrid drone

A special kind of drones are so-called *remotely-piloted aircraft systems* (RPAS), which are controlled by a remote pilot. An exemplary UAV of this group is Airbus' Quadcruiser (QC). With a maximum take-off weight of approximately $500kg$, the QC is realized in a hybrid configuration with eight electric redundant lift motors for vertical take-off and landing combined with a fixed wing system with a pusher motor at the rear to allow an aerodynamic flight [MMT18]. The QC is especially designed for search and rescue, surveillance and inspection (e.g. pipelines) missions, where the aircraft needs to fly long distances, while also having the capability for hoovering and a vertical take-off and landing (VTOL). A schematic illustration of the QC is given in Figure 4.1.

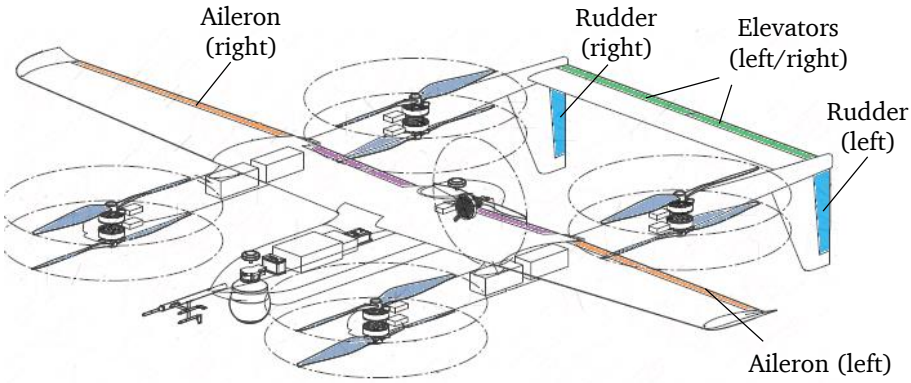


Figure 4.1.: Illustration of Airbus' Quadcruiser with the control surfaces being highlighted [MMT18][2]

The QC RPAS is considered being a complex system as it fulfills many of the stated characteristics as given in Section 1.1. In that way, the QC

- is composed of different systems and subsystems, which are even geographically distributed as the pilot remains on ground,
- has two different operating modes, the *aerodynamic flight* as well as *vertical lift* (hovering) mode,
- combines different subsystems from different technical domains,
- has embedded uncertainties due to high variability of possible flight missions
- and includes a high level of automation with a remote operator.

Especially the different operating modes as well as the remote pilot of the QC, make this system an interesting candidate for the application of the proposed dynamic hybrid reliability model, which is due to the following reasons:

1. Realized in a hybrid configuration, the QC has an inherent redundant setup in which, at least to some degree, the operating mode can be changed in the occurrence of a subsystem failure.
2. Being remotely piloted, the pilot in command relies on an automated and accurate system feedback about the QC's current capabilities to incorporate that information into his decision-making during the current mission.
3. Considered as a multirole drone, the QC is expected to be operated in varying system scenarios, leading to different degradation patterns. From an operators point of view it is therefore interesting to precisely estimate the remaining performance of an individual aircraft in order to plan upcoming missions and schedule maintenance tasks.

4.1.2. System under study – the Control Surface Actuation System (CSAS)

With the above considerations the Control Surface Actuation System (CSAS) of the QC, as the main control system during the aerodynamic flight phase, is chosen as the system under study in this work. This actuation system itself is part of a larger closed-loop control system of the aircraft as depicted in Figure 4.2. Key elements of this control-loop include the flight

[2]With kind permission of the authors of [MMT18].

control computer, the actuation system, the sensors as well as external disturbances and the resulting aircraft dynamics. The flight control computer receives a reference input from either the pilot in command or a flight path controller (not shown in the figure) and compares this to the system feedback obtained from sensors. The calculated output is then sent to the actuation system, which translates the information into acting forces. These forces together with external disturbances lead to the overall aircraft dynamics. For this work, it is chosen to focus on the actuation system, neglecting the sensors and flight control computer. The rationale for this decision is that these components do not cause a step-wise change in system performance. Both the FCS and the sensors provide digital outputs. Therefore, omission of these systems results in a complete loss of control of the aircraft. This is different for the actuation system, which is characterized by different degradation and performance states. Depending on the combination of failures, the aircraft can still be controlled, but is limited in its maneuverability and actuation forces. Thus, the application of the previously introduced DHRM makes much more sense for the actuation system itself.

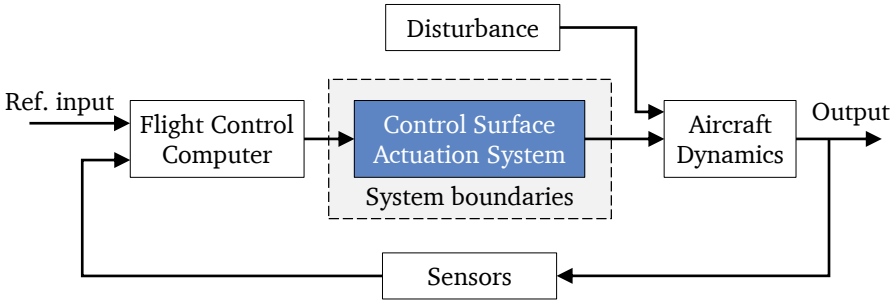


Figure 4.2.: Block diagram of the closed flight control-loop with the system under study, the Control Surface Actuation System, being highlighted

The CSAS is composed of six identical electromechanical actuators, which are used to actuate six independent control surfaces to generate the pitch, roll and yaw moments required to control the aircraft. As there are always two EMAs used to act on the same axis, e.g. pitch, roll and yaw, the CSAS itself is inherently redundant and failure tolerant.

The used EMAs within the CSAS are rotary actuators completely embedded in a housing similar to a servo drive. For safety reasons, each motor has two brushless direct current (BLDC) motors, which are configured in a redundant setup and are connected by a single

gearbox. Thus, each EMA is still operable after one BLDC motor failure (electrically fail/op, mechanically fail/safe).

To allow a precision positioning each EMA is further equipped with three position sensors, which are governed by a voter monitor scheme to detect any failures. Finally, each EMA has two distinct electronic control units (ECU), which run on different power lines and bus systems two allow a redundant control of the actuator. In case of a failure event the active controller is automatically changed (hot standby). In Figure 4.3, a prototype EMA of the QC is shown, installed on a test-rig at the TU Darmstadt. On the left-hand side, the two connector cables for the redundant controllers can be seen, on the right-hand side the output shaft of the EMA is connected to an active load simulator. The break-out connectors in the center of the EMA have been additionally added to allow the monitoring of the motor currents and are not part of the commercial of the shelf product.

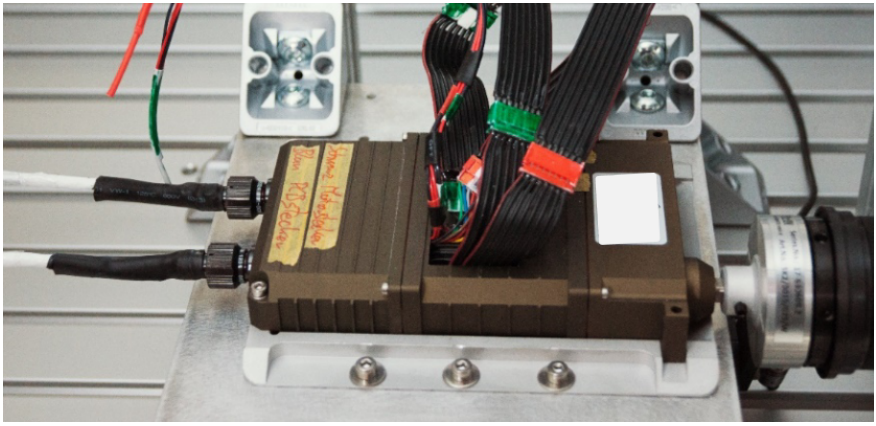


Figure 4.3.: A QC's EMA installed on a test-rig at TU Darmstadt

4.2. Application process for the Dynamic Hybrid Reliability Model

In the following section the developed DHRM is applied to the use-case with the Control Surface Actuation System (CSAS) of the Quadcruiser as the system under study. For the

application of the method, the following generic process is proposed, which is system independent:

The first step in this process is the identification of relevant parts and their dependencies as it is usually performed in most reliability assessments. This *pre-analysis* can be based on a part list or an FMEA from which the part's functions as well as their relevance can be concluded. Afterwards a *part-analysis* is conducted. The aim of this step is to identify the general degradation mechanism associated with each part and to define an appropriate modelling strategy. In relation to the DHRM approach this means to identify, if dynamic PHM data are available for a monitored part or not. Once this is defined, the *component-analysis* is conducted, in which the part dependencies of each component are identified. In this work, the DHRM approach and a Fault Tree model is used for that. In the fourth step, the *system-analysis*, the individual failure combinations of all system components are elaborated in order to derive the resulting failure modes and their severities. Once this is known, the DHRM can be defined and is prepared for the later calculation with input data.

The overall process is shown in Figure 4.4 and will be conducted in the following. At this point it shall already be noted that the further evaluation of the model will be based on simulated data. The decision to use simulated data rather than real system failure data in this work was made due to the following circumstances:

1. The QC as well as its CSAS are both prototypes themselves, which makes it difficult to obtain a large representative dataset of real failure and degradation data.
2. Due to the complexity of the system composed of six actuators, accelerated life-tests would have been time-consuming and expensive.
3. Based on simulated data a broad range of parameter combinations can be easily evaluated, including extreme value scenarios which are rare to find in real data.

As simulated data is used, the process steps in Figure 4.4 are also used to investigate the underlying degradation mechanisms and to motivate the considerations for the used degradation models and the generation of run-to-failure data in the upcoming sections.

4.2.1. Identification of relevant parts (pre-analysis)

As required for the DHRM approach, at first all relevant parts of the CSAS are identified. As previously described, the system consists of six identical EMAs. Each of them is composed of two controllers, two BLDC motors, one connected gearbox as well as three identical rotary position sensors. All of these parts are functional relevant for each EMA and thus

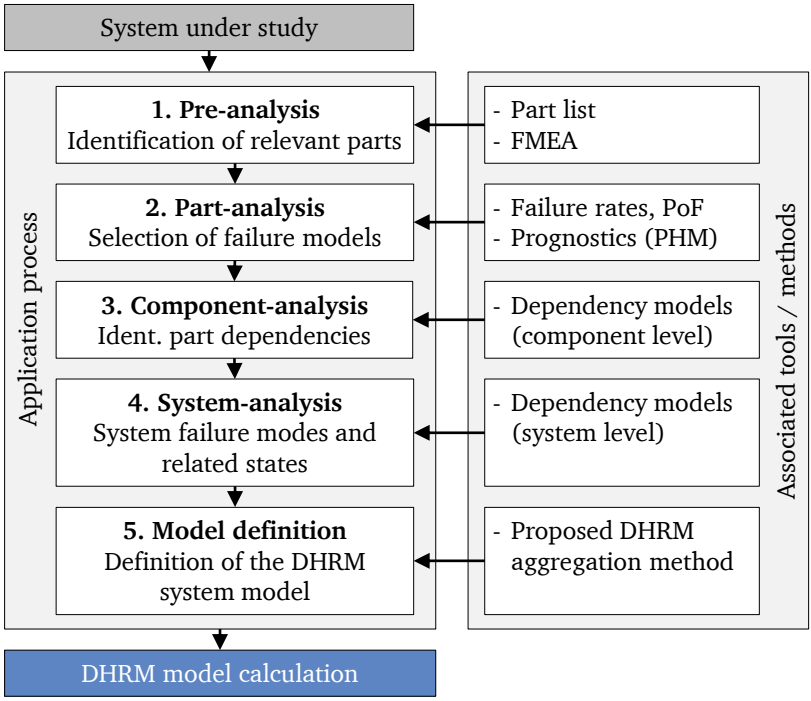


Figure 4.4.: Defined process and associated tools and methods to develop the required DHRM for a given use-case / technical system

for the CSAS to work correctly. Still there also a few parts that are neglected in this work. This includes the housing of each EMA, as it is assumed to be over dimensioned with comparably much lower failure rates compared to all other items, as well as any wiring within and between the EMAs. The latter is argued as a wiring failure will lead to a lost controller, motor or sensor and thus can be attributed to those part's failure models.

4.2.2. Selection of failure models (part-analysis)

In this step, the identified component parts are further analyzed with respect to their degradation mechanisms and suitable monitoring schemes. This builds the basis to choose representative failure models for the later simulation of the system.

Electronics (controllers and sensors)
A portion of parts used in each EMA are electronics. This includes both controllers and the three identical position sensors. The controllers and sensors play a crucial role for the EMA as they are used to actively control the position of the servo. As stated in Chapter 2, monitoring of electronic parts can be cumbersome because of their tiny size and high level of integration. Finally, most sensors used for health monitoring require electronic-rich parts themselves, which makes it further impractical to apply PHM to electronics. Nevertheless, a few options have been investigated in literature and research. An overview of applicable methods is given in [PK18] and [Pri+17]. Many of the described methods are based on the Physics of Failure (PoF) approach as outlined in Section 2.1, which can be used to describe the general mechanisms which cause an electrical item to fail. They are suited to support a design for reliability but do not cover an active feedback loop assessing the current health level as it is required for an individual part's prognosis. Other methods, being discussed in research, include so-called *canary devices* [MOP12; Das+10]. These additionally integrated circuit parts are predetermined breaking points in an electrical form and usually consist of small conducting paths, which are intended to break for high stress.

Although these concepts exist, it is still impractical for most applications to implement an active monitoring scheme for electronic parts as it is technically or economically not feasible. Therefore, and in compliance with the proposed aggregation method, it is considered in this work that these parts are not monitored by a PHM system and must be modeled by means of conventional reliability methods based on failure rates. Nevertheless, as failure rates can significantly vary for changing environmental factors, it is proposed to allow for a dynamic adaption of failure rates.

Gearbox
A common cause for gear failure is bending and contact fatigue, which results from large amount of cycles and, especially for servo actuators, the alternating direction of rotation [ZZX16; Err02]. This continuous wearout leads to (micro-) pitting, spalling and cracks, which causes an increased acoustic noise and vibrations. Fatigue life prediction has been studied in literature over a long time and can be broadly split into models with a constant load as well as those assuming a varying amplitude. Representatives of the first group include the *Wöhler-curve*, which is based on a stress-life $S - N$ curve, as well as the

Paris-Erdogan-law, a model used to describe the growth-rate of a crack. Regarding the varying load models, the *Miner-Palmgrem model* is widely accepted [HL97]. The hypothesis behind this linear damage model is, that each material can absorb a certain amount of accumulated energy, which is directly connected to the stress it takes. In this model, the ratio between observed cycles n and the total number of cycles N are accumulated over all (distinct) stress levels i to describe the current damage fraction D:

$$\sum \frac{n_i}{N_i} = D \tag{4.38}$$

For $D = 1$ the material is considered to have failed. In this way, the correlations from the $S - N$ curve for static amplitudes can be extended for varying loads, which is more realistic in most usage scenarios.

Monitoring schemes for gears have been widely studied in the literature and can be mainly realized by vibration monitoring or load measuring [Lee+14; Lan01]. For the EMA in this use-case it is therefore assumed that a gearbox monitoring is available. Further, a linear degradation characteristic is considered as main degradation mechanism as given by the Miner's rule for this part.

Motors
BLDC motors are very efficient and powerful electrical motors, which is the reason why they gain more and more importance in the aerospace domain. As given in [DSK11; SSG14] most relevant motor failures are bearing (41%), stator (37%) and rotor (10%) related failures.

The most often found failures for BLDCs are bearing failures. Bearings are used to hold the rotating rotor bar in position. Due to excessive cycling, mechanical overload as well as environmental aspects such as high humidity (corrosion), bearings will deteriorate over time [OSP12]. Typical failure modes include seizure, spalling or brinell marks, which will cause increased vibrations and noise and can propagate till a complete collapse or blockage of the bearing.

Next to the bearings are stator faults, rated as second most occurring failures. Stator faults are mainly winding shorts, which can be further split into coil-to-coil, phase-to-phase, coil-to-ground and turn-to-turn faults. These faults occur due to high temperatures or vibrations, which cause the thin insulation film of the coils to deteriorate [SSG14]. As a result the overall performance of the motor will be reduced as the magnetic field strength is decreased.

The third most often failure includes rotor faults. These include static eccentricity, dynamic eccentricity and flux disturbances due to defects of the used permanent magnets [LHH03]. While the first two failure modes originate from external loads, a misalignment or bent rotor shaft, the latter occurs due to high temperatures, physical damage and aging effects [UH18]. All together those failures will result in vibration and acoustic noise due to the imbalanced rotation and magnetic field. A (partial) demagnetization of the used magnets will further reduce the overall efficiency of the motor.

Being an often used item, the monitoring of electrical motors, including BLDCs, have been widely studied in the literature. Common diagnosis methods to assess the current level of degradation of these parts are summarized in [DSK11], [Ise11] and [Cho+19]. Besides approaches based on an observer model (physical model) as described in [Ise11], in recent years more and more data driven methods have evolved, which use machine learning and pattern recognition techniques to assess the current level of degradation from online data. These methods are based on a signal analysis, which includes vibration, acoustic and torque monitoring as well as the motor current signal analysis (MCSA) [Cho+19]. In Table 4.1, an overview of the most important failure modes and possible detection schemes for BLDC motors is given.

From the above insights it is concluded that a BLDC monitoring in terms of prognostics is technical feasible. Further research papers such as [SJ20] and [Xua+17] describe non-linear degradation schemes for BLDC motors. Therefore, in this work both BLDC motors are considered to be monitored by means of a PHM system. The underlying degradation mechanism is assumed to follow an exponential failure model as suggested in literature.

4.2.3. Identification of part dependencies (component-analysis)

With the parts and their corresponding failure models identified, the CSAS components are analyzed next. With the system being composed of six identical EMAs, this analysis is carried out only once.

As stated before, each EMA has two controllers in a hot standby configuration. The complete EMA thus will be fully functional unless both controllers have failed. Further, there are two identical BLDC motors installed in each EMA. Being connected to the same axis, the EMA is still functional after the first BLDC motor failure and will continue to work until both motors have failed. The connected gearbox is the link between both motors and the output lever of the EMA. Being singular, the EMA will fail as soon as the gearbox becomes non-functional e.g. blocked or torn off. The remaining three position sensors are configured in a voter-monitor scheme in which the sensor signals are compared to each

Table 4.1.: Failure modes, effects and detection methods for common BLDC failures

	Failure mode	Failure causes	Fail. mechanism	Detection methods
Bearing	Seizure	Thermal overload	Lubricant deterioration	MCSA, vibration analysis
	Spalling	Cyclic loading	Fatigue	
	Small furrows	Moisture	Corrosion	
	Brinell mark	Mechanical overload	Yielding	
Stator	Winding shorts	Thermal or mechanical overload	Aging of insulating material	MCSA, Direct Flux Monitoring, Temperature Monitoring, Current Voltage Model
Rotor	Eccentricity (static)	Static load	Fatigue	MCSA, vibration analysis, Direct Flux Monitoring
	Eccentricity (dynamic)	Misalignment, bend rotor	Fatigue	
	Partial demagnetization	Thermal stress	Aging of magnets	MCSA, Direct Flux Monitoring, Current Voltage Model

other being able to detect and discriminate any faulty sensor. This concept works with at least two sensors being intact.

Accordingly, the EMA's part dependencies are modelled by a combination of *AND* blocks (motors and controllers), one *2-out-of-3 block* for the sensors, as well as one large *OR* connection considering all grouped parts together. The resulting dependency model in form of a Fault Tree model is shown in Figure 4.5. Those parts considered to be monitored by PHM as outlined before are highlighted.

For the sake of completeness, it should be noted at this point that in this work it is assumed for simplicity that the failure of sensors or controllers are always detected correctly and in time by the system.

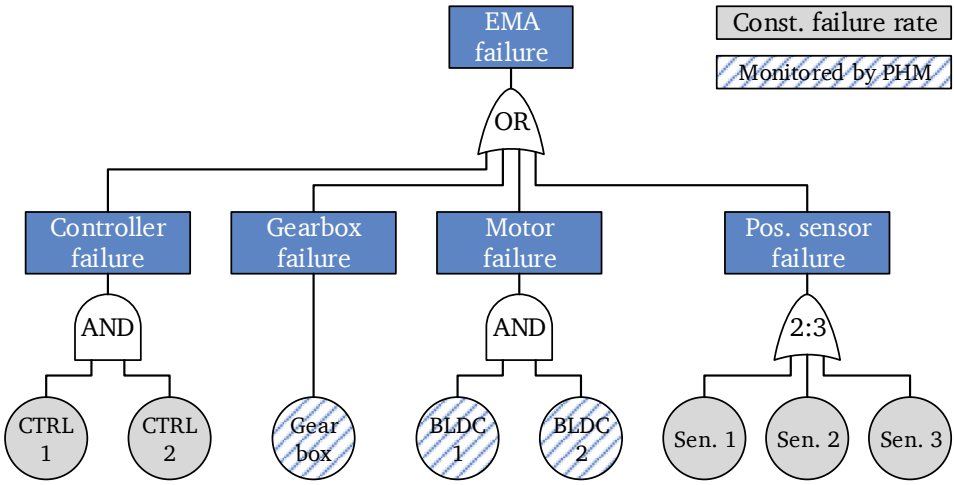


Figure 4.5.: Resulting Fault Tree model used to model the dependencies of all considered parts within the EMA

4.2.4. System failure modes and related states (system-analysis)

Excluding the power supply as well as the flight controller from the considered system, the CSAS consist of six EMAs. As each EMA is self-contained and independent of the others, they are considered as the components of the CSAS. With always two EMAs acting on the same steering axis, the system consists of a left/right aileron EMA, a left/right rudder EMA as well as a left/right elevator EMA as shown in Figure 4.1.

With the assumption that each actuator can fail individually, this leads to exactly 2^6 failure combinations respectively system states. The overall number of these inherent system states can be greatly reduced by grouping states with same severity together. This procedure shall be illustrated with an example: We consider a failure affecting the x-axis maneuverability of the aircraft. This will be the case if either the left or the right aileron EMA of the aircraft has failed. Although being distinct failure combinations from a technical point of view, the effect as well as the severity of each failure mode will be the same, leading to a reduced control moment around the x-axis. Thus, it is reasonable to represent both system states as one:

$$P\left(S_{xaxis}\right) = P\left(S_{aileron,left}\right) + P\left(S_{aileron,right}\right) \tag{4.39}$$

This concept can be further extended as from an operator's point of view it might not be relevant, which exact axis is degraded as the aircraft's overall maneuverability is already affected.

In Table 4.2, the initial system states have been broken down to a manageable amount. In total there remain five distinguishable meaningful system states, which are differentiated by the amount of axis affected by degradation and the associated effect. The most severe state therefore is S_5, which refers to a loss of control, which will happen as soon as two EMAs acting on the same axis fail. The ideal and desired state is S_1, where no failure is present and the CSAS has its full specified operating power. States S_2 to S_4 describe those states, where one or more axes are affected by exactly one failed EMA. Thus, the overall steering power on the affected axes are reduced as only one EMA is working. Still the aircraft remains maneuverable. Finally, there is state S_4, which describes a single failed EMA on each axis. As it will be shown later, this state is very unlikely.

Table 4.2.: Derived (grouped) system states for the SCAS

State	Failure mode	Effect	Criticality
S_1	no failure	none	none
S_2	1-axis degraded	reduced performance	low
S_3	2-axis degraded	reduced performance	medium
S_4	3-axis degraded	reduced performance	medium (unlikely)
S_5	n-axis failed	loss of control	high

4.2.5. DHRM and reference model definition

Based on the previous system analysis the DHRM as well as a reference model are constructed. Both will be used throughout the remainder of this work to compare and evaluate the proposed DHRM.

Each model is composed of six EMA Fault Trees, as described in Section 4.2.3. Each of them is modelled with the custom Simulink Fault Tree library from Section 3.4.1. Both controllers and all three sensors are modelled as basic events with fixed failure rates. The two motors as well as the gearbox are modelled with a dynamic input gate to allow

time-variant input data. The resulting Simulink Fault Tree model is given in the Appendix C.1.

All six EMA Fault Tree models are then connected to the state-space model, which incorporates the different distinguishable system states as elaborated in the previous Section 4.2.4. The EMA Fault Trees as well as the state-space model are embedded in the Aggregation Tool as described in Section 3.4.2 and thus provides all necessary in- and output ports to run the simulation. For each simulation, the input data as well as the static parameters can be configured and modified.

In Figure 4.6 the structure of both models is illustrated. It shall be noted, that they rely on the same structure, but are connected to different input streams. While the reference model will be feed by a constant failure distribution, the DHRM will receive frequently updated failure probabilities for the motors as well as the gearbox in form of dynamic changing RUL-CDFs.

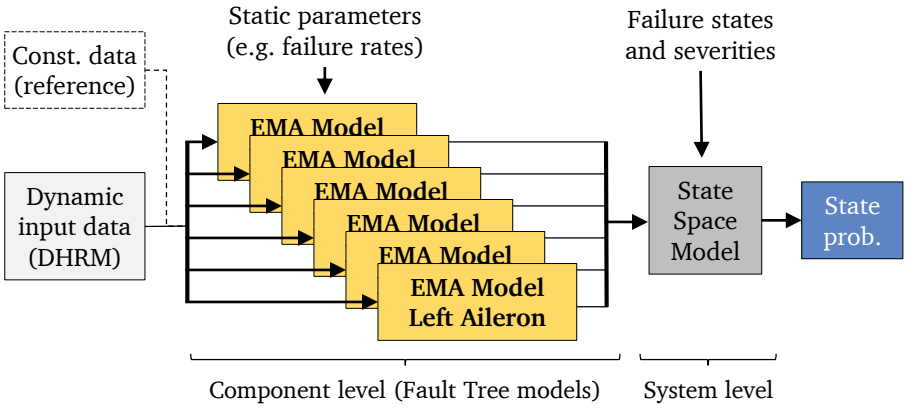


Figure 4.6.: Illustration of the used reference and DHRM

4.3. Simulation environment

To test and evaluate the proposed DHRM, one would require one (or ideally more) existing systems equipped with a functional PHM system that creates RUL-CDFs for the degrading

parts. For those parts subject to random failures (e.g. electronics), failure times would have to be logged and statistically evaluated to derive the corresponding failure rates. As outlined at the beginning of this chapter, this is an expensive and cumbersome task if no existing data is available. Therefore, it is chosen to simulate the required input data for the DHRM within a simulation environment for this thesis and to compare the defined DHRM against a reference model based on that data.

In Figure 4.7, all elements of the simulation environment are shown in comparison to a real world application. The used models, the DHRM respectively the Weibull reference model, are highlighted in the figure. The simulation environment thus is used to substitute the degradation process of a real-world application with empirical data respectively generated degradation data depending on the part's failure behavior.

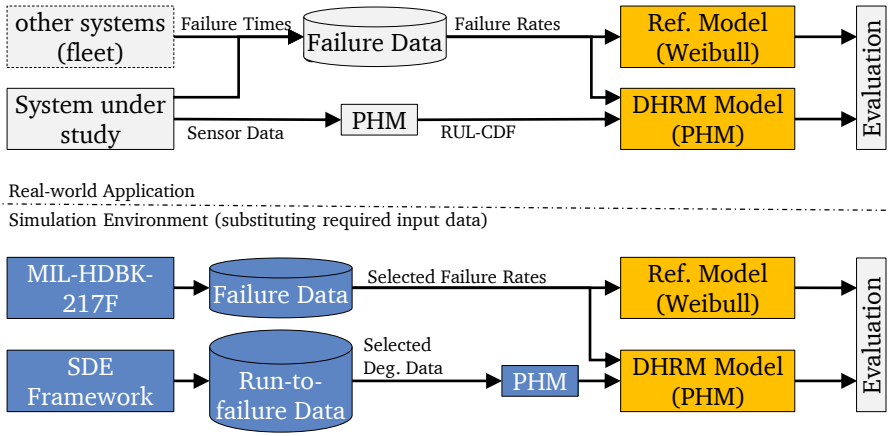


Figure 4.7.: Used simulation environment to generate and describe required input data for the later evaluation of the DHRM and comparison against a reference model

The overall data generation is based on the following approach: The electronic parts of the EMA, the controllers and sensors, are expected to fail on a rather random basis and thus are modelled with failure models as given in the MIL-HDBK-217F. To allow the consideration of different usage scenarios, environmental factors are further varied to create a broad parameter data basis. This will be further outlined in Section 4.3.1. For the motors and the gearbox, which are expected to degrade in a continuous manner with measurable

health indicators, a common framework based on stochastic differential equations (SDE Framework) is introduced in Section 4.3.2 that will be used to create realistic run-to-failure data. This data will then be evaluated in a statistical manner for the reference case using a Weibull distribution in Section 4.3.3. Further, a prognosis algorithm is developed and applied to the data in Section 4.3.4 to create the required dynamic RUL-CDFs for the DHRM approach.

4.3.1. Selection of failure rates

In order to define realistic failure rates for the electrical parts of the CSAS, the military handbook MIL-HDBK-217F is consulted. The two servo controllers as well as the position sensors are represented by exponential failure models in analogy to the servo model given in [Yuy+15]. The failure rates used to model these parts are derived from the MIL-HDBK-217F as presented in Section 2.2.2. Both controllers and the three positions sensors are considered to be identical.

Position sensors The sensors of the servo are modelled as rotary encoder as given in MIL-HDBK-271F Section 12.2. The part failure rate is given as:

$$\lambda_p = \lambda_b \pi_S \pi_N \pi_E \frac{Failures}{10^6 Hours} \tag{4.40}$$

The factors $\pi_S = 2.25$ and $\pi_N = 1.4$ are taken from the military handbook and product specific. They remain constant during operation. The environmental factor π_E however is subject to changes during the mission-phase. The base failure rate λ_b finally accounts for the current temperature T and is defined as:

$$\lambda_b = 0.00535 \exp\left(\frac{T + 273}{334}\right)^{8.5} \tag{4.41}$$

In Figure 4.8, the general parameter-space for the sensor's failure rate is shown for varying temperatures and different environmental classes based on Equation 4.40.

Servo controllers
In analogy to the position sensors, the failure rates for both controllers are derived based on the MIL-HDBK-217F. Both controllers are considered to have a CMOS design in form of a digital gate array, a common design for microcontrollers and chips. The data used in the model is taken from the given example in MIL-HDBK-217F in Section 5.13. As given in

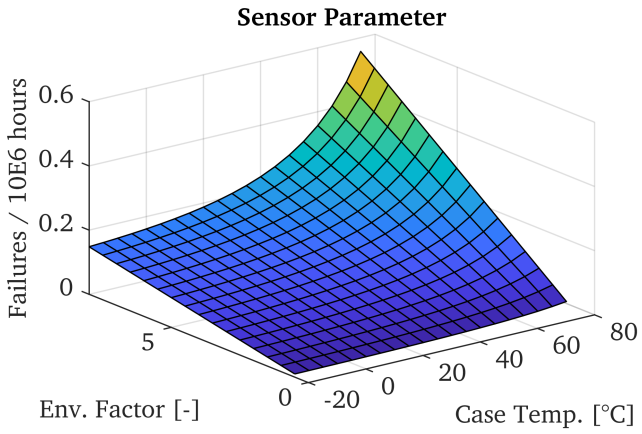


Figure 4.8.: Failure rate of the position sensor for different environments and temperatures

the handbook, the failure rate for such microcircuits is predicted based on the following function:

$$\lambda_b = (C_1 \pi_T + C_2 \pi_E)\, \pi_Q \pi_L \tag{4.42}$$

In this function, the parameters $C_1 = 0.02$ and $C_2 = 0.011$ account for the complexity of the part. The parameters $\pi_Q = 3.1$ and $\pi_L = 1$ are the quality and learning factor and considered to be constant throughout the flight mission. The parameters π_T and π_E account for the case temperature and the current environment the part is used in. These parameters are foreseen to be adjusted to the current mission. With the above formulation the failure rate space can be drawn, which is shown in Figure 4.9 for different environments and temperature ranges.

Scenarios and parameter definition
For the further application and the conducted simulations, a set of failure rates are selected according to different scenarios, which directly affect these parameters.

The first parameter to vary is the temperature, which greatly affects the derived failure rates. In this work, the following different scenarios are assumed: *Scenario I* considers a cold environment around $T = 0°C$, *scenario II* a mild environment around $T = 20°C$,

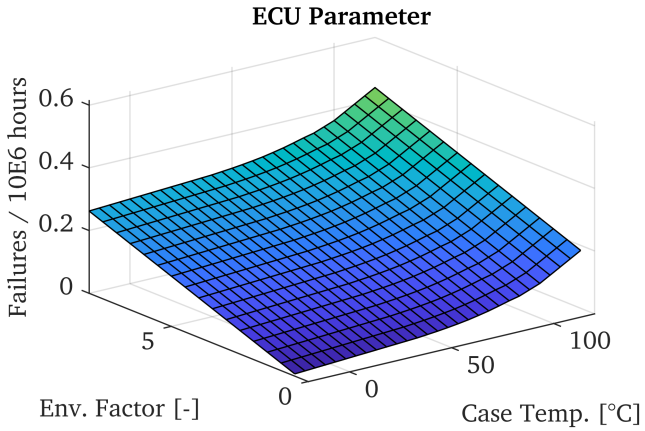


Figure 4.9.: Parameter space of the ECU's failure rate in dependence of the temperature and environment

scenario III a hot environment with $T = 40°C$ and *scenario IV* an extreme one with temperatures of $T = 60°C$. The specified temperatures further correspond to the case temperature of the considered parts.

Besides the temperature, also the overall flight mode shall be considered. According to the MIL-HDBK-217F it is distinguished between installations on a fixed wing aircraft ($A_{IC}, \pi_E = 4.0$) and installations on rotary wing aircraft ($A_{RW}, \pi_E = 8.0$). The considered hybrid UAV in this work can be operated in two modes, aerodynamic flight and vertical take-off and landing, thus these two modes shall be distinguished further in the following to account for the additional vibrations induced by the lift-rotors during hovering. The missions are therefore classified according to the ratio between aerodynamic flight time and hovering. The environmental factor thus is chosen accordingly to vary between 4 and 8. Based on these considerations, the following parameter sets are drawn from the above equations for the different operation modes and environments, summarized in Table 4.3 and 4.4. It shall be noted, that the generated failure rates vary between $\lambda_{Sen} = 0.08 \times 10^{-6}/h$ and $\lambda_{Sen} = 0.35 \times 10^{-6}/h$ for the sensors and $\lambda_{ECU} = 0.13 \times 10^{-6}/h$ and $\lambda_{ECU} = 0.3 \times 10^{-6}/h$ for the ECU.

Table 4.3.: Sensor failure rate selection for different scenarios in failures / 10E6 hours

Environment	cold $T = 0°C$	mild $T = 20°C$	hot $T = 40°C$	extreme $T = 60°C$
Aerodynamic flight $\pi_E = 4$	$\lambda_{Sen} = 0.08$	$\lambda_{Sen} = 0.09$	$\lambda_{Sen} = 0.12$	$\lambda_{Sen} = 0.18$
Mixed flight mode $\pi_E = 6$	$\lambda_{Sen} = 0.12$	$\lambda_{Sen} = 0.14$	$\lambda_{Sen} = 0.18$	$\lambda_{Sen} = 0.27$
Hovering only $\pi_E = 8$	$\lambda_{Sen} = 0.16$	$\lambda_{Sen} = 0.18$	$\lambda_{Sen} = 0.24$	$\lambda_{Sen} = 0.35$

Table 4.4.: ECU failure rate selection for different scenarios in failures / 10E6 hours

Environment	cold $T = 0°C$	mild $T = 20°C$	hot $T = 40°C$	extreme $T = 60°C$
Aerodynamic flight $\pi_E = 4$	$\lambda_{ECU} = 0.13$	$\lambda_{ECU} = 0.14$	$\lambda_{ECU} = 0.15$	$\lambda_{ECU} = 0.16$
Mixed flight mode $\pi_E = 6$	$\lambda_{ECU} = 0.2$	$\lambda_{ECU} = 0.21$	$\lambda_{ECU} = 0.22$	$\lambda_{ECU} = 0.23$
Hovering only $\pi_E = 8$	$\lambda_{ECU} = 0.27$	$\lambda_{ECU} = 0.28$	$\lambda_{ECU} = 0.28$	$\lambda_{ECU} = 0.30$

4.3.2. Simulated degradation data based on SDE

For the electromechanical parts of the CSAS, the BLDC motor as well as the gearbox, it is chosen to generate artificial run-to-failure data. Therefore, after deriving the theoretical background, a stochastic differential equation (SDE) framework is used to build a common database with a large set of failure curves.

In literature there exists different approaches on how to create artificial and realistic failure data based on mathematical models. A common approach is to use models that generate point estimates, respectively distributions of point estimates, which describe the event of failure and is similar to the conventional reliability concepts, working with distributions and statistics. However, another approach is the artificial generation of a degradation trajectory, which describes the degradation process over time (or a number of cycles). This is much closer to the concept of PHM, as it is capable of describing a health state of a part at any time between its first use and the time of failure. Thus, the latter, trajectory-based approach, is chosen in this work for the artificial generation of run-to-failure data.

As given in [MHE04], trajectory-based models can be further split into parametric and stochastic models. The parametric models are often referred to as *general path* (degradation) models, as defined in [LM93]. This model type uses a combination of fixed and random parameters. While the first are used to form the general shape of the degradation curve and thus the overall degradation behavior of the considered part, the latter are used to introduce some variance accounting for the part-specific factors, such as individual usage, fabrication tolerances or deviations in the material. Being a parameterized model, failure times are predicted by transposing the model function. For some special simple cases the failure time distribution can then even be derived in a closed form. The general path model is defined as

$$x_{ij} = \eta(t_k, \mathbf{\Phi}, \mathbf{\Theta_i}) + \epsilon_{ij}, i = 1, 2, \ldots, n \tag{4.43}$$

In Equation 4.43, the degradation of the i-th part at time step t_k is described by the general path function $\eta(t_k, \mathbf{\Phi}, \mathbf{\Theta_i})$, which is superposed by a standard measurement error $\epsilon_{ik} = N(0, \sigma^2)$. The general path function is composed of the fixed parameters $\mathbf{\Phi}$, and those subject to variances $\mathbf{\Theta_i}$.

An alternative to this deterministic modeling approach is to model the degradation as a stochastic process, in which the evolution of degradation is iteratively described. Commonly used time continuous processes include the *Wiener process*, also known as *Gaussian* or *Brownian motion*, the *Gamma Process* as well as the *Geometric Brownian Motion* (GBM)

Table 4.5.: Configuration of defined stochastic processes

Process	$a(\cdot)$	$b(\cdot)$	D_t
Wiener	$\mu \cdot dt$	σ^2	$D_t \sim \mathcal{N}(\mu, \sigma^2)$
Gamma	0	β	$D_t \sim \Gamma(\alpha, \beta)$
GBM	$\mu \cdot X_t \cdot dt$	$\sigma^2 \cdot X_t$	$D_t \sim \mathcal{N}(\mu, \sigma^2)$
GGP	$\alpha \cdot X_t \cdot dt$	$\beta \cdot X_t$	$D_t \sim \Gamma(\alpha, \beta)$

and *Geometric Gamma Process* (GGP). The use of these processes to model degradation has been studied in [LC04; PP05] and [Whi95]. The idea behind this approach is to model the degradation with small damage increments at discrete time points. These increments are drawn from defined distributions such as the normal distribution for the Wiener process or the Gamma Distribution for the Gamma Process. An important property of these processes is that the drawn increments are independent of any previous samples.

A general definition of such processes used for degradation modelling is described in [PP05] and can be formulated as stochastic differential equation (SDE) as given below

$$dX_t = a(X_t, t)dt + b(X_t, t)dD_t. \tag{4.44}$$

In Equation 4.44, D_t describes the underlying stochastic process, while $a(\cdot)$ and $b(\cdot)$ describe the general shape of the found solution. Here $a(\cdot)$ is the damage accumulation function, which accounts for any previous degradation. Mathematically this is also known as drift, as SDEs with $a(\cdot) > 0$ will tend to increase, while SDEs for $a(\cdot) < 0$ tend to decrease over time. Further, $b(\cdot)$ describes the general form of the used damage model. Depending on the initial condition and parameter selection, Equation 4.44 can be used to model various stochastic processes such as the Wiener, Gamma or Geometric Brownian Motion process. In Table 4.5, a general overview of parameter settings is given, on which the described processes can be modelled.

Generation of run to failure data
Based on the given overview, it is chosen to use the more generic SDE framework to generate run to failure data in this work. This is chosen as no exact model, as required for the general path model, can be stated at this point. The SDEs instead is implemented in a very generic way and used to sample different forms of failure curves. The implementation of Equation 4.44 is realized in recursive way as follows:

$$dX = a(X_k)dt + b(X_k)dD_t \tag{4.45}$$

$$X_{k+1} = X_k + dX + \varepsilon \tag{4.46}$$

Herein X_k describes the degradation of a component at time instance k. In addition to 4.44, the parameter ε was added to 4.46, which is used to account for any observation and measurement errors on the true degradation X. It is modelled as white noise with $\varepsilon = \mathcal{N}(0, \sigma_\epsilon^2)$, where σ_ϵ^2 can be used to artificially deteriorate the measurement's accuracy.

Based on the above formulations, different generic failure trajectories are generated and stored in a database, which is used as data input for the evaluation and analysis of the proposed method. The resulting run-to-failure *Degradation DB* is organized into four different failure groups $D_W, D_G,\ D_{GBM},\ D_{GGP}$ according to the underlying stochastic process. Further, each group contains multiple failure families $D_{i,j}$ with varying parameters to account for the overall degradation rate (drift parameters α and μ) and the general shape of the process (parameters β and σ^2). The observation noise variance σ_ϵ^2 is set equally for all families and groups. Finally, each family contains a set of 100 individual failure trajectories which represent the individual parts. In Figure 4.10, the generation of the Degradation DB is illustrated.

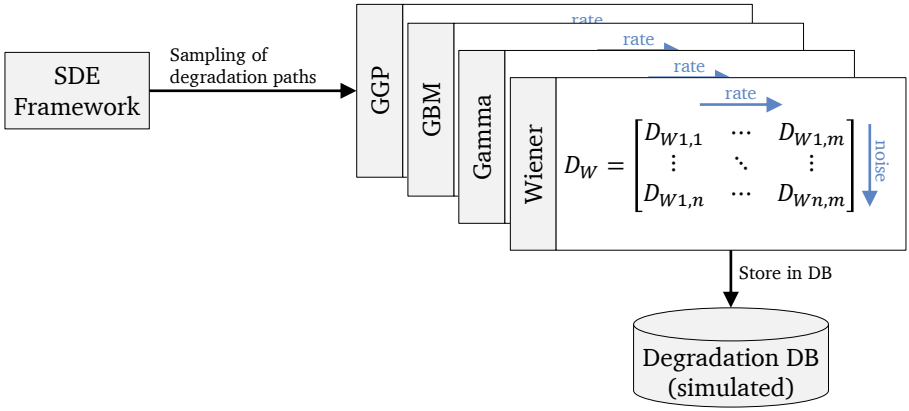


Figure 4.10.: Generation process of the simulated degradation data

In Figure 4.11, different failure families of the four groups are shown. It shall be noted that the degradation trajectories are normalized in a way that the failure event occurs for a degradation level of $x = 100$ with a median time to failure of 100 time steps k. Thus, this general data can be easily adjusted to various time and degradation scales.

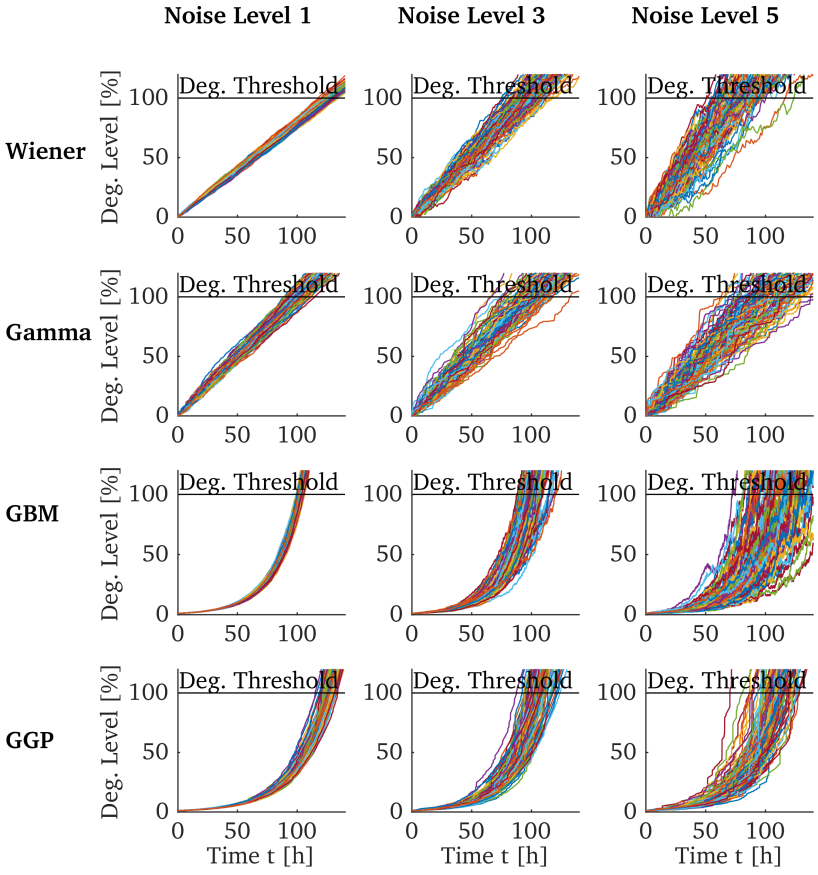


Figure 4.11.: Exemplary failure families of different groups (row-wise) and different noise levels (column-wise)

Selection of degradation data
With the artificially generated degradation data in the database, the ground truth failure trajectories for the further application are chosen. To avoid any predictions with training data, each dataset is split into a training, validation and evaluation subset with shares of 20%, 20% and 60%. While training data are only used for the training of any prognosis algorithms, the validation data are used for the performance assessment of the resulting algorithms. The remaining evaluation data are then available for the later application in the use-case.

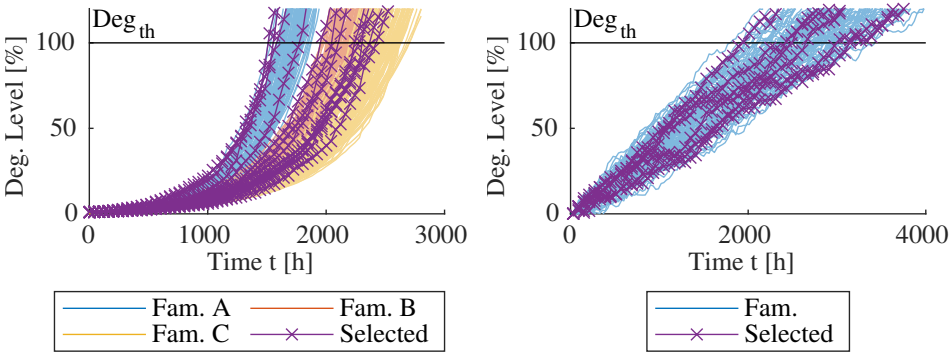
In Table 4.6, the general selection of data is shown. The data for the gearbox is selected from the Wiener Process model as it is expected to follow a rather linear degradation scheme as outlined in Section 4.2.2, while the data for the motors is selected from the Geometric Gamma Process, representing an exponential degradation behavior (compare Section 4.2.2). For the Gearbox, all degradation data are chosen from Wiener Process data with a degradation rate of 2 and a degradation noise of 4. The data for the motors are chosen from the Geometric Gamma Process data, from failure families with a degradation of 1, 2 and 3 having a degradation noise of level 2. As all datasets are normalized by default to fail with a median of $100h$, the time vector is further rescaled with a scaling factor υ, to better reflect a realistic failure time. For the motor data a scaling factor of $\upsilon = 20$ is used, while for the gearbox a factor of $\upsilon = 30$ is used.

Table 4.6.: Selected degradation data for the evaluation of the DHRM approach

Component	Failure Group	Family		Data Split (in %)		
		Rate	Noise	Training	Validation	Evaluation
Gearbox	Wiener Process	2	4	1-20	21-40	41-100
BLDC Motor	Geometric Gamma Process	1, 2, 3	2	1-20	21-40	41-100

Based on this concept samples are drawn for all components, which are used as ground truth data during the evaluation phase. For the Gearbox, six independent trajectories are randomly chosen from the Wiener Process data from the evaluation set, while for the motors, twelve trajectories are taken from the GGP as each EMA has two BLDC motors. In Figure 4.12, the selected complete failure dataset for both, the motor and the gearbox, are shown. Also, the selected ground truth data are highlighted.

With the ground truth data chosen, each failure trajectory is assigned to one component of the CSAS, which is done again on a random basis. In Table 4.7, the resulting EoL times for each component and part are given. Also, the overall failure time of each EMA is stated and the contributing EoL failure is highlighted.



(a) Motor failure curves (GGP)

(b) Gearbox failure curves (Wiener Process)

Figure 4.12.: Select failure trajectories set as true failure curves during the evaluation phase

Table 4.7.: Selected failure times for each EMA in [h]

Component	BLDC 1	BLDC 2	Gearbox	t_{EoL}	F Ord.
EMA Aileron left	1542	**1584**	1950	1584	1.
EMA Aileron right	2116	**2240**	2520	2240	3.
EMA Elevator left	1498	**1776**	3000	1776	2.
EMA Elevator right	**2276**	1944	2430	2276	4.
EMA Rudder left	2300	**2368**	3150	2368	5.
EMA Rudder right	2054	**2420**	2670	2420	6.

4.3.3. Weibull reference model (base-line approach)

With the input data being defined and in accordance with Figure 4.7, a reference model is defined in the following, which will be used to compare and benchmark the proposed DHRM against in the remainder of this thesis. The reference model is based on the assumption, that failure rates are known (as defined in Sections 4.3.1) and that samples from the run-to-failure data are available for a statistical evaluation. In that way, the reference model uses only statistical models as inputs without any time-varying data as it would be the case with any PHM system in place.

The reference model is defined as follows: Based on the rescaled failure data a statistical evaluation of the failure times is carried out. This evaluation is considered as base-line approach and is used to feed input data to the reference model. In comparison to a real world application, only a small portion of observations is considered to be available for the statistical evaluation. Therefore, the available data from the training subset (20%) is taken, while the complete dataset of EoL values is considered as the overall ground truth.

In a next step, a maximum likelihood estimator is used to fit the data against a predefined distribution. Because of its universality the Weibull distribution is chosen in this case. The estimated values are the scale and shape parameters η and β of the Weibull distribution as well as the expectancy value $E\left(t_{EoL}\right)$ and the 95% confidence intervals CI_{low} and CI_{high}. In Table 4.8, the estimated distribution parameters are shown together with the deviations from the ground truth distribution. It can be seen that even for a small subsampling size, the estimated parameters are close to those of the ground truth distribution. However, it can be also seen that the overall failure times are widely spread as shown in Figure 4.13. In a conservative scenario, where a high reliability of the components is required, it would be reasonable to choose the lower CI bound CI_{low} as the MTTF. In that case a mean value of roughly $730h$ of unused lifetime would be lost for the motor, respectively $860h$ for the gearbox, while having a 97.5% chance to avoid an unforeseen failure event. With respect to Figure 2.6, this is considered to be the available PHM potential for these parts.

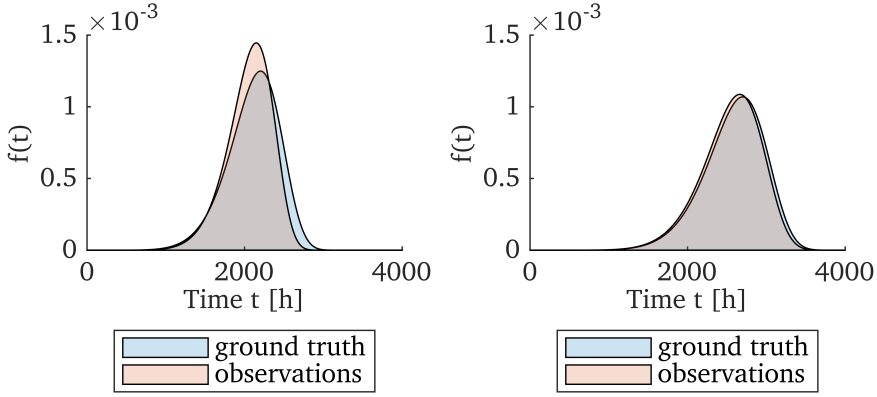
With the above evaluation, the reference failure models for the BLDC as well as the gearbox are derived. The resulting failure probabilities are

$$F^{*}_{BLDC}\left(t\right) = 1 - exp\left(\frac{-t}{2180}\right)^{8.5} \tag{4.47}$$

and

Table 4.8.: EoL characteristics of the used data for the BLDC motors and the gearbox

	BLDC			Gearbox		
	g. truth [h]	obs. [h]	dev. [%]	g. truth [h]	obs. [h]	dev. [%]
$E(t_{EoL})$	2108	2059	2.34	2586	2550	1.37
CI_{low}	1379	1415	-2.55	1727	1705	1.30
CI_{high}	2669	2541	4.78	3239	3193	1.42
η	2245	2180	2.90	2747	2709	1.39
β	7.55	8.50	-12.63	7.92	7.94	-0.23



(a) BLDC EoL

(b) Gearbox EoL

Figure 4.13.: End of life distribution for the BLDC and the gearbox - observations vs. ground truth

$$F^*_{gear}(t) = 1 - exp\left(\frac{-t}{2708.8}\right)^{7.94}. \tag{4.48}$$

4.3.4. Development of a GPR-based prognosis algorithm

As initially described and shown in Figure 4.7, a PHM algorithm is required that transforms the run-to-failure data into the RUL-CDFs, which are required by the DHRM. Therefore, this section is dedicated to the definition of a very basic prognosis algorithm that is used in the remainder of this work to create the required dynamic input data for the DHRM. The developed prognosis algorithm is based on a Gaussian Process Regression (GPR) model, which is used to learn the degradation-rate at distinct degradation levels from the observed data. This is a purely data-driven approach, similar to the concept shown in [PAK14].

The general concept of the GPR-based prognosis algorithm is shown in Figure 4.14. Based on historical degradation data, as described in Section 4.3.2, the underlying GPR model is initially trained. During the prognosis resp. validation of the algorithm, new degradation data is then fed into the model. Based on an MCS, n samples are then drawn to represent the current degradation x_k at the time of the prognosis t_P. These samples are then populated by the GPR model until all samples have reached a predefined degradation threshold x_{th}.

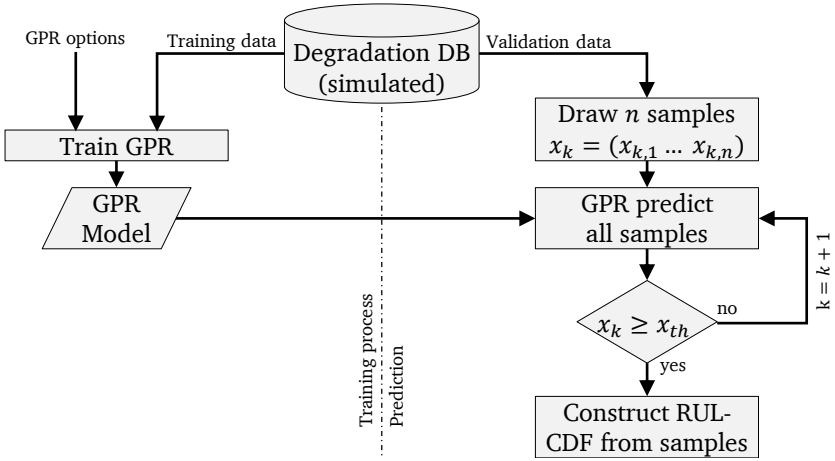


Figure 4.14.: Concept of the prognosis algorithm

After describing the basics of the GPR and how it can be used for predictions, a *simplified* and *improved* version of the above described concept are developed in the following and evaluated in terms of their performance.

Basics of the Gaussian-Process regression model

The GPR is a supervised learning algorithm based on the Gaussian Process (GP) that is used for regression and classification problems [RW06]. The GP itself is fully defined by its mean function $m(\boldsymbol{x})$ and its covariance function $k(\boldsymbol{x}, \boldsymbol{x}^{'})$. The process $f(\boldsymbol{x})$ thus can be written as

$$f(\boldsymbol{x}) \sim GP(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}^{'})), \tag{4.49}$$

in which

$$m(\boldsymbol{x}) = E\left[f(\boldsymbol{x})\right] \tag{4.50}$$

$$k(\boldsymbol{x}, \boldsymbol{x}^{'}) = E\left[(f(\boldsymbol{x}) - m(\boldsymbol{x}))(f(\boldsymbol{x}^{'}) - m(\boldsymbol{x}^{'}))\right]. \tag{4.51}$$

The aim of the regression problem is to find an arbitrary function $f(\boldsymbol{x})$, which maps a given input $\boldsymbol{x} = [x_1,\ x_2, \dots x_n]$ of size n to an observed output vector $\boldsymbol{y} = [y_1,\ y_2, \dots y_n]$. As in reality the true value of $f(\boldsymbol{x})$, e.g. the current degradation level, can often not be assessed directly or is subject to noise, observations $\boldsymbol{y}$ are considered to contain an additional noise term ϵ, which is assumed to be normally distributed with a zero mean $\mathcal{N}(\boldsymbol{0}, \sigma^2)$. Any observations are thus written as

$$y = f(\boldsymbol{x}) + \epsilon, \tag{4.52}$$

with $f(\boldsymbol{x}) = \boldsymbol{x}\beta$ for a linear model. The general distribution of possible solutions $\boldsymbol{f}_*$ that apply for an arbitrary input $\boldsymbol{x}_*$, can then be drawn from the GP as

$$\boldsymbol{f}_* \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{K}(\boldsymbol{X}_*, \boldsymbol{X}_*) + \sigma_n^2 \boldsymbol{I}). \tag{4.53}$$

Equation 4.53 is also known as the prior of the GP. Herein, $\sigma_n^2 \boldsymbol{I}$ describes the noise variance as initially induced by ϵ and $\boldsymbol{K}(\boldsymbol{X}_*, \boldsymbol{X}_*)$ is the kernel matrix, which is defined as

$$\boldsymbol{K}(\boldsymbol{X}_*, \boldsymbol{X}_*) = \begin{pmatrix} k(x_{1^*}, x_{1^*}) & \cdots & k(x_{1^*}, x_{n^*}) \\ \vdots & \ddots & \vdots \\ k(x_{n^*}, x_{1^*}) & \cdots & k(x_{n^*}, x_{n^*}) \end{pmatrix}. \tag{4.54}$$

The kernel matrix is an important property of the GP as it defines the general shape of possible solutions. As it can be assumed that output values y_i will be similar for predictor values x_i in proximity, the covariance is a measure of similarity among data points and governed by the used covariance function. Common covariance functions include the exponential, the squared exponential and the Matérn covariance function. In this work, a squared exponential kernel will be used, which is defined as

$$k(\boldsymbol{x}, \boldsymbol{x}^{'}|\boldsymbol{\Theta}) = \sigma_f^2 \exp\left[-\frac{1}{2}\frac{(\boldsymbol{x} - \boldsymbol{x}^{'})^T(\boldsymbol{x} - \boldsymbol{x}^{'})}{\sigma_l^2}\right]. \tag{4.55}$$

The kernel's parameter $\boldsymbol{\Theta}$ includes the standard deviation σ_f as well as the characteristic length scale σ_l and are optimized together with the function's coefficients β and the noise variance σ^2 during the training of the model. In order to account for any previous observations $\boldsymbol{X}$ and $\boldsymbol{y}$, the GP prior distribution (eq. 4.53) is conditioned to this training data, which leads to the GP posterior distribution given as

$$\boldsymbol{f}_*|\boldsymbol{X}_*, \boldsymbol{X}, \boldsymbol{y} \sim \mathcal{N}(\overline{\boldsymbol{f}}_*, cov(\boldsymbol{f}_*)). \tag{4.56}$$

In Equation 4.56, the mean and covariance are

$$\overline{\boldsymbol{f}}_* = \boldsymbol{K}(\boldsymbol{X}_*, \boldsymbol{X})\left[\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \sigma_n^2\boldsymbol{I}\right]^{-1}\boldsymbol{y} \tag{4.57}$$

$$cov(\boldsymbol{f}_*) = \boldsymbol{K}(\boldsymbol{X}_*, \boldsymbol{X}_*) - \boldsymbol{K}(\boldsymbol{X}_*, \boldsymbol{X})\left[\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \sigma_n^2\boldsymbol{I}\right]^{-1}\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}_*). \tag{4.58}$$

Based on the above definitions the GPR is suited to fit an arbitrary function $f(\boldsymbol{x})$ in the presence of noise. During training, the GP prior is conditioned with the given observations and the kernel's parameter $\boldsymbol{\Theta}$ are optimized. The advantages of the GPR over other regression tools is its non-parametric form and its inherent ability to provide an uncertainty measure for each prediction.

Description of the simplified prognosis algorithm

Based on the GPR, a data-driven RUL prediction algorithm is constructed. The general idea of the algorithm is to use the GPR to model the degradation rate $\dot{x}$ in dependence of the current degradation level x based on historic data. This can be written as

$$\dot{x} = f(\boldsymbol{x}) + \epsilon. \tag{4.59}$$

After training, the GP posterior mean $\overline{\boldsymbol{f}}(x_*)$ and posterior covariance function $cov(\boldsymbol{f}(x_*))$ of the GP posterior are known. Thus, for a given x_* the corresponding $\dot{x}_*$ can be drawn with

$$\dot{x}_* \sim \mathcal{N}(\overline{\boldsymbol{f}}(x_*), cov(\boldsymbol{f}(x_*))). \tag{4.60}$$

Assumed that the degradation level x_k at time instance k is accurately known, the future degradation path from this point on can be estimated by means of an MCS, in which a set of n individual trajectories is iteratively sampled with help of the GPR model, given as

$$x_{n,k+1} = x_{n,k} + \dot{x}_*. \tag{4.61}$$

This prediction step is carried out for all samples until the smallest solution in the current set exceeds the defined failure threshold x_{th}:

$$\min(x_{n,k+1}) \geq x_{th}. \tag{4.62}$$

Once reached, the individual samples are reshaped to find the RUL-CDF. For a sufficient large sampling size N, Equation 3.11 can be approximated by its discrete function

$$F_l(t) = \int_{x_{\lim}}^{\infty} p_x(x|t)dx \approx F_l(k) = \frac{1}{N}\sum x_{i,k},\ (x_{i,k} \geq x_{th}) \tag{4.63}$$

for $t = k$.

Training and Prediction results

With the definitions above, the developed GPR algorithm is now trained on the selected datasets. As given in Table 4.6, the Wiener Process data are used to train the gearbox

model, while the Geometric Gamma Process data are used for the motor model. As the size and computation time of the GPR-model increases exponentially with the training data, a moderate training dataset with five failure curves for both models is chosen, which are taken from the training subset. Both models use a linear basis function for the GPR and a squared exponential kernel function. The training configurations of both models are given in Table 4.9.

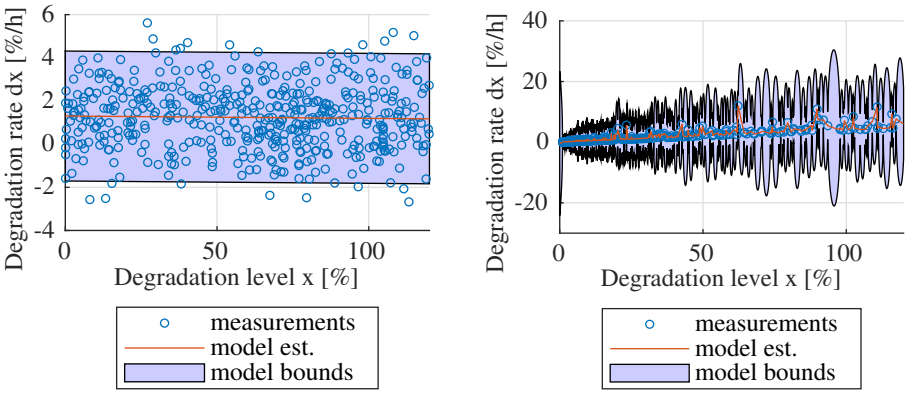
Table 4.9.: Trained GPR models

Parameter	Motor Model	Gearbox Model
Input vector x	Linear deg. model (Wiener Process)	Exp. deg. model (Geometrics Gamma Process)
Training Size n	5	
Basis function	linear	
Kernel function	Squared exp.	

Exemplary training results for the motor and gearbox model are shown in Figure 4.15. In both plots, the original training data in form of measurement points and the output of the GPR model in form of the GP mean (model estimation) and GP covariance (bounds) are shown. Therefore, the model's output, the degradation rate is plotted against the input, the degradation level. It can be seen that model mean of the linear GPR data are rather constant, while the mean of the exp. GPR follows a positive gradient, which was expected as the degradation rate is modelled. Further, it can be seen that the uncertainty bounds of the exp. GPR model vary much more than those of the linear one, which is an indicator of bigger jumps in the trained degradation data.

Based on the trained models, multiple predictions are carried out for different prediction times t_P. The purpose of these predictions is two-fold: On one side they are used to assess the performance of the prognosis algorithm itself, on the other side they are used as online data for the later application of the DHRM method in the considered use-case. Therefore, it is chosen to use ten individual datasets for each model to assess its performance, which are drawn from the validation subset as stated in Table 4.6. For the evaluation phase, additional twelve, respectively six, trajectories are chosen to represent the motor and gearbox failure during the use-case. These trajectories correspond to those shown in Figure 4.12.

For each prediction, the degradation data up to t_P is used as observation data, being the input data of the prognosis model. From the last observation point on, the GPR model is



(a) Gearbox (b) BLDC

Figure 4.15.: Training results of the GPR models

then used in an iterative way as shown in Figure 4.14 to predict the further evolution of the trajectory. This is done for all samples.

In Figure 4.16, the outputs of the GPR are shown for two exemplary datasets from the Wiener and the Geometric Gamma Process. Both predictions were calculated with an MCS sampling size of $n = 30$. Next to the resulting RUL-CDF also the true as the predicted (mean) of the degradation t_{EoP} are shown, giving a first impression of the accuracy of the algorithm. Besides a small increasing deviation between the predicted and the true degradation curve, it can be seen, that the true curve is fully covered by the bounds of the model.

Description of the improved prognosis algorithm with pre-classification

To further investigate the effects of varying algorithm performances in the remainder of this work, an improved prognosis algorithm is additionally developed that extends the concepts of the standard GPR as shown before. The improved algorithm is based

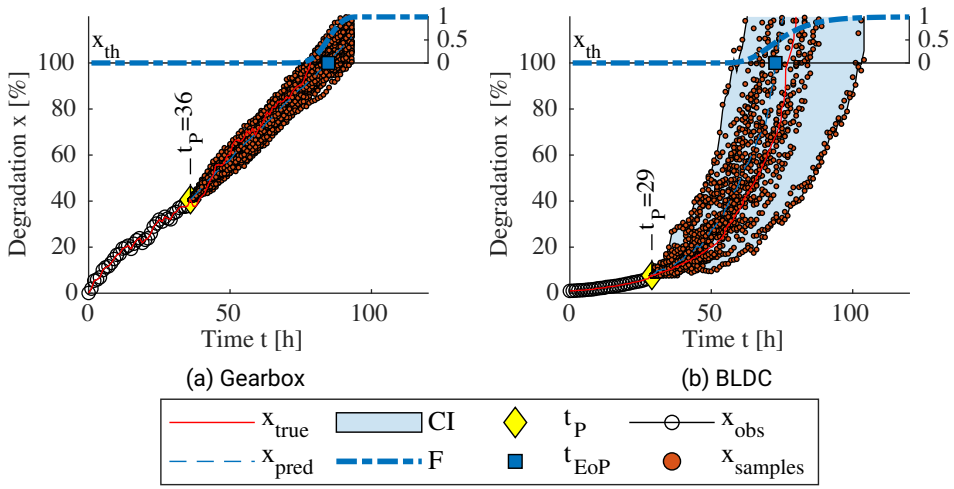


Figure 4.16.: Exemplary prediction results of both models

on a multiple-model-approach (MMA), in which multiple prediction models are used at the same time to accurately predict the future degradation curve based on the observed degradation data.

The historic failure data of the motor are therefore separated into three different groups A, B and C based on their true failure times (early, normal, late) and three individual GPR models are trained for each of these groups. In this way, different degradation rates can be considered by the algorithm. During the prediction process, the GPR model with the highest agreement to the observed data is then chosen to estimate the further evolution of the degradation curve and the resulting RUL.

To select the best fitting model, the MMA is coupled with a k-nearest-neighbor (KNN) algorithm, a supervised learning algorithm used for classification problems. The KNN is trained on the same data basis as the GPR models. Therefore, each known failure trajectory is labelled according to its group A, B or C and trained at different prognoses times t_P. During the prediction process the KNN is then used to classify the observed data. Once the group is classified, the corresponding GPR model is chosen and used to predict the further evolution of the degradation curve. This is done at every prediction point t_P. This

extended concept is shown in Figure 4.17. Further details on the KNN's accuracy can be found in the Appendix C.2.

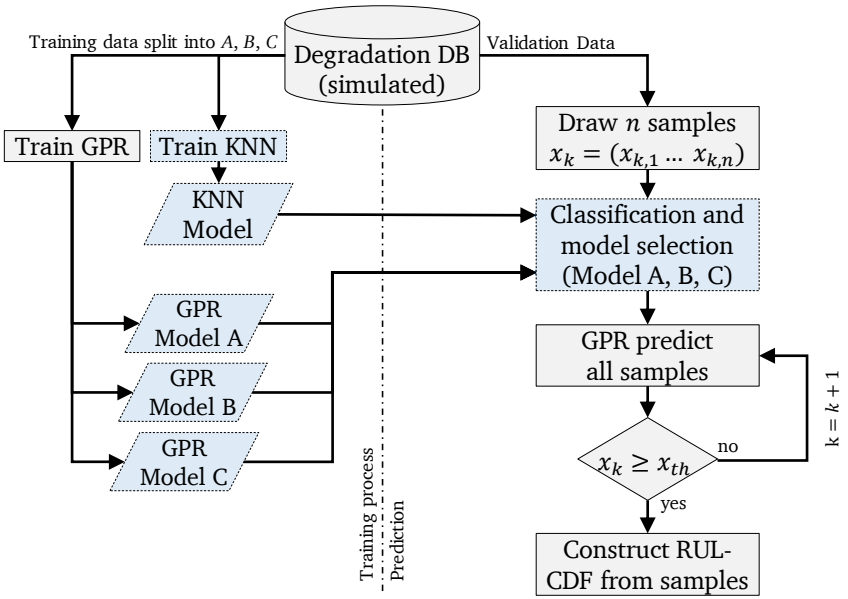


Figure 4.17.: Concept of the improved prognosis algorithm with extensions being highlighted

Algorithm performance

After generating first predictions, all algorithms are evaluated based on the performance metrics introduced in Section 3.3.3. Therefore, only the predictions based on the validation subset data are considered. In Figure 4.18, the results of the MAE, $MAPE$, MAD and S metric are given for all derived prediction models at different degradation levels. It can be seen, that all models improve their absolute metrics for predictions started at a higher degradation level and thus being closer to the true EoL. Only the relative $MAPE$ metric worsens for predictions at a higher degradation (ideal value is $MAPE = 0$). Further, it can be seen that the improved MMA-GPR outperforms the standard GPR on all metrics.

This is crucial, as the influence of the prognosis algorithm's performance will be discussed later during the evaluation of the proposed DHRM method.

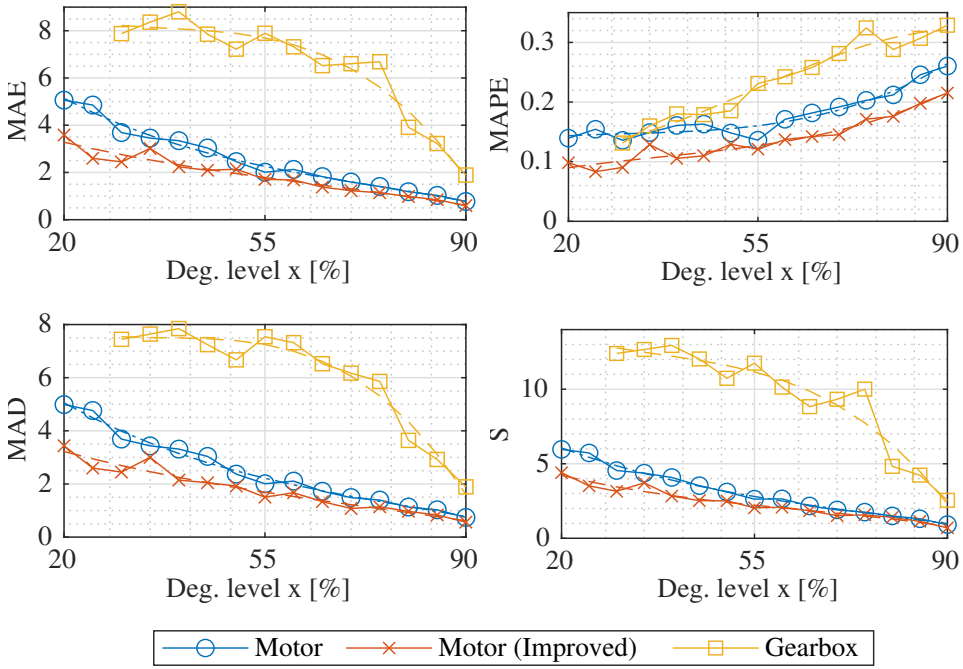


Figure 4.18.: Degradation based performance metrics MAE, MAPE, MAD and S for both motor prediction models (standard and improved) as well as for the gearbox prediction algorithm

To fully quantify and compare all prognosis algorithms, the aggregated values for each metric are calculated. Therefore, the area under the curve is calculated for each metric and divided by the number of evaluation points, giving an average value. The overall aggregated metrics for all algorithms are shown in Table 4.10. Based on the relative data it can be concluded that the improved MMA-GPR is roughly 25% better than the standard algorithm in accuracy as well as precision.

Table 4.10.: Algorithm performance metrics for the BLDC prognosis models *simple* and *improved*

Metric	Simple Alg.	Improved Alg.	diff.	rel. [%]
MAE	2.52	1.84	-0.68	26.87
MAPE	0.17	0.14	-0.04	22.66
MAD	2.48	1.78	-0.70	28.30
S	3.07	2.33	-0.75	24.23

4.4. System state prediction

In the previous sections of this chapter, a representative use-case for the application of the DHRM in form of the control surface actuation system of a hybrid drone was introduced. After a thorough system analysis, a reference as well as the novel DHRM were defined. For both models the required input data was then created based on artificially run-to-failure data for degrading parts (motors and gearbox) and predefined static failure models for those parts suffering a spontaneous failure (controllers and sensors). In this section, both models are calculated to obtain first results. Therefore, for the reference mode the failure probability for the degrading parts are taken from the static Weibull distribution as given in Section 4.3.3. For the DHRM, the dynamic CDF-RULs from the introduced PHM algorithm from Section 4.3.4 are used. All other parts are modelled with the rather static failure rates obtained from the Military Handbook 217F (see Section 4.3.1). The results are then compared in a first evaluation. In Figure 4.19, the overall approach is shown.

4.4.1. Base-line reference calculation

At first, the reference model based on the static Weibull data is calculated. For this simulation, a standard scenario at 20°C is considered, with the corresponding failure rates for the controllers and sensors as stated in Table 4.3 and 4.4. Further, for this first calculation, any uncertainties and parameter variations are neglected. Also, the state predictions do not consider any replacements and all components are considered to be as new for $t = 0\ \mathrm{h}$.

Based on the given input data as presented in Table 4.8, the EMA's individual failure times and thus the overall system states and transitions are derived. An overview of the first three failures up to the loss of control state is given in Table 4.11:

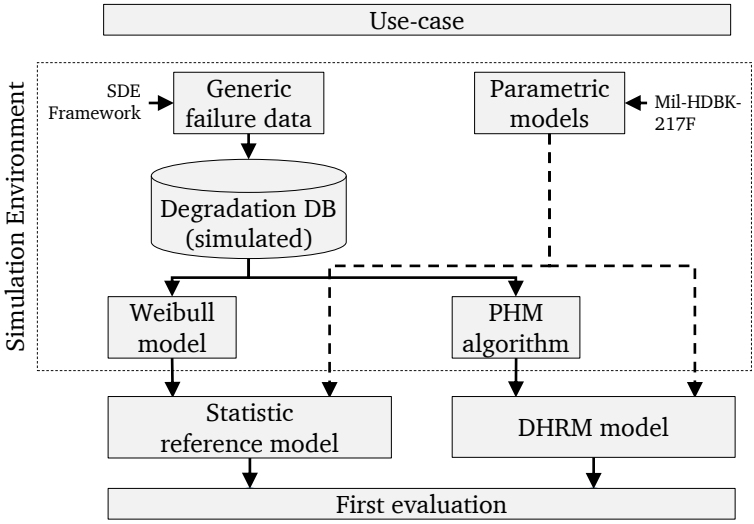


Figure 4.19.: Application concept for the further evaluation of the DHRM and comparison to a reference case

Table 4.11.: Ground truth failure transitions and result states based on the selected run-to-failure data

Event time [h]	Failed Component	Resulting new state
t=0	none	S1 - Initial state
t=1584	EMA Aileron left	D1 - Degraded state
t=1776	EMA Elevator left	D2 - Degraded state
t=2240	EMA Aileron right	LOC – Loss of control

With the above configuration the reference model is calculated for the static input data. In Figure 4.20, the output of the aggregation method for this scenario with constant distributions is shown. While on the left, the visualization is given in a linear scale, on the right side a double logarithmic scale is used. Both plots show the probability for each state ($S1$, $D1$, $D2$, $D3$, LOC) after a given total mission time t. The true state transitions, based on the real failure of components are indicated by the vertical lines. Relevant reliability

thresholds are indicated by the horizontal lines at $F = 10E-6$ and $F = 10E-3$.

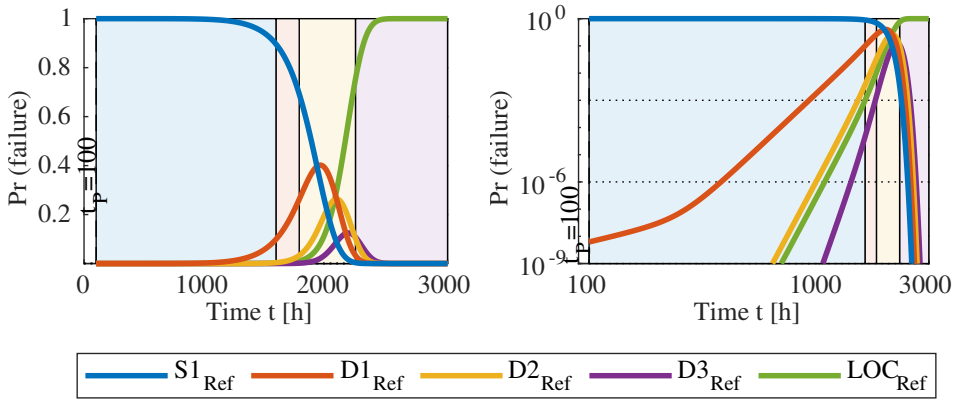


Figure 4.20.: State estimations for the QCAS based on the Weibull approach with model outputs in linear (left) and log scale (right)

The above state probabilities remain constant for all times as in the conventional reliability approach no new information is considered at any times. From the results, the following key characteristics for the base-line approach are observed:

1. Because of the constant parameter values and fixed distributions, the state predictions remain constant in time. Thus, probabilities do only depend on the time vector t.
2. Although the predicted states are visually distinguishable, there is no clear manifestation of them except for states $S1$ and LOC. It is expected that for systems with even more states, transitions become even more imprecise as the discriminatory power is reduced as each state will at least have a very small likelihood to be entered.

4.4.2. DHRM calculation

In the following, the DHRM is calculated. Therefore, the same simulation settings are chosen as for the reference case to maintain comparability. Still, instead of using constant failure distributions for the non-electronic parts motor and gearbox, this time PHM results based on the introduced prognosis algorithm from Section 4.3.4 are used, as proposed

by the novel aggregation method. The results are based on the improved prognosis algorithm with pre-classification. To capture the expected dynamic of the model, it is further calculated at different prognosis times t_P until the complete failure of the system (loss of control) is reached.

In Figure 4.21, the results of the DHRM are shown for two prognosis times $t_P = 1000\ \mathrm{h}$ and $t_P = 1400\ \mathrm{h}$. For better comparability, all predictions are again shown in a linear and a log-scale. From the plot in linear scale on the left it can be seen, how the algorithm adapts the state probabilities for later predictions. While the first prediction at $t_P = 1000\ \mathrm{h}$ still is a rough estimate, the later prediction at $t_P = 1400\ \mathrm{h}$ already gives a more accurate prediction regarding upcoming state transitions as the predicted state probabilities converge to the true transitions. Also, it can already be seen that state $D3$ is very unlikely to be entered at all and that the system will directly go to the LOC state. On the right side, the state probabilities in log scale are given. In this visualization, the steep change within the state probability for $D1$ shall be highlighted. After a nearly constant increase, this state probability significantly increases around $t = 1450\ \mathrm{h}$.

4.5. Conclusions from use-case

Within this chapter a representative use-case from the aerospace domain was introduced to further show and investigate the applicability of the proposed aggregation method. As system to analyze the control surface actuation system of Airbus' Quadcruiser, a hybrid drone, was chosen. Based on simulated degradation data the novel aggregation method was compared to a baseline scenario, following a rather traditional reliability estimation process with constant failure distributions.

From the results above it can be concluded that the proposed method is suited to be also applied to rather complex systems, consisting of multiple components from different technical domains and having a large set of internal system states. It was further shown, that RUL-predictions can be included into the aggregation to generated adaptive state predictions. In contrast to the reference scenario with fixed lifetime distributions the DHRM method results in optimized accuracy regarding state changes and, even more important, is able to indicate which state the system is likely to enter next.

With respect to the described use-case, it can be said that the operator of the UAV receives a decision support that continuously estimates the current as well as future system state of the aircraft's control surface actuation system. Although the UAV is controlled remotely, the operator receives a detailed state estimation and can decide early on whether to abort

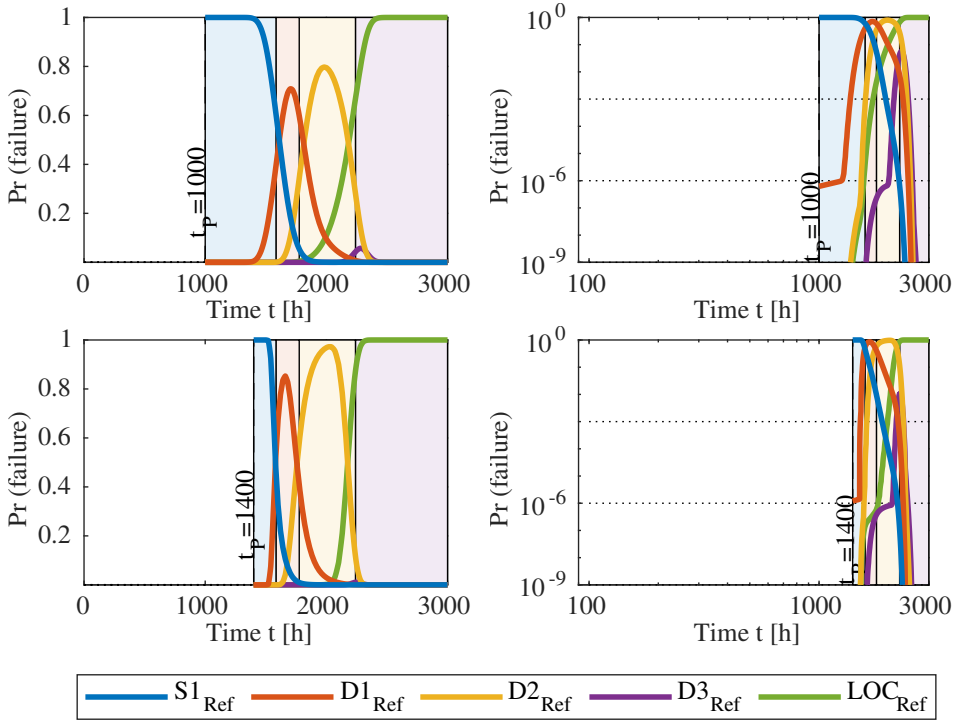


Figure 4.21.: State estimations for the QCAS based on the DHRM approach at different prognoses times in linear and log scale

or continue the mission in the event of an impending system state change. In this way, upcoming missions as well as maintenance activities can be better planned, leading to an increased safety and availability of the aircraft.

In order to further investigate the general advantages of the developed method, it will be further evaluated in the next chapter.

5. Evaluation and discussion of the developed method

This chapter is dedicated to the further evaluation of the proposed DHRM approach in a quantitative and qualitative way. Therefore, the capabilities as well as limits are identified and compared to the baseline approach in order to derive the method's overall advantages and disadvantages. This evaluation is mainly based on the developed use-case from the previous chapter and uses a set of selected metrics and different parameter constellations. Key aspects of this evaluation are:

- **Prediction performance:** The scope of this evaluation is to quantify the quality of the derived state predictions in comparison to a (static) reference case. This assessment will answer, whether the new approach is beneficial from a functional point of view.
- **Calculation efficiency:** The scope of this analysis is to derive the method's computational demand and to derive first expectations regarding its scalability. The consideration of the calculation efficiency will investigate, whether the approach is also technical feasible for larger systems or constrained calculation power.
- **Economic evaluation:** The economic considerations will investigate the key drivers for monetary advantages and limits of the proposed method.

All three aspects are further outlined in the following, before a comprehensive discussion and summary of the results is presented in Section 5.4 of this chapter.

5.1. Performance evaluation

The following section is dedicated to the evaluation of the prediction performance of the proposed method. The scope is to quantify the accuracy and precision of the DHRM's

state predictions compared to a reference case. The evaluation is based on predefined performance metrics that are applied to the model's output for different scenarios and parameter variations.

5.1.1. Prediction performance metrics

An important measure for every prediction model is a statement about the accuracy of the delivered result. A common approach to assess the quality of prediction and classification algorithms is the confusion matrix [GBV20]. This matrix maps each prediction output to the true problem class. In an ideal world, this matrix would become the identity matrix $\mathbf{I}$, stating that every prediction outcome is mapped correctly to the according class. Within this work, the confusion matrix is used to quantify the accuracy of the state predictions for each prediction for a given t_P. This is done by calculating the area under the curve (AUC) of each state probability estimation and comparing its value against the AUC of the true system state transition. Based on this concept an $n \times n$ matrix $\mathbf{C}$ is created, where n reflects the number of system states. Each element $c_{i,j}$ in this matrix represents the ratio between the true (index j) and the prediction (index i) of that state. In Table 5.1, an exemplary confusion matrix of a system with three different states is given. Within the confusion matrix, the diagonal elements correspond to a correct assignment, whilst the lower triangular matrix represents too early predicted state transitions and the upper right for too late predicted state transitions. More details on the confusion matrix and its calculation are also given in the Appendix D.1.

Table 5.1.: Exemplary confusion matrix of a system with three distinguishable states

		Reference Dist. (ground truth)		
		S_1	S_2	S_3
AUC of predicted state probabilities	$S_{1,est}$	$c_{1,1}(t_P)$	$c_{1,2}(t_P)$	$c_{1,3}(t_P)$
	$S_{2,est}$	$c_{2,1}(t_P)$	$c_{2,2}(t_P)$	$c_{2,3}(t_P)$
	$S_{3,est}$	$c_{3,1}(t_P)$	$c_{3,2}(t_P)$	$c_{3,3}(t_P)$

It shall be noted that the confusion matrix always corresponds to a given prediction at a certain prognosis time t_P. Thus, metrics based on this concept will be time-dependent. The following metrics are defined based on the confusion matrix:

Accuracy (ACC)
The accuracy metric measures the overall accuracy of the output, considering all predicted state probabilities. It is therefore defined as the sum of the correct assigned AUCs, divided by the total sum of all ratios:

$$ACC(t_P) = \frac{\sum_{i=1}^{n} c_{i,i}(t_P)}{\sum_{i=1}^{n}\sum_{j=1}^{n} c_{i,j}(t_P)}$$

The ideal value of the ACC metric is 1, the overall metric falls into the range of $[0, 1]$.

Early/Late-Ratio (ELR)
The ELR metric is the ratio between the lower triangular sum divided by the upper triangular sum of $\mathbf{C}$. Thus, the ELR is the quotient between falsely too early and too late predictions and can be used as an indicator, if the algorithm's output is too conservative or too optimistic. The ERL is defined as:

$$ELR(t_P) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} l_{i,j}(t_P)}{\sum_{i=1}^{n}\sum_{j=1}^{n} u_{i,j}(t_P)} \tag{5.64}$$

where: $\Delta u_{i,j}(t_P) = \begin{cases} c_{i,j}\,(t_P)\,, & \text{if } j \geq i \\ 0, & \text{otherwise} \end{cases}$

$\Delta l_{i,j}(t_P) = \begin{cases} c_{i,j}\,(t_P)\,, & \text{if } i \geq j \\ 0, & \text{otherwise} \end{cases}$

This metric has an ideal value of $ELR = 1$ and falls into the range $[0, \infty]$. While an $ELR < 1$ indicates a trend towards being too optimistic an $ELR > 1$ states too conservative predictions. However, it shall be highlighted that this metric gives no indication of the overall accuracy of the method, as equally distributed false predictions will compensate each other in this metric. Thus, the ELR should be always used together with the ACC metric.

True Positive Rate (TPR)
The TPR metric describes to which degree a given state was estimated correctly. Therefore, the ratio of correctly assigned AUC and falsely estimated values is considered for each prediction. The TPR for a state i is defined as:

$$TPR_i(t_P) = \frac{c_{i,i}(t_P)}{\sum_{j=1}^{n} c_{i,j}(t_P)}$$

Again the ideal value of this metric is 1, with a value range of $[0, 1]$.

5.1.2. Aggregated prediction performance metrics

As the above presented metrics consider results only for one distinct t_P, they are further aggregated to allow a prognosis-time independent evaluation. Therefore, aggregated performance metrics are introduced, which are based on the ACC, ELR and TPR metric and extend them to represent all considered prediction times t_P. In this way, each metric is aggregated into one key performance indicator allowing a better comparison. These metrics are defined as follows:

Aggregated Accuracy (AACC)
The AACC is defined as the area under the ACC curve, normalized by the time between the first prediction t_0 and the last t_{end}.

$$AACC = \int_{t_0}^{t_{end}} ACC\,(t_P)\,dt/(t_{end} - t_0)$$

The range of this score is given as $AACC \in [0, 1]$ with 1 being the best score. The AACC indicates to which degree the states and their transitions have been predicted correctly over all times for a system.

Aggregated Early/Late-Ratio (AELR)
Similar to the AACC also the ELR can be aggregated over all prediction times leading to the aggregated ELR. This metric is defined as:

$$AELR = \int_{t_0}^{t_{end}} |ELR\,(t_P) - 1|\,dt/(t_{end} - t_0)$$

In this metric $|ELR\,(t_P) - 1|$ describes the absolute deviation of the result from the ideal curve. The AELR falls then into the range of $AELR \in [0, \infty]$, with 0 being the best score. It shall be noted that the AELR, in contrast to the ELR, gives no indication of to early or too late predictions as the absolute ELR value in Equation 5.1.2 is used to avoid error compensation.

Aggregated True Positive Rate (ATPR):
ATPR metric is similar to the AACC and shows the overall prediction accuracy per state. The ATPR for a state i is thus defined as:

$$ATPR_i = \int_{t_0}^{t_{end}} TPR_i\left(t_P\right) dt / (t_{end} - t_0)$$

Again the range of this metric is given as $ATPR \in [0, 1]$ with 1 being the best score, indicating that a state has been predicted correctly at all times.

5.1.3. Scenario evaluation

With the evaluation metrics defined, different experiments with varying configurations are conducted. Below, the considered scenarios and the corresponding research questions are listed, before results are presented in the following.

- **Standard Scenario:**
  The DHRM is calculated with the improved prognosis algorithm from Section 4.3.4 and then compared to the reference case. Scope: The overall performance of the novel method is quantified.
- **Scenario with simplified PHM algorithm:**
  This experiment is similar to the first, but uses the simplified PHM algorithm that was introduced in Section 4.3.4 instead. The scope of this experiment is to quantify the effect of varying PHM algorithm performance.
- **Failure rate variation:**
  The scope of this experiment is to evaluate the impact of varying failure rates and to investigate the uncertainty propagation within the model.

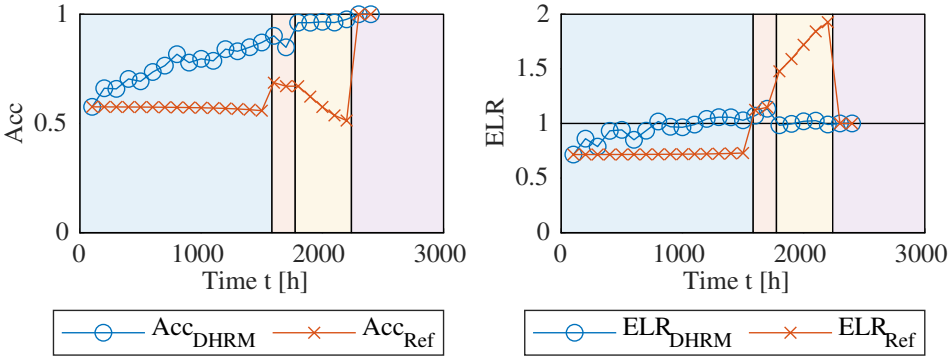
Standard Scenario
The standard scenario experiment uses the results from Subsection 4.4.2, where the DHRM is calculated with the improved PHM-algorithm, and those from Subsection 4.4.1, where the static reference case is shown. For both experiments, the previously defined performance metrics are then applied to compare and quantify the performance of the DHRM and the reference case. Therefore, all metrics are evaluated at discrete prediction points, ranging from $t_P = 100$ h to $t_P = 2400$ h.

In Figure 5.1 on the left, the evolution of the accuracy metric ACC is shown for both cases over different prediction times. For the reference case, this metric starts with an accuracy of 57%, which remains nearly constant until the first state transition is reached. The maximum accuracy for the reference case is reached shortly after the first transition with 68% and the minimum just before the transition to the final state with 51%. The jump to 100% when reaching the LOC state is reasonable as the prediction of the other states immediately stops, when they fail.

In contrast to the reference approach, where only few abrupt changes in the metric are visible, it can be seen that the ACC metric for the DHRM approach is constantly increasing for later prediction times. The absolute minimum is 57% for this metric, reached in the very first prediction. This is exactly the same value as for the reference, which is plausible as for the first state prediction no PHM data is available, forcing the algorithm to fall back to the standard approach with static failure distributions, which is equal to the reference given here. In the first state (nothing failed), the DHRM has a maximum accuracy of 90%, which is shortly reached before the first state transition. During the second state (first component has failed), an accuracy between 85% and 95% is obtained, while the accuracy for the third state (two components have failed) is over 95% at all times.

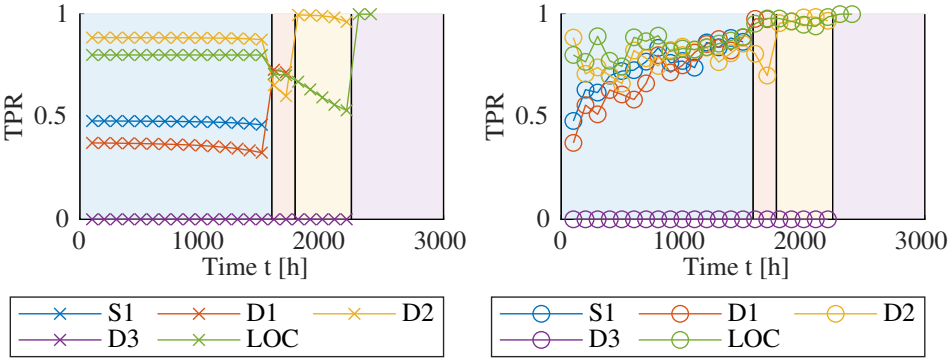
On the right-hand side of Figure 5.1 the ELR metric is given for both approaches. For the reference case, this metric starts with a value of 0.7, which remains nearly constant until the first state transition. Thus, the prediction for the reference case is too optimistic during the first state (a later state transition is expected). After the first state transition, the ELR metric becomes greater than one and continues to increase until a maximum of 1.9 is reached shortly before the last state transition, meaning that the state predictions were too conservative for later predictions.

In Figure 5.2, the TPR for all states is shown for the reference case on the left, as well as for the DHRM on the right. For the reference case, it can be seen that this metric remains again nearly constant for all states until the first transition. The highest TPR is reached for the $D2$ and LOC state with 88% and 80%, respectively. The state predictions for state $S1$ and $D1$ are much lower with around 48% and 36%. After the first transition, the predictions for states $D1$, $D2$ and LOC remain all in a range between 60% and 70%. After the second transition, the predictions for states $D2$ and LOC are left with a TPR of 99% and 67% to 53%. Regarding the results of the DHRM shown on the right of Figure 5.2, it can be said that all TPR values start with the same values as for the reference case, but improve immediately afterwards. Towards the end of state $S1$, all TPR values for each state are already above 85%. During $D1$ the TPRs for states $D1$ and LOC are above 95%, while only the prediction for state $D2$ falls back to 70%. After the second transition the remaining state predictions for $D2$ and LOC are above 90% at all times.



(a) ACC metric (b) ELR metric

Figure 5.1.: Comparison of the accuracy (ACC) metric and Early-Late-Ratio (ELR) metrics for different prognoses times for the DHRM and the reference case (Ref)



(a) Reference (b) DHRM

Figure 5.2.: Comparison of the True Positive Rate (TPR) metric for the reference case and the DHRM

From all four metrics, it can be seen that the DHRM approach delivers better results compared to the reference case as soon as PHM data becomes available and is considered for the state predictions. Further, all metrics converge towards the ideal value with the

DHRM for later predictions. This finding is plausible as the underlying PHM data is also expected to gain in accuracy for later RUL predictions, leading to improved results. Finally, from the TPR metric it is concluded that the DHRM method not only gives an overall better result in terms of accuracy, but is also more accurate on a per-state comparison. This could be crucial for the application of this method, as decision makers are expected to be interested in an accurate prediction of a specific state rather than the overall state evolution.

To provide a final assertion, the aggregated performance metrics as introduced in Subsection 5.1.2 are calculated and compared to each other in Table 5.2. Therefore, the absolute and relative values are calculated for each metric. Based on this, the overall improvement with the DHRM approach is quantified. The results show that for all metrics except APTR_D2 [1] an improvement is achieved using the DHRM method. The improved metrics range from 17% for the ATPR_LOC up to 86% for the ATRP_D1 metric. It shall be noted that the ATPR_D3 metric cannot be calculated for this scenario, as the state D3 (=one EMA failed on each axis) is never reached.

Table 5.2.: Evaluation and comparison of the aggregated performance metrics between the DHRM and the reference case.

Metric	DHRM	Ref.	Δ Abs.	Δ Rel. %
AACC	0.80	0.59	0.21	36
AELR	0.06	0.33	- 0.27	-82
ATPR_S1	0.70	0.44	0.26	58
ATPR_D1	0.69	0.37	0.32	86
ATPR_D2	0.78	0.84	- 0.06	-7
ATPR_D3	-	-	-	NaN
ATPR_LOC	0.85	0.73	0.12	17

Scenario with the simplified PHM algorithm
In analogy to the previous experiment based on the improved PHM-algorithm, the same experiment is conducted with the simplified PHM-algorithm. This experiment is intended to investigate the influence of the algorithm performance on the overall aggregation method.

In Figure 5.3, the results of all metrics are shown for the reference case (Ref.), the DHRM

[1] The optimal score for AELR is zero, thus a negative relative score indicates an improvement.

with the simplified PHM algorithm (DHRM (s)) and the DHRM with the improved PHM algorithm (DHRM (i)). The comparison of all three metrics ACC, ELR and TPR show that the results based on the improved PHM-algorithm outperforms the simplified as well as the reference case for early t_P. However, for later state predictions, the results of the simplified version converges to those obtained by the improved PHM-algorithm. Either way, both DHRM scenarios start with the same performance as the reference case for $t_P = 0\ \mathrm{h}$ and deliver better results for later state predictions.

To be able to further quantify, in Table 5.3 the aggregated performance metrics of the simplified case are given together with the those for the improved PHM algorithm. The results reveal that the DHRM with the improved PHM algorithm (i) outperforms the one with the simplified algorithm (s) for all metrics except ATPR_D2.

Table 5.3.: Evaluation and comparison of the aggregated performance metrics between the DHRM using the improved (i) and the simplified (s) PHM algorithm.

Metric	DHRM (s)	DHRM (i)	Δ Abs.	Δ Rel. %
AACC	0.71	0.80	-0.09	-11
AELR	0.15	0.06	0.09	150
ATPR_S1	0.51	0.70	-0.19	-27
ATPR_D1	0.45	0.69	-0.24	-35
ATPR_D2	0.91	0.78	0.13	17
ATPR_D3	-	-	-	NaN
ATPR_LOC	0.80	0.85	-0.05	-6

From the given experiment it can be concluded that an improvement of the underlying PHM-algorithm(s) mainly affects the accuracy of earlier predictions. Accordingly, the proposed DHRM's prediction accuracy can be improved by using more accurate PHM-algorithms, which in turn increases the reaction time for an operator as precise information on a system state transition can be detected earlier.

Scenario with varying failure rates

As argued in Section 3.2.3, conventional failure rates are often parameterized according to the environment the parts are foreseen to be used in. Also, the real usage of a system may change over time. Thus, it was suggested in Section 3.2.3 of this work to allow changes of these parameters to be able to adjust them to the real usage.

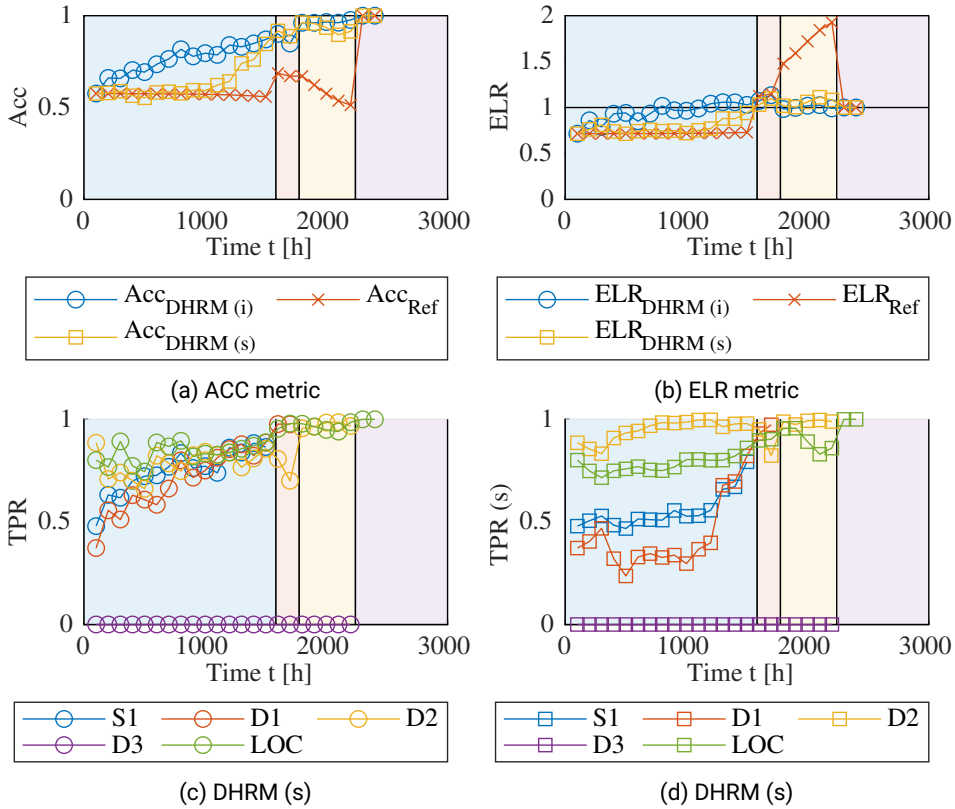


Figure 5.3.: Comparison off all metrics for the DHRM with improved PHM-algorithm (i), with simplified PHM-algorithm (s) and the reference case

In the following experiment, the configuration of the standard scenario is taken and calculated for two extreme environments with adjusted failure rates (compare tables 3.2.3 and 4.4). While the first parameter set represents an aerodynamic flight mode at cold temperatures, the second parameter set represents a hot environment with high vibrations (e.g. hovering mode). The aim of this experiment is to identify the influence of the parameter changes of the non-degrading parts on the state predictions in the DHRM.

In Table 5.4, the chosen parameter sets for the extreme environments *cold aerodynamic* and *hot hovering* are given and compared to the *normal* parameter set.

Table 5.4.: Failure rates in failure/10E6 hours for the two extreme value scenarios

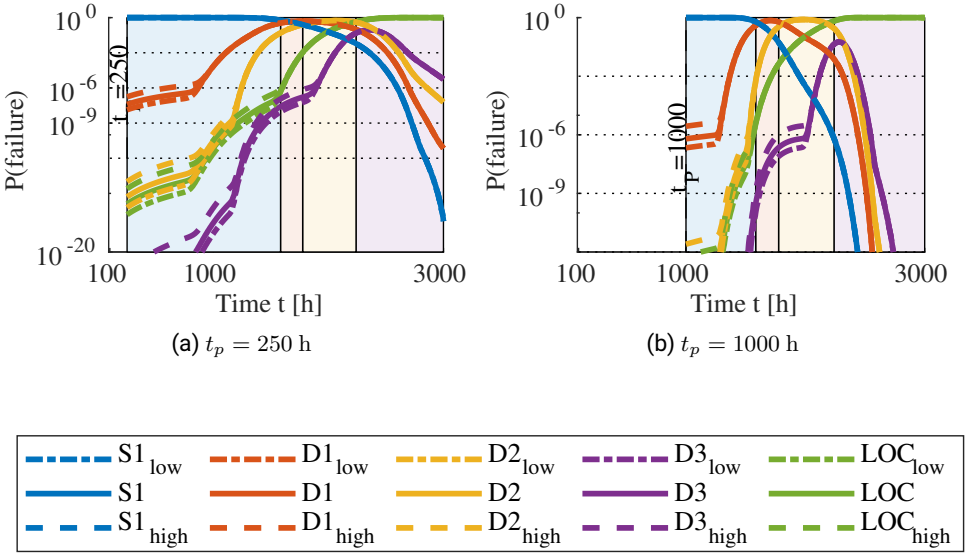
Part	cold aerodynamic (extreme low)	normal	hot hovering (extreme high)
ECU	$\lambda_{ECU} = 0.13$	$\lambda_{ECU} = 0.21$	$\lambda_{ECU} = 0.30$
Sensor	$\lambda_{Sen} = 0.08$	$\lambda_{Sen} = 0.14$	$\lambda_{Sen} = 0.35$

In Figure 5.4, the results of all three simulations are shown and overlaid for prediction times $t_P = 250\ \mathrm{h}$ and $t_P = 1000\ \mathrm{h}$. The influence of the different failure rates is especially visible during the first hours of operation before the dominant parts suppress the less dominant ones when coming closer to the state transition. The seen behavior is reasonable as the individual contribution of each part to the overall system's state probability changes over time. While the contribution of the non-monitored parts are much higher during the first hours of operation, the failing components (here motor and gearbox) become more dominant later. Thus, the induced PHM data suppresses the conventional failure rates at a certain point, becoming the main contributor to the system's state probabilities. This effect already reveals an important characteristic of the proposed aggregation method: The advantage of the DHRM is not only based on the accuracy of the underlying PHM algorithms, but also depends on the ratio between PHM-monitored and non-monitored parts in the considered system and their individual contribution to the failure states.

5.2. Calculation efficiency

Besides the accuracy and precision of the method's results, also the computational effort required to obtain these results is an important measure. Only if the method can be calculated in an appropriate time frame and with reasonable hardware resources, the method becomes practicable.

A general methodology of assessing the time-complexity of an algorithm is the use of the *Big-O* notation [RRH00]. The Big-O notation foresees to compare a given function $f(x)$ with a comparison function $g(x)$. Assumed that both functions are defined on an unbounded set with strictly positive values, one writes



(a) $t_p = 250$ h (b) $t_p = 1000$ h

Figure 5.4.: State prediction for two different prognosis times with varying failure rates

$$f(x) = O(g(x)), \text{ for } x \to \infty \tag{5.65}$$

if there is exists a real number M such that $|f(x)| \leq Mg(x)$ for all $x \geq x_0$. This basically means that $f(x)$ stays below the selected comparison function $g(x)$ multiplied with M for all times. In terms of algorithm efficiency usually the runtime $t_{Alg}(n)$ of an algorithm in dependence of the input (problem) size n is considered. Typical comparison functions include the logarithmic, linear, polynomial or exponential function. In Table 5.5 an overview of common Big-O metrics and there meaning is given:

In order to assess the calculation efficiency of the proposed method, the time-complexity of the individual modules of the DHRM are considered.

In correspondence with Figure 3.8, the DHRM consists of the part-level, the component-level and the system-level. The part-level's time complexity is neglected at this point, as the part's failure rates and results from any PHM-algorithms are considered as inputs of the

Table 5.5.: Big-O notation and their interpretation, based on [Kne20]

Notation	Order	Technical feasibility
$t_{Alg} \in O(\log n)$	logarithmic	Easily scalable
$t_{Alg} \in O(n)$	linear	Scalable for large and very large n
$t_{Alg} \in O\left(n^c\right)$	polynomial	Technical cumbersome for large n
$t_{Alg} \in O\left(c^n\right)$	exponential	To avoid, cannot be solved for large n

DHRM. The time-complexity on this level will mainly depend on the used PHM-algorithms and their calculation efficiency, which cannot be known in advance.

On component-level, there are n fault trees to solve, where n is the number of components in the system. As given in [DWG15] the fault tree analysis is a non-deterministic polynomial-time hard problem, in which the problem size depends on the number of nodes and basic events. Accordingly, the calculation of the DHRM is expected to become slow for larger fault tree models. However, as given in Table 5.5 these models can still be solved for larger problem sizes as they are of a complexity of $O\left(n^c\right)$. Also, as the fault tree method is well explored, there exists alternative solving approaches such as approximation techniques that can be used to reduce the computational effort [Yev10].

On the highest aggregation level, the system-level, the resulting probabilities of all components, respectively fault trees, are used to derive the overall probability of all states. This is achieved based on a state-space model as presented in Section 3.2.5. The difficulty with state-space models is that their complexity scales exponential to the problem size, which is referred to as state-space explosion [Mey12, p. 10]. Even for problems with two states (e.g. failed and non-failed), which is the case for most reliability problems including the one described in this thesis, the growth rate is of order $O\left(2^n\right)$. According to this, the run-time of the DHRM is expected to increase exponentially for numerous components, which shall be avoided at all costs.

Despite the component and system-level calculations, the time complexity of the DHRM has two additional dimensions, namely the uncertainty handling (the fault trees as well as the state model are solved by means of an MCS) and the evaluation on a discrete time space.

The DHRM evaluates all estimates on a discrete time space reaching from the first point of prediction t_P till a predefined simulation end time t_{end} (e.g. until a state have been reached where all components have failed). Thus, the required calculations have to be

performed K times, where K is the number of discrete time points, which results in a time complexity of $O(n)$. An important aspect here is that the calculation time will decrease for later predictions times t_P as the model has to be solved on fewer time points, which is handy as also the available time for decision-making is less for later t_P.

Regarding the uncertainty handling, all calculations on component and system-level have to be performed n-times where n is the sampling size of each MCS. Thus, these additional calculations are also covered by a complexity of $O(n)$.

Thus, the overall time-complexity of the DHRM not only results from the complexity of the modelled system (and its number of components and parts), but also depends on the sampling size within the MCS and the number of discrete points in time, for which the model is calculated.

The overall contributing factors for defining the time complexity of the DHRM approach are summarized in Figure 5.5. In this figure it is important to note that a part of the complexity is given inherently by the analyzed system and the number of parts and components and thus cannot be changed. Other elements of the time complexity instead result from the sample sizes as well as time discretization and can be further optimized.

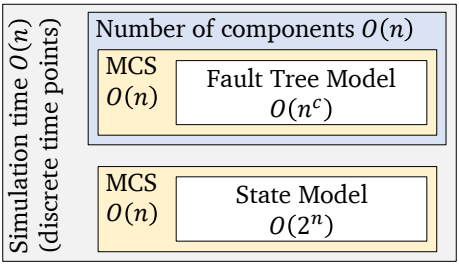


Figure 5.5.: Overview the elements contributing to the calculation time

To verify the above considerations, different experiments are conducted in which the DHRM from the standard scenario is simulated in different configurations. For comparability reasons all experiments are run on the same computer, a standard notebook with a quad-core Intel processor having 2.7 Ghz.

At first, the sample size of the embedded MCS is varied to run with $n = 50$ and $n = 500$ samples to further analyze the calculation efficiency on component-level. In a second experiment, additional EMAs are added to the system. Although the CSAS of the Quad

Cruiser is composed of exactly six EMAs, this test is used to theoretically assess the effect of upscaling the system to contain more components. Accordingly, the model is simulated with $n = 6$ (default), $n = 9$ and $n = 12$ identical EMA models. In Figure 5.6, the average calculation time on component-level with varying samples sizes as well as the calculation time on system-level for a different number of components is shown in a log-scale plot for prognoses times reaching from $t_P = 100\ \mathrm{h}$ to $t_P = 2400\ \mathrm{h}$.

From Figure 5.6a (a), showing the average component calculation time, it can be concluded that the calculation time scales linear with the sampling size of the MCS. While the first component simulation at $t_P = 100\ \mathrm{h}$ took in average $0.56\ \mathrm{s}$ for a sampling size of $n = 50$, it took $6.14\ \mathrm{s}$ for a sampling size of $n = 500$. According to this, the model can be easily calculated as long as the underlying fault tree can be solved in reasonable time. In Figure 5.6b (b), the calculation time of the internal state-space model is shown. Here, it can be seen that the overall calculation time increases significantly with each component added. While the calculation takes $0.054\ \mathrm{s}$ for a system with $n = 6$ components, it already takes $2.95\ \mathrm{s}$ for a system with $n = 12$ components, which is roughly two magnitudes longer. Thus, the state model scales with an exponential time function.

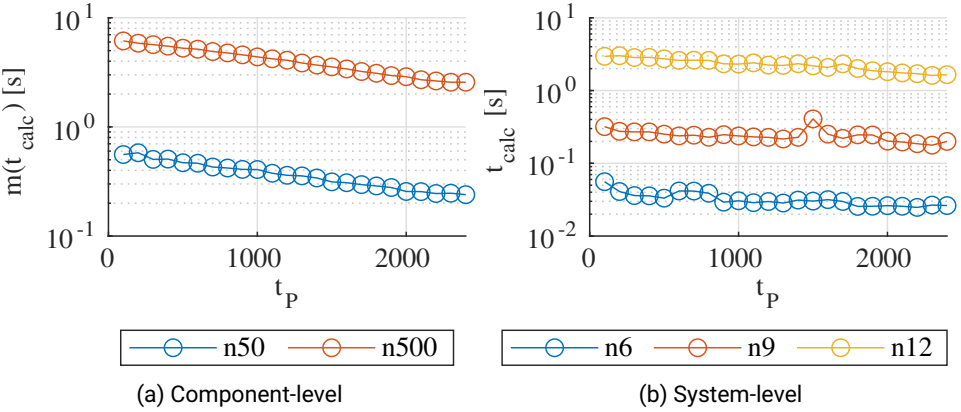
Finally, all experiments show a linear decreasing calculation time for later t_P as expected, with only few outliers.

Both experiments support the initially stated considerations for the time-complexity. However, they also proof that the DHRM is applicable and technical feasible with a reasonable calculation time for systems with a reasonable amount of failure states.

5.3. Economic considerations

In the following section, the economic considerations of the method are presented. However, it should be noted at this point that the considerations in the following are generic, as an exact calculation is only feasible for a specific system with known cost factors.

Two widely used metrics for economic analyses are the cost-effectiveness and the return on investment (ROI). While the cost-effectiveness links the financial expenditure, e.g. for the operation of a system, with the performance obtained [Bir10, p. 13], the ROI considers the ratio between an investment and its expected return [FSJ08]. Since the method proposed in this paper is a new one, an ROI analysis is performed with the aim to make the benefit of an investment visible (business case).



(a) Component-level (b) System-level

Figure 5.6.: Calculation times for different t_P on the component-level (a) and the system-level (b)

As given in [FSJ08], the ROI is defined as the ratio between the net return and the initial investment. In the field of PdM this concept usually is reverted to represent the avoided costs based on a given investment. This relation is given as:

$$ROI = \frac{Return - Investment}{Investment} = \frac{Avoided\ Cost}{Investment} - 1 \tag{5.66}$$

For a $ROI > 0$ an investment is considered to be profitable otherwise it is a financial loss (or equal for $ROI = 0$). In the context of PHM, e.q. 5.66 can be written as:

$$ROI = \frac{C_{Ref} - (C_{PHM} - I)}{I} - 1 \tag{5.67}$$

In this equation, C_{Ref} represents the reference cost to be expected if no PHM is used, while C_{PHM} represents costs that are achieved by using PHM. As reported in [FSJ08], the investment I finally represents the initial amount needed to implement PHM monitoring and can be further segregated into non-recurring costs C_{NRE}, recurring costs C_{REC} and those costs associated with the continuous operation of required infrastructure C_{INF} (e.g. data warehouse etc.):

$$I = C_{NRE} + C_{REC} + C_{INF} \tag{5.68}$$

In order to quantify and compare the economic benefits the reference costs C_{Ref} and the PHM costs C_{PHM} are further broken down and linked to the (state) classification problem as introduced in Section 5.1 of this chapter. This approach is largely based on [Lil18]. However, for this work the concept from [Lil18] is extended to fit a multi-class classification problem. Accordingly, the confusion matrix from Table 5.1 is used again. Being a classification problem, all state predictions are set into relation to the true state at a given time, which represents whether the classification was correct or not. This leads to the following general four combinations, which can be derived for every state i of the model:

(a) The model predicts state i while the system is in state i (correct prediction) $\rightarrow$ true positive, TP_i.

(b) The model predicts another state than i while the system's true state is i (false prediction) $\rightarrow$ false negative, FN_i.

(c) The system is not in state i, but state i was predicted $\rightarrow$ false positive, FP_i

(d) The system is not is state i and other states are predicted $\rightarrow$ true negative, TN_i

By putting (a) to (d) into relation to all possible outcomes, the probabilities (referred to as rates) of each combination are obtained. They are defined as:

$$\begin{aligned} TPR_i &= \frac{TP_i}{TP_i + FN_i} \\ FNR_i &= \frac{FN_i}{TP_i + FN_i} = 1 - TPR_i \\ FPR_i &= \frac{FP_i}{TN_i + FP_i} \\ TNR_i &= \frac{TN_i}{TN_i + FP_i} = 1 - FPR_i \end{aligned} \tag{5.69}$$

A formal definition of the TP, FN, FP and TN rates for a multi-class confusion matrix is also given in the Appendix D.1 of this work.

As given in [Lil18], each possibility for an outcome is then assigned to a corresponding cost factor. Consequently, the following cost factors are defined:

(a) C_{TP}: Costs that arise for a correct prediction of a state. Usually it is expected that these costs are the normal operating costs that exist.

(b) C_{FN}: Costs that arise through a wrong prediction of the current state. This can relate to unplanned maintenance costs, costs due to a loss, financial penalties that have to be paid as a service was not available (e.g. in a PBC scenario) or indirect costs e.g. from reputational damage.

(c) C_{FP}: Costs that arise through the wrong prediction of an upcoming state. This can relate to costs as false decisions were made. E.g. the system was stopped, or an expensive maintenance action was carried, although there was no need for this.

(d) C_{TN}: Costs linked to true negatives: In accordance with [Lil18], it is assumed that $C_{TN} = 0$ for most cases as the true negative costs are also represented by the C_{TP}, C_{FN} and C_{FP} of the other states.

With the above definitions, the total costs for the correct respectively wrong state prediction for a given state i is defined as:

$$C_i = C_{TP,i} \cdot TPR_i + C_{FN,i} \cdot FNR_i + C_{FP,i} \cdot FPR_i \tag{5.70}$$

Assuming that the individual cost factors C_{TP}, C_{FN} and C_{FP} are constant regardless of whether PHM was used or not and only the prediction rates vary between a PHM approach and the reference case, Equation 5.67 can be rewritten as:

$$ROI = \frac{\sum_{i=1}^{n} \left(C_{TP,i}\Delta TPR_i + C_{FN,i}\Delta FNR_i + C_{FP,i}\Delta FPR_i\right) - I}{I} - 1 \tag{5.71}$$

where: $\Delta TPR_i = TPR_{Ref,i} - TPR_{PHM,i}$
$\Delta FNR_i = FNR_{Ref,i} - FNR_{PHM,i}$
$\Delta FPR_i = FPR_{Ref,i} - FPR_{PHM,i}$

With the aim of achieving a positive ROI it becomes clear from the above equations that the DHRM approach becomes economical beneficial when the following conditions are met:

- There is a large cost-saving possible through the correct prediction of system states (large C_{TP}, C_{TN} or C_{FP}) while the prediction performance is slightly better than a reference case ($\Delta TPR_i > 0$, $\Delta FNR_i > 0$ and $\Delta FPR_i > 0$).

- The cost-savings are moderate but the DHRM approach is much better than a conventional approach (low to moderate C_{TP}, C_{TN} and C_{FP}) but high prediction performances ($\Delta TPR_i > 0$, $\Delta FNR_i > 0$ and $\Delta FPR_i > 0$) compared to reference.
- The investment I is small compared to possible savings.

5.4. Discussion of results

With the evaluation being completed, the results of this work are discussed in the following section with respect to the initially identified research gap and need. The discussion of the results are again structured into performance, efficiency and economic considerations.

Performance aspects
As initially stated in Chapter 1, an accurate assessment of the current and future system capabilities is of important matter for operators. The relevant keyword in that regard is the performability of a system. As it was shown in Figure 1.1, two key performance indicators of performability are the performance of a system, defined as the ability of a system to accomplish an intended service within given constraints, and the availability, which measures the readiness for correct service when demanded. With the presented DHRM a novel approach has been introduced that helps to assess and relate both metrics. In that regard, the different performance levels of a system are considered as its states. The availability is then expressed as the probability that the system will be in a certain state at a given time. With this combination of performance level prediction, the operator of a complex system obtains a decision support that helps to precisely estimate the future capabilities of a given system. As it has been shown throughout Section 5.1 of this chapter, the DHRM approach outperforms the considered reference case where only static input data is used as shown with the ACC metric in Figure 5.1. By incorporating PHM data that captures the real health state of the underlying system components and on top provides an estimation of their end of life times, the overall assessment becomes much more accurate as the state predictions are adjusted with every recalculation of the model. In addition to this, it has been shown by the TPR metric in Figure 5.2 that not only the transition time is predicted accurately but also the next state the system is foreseen to enter. This capability is considered as one of the main advantages of the DHRM approach as it allows operators to plan ahead and prepare for the upcoming new state or respectively to prevent the state transition (e.g. through a maintenance action). Regarding the uncertainty handling it can be said that the model is equipped with all means to propagate uncertainties resulting from the underlying prognosis algorithms as well as failure rates. In general, the influence

of uncertainty is especially visible for the dominant factors. E.g. for parts that contribute only very little to the overall reliability of a component, the uncertainty has a minor impact on the overall calculation as shown in Figure 5.4.

However, the proposed method also comes with limitations. Regarding the prediction performance it can be said that the model's results mainly depend on the accuracy of the underlying PHM algorithms as well as the overall contribution of the monitored parts towards failure events of the belonging components. Accordingly, a system without any PHM-monitoring will fall back to a conventional approach with static failure rates. Instead, a system with many monitored parts or a system where the failure event contribution of monitored parts is high, will benefit from the proposed approach as the individual failure events and state transitions become predictable – at least provided that the PHM algorithms are of sufficient accuracy. Also, it is worthy to note that the accuracy of the state predictions will depend on the update-frequency of the DHRM calculation and PHM data as well as the distance between consecutive state transitions. E.g. if the true state transitions are close together, but the update frequency of the model is too far apart, a clear distinction between the following states will not be possible.

A further constraint of the DHRM lies in the fact that it does not consider any inference between the degradation of the individual components. Accordingly, it is assumed that the degradation level of a given component has no effect of the degradation rate of another component, which is not true for all systems. Despite this limitation, it is assumed that the inference can be captured by the (PHM) monitoring of the affected other components to a certain degree for compensation. Last but not least, it must be said that the DHRM approach only is viable for systems with different distinguishable performance states. For a rather simple system with a binary state-space (functional vs. failed) the shown concept provides no real benefit.

Calculation aspects

As it was shown in Section 5.2, the overall calculation of the DHRM is not trivial as probabilities and uncertainties have to be propagated at the same time. Also, the calculation efficiency depends on several parameters such as the number of parts, components and systems states. The key-strategy used in this work to reduce the calculation effort was to avoid any exponential time complexity as long as possible. Therefore, it was chosen to use the fault tree model with a polynomial time complexity to model the failure propagation on component-level, where most parts – and thus calculations – are expected (compare Section 3.2.4). This already helps to reduce the required calculation effort and is viable as long as every component can be represented by a two-state model, being either functional or non-functional (compare Table 3.3). Accordingly, only on system-level the much more time-consuming state-space model was used. However, this model had to be used as it is

required to represent multiple states. Nevertheless, the critical state-space explosion can be further avoided by grouping similar states (in terms of their severity) together as it was shown in Section 3.2.5.

Regarding the uncertainty propagation it can be said that the main contributing factor is the number of used samples. Thus, it is important to choose a sampling size that is as small as possible, but still high enough to capture the uncertainty effects. With the replica-based approach as shown in Section 3.3.3 a measure is provided that can be used to quantify the quality of the MSC. As it is expected that the required number of samples varies depending on the component complexity, the sampling size can be individually chosen. Finally, it is also important to highlight that the calculation time of the DHRM decreases linearly for later prediction times. This is crucial as it helps to provide enough preparation time for the decision maker. Also, the relation between operating time and calculation time has to be kept in mind. For a system that is planned to be operated for several thousand hours, a calculation time of a few hours might be still acceptable.

Economic aspects
The conducted economic consideration in Section 5.3 has shown that the DHRM method has the potential to save on expanses during system operation as it is assumed that a precise state prediction can lead to cost saving e.g. trough and optimized maintenance or operating scheme. As key-drivers to increase the ROI of the method, the accuracy measures (compared to a reference case), the potential cost-savings per outcome (TN, FN, TP, FP metrics) as well as the overall investment I were identified. As the costs are directly linked to the correct assessment of the individual system states, the shown approach is suited to be applied in a PBC scenario. Lastly, the shown ROI assessment can also be used to identify for which states/components of a system a monitoring would be favorable or to specify requirements for the underlying PHM-algorithms, similar to the approach shown in [Käh17].

Despite the purely economic consideration as provided in Section 5.3, the presented DHRM method can also contribute to improve on sustainability. A proper overview of the link between (predictive) maintenance and sustainability is given in [Fra+18]. As shown in [Pol+21], it is assumed that a PdM strategy also has the potential to improve a company's sustainability scores in the long term. This is justified by the assumption that not only spare parts but also other resources can be saved through the use of predictive maintenance, leading to a positive effect on the sustainability KPIs. With this in mind, even a small or slightly negative ROI can be beneficial for a corporate in a greater context.

Further considerations
The proposed method obviously is only applicable for systems for which end-of-life predic-

tions of individual parts are available and can be centrally processed. Accordingly, there must be means in place to exchange that information and process it in a standardized way. With this in mind, the aggregation might be easy in theory but can easily become cumbersome in practice if the following circumstances apply:

- Components/parts are provided by different suppliers with different interfaces etc.
- Components of a complex system are geographically distributed
- Data transport and interfaces are not harmonized
- Data volumes are very large
- Update rates vary strongly between the monitored components
- Different security constraints among different components

With this said, from an implementation point of few the question arises, how to practically bring the data together and evaluate it efficiently. To accomplish this, PHM frameworks and IT architectures are required that support the handling of mass data. Research in this regard have been conducted in [AGB18] and [LVC20], where such frameworks are described. In addition to this, with the rise of the Internet of Things (IoT), a technology that interconnects smart machinery and components via the internet, an important building block for the further realization exists. An exhaustive overview of available platforms and solutions is presented in [Kra+17].

6. Summary and outlook

The following is the final and concluding chapter of this work. In this chapter, the results of this thesis are summarized, and the main findings are outlined. Afterwards, an outlook on a possible continuation of the research in this area is given.

6.1. Summary

As it was shown in Chapter 1, the proper future availability assessment of a complex system plays a crucial role for system operators as it supports their decision-making regarding upcoming missions, maintenance actions or business targets. However, this is not an easy task especially for complex technical systems, which are composed of different components, combine different technologies or are used in varying scenarios, as all this will lead to very individual degradation patterns and system states. Statistics based reliability methods, which are considered to be the state of the art, reach their limits here as they struggle to capture the true (health) state of the system and thus become inaccurate. With the novel discipline of PHM, there is an alternative to the traditional reliability engineering that is based on a direct system feedback trough online measurements that allows to obtain very accurate estimates on the remaining useful lifetime of monitored parts. Being an emerging technology, there is still the question on how to aggregate and integrate PHM results within multi-component systems.

A comprehensive literature search on this topic in Section 2.3.6 revealed that there are already initial research approaches to extend the PHM approach on a system-level. However, this also revealed that most of the presented approaches expand the concept of remaining useful lifetime prediction to a system-wide S-RUL, without making a distinction according to different performance levels. From a practical point of view, however, this is exactly what is important for an operator in order to be able to better plan the use of the system. Accordingly, it was defined as the goal of this work to develop an aggregation method that

estimates the current and future states of a complex system based on prognostics and in that way can be used as decision support for operators to increase the availability of the system. In this context, the feasibility, integration and performance of such a method were defined as accompanying questions to focus on throughout the work.

Within Chapter 3 of this work, a novel method to accomplish the defined goals was conceptualized. The so-called DHRM was developed based on the V-model approach and combines methods of classical Reliability Engineering with feedback-based PHM information. According to the technical composition of (complex) systems, the model was defined and described for the three levels *part*, *component* and *system*. For the lowest level (part-level), a RUL-CDF was defined, which can be used and interpreted analogously to classical failure distributions. On the component-level, it was then shown how individual component failures can be propagated with the aid of a fault tree. Finally, on the system-level, a state-space model was used to represent the failures of individual system components and link them to a predefined system state. To account for possible uncertainties, the respective sources were first identified and then described for all three levels how they can be propagated. For this a two-tiered MCS approach was chosen. Finally, the entire DHRM was implemented in the form of a prototype in Matlab/Simulink and verified with the aid of sample data.

In Chapter 4, the created DHRM was then applied to a use-case from the aviation industry. For this purpose, the CSAS of a hybrid drone was examined. The entire system, consisting of six identical EMAs, was investigated for the underlying wear mechanisms. While the electronic components (sensors and controller) were modeled using conventional failure rates, the electromechanical components (BLDC and gearbox) were assumed to have a continuous wear process and PHM monitoring. The damage progression of the respective motors and gearboxes was finally simulated using an SDE framework. Subsequently, a prediction algorithm was created based on this artificially created degradation database. The GPR-based algorithm was afterwards used to generate RUL forecasts, which were used as input to the DHRM. Finally, a reference model was created that did not use PHM information to allow a later comparison and evaluation of the novel method.

In Chapter 5, the created DHRM was then evaluated to derive its overall performance and allow a discussion of advantages and disadvantages of this method. For this purpose, a set of comparison metrics was defined at first. Afterwards the model was simulated with different parameter combinations and compared to the initial reference model. In this context, the three aspects performance, computational efficiency as well as economic efficiency of the new approach were evaluated. The results of this evaluation showed that the novel DHRM outperforms the reference model in several areas. Not only does it provide a more accurate estimate of the time of a state transition, but it also provides an accurate estimate of the subsequent state that will occur. Consideration of the time complexity

revealed that the model behaves in a maximal polynomial fashion except for the merge at the last level, the system-level. However, the computation time can be shortened by merging several states of the same severity. Based on an ROI analysis it was shown, that the new method's economic advantages depend on the associated cost-savings per correct state prediction as well as on the accuracy of the state predictions.

6.2. Conclusions and key findings

With respect to the original stated research objectives, it can be said that this work has demonstrated and proven that it is technically feasible to aggregate PHM information within a multi-component system to assess and predict current and future system performance. Therefore, it was shown how PHM results can be combined with existing models from reliability engineering. The combination of both disciplines is considered to be an essential advantage and innovation of the resulting method, since the strengths of both disciplines come into play. These are namely the existence of established models for the aggregation and updating of reliabilities based on Reliability Engineering and the generation of a highly accurate condition assessment by PHM.

The results from Section 5.1 have further shown that with the presented approach not only the time of a state change, but also the subsequent state can be estimated accurately. This is seen as a major advantage over other methods where either no state differentiation is made or a clear distinction of the next state is not possible due to the lack of active system feedback. Thus, the operator of the system receives a decision support which helps him to react to upcoming events.

The final conclusion of this work is that the proper aggregation of PHM information in a multi-component context has the potential to be a great support in the decision-making process for operators of complex systems as it provides an accurate assessment about the current and future system capabilities. At the same time, however, it must be said that the advantages of this method can only be realized if the following conditions are met:

- The system is composed of multiple components with independent degradation.
- The system has several distinguishable performance levels in relation to failed components.
- PHM monitoring is available for failure dominant parts.
- The used PHM algorithms are sufficiently accurate.

- All calculations can be performed in a sufficiently short interval, leaving enough time for action.

From a practical point of view, it also has to be mentioned that there will be more efforts to be undertaken in the future to further standardize and harmonize the interaction and communication between multiple PHM-capable parts and components. Also, the technical means in terms of IT platforms needs to be evaluated to foster the use of such an aggregation method.

6.3. Outlook

In the context of this work, a first step was taken to aggregate prognostic data of a complex system in order to provide a decision support to the operator. The scope of the work is limited to the concept formulation and an experimental proof of concept (Technology Readiness Level 1-3). For further development and validation of the approach, an experimental investigation in the form of a physical demonstrator is proposed. A particular challenge and related research question is seen in the exchange of information between the respective components. In particular, this rises the question of suitable technologies for data transport and exchange, uniform interfaces and data protocols. The data must also be consolidated and evaluated on a central platform. The corresponding architecture and interfaces must be defined and existing technologies such as IoT must be investigated.

In relation to the PHM environment, the overall question is how data exchange and the interaction of algorithms can be harmonized and standardized in the future. Particularly in the case of applications within multi-component systems, it can be assumed that individual components will be contributed by different suppliers. In order to enable communication between individual devices, standards and conventions must be created that can be applied throughout the industry to enable the smooth exchange of information.

As far as the aggregation method itself is concerned, it is limited to pure decision support. Under the premise that the method is applied to a concrete system, it could be extended to a decision automation, in which the system itself takes action. In this context, the system operator would not take an active part and would only assume a monitoring function. Such an automation would be especially interesting for highly autonomous systems, where a direct intervention is only possible with difficulty or with a significant time delay, as is the case for space applications, for example.

References

[AGB18] P. P. Adhikari, H. Gururaja Rao, and M. Buderath. „Machine Learning based Data Driven Diagnostics & Prognostics Framework for Aircraft Predictive Maintenance“. In: *10th International Symposium on NDT in Aerospace* (2018).

[And15] Andrew Cook - University of Westminster. *European airline delay cost reference values - updated and extended values (Version 4.1)*. 2015.

[Ang18] Christoph Anger. „Hidden semi-Markov Models for Predictive Maintenance of Rotating Elements“. PhD thesis. Darmstadt, Germany: TU Darmstadt, 2018.

[AL77] G. Apostolakis and Y. T. Lee. „Methods for the Estimation of Confidence Bounds for the Top-Event Unavailability of Fault Trees“. In: *Nuclear Engineering and Design* 41 (1977), pp. 411–419. ISSN: 00295493.

[BDN11] A. C. Bertolino, F. Di Giandomenico, and N. Nostro. „Dependability and Performance Assessment of Dynamic CONNECTed Systems“. In: *11th International School SFM* (2011), pp. 350–392.

[Bir10] Alessandro Birolini. *Reliability engineering: Theory and practice*. 6th ed. Heidelberg, New York: Springer, 2010. ISBN: 9783642149511.

[Bit+86] Peter Bitter et al., eds. *Technische Zuverlässigkeit: Problematik · Mathematische Grundlagen Untersuchungsmethoden · Anwendungen*. Dritte, neubearbeitete und erweiterte Auflage. Berlin, Heidelberg: Springer Berlin Heidelberg, 1986. ISBN: 9783540167051. DOI: 10.1007/978-3-642-88364-4.

[Bro+09] Douglas W. Brown et al. „Prognostics Enhanced Reconfigurable Control of Electro-Mechanical Actuators“. In: *Annual Conference of the Prognostics and Health Management Society* (2009).

[BKL08] M.D Bryant, M.M Khonsari, and F.F Ling. „On the thermodynamics of degradation“. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 464 (2008), pp. 2001–2014. ISSN: 1364-5021. DOI: 10.1098/rspa.2007.0371.

[Bun19] Bundesministerium der Verteidigung. *Bericht zur Materiellen Einsatzbereitschaft der Hauptwaffensysteme der Bundeswehr*. 2019.

[Bun21] Bundesministerium der Verteidigung. *14. Bericht des Bundesministeriums der Verteidigung zu Rüstungsangelegenheiten - Teil 1*. 2021.

[Bun18] Bundesverband der Deutschen Luft- und Raumfahrtindustrie e.V. *Positionspapier: Performance Based Logistics (PBL) für fliegende Waffensysteme der Bundeswehr im Inland*. 2018.

[ČM02] Marko Čepin and Borut Mavko. „A dynamic fault tree“. In: *Reliability Engineering and System Safety* 75 (2002), pp. 83–91. DOI: 10.1016/S0951-8320(01)00121-1.

[Chi+11] F. Chiacchio et al. „Dynamic fault trees resolution: A conscious trade-off between analytical and simulative approaches“. In: *Reliability Engineering and System Safety* 96 (2011), pp. 1515–1526. DOI: 10.1016/j.ress.2011.06.014.

[Cho+19] Anurag Choudhary, Deepam Goyal, Sudha Letha Shimi, and Aparna Akula. „Condition Monitoring and Fault Diagnosis of Induction Motors: A Review“. In: *Archives of Computational Methods in Engineering* 26 (2019), pp. 1221–1238. DOI: 10.1007/s11831-018-9286-z.

[DSK11] Yao Da, Xiaodong Shi, and Mahesh Krishnamurthy. „Health monitoring, fault diagnosis and failure prognosis techniques for Brushless Permanent Magnet Machines“. In: *IEEE Vehicle Power and Propulsion Conference* (2011), pp. 1–7. DOI: 10.1109/VPPC.2011.6043248.

[DG11] M. Daigle and K. Goebel. „A Model-Based Prognostics Approach Applied to Pneumatic Valves“. In: *International journal of prognostics and health management* 2 (2011), pp. 84–99.

[DK13] M. Daigle and C. S. Kulkarni. „Electrochemistry-based Battery Modeling for Prognostics“. In: *Annual Conference of the Prognostics and Health Management Society* (2013). DOI: 10.36001/phmconf.2013.v5i1.2252.

[DBR12] Matthew Daigle, Anibal Bregon, and Indranil Roychoudhury. „A Distributed Approach to System-Level Prognostics“. In: *Annual Conference of the Prognostics and Health Management Society* (2012).

[Das+10] Abhijit Dasgupta et al. „The Use of “Canaries” for Adaptive Health Management of Electronic Systems“. In: ADAPTIVE 2010, IARIA Conference (2010).

[DWG15] Yunli Deng, He Wang, and Biao Guo. „BDD algorithms based on modularization for fault tree analysis“. In: *Progress in Nuclear Energy* 85 (2015), pp. 192–199. ISSN: 01491970. DOI: 10.1016/j.pnucene.2015.06.019.

[Dep91] Department of Defense. *Militiary Handbook (MIL-HDBK-217F): Reliability Prediction of Electronic Equipment*. 1991.

[DDH19] H. Heath Dewey, Derek R. DeVries, and Scott R. Hyde. „Uncertainty Quantification in Prognostic Health Management Systems“. In: *2019 IEEE Aerospace Conference* (2019), pp. 1–13. DOI: 10.1109/AERO.2019.8741821.

[Dur+09] K. Durga Rao et al. „Dynamic fault tree analysis using Monte Carlo simulation in probabilistic safety assessment“. In: *Reliability Engineering & System Safety* 94.4 (2009), pp. 872–883. ISSN: 09518320. DOI: 10.1016/j.ress.2008.09.007.

[Eng+00] S. J. Engel, B. J. Gilmartin, K. Bongort, and A. Hess. „Prognostics, the real issues involved with predicting life remaining“. In: (2000), pp. 457–469. DOI: 10.1109/AERO.2000.877920.

[Err02] Robert Errichello. „How to analyze gear failures“. In: *Practical Failure Analysis* 2.6 (2002), pp. 8–16. ISSN: 1529-8159. DOI: 10.1007/BF02715492.

[EB03] M.H.C. Everdij and H.A.P. Blom. *Petri-nets and hybrid-state Markov processes in a power-hierarchy of dependability models*. 2003.

[ELB18] Mark E. Ewing, Brian C. Liechty, and David L. Black. „A General Methodology for Uncertainty Quantification in Engineering Analyses Using a Credible Probability Box“. In: *Journal of Verification, Validation and Uncertainty Quantification* 3.2 (2018). ISSN: 2377-2158. DOI: 10.1115/1.4041490.

[Fed16] Federal Aviation Administration. *FAA Aerospace Forecast: Fiscal Year 2016-2036*. 2016.

[Fei18] A. Feinberg. „Physics of Failure Laws Using Thermodynamic Degradation Science“. In: (2018).

[FSJ08] Kiri Feldman, Peter Sandborn, and Taoufik Jazouli. „The analysis of Return on Investment for PHM applied to electronic systems“. In: *2008 International Conference on Prognostics and Health Management* (2008), pp. 1–9. DOI: 10.1109/PHM.2008.4711415.

[FFP17] Eduardo de Francesco, Ruggero de Francesco, and Enrico Petritoli. „Obsolescence of the MIL-HDBK-217: A critical review“. In: *2017 IEEE International Workshop on Metrology for AeroSpace (MetroAeroSpace)* (2017), pp. 282–286. DOI: 10.1109/MetroAeroSpace.2017.7999581.

[Fra+18] Chiara Franciosi, Benoit Iung, Salvatore Miranda, and Stefano Riemma. „Maintenance for Sustainability in the Industry 4.0 context: a Scoping Literature Review“. In: *IFAC-PapersOnLine* 51.11 (2018), pp. 903–908. ISSN: 24058963. DOI: 10.1016/j.ifacol.2018.08.459.

[Glo+10] William Glover et al. „The Use of Prognostic Health Management for Autonomous Unmanned Air Systems“. In: *Annual Conference of the Prognostics and Health Management Society* (2010).

[GSB08] A. Gorod, B. Sauser, and J. Boardman. „System-of-Systems Engineering Management: A Review of Modern History and a Path Forward: A Review of Modern History and a Path Forward“. In: *IEEE Systems Journal* 2.4 (2008), pp. 484–499. ISSN: 1932-8184. DOI: 10.1109/JSYST.2008.2007163.

[GBV20] Margherita Grandini, Enrico Bagli, and Giorgio Visani. „Metrics for Multi-Class Classification: an Overview“. In: (2020).

[Gua+12] Jose A. Guajardo, Morris A. Cohen, Sang-Hyun Kim, and Serguei Netessine. „Impact of Performance-Based Contracting on Product Reliability: An Empirical Analysis“. In: *Management Science* 58.5 (2012), pp. 961–979. ISSN: 0025-1909. DOI: 10.1287/mnsc.1110.1465.

[Hei19] Henrik Heier. „Uncertainty Propagation in a PHM Enhanced Dynamic Reliability Model“. In: *2019 IEEE Aerospace Conference*. IEEE, 2019, pp. 1–11. ISBN: 978-1-5386-6854-2. DOI: 10.1109/AERO.2019.8742154.

[Hes02] A. Hess. „Prognostics, from the need to reality-from the fleet users and PHM system designer/developers perspectives“. In: *Proceedings, IEEE Aerospace Conference* 6 (2002), pp. 2791–2797. DOI: 10.1109/AERO.2002.1036118.

[HL97] R. K. Holman and P. K. Liaw. „Methodologies for predicting fatigue life“. In: *JOM* 49.7 (1997), pp. 46–52. ISSN: 1047-4838. DOI: 10.1007/BF02914767.

[IH88] Ronald L. Iman and Jon C. Helton. „An Investigation of Uncertainty and Sensitivity Analysis Techniques for Computer Models“. In: *Risk Analysis* 8.1 (1988), pp. 71–90. ISSN: 0272-4332. DOI: 10.1111/j.1539-6924.1988.tb01155.x.

[Ise11] Rolf Isermann. *Fault-Diagnosis Applications: Model-Based Condition Monitoring: Actuators, Drives, Machinery, Plants, Sensors, and Fault-tolerant Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. ISBN: 9783642127663. DOI: 10.1007/978-3-642-12767-0.

[JHY82] P. S. Jackson, R. W. Hockenbury, and M. L. Yeater. „Uncertainty analysis of system reliability and availability assessment“. In: *Nuclear Engineering and Design* 68.1 (1982), pp. 5–29. ISSN: 00295493. DOI: 10.1016/0029-5493(82)90037-1.

[Jam08] M. Jamshidi. „System of systems engineering - New challenges for the 21st century“. In: *IEEE Aerospace and Electronic Systems Magazine* 23.5 (2008), pp. 4–19. ISSN: 0885-8985. DOI: 10.1109/MAES.2008.4523909.

[JS11] Taoufik Jazouli and Peter Sandborn. „Using PHM to meet availability-based contracting requirements“. In: *2011 IEEE Conference on Prognostics and Health Management* (2011), pp. 1–12. DOI: 10.1109/ICPHM.2011.6024317.

[JK95] Jianan Xue and Kai Yang. „Dynamic reliability analysis of coherent multistate systems“. In: *IEEE Transactions on Reliability* 44.4 (1995), pp. 683–688. ISSN: 0018-9529. DOI: 10.1109/24.476002.

[JM88] Allen M. Johnson and Miroslaw Malek. „Survey of software tools for evaluating reliability, availability, and serviceability“. In: *ACM Computing Surveys* 20.4 (1988), pp. 227–269. ISSN: 03600300. DOI: 10.1145/50020.50062.

[Käh17] A. Kählert. „Specification and Evaluation of Prediction Concepts in Aircraft Maintenance“. PhD thesis. Darmstadt, Germany: TU Darmstadt, 2017.

[KP14] Kailash C. Kapur and Michael Pecht, eds. *Reliability Engineering*. Wiley Series in Systems Engineering and Management. Hoboken, NJ, USA: John Wiley & Sons, Inc, 2014. ISBN: 9781118841716. DOI: 10.1002/9781118841716.

[KBS16] Hamed Khorasgani, Gautam Biswas, and Shankar Sankararaman. „Methodologies for system-level remaining useful life prediction“. In: *Reliability Engineering & System Safety* 154 (2016), pp. 8–18. ISSN: 09518320. DOI: 10.1016/j.ress.2016.05.006.

[KAC17] N. Kim, D. An, and J. Choi. *Prognostics and Health Management of Engineering Systems*. Cham: Springer International Publishing, 2017. DOI: 10.1007/978-3-319-44742-1.

[Kne20] Helmut Knebl. *Algorithms and Data Structures: Foundations and Probabilistic Methods for Design and Analysis*. 1st ed. 2020. Springer eBook Collection. Cham: Springer International Publishing and Imprint Springer, 2020. DOI: 10.1007/978-3-030-59758-0.

[Kra+17] Tobias Krause et al. *IT-Plattformen für das Internet der Dinge (IoT): Basis intelligenter Produkte und Services*. Stuttgart, Germany, 2017.

[KTB11] Dirk P. Kroese, Thomas Taimre, and Zdravko I. Botev. *Handbook of Monte Carlo Methods*. Hoboken, NJ, USA: Wiley, 2011. ISBN: 9781118014967. DOI: 10.1002/9781118014967.

[Lan01] J. E. Land. „HUMS-the benefits-past, present and future“. In: *2001 IEEE Aerospace Conference Proceedings* (2001), pp. 3083–3094. DOI: 10.1109/AERO.2001.931326.

[LC04] Jerry Lawless and Martin Crowder. „Covariates and random effects in a gamma process model with application to degradation and failure“. In: *Lifetime data analysis* 10.3 (2004), pp. 213–227. ISSN: 1380-7870. DOI: 10.1023/b:lida.0000036389.14073.dd.

[LHH03] W. Le Roux, R. G. Harley, and T. G. Habetler. „Detecting rotor faults in permanent magnet synchronous machines“. In: *4th IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics and Drives* (2003), pp. 198–203. DOI: 10.1109/DEMPED.2003.1234573.

[Lee+14] Jay Lee et al. „Prognostics and health management design for rotary machinery systems—Reviews, methodology and applications“. In: *Mechanical Systems and Signal Processing* 42.1-2 (2014), pp. 314–334. ISSN: 08883270. DOI: 10.1016/j.ymssp.2013.06.004.

[LVC20] Rui Li, Wim J.C. Verhagen, and Richard Curran. „A systematic methodology for Prognostic and Health Management system architecture definition“. In: *Reliability Engineering & System Safety* 193 (2020), p. 106598. ISSN: 09518320. DOI: 10.1016/j.ress.2019.106598.

[LK14] Linxia Liao and Felix Kottig. *Review of Hybrid Prognostics Approaches for Remaining Useful Life Prediction of Engineered Systems, and an Application to Battery Life Prediction*. 2014. DOI: 10.1109/TR.2014.2299152.

[Lil18] Lily Koops. „ROC-based Business Case Analysis for Predictive Maintenance-Applications in Aircraft Engine Monitoring“. In: *European Conference of the Prognostics and Health Management Society* (2018).

[LLZ15] Yan-Hui Lin, Yan-Fu Li, and Enrico Zio. „Fuzzy Reliability Assessment of Systems With Multiple-Dependent Competing Degradation Processes“. In: *IEEE Transactions on Fuzzy Systems* 23.5 (2015), pp. 1428–1438. ISSN: 1063-6706. DOI: 10.1109/TFUZZ.2014.2362145.

[LZ16] J. Liu and E. Zio. „Dynamic reliability assessment and prognostics with monitored data for multiple dependent degradation components“. In: *European Safety and Reliability Conference* (2016).

[LM93] C. J. Lu and William Q. Meeker. *Using Degradation Measures to Estimate a Time-to-Failure Distribution*. 1993.

[MMT18] M. Hanel, M. Haimerl, and T. Bienert. „Flight Control Laws for the Quadcruiser RPA“. In: *Deutscher Luft- und Raumfahrtkongress* (2018). DOI: 10.25967/480033.

[MT94] M. Malhotra and K. S. Trivedi. „Power-hierarchy of dependability-model types“. In: *IEEE Transactions on Reliability* 43.3 (1994), pp. 493–502. ISSN: 0018-9529. DOI: 10.1109/24.326452.

[MOP12] Sony Mathew, Michael Osterman, and Michael Pecht. „A canary device based approach for prognosis of Ball Grid Array packages“. In: *2012 IEEE Conference on Prognostics and Health Management* (2012), pp. 1–5. DOI: 10.1109/ICPHM.2012.6299546.

[McP10] J. W. McPherson. *Reliability Physics and Engineering: Time-To-Failure Modeling*. Boston, MA: Springer Science+Business Media LLC, 2010. ISBN: 978-1-4419-6347-5. DOI: 10.1007/978-1-4419-6348-2.

[MHE04] William Meeker, Yili Hong, and Luis Escobar. „Degradation Models and Analyses“. In: *Encyclopedia of Statistical Sciences* (2004), pp. 1–23. DOI: 10.1002/0471667196.ess7148.

[Mey12] Bertrand Meyer, ed. *Tools for practical software verification: LASER, international summer school 2011, Elba Island, Italy ; revised tutorial lectures*. Vol. 7682. Lecture notes in computer science Programming and software engineering. Berlin and Heidelberg: Springer, 2012. ISBN: 978-3-642-35745-9.

[Mey80] J. F. Meyer. „On Evaluating the Performability of Degradable Computing Systems“. In: *IEEE Transactions on Computers* C-29.8 (1980), pp. 720–731. DOI: 10.1109/TC.1980.1675654.

[Mey92] John F. Meyer. „Performability: a retrospective and some pointers to the future: A retrospective and some pointers to the future“. In: *Performance Evaluation* 14.3-4 (1992), pp. 139–156. ISSN: 01665316. DOI: 10.1016/0166-5316(92)90002-X.

[Mis08] Krishna B. Misra. *Handbook of Performability Engineering*. London: Springer London, 2008. ISBN: 978-1-84800-130-5. DOI: 10.1007/978-1-84800-131-2.

[MR93] S. F. Morris and J. F. Reilly. „MIL-HDBK-217-A favorite target“. In: *Annual Reliability and Maintainability Symposium 1993 Proceedings* (1993), pp. 503–509. DOI: 10.1109/RAMS.1993.296808.

[NP09] Gang Niu and Michael Pecht. „A framework for cost-effective and accurate maintenance combining CBM RCM and data fusion“. In: *2009 8th international conference on reliability, maintainability and safety* (2009), pp. 605–611. DOI: 10.1109/ICRMS.2009.5270119.

[NYP10] Gang Niu, Bo-Suk Yang, and Michael Pecht. „Development of an optimized condition-based maintenance system by data fusion and reliability-centered maintenance“. In: *Reliability Engineering & System Safety* 95.7 (2010), pp. 786–796. ISSN: 09518320. DOI: 10.1016/j.ress.2010.02.016.

[OSP12] Hyunseok Oh, Tadahiro Shibutani, and Michael Pecht. „Precursor monitoring approach for reliability assessment of cooling fans“. In: *Journal of Intelligent Manufacturing* 23.2 (2012), pp. 173–178. DOI: 10.1007/s10845-009-0342-2.

[OV09] M. E. Orchard and G. J. Vachtsevanos. „A particle-filtering approach for on-line fault diagnosis and failure prognosis“. In: *Transactions of the Institute of Measurement and Control* 31.3-4 (2009), pp. 221–246. ISSN: 0142-3312. DOI: 10.1177/0142331208092026.

[PP06] C. Park and W. J. Padgett. „Stochastic Degradation Models With Several Accelerating Variables“. In: *IEEE Transactions on Reliability* 55.2 (2006), pp. 379–390. ISSN: 0018-9529. DOI: 10.1109/TR.2006.874937.

[PP05] Chanseok Park and W. J. Padgett. „Accelerated degradation models for failure based on geometric Brownian motion and gamma processes“. In: *Lifetime data analysis* 11.4 (2005), pp. 511–527. ISSN: 1380-7870. DOI: 10.1007/s10985-005-5237-8.

[PK88] M. Pecht and Wen-Chang Kang. „A critique of Mil-Hdbk-217E reliability prediction methods“. In: *IEEE Transactions on Reliability* 37.5 (1988), pp. 453–457. ISSN: 0018-9529. DOI: 10.1109/24.9859.

[PK18] Michael G. Pecht and Myeongsu Kang. *Prognostics and Health Management of Electronics*. Chichester, UK: John Wiley and Sons Ltd, 2018. ISBN: 9781119515326. DOI: 10.1002/9781119515326.

[Pol+21] Francesco Polese, Carmen Gallucci, Luca Carrubbo, and Rosalia Santulli. „Predictive Maintenance as a Driver for Corporate Sustainability: Evidence from a Public-Private Co-Financed R&D Project“. In: *Sustainability* 13.11 (2021), p. 5884. DOI: 10.3390/su13115884.

[PAK14] C. Preusche, C. Anger, and U. Klingauf. „Evaluation of the Training Process of three different Prognostic Approaches based on the Gaussian Process“. In: *European Conference of the Prognostics and Health Management Society* (2014).

[Pri+17] Alexandru Prisacaru et al. „Prognostics and Health Monitoring of Electronic System: A Review“. In: *18th International Conference on Thermal, Mechanica1 and Mu1ti-Physics Simulation and Experiments in Microe1ectronics and Microsystems* (2017), pp. 1–11. DOI: 10.1109/EuroSimE.2017.7926248.

[RPS15] Leonardo Ramos Rodrigues, Ivo Paixao de Medeiros, and Christian Strottmann Kern. „Maintenance cost optimization for multiple components using a condition based method“. In: *Annual IEEE Systems Conference (SysCon) Proceedings* (2015), pp. 164–169. DOI: 10.1109/SYSCON.2015.7116746.

[RW06] C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. Cambridge, Mass.: MIT Press, 2006. ISBN: 0-262-18253-X.

[RV91] A. L. Reibman and M. Veeraraghavan. „Reliability modeling: An overview for system designers: An overview for system designers“. In: *Computer* 24.4 (1991), pp. 49–57. ISSN: 0018-9162. DOI: 10.1109/2.76262.

[RRH00] Edwin D. Reilly, Anthony Ralston, and David Hemmendinger. *Encyclopedia of computer science*. 4th ed. London and New York: Nature Pub. Group, Published in the United States and Canada by Grove's Dictionaries, 2000. ISBN: 9781561592487.

[Rod17] Leonardo R. Rodrigues. „Remaining Useful Life Prediction for Multiple-Component Systems Based on a System-Level Performance Indicator“. In: *IEEE/ASME Transactions on Mechatronics* (2017). DOI: 10.1109/TMECH.2017.2713722.

[Rv02] J.L Rouvroye and E.G van den Bliek. „Comparing safety analysis techniques“. In: *Reliability Engineering & System Safety* 75.3 (2002), pp. 289–294. ISSN: 09518320. DOI: 10.1016/S0951-8320(01)00116-8.

[SAE96] SAE International. *Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment (ARP4761)*. 400 Commonwealth Drive, Warrendale, PA, United States, 1996. DOI: 10.4271/ARP4761.

[SM06] J. H. Saleh and K. Marais. „Highlights from the early (and pre-) history of reliability engineering“. In: *Reliability Engineering & System Safety* 91.2 (2006), pp. 249–256. ISSN: 09518320. DOI: 10.1016/j.ress.2005.01.003.

[SG13] S. Sankararaman and K. Goebel. „Why is the Remaining Useful Life Prediction Uncertain?“ In: *Annual Conference of the Prognostics and Health Management Society* (2013), pp. 1–14. DOI: 10.1002/9781118170229.ch1.

[SCS09] A. Saxena, J. Celaya, and Saha, B. et. al. „Evaluating Algorithm Performance Metrics Tailored for Prognostics“. In: *Proceedings of IEEE Aerospace Conference* (2009).

[Sax+08] A. Saxena et al. „Metrics for Evaluating Performance of Prognostics Techniques“. In: *1st International Conference on Prognostics and Health Management* (2008).

[Sch05] Mark Schwabacher. „A Survey of Data-Driven Prognostics“. In: *Infotech@Aerospace* (2005). DOI: 10.2514/6.2005-7002.

[Sen22] Senseye. *Report: The True Cost of Downtime 2022: How much do leading manufacturers lose through inefficient maintenance?* 2022.

[SES16] SESAR Joint Undertaking. *European Drones Outlook Study: Unlocking the value for Europe*. 2016.

[ST07] D. M. Shalev and J. Tiran. „Condition-based fault tree analysis (CBFTA): A new method for improved fault tree analysis (FTA), reliability and safety calculations“. In: *Reliability Engineering & System Safety* 92 (2007), pp. 1231–1241. ISSN: 09518320. DOI: 10.1016/j.ress.2006.05.015.

[Sha+16] Dan M. Shalev, Joseph Tiran, David Katoshevski, and Jacob Bortman. „Condition Based Reliability, Availability, Maintainability, and Safety (CB-RAMS) model: Improving RAMS predictions by combining condition-monitoring (CM) data with RAMS calculations“. In: *PHM Society European Conference* 3.1 (2016).

[SJ20] Tanvir Alam Shifat and Hur Jang-Wook. „Remaining Useful Life Estimation of BLDC Motor Considering Voltage Degradation and Attention-Based Neural Network“. In: *IEEE Access* 8 (2020), pp. 168414–168428. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.3023335.

[SSG14] Siddiqui, K., M., K. Sahay, and V. K. Giri. „Health Monitoring and Fault Diagnosis in Induction Motor- A Review“. In: *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering* 3.1 (2014), pp. 6549–6565.

[SHM11] J. Z. Sikorska, M. Hodkiewicz, and L. Ma. „Prognostic modelling options for remaining useful life estimation by industry“. In: *Mechanical Systems and Signal Processing* 25.5 (2011), pp. 1803–1836. ISSN: 08883270. DOI: 10.1016/j.ymssp.2010.11.018.

[SV02] M. Stamatelatos and W. E. Vesely. *Fault Tree Handbook with Aerospace Applications*. 2002.

[Sun+12] Bo Sun, Shengkui Zeng, Rui Kang, and Michael G. Pecht. „Benefits and Challenges of System Prognostics“. In: *IEEE Transactions on Reliability* 61.2 (2012), pp. 323–335. ISSN: 0018-9529. DOI: 10.1109/TR.2012.2194173.

[SBR96] P. V. Suresh, A. K. Babar, and V.Venkat Raj. „Uncertainty in fault tree analysis: A fuzzy approach“. In: *Fuzzy Sets and Systems* 83.2 (1996), pp. 135–141. ISSN: 01650114. DOI: 10.1016/0165-0114(95)00386-X.

[Swa01] Laura Swanson. „Linking maintenance strategies to performance“. In: *International Journal of Production Economics* 70.3 (2001), pp. 237–244. ISSN: 09255273. DOI: 10.1016/S0925-5273(00)00067-0.

[SL06] R. Szczepanik and A. Leski. *Evolution of Aircraft Maintenance/Support Concepts with Particular Reference to Aircraft Availability – Polish Air Force Perspective*. Ed. by NATO. Vilnius, Lithuania, 2006.

[TAC99] Ann T. Tai, Leon Alkalai, and Savio N. Chau. „On-board preventive maintenance: A design-oriented analytic study for long-life applications“. In: *Performance Evaluation* 35.3-4 (1999), pp. 215–232. ISSN: 01665316. DOI: 10.1016/S0166-5316(99)00006-1.

[TNM18] Ferhat Tamssaouet, Thi Phuong Khanh Nguyen, and Kamal Medjaher. „System-level Prognostics Based on Inoperability Input-output Model“. In: *Annual Conference of the PHM Society* 10.1 (2018). ISSN: 2325-0178. DOI: 10.36001/phmconf.2018.v10i1.487.

[Tch+14] Pierre Tchakoua et al. „Wind Turbine Condition Monitoring: State-of-the-Art Review, New Trends, and Future Challenges“. In: *Energies* 7.4 (2014), pp. 2595–2630. ISSN: 1996-1073. DOI: 10.3390/en7042595.

[Tia+12] Kossi Tiassou et al. „Impact of Operational Reliability Re-assessment during Aircraft Missions“. In: *31st International Symposium on Reliable Distributed Systems* (2012), pp. 219–224. DOI: 10.1109/SRDS.2012.37.

[UH18] Zia Ullah and Jin Hur. „A Comprehensive Review of Winding Short Circuit Fault and Irreversible Demagnetization Fault Detection in PM Type Machines“. In: *Energies* 11.12 (2018), p. 3309. ISSN: 1996-1073. DOI: 10.3390/en11123309.

[vP13] Adriaan van Horenbeek and Liliane Pintelon. „A dynamic predictive maintenance policy for complex multi-component systems“. In: *Reliability Engineering & System Safety* 120 (2013), pp. 39–50. ISSN: 09518320. DOI: 10.1016/j.ress.2013.02.029.

[VDI21] VDI. *VDI 2206: Entwicklung mechatronischer und cyber-physischer Systeme*. 2021.

[Voh+08] Sonia Vohnout et al. „Electronic Prognostics System Implementation on Power Actuator Components“. In: *IEEE Aerospace Conference* (2008), pp. 1–11. DOI: 10.1109/AERO.2008.4526607.

[Whi95] G. A. Whitmore. „Estimating degradation by a Wiener diffusion process subject to measurement error“. In: *Lifetime data analysis* 1.3 (1995), pp. 307–319. ISSN: 1380-7870. DOI: 10.1007/BF00985762.

[WTC13] Bairong Wu, Zhigang Tian, and Mingyuan Chen. „Condition-based Maintenance Optimization Using Neural Network-based Health Condition Prediction“. In: *Quality and Reliability Engineering International* 29 (2013), pp. 1151–1163. ISSN: 07488017. DOI: 10.1002/qre.1466.

[Xua+17] Jinquan Xuan, Xiaohong Wang, Dawei Lu, and Lizhi Wang. „Research on the safety assessment of the brushless DC motor based on the gray model“. In: *Advances in Mechanical Engineering* 9.3 (2017), pp. 1–15. ISSN: 1687-8140. DOI: 10.1177/1687814017695438.

[Yan11] Jing Yang. „Convergence and uncertainty analyses in Monte-Carlo based sensitivity analysis“. In: *Environmental Modelling & Software* 26.4 (2011), pp. 444–457. ISSN: 13648152. DOI: 10.1016/j.envsoft.2010.10.007.

[Yev10] Olexandr Yevkin. „An improved monte carlo method in fault tree analysis“. In: *Proceedings / Annual Reliability and Maintainability Symposium (RAMS), 2010*. Piscataway, NJ: IEEE, 2010, pp. 1–5. ISBN: 978-1-4244-5102-9. DOI: 10.1109/RAMS.2010.5447989.

[Yuy+15] Cao Yuyan, Wang Jian, Xie Rong, and Wang Xinmin. „Fault tree analysis of electro-mechanical actuators“. In: *Proceedings of the 34th Chinese Control Conference* (2015), pp. 6392–6396. DOI: 10.1109/ChiCC.2015.7260646.

[ZZX16] Bin Zhang, Lijun Zhang, and Jinwu Xu. „Degradation Feature Selection for Remaining Useful Life Prediction of Rolling Element Bearings“. In: *Quality and Reliability Engineering International* 32.2 (2016), pp. 547–554. DOI: 10.1002/qre.1771.

[Zio09] E. Zio. „Reliability engineering: Old problems and new challenges“. In: *Reliability Engineering & System Safety* 94.2 (2009), pp. 125–141. ISSN: 09518320. DOI: 10.1016/j.ress.2008.06.002.

Appendices

A. Appendix Chapter 2 (state of the art)

A.1. PHM algorithm performance metrics

Table A.1.: Overview of common algorithm performance metrics, based on [Sax+08]. It shall be noted that the perfect score for all metrics is zero.

	Metric	Definition	Range
Accuracy	RUL error	$\Delta_l = r_l^*(k) - r_l(k)$	$(-\infty, \infty)$
	Average bias	$B(k) = \frac{1}{L}\sum_{l=1}^{L} \Delta_l(k)$	$(-\infty, \infty)$
	False positive	$FP(r_l^*(k)) = \begin{cases} 1 & \text{if } \Delta_l(k) > t_{FP} \\ 0 & \text{otherwise} \end{cases}$	$[0, 1]$
	False negative	$FN(r_l^*(k)) = \begin{cases} 1 & \text{if } \Delta_l(k) < t_{FN} \\ 0 & \text{otherwise} \end{cases}$	$[0, 1]$
	Mean absolute percentage error	$MAPE(k) = \frac{1}{L}\sum_{l=1}^{L} \left\lvert \frac{100\Delta_l(k)}{r_l^*(k)} \right\rvert$	$[0, \infty)$
	Mean squared error	$MSE(k) = \frac{1}{L}\sum_{l=1}^{L} \Delta_l(k)^2$	$[0, \infty)$
	Mean absolute error	$MAE(k) = \frac{1}{L}\sum_{l=1}^{L} \lvert\Delta_l(k)\rvert$	$[0, \infty)$
Precision	Sample standard deviation	$S(k) = \sqrt{\frac{\sum_{l=1}^{L}(\Delta_l(k)-M)^2}{L-1}}$ M is the sample mean of the error	$[0, \infty)$
	Mean absolute deviation from the sample	$MAD(k) = \frac{1}{L}\sum_{l=1}^{L} \lvert\Delta_l(k) - M\rvert$ M is the sample median of the error	$[0, \infty)$

B. Appendix Chapter 3 (conception)

B.1. Fault tree calculation formulas

Table B.1.: Used fault tree calculation formulas in the custom Simulink fault tree library, based on [Bit+86]

Event / node type	formula
Failure probability (CDF)	$F(t)$
Reliability probability (CDF)	$R(t) = 1 - F(t)$
Parallel connection (logical AND)	$F(t) = \prod F_i(t)$
Series connection (logical OR)	$F(t) = 1 - \prod R_i(t)$
Exactly k out of n	$F(t) = \binom{n}{i} F_0(t)^{n-1} R_0(t)^i$
At least k out of n	$F(t) = \sum_{i=0}^{n-k} \binom{n}{i} F_0(t)^{n-1} R_0(t)^i$
Exactly one out of n (Exclusive OR)	$F(t) = \sum_{i=1}^{n} \left(\prod_{j=1}^{n} F_j \right) \frac{R_i}{F_i}$

Note: The formulas for *exactly k out of n* and *at least k out of n* can only be applied, if all parts of the redundant configuration are identical and thus have the same failure rate.

B.2. Verification of the DHRM implementation

In the following, the results from the verification of the DHRM implementation within the Matlab environment are outlined. The verification is based on a simplified system model composed of two identical EMA, where each EMA is based on the reference model from [Yuy+15].

B.2.1. Verification of the fault tree implementation in Matlab Simulink

The custom fault tree implementation in Matlab Simulink is verified against the proprietary software *Reliability Workbench* from Isograph. Figure B.1 shows the EMA reference fault tree model in *Reliability Workbench*, Figure B.2 the equivalent model in Simulink.

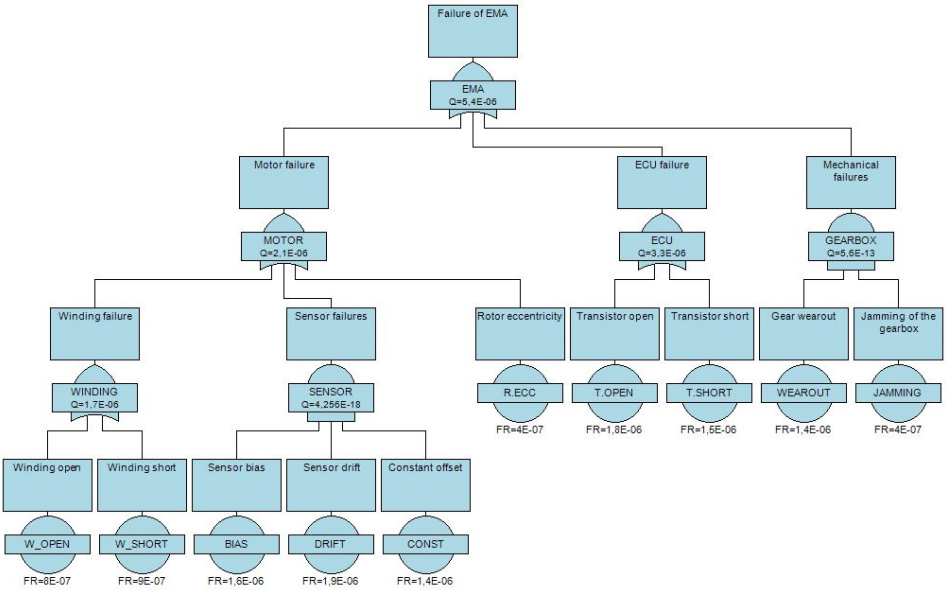


Figure B.1.: Reference EMA model, modelled in the proprietary software *Reliability Workbench*

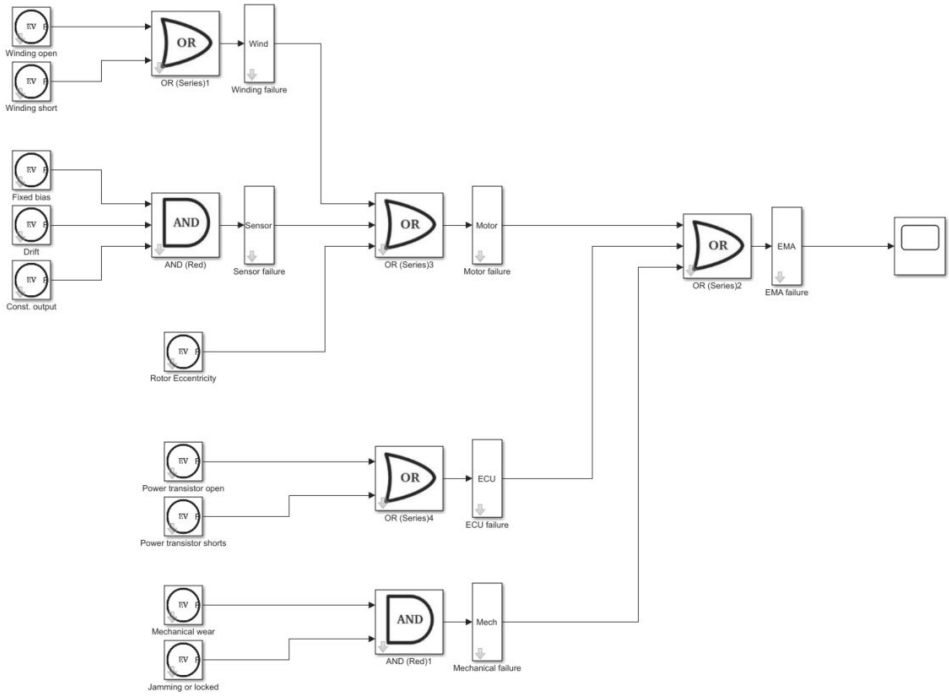


Figure B.2.: Reference EMA model created with the custom fault tree library in Simulink

Note: Being a simulation tool, the fault tree model is calculated in Simulink at discrete points in time. Thus, the user can define the start and end time of the simulation and change the result's resolution by altering the step-size. Via Simulink's input/output blocks all results can be im- and exported to the Matlab environment for further processing, which is considered as the main advantage of this setup compared to proprietary software.

Both models from Figure B.1 and B.2 are calculated with the parameters stated in Table B.2. The results from the proprietary software (RL_WB) are then compared to those from the custom fault tree setup for different step-sizes h in Table B.3.

Table B.2.: Used parameters for the calculation of the reference EMA fault tree model for the proprietary software Reliability Workbench (RL_WB) and the custom fault tree implementation. Inputs (λ) given in failures per hour.

Event	Failure mode	λ RL_WB	λ cust. FT
$E1$	Winding open	8.0E-07	8.0E-07
$E2$	Winding short	9.0E-07	9.0E-07
$E3$	Sensor bias	1.6E-06	1.6E-06
$E4$	Sensor drift	1.9E-06	1.9E-06
$E5$	Sensor offset	1.4E-06	1.4E-06
$E6$	Rotor eccentricity	4.0E-07	4.0E-07
$E7$	Power transform. Open	1.8E-06	1.8E-06
$E8$	Power transform. Short	1.5E-06	1.5E-06
$E9$	Mechanical wear	1.4E-06	1.4E-06
$E10$	Mechanical jam	4.0E-07	4.0E-07

Table B.3.: Results from both software programs with deviation ΔF for different step-sizes h

unreliability F (per cutset)	RL_WB	cust. FT	ΔF, $h = 1$	ΔF, $h = 10$	ΔF, $h = 100$
F_{EMA}	5.4E-06	5.4E-06	0	-2.00E-10	1.50E-09
F_{Motor}	2.1E-06	2.1E-06	0	0	-2.00E-10
$F_{Windings}$	1.7E-05	1.7E-05	0	0	-1.00E-10
$F_{Sensors}$	4.3E-18	4.3E-18	0	-4.21E-16	-4.25E-14
F_{ECU}	3.3E-06	3.3E-06	0	-1.00E-10	-5.00E-10
$F_{Mech\_Failure}$	5.6E-13	5.6E-13	0	-5.04E-12	-5.54E-11

B.2.2. Verification of dynamic input (RUL-CDF)

To test the behavior of the model with dynamic input data in form of the required RUL-CDFs, event node $E6$ of the exemplary fault tree model is modified to ingest dynamic data and to compare the model's outputs versus the scenario with only constant failure rates. In Figure B.3 and B.4 the reference EMA model from [Yuy+15] is shown with the modified basic event $E6$ to account for the dynamic PHM data. While the Simulink solution automatically chooses the correct input data based on the given t_P, the comparison model in Reliability Workbench was manually calculated at different prognoses times (with a manually changed parameter). The used input data, in relation to the EoL is given in Table B.4, the results of both models at different prognoses times are given in Table B.5.

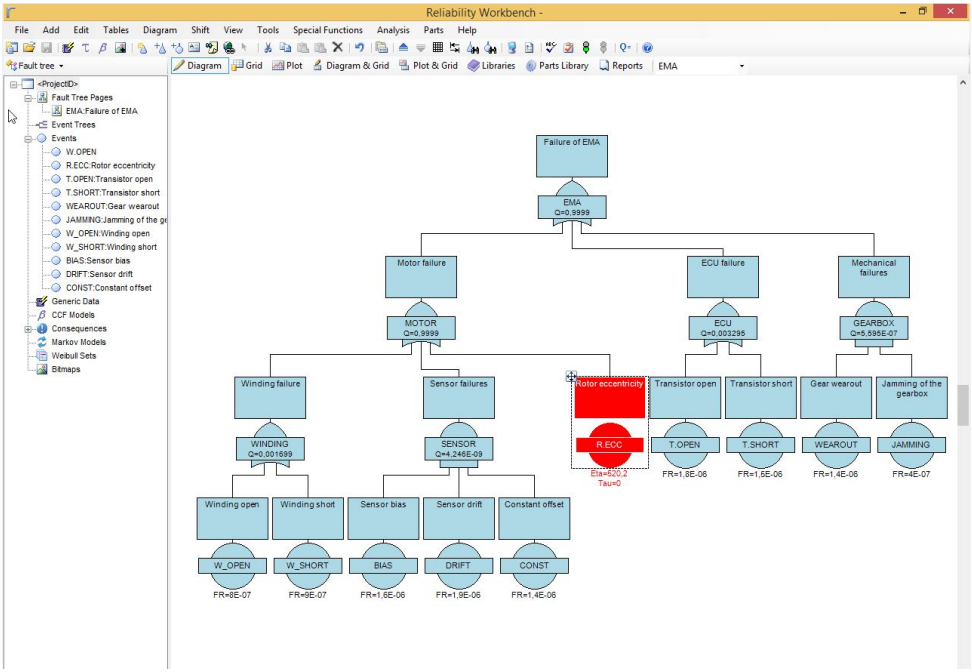


Figure B.3.: Reference fault tree model in Reliability Workbench with input parameter for basic event $E6$ chosen for iteration $k = 2$.

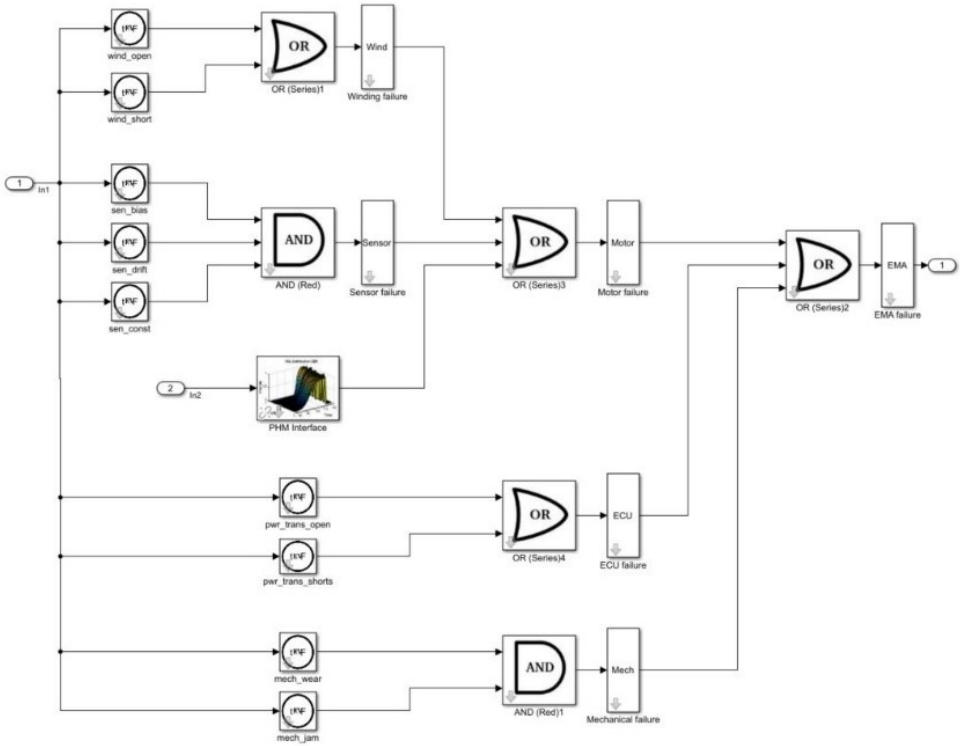


Figure B.4.: Reference fault tree model, modelled in the custom fault tree library in Simulink with a dynamic input event for event $E6$ to ingest dynamic PHM date (RUL-CDF).

Table B.4.: Iterative generation of dynamic input for node $E6$ in the fault tree model based on standard Weibull distribution with changing scale (η) and shape (β) parameter to be used at different prognosis times (represented with $\Lambda = t_P/EoL$)

Input Data Generation (Weibull)				RL_WB		cust. FT	
Iteration k	λ	η	β	η	β	η	β
0	n.a.	500.00	2.40			500.00	2.40
1	0	510.00	2.88			510.00	2.88
2	0.1	520.20	3.46	520.20	3.46	520.20	3.46
3	0.2	530.60	4.15			530.60	4.15
4	0.3	541.22	4.98			541.22	4.98
5	0.4	552.04	5.97			552.04	5.97
6	0.5	563.08	7.17			563.08	7.17
7	0.6	574.34	8.60			574.34	8.60
8	0.7	585.83	10.32			585.83	10.32
9	0.8	597.55	12.38	597.55	12.38	597.55	12.38
10	0.9	609.50	14.86			609.50	14.86

Table B.5.: Calculation results of the top node failure probability F_{EMA} evaluated at different prognoses times t_P with the proprietary software Reliability Workbench and the custom fault tree implementation in Simulink.

Node unreliability F	Prediction time	RL_WB	cust. FT
F_{EMA}	$t_P = 100$ h	0.003842	0.0038416
F_{EMA}	$t_P = 500$ h	0.1064	0.10643

B.2.3. Model parameter uncertainty (UD_{REL})

To test the uncertainty handling within the DHRM, the previously considered example EMA model from [Yuy+15] is used and calculated by means of the embedded MCS. The model's input parameters (failure rates) are chosen from two lognormal distributions A and B with varying variances σ^2. The used configuration for this experiment is listed in Table B.6. The drawn parameters are visualized in Figures B.5 and B.6.

Table B.6.: Used parameters to investigate the model's outputs under the presence of uncertainty

Failure mode	Parameter	Distribution	$log(\mu)$	σ_A^2	σ_B^2
Winding open	wind_open_lam	lognorm	8.0E-07	0.1	0.3
Winding short	wind_short_lam	lognorm	9.0E-07	0.1	0.3
Sensor bias	sen_bias_lam	lognorm	1.6E-06	0.1	0.3
Sensor drift	sen_drift_lam	lognorm	1.9E-06	0.1	0.3
Sensor offset	sen_const_lam	lognorm	1.4E-06	0.1	0.3
Power transform. Open	pwr_trans_open_lam	lognorm	1.8E-06	0.1	0.3
Power transform. Short	pwr_trans_shorts_lam	lognorm	1.5E-06	0.1	0.3
Mechanical wear	mech_wear_lam	lognorm	1.4E-06	0.1	0.3
Mechanical jam	mech_jam_lam	lognorm	4.0E-07	0.1	0.3

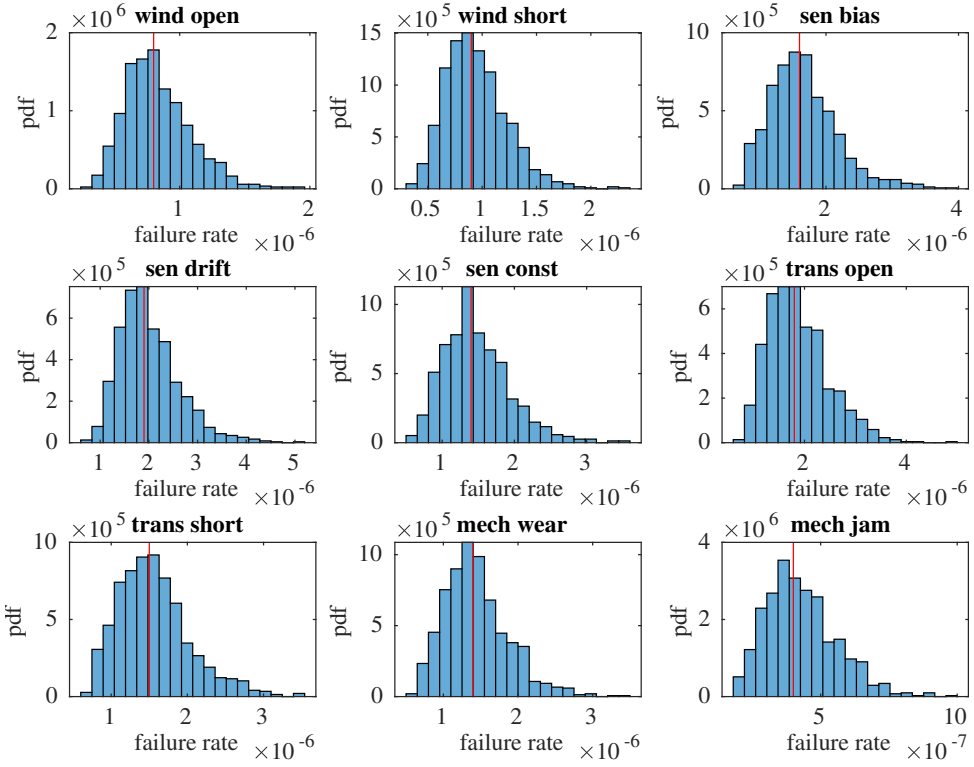


Figure B.5.: Drawn parameters for dataset A, reference values highlighted. Failure rate given in failures per hour.

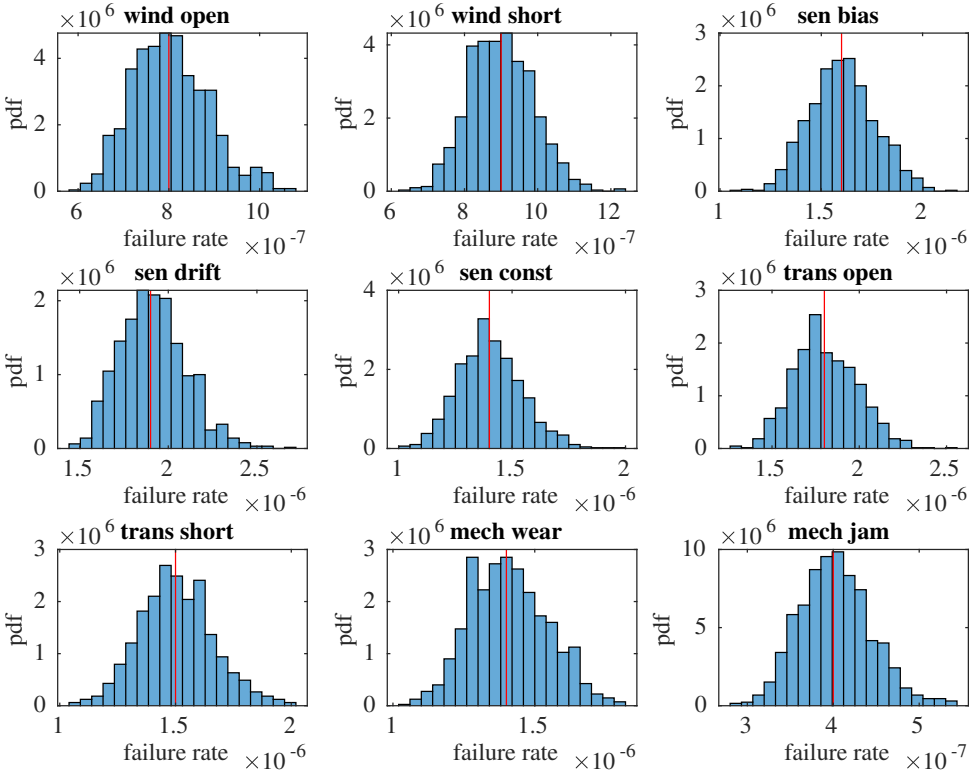


Figure B.6.: Drawn parameters for dataset B, reference values highlighted. Failure rate given in failures per hour.

C. Appendix Chapter 4 (application)

C.1. Used EMA model for the use-case

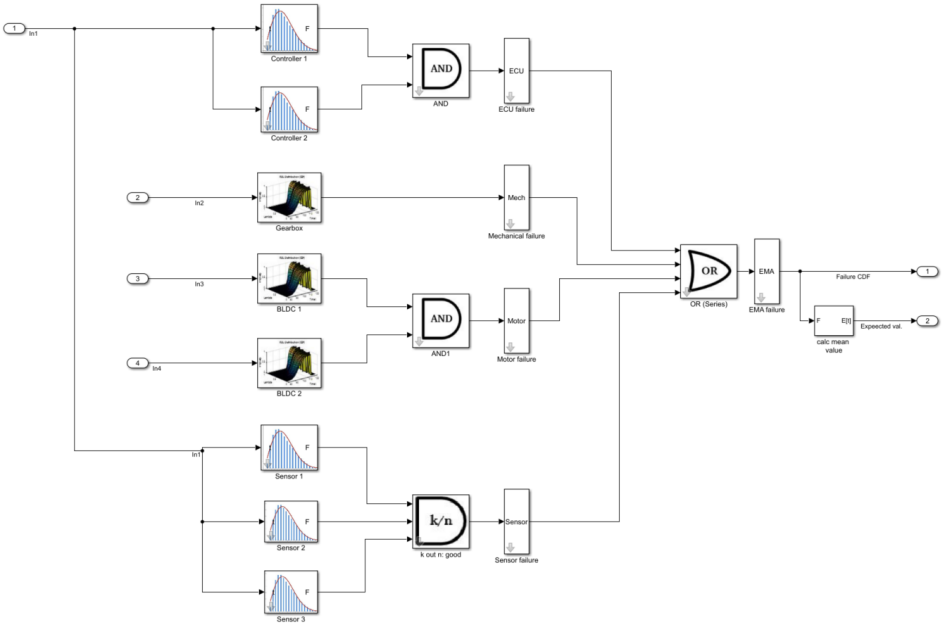


Figure C.1.: The EMA fault tree model implemented with the custom Simulink fault tree library, with dynamic inputs (PHM data) for the gearbox and both BLDC motors.

C.2. KNN validation

The KNN classifier is used for the improved prognosis algorithm. Therefore, all degradation data is clustered into three distinguishable classes with different degradation rates (A, B, C). The KNN classifier is trained on the available historic data. During the prognosis, the KNN is placed before the GPR model(s) and helps to classify the observed data first. Based on the results of the KNN classifier, the corresponding GPR model (A, B, C) is then chosen. In Figure C.2, the KNN classification for different observed degradation curves up to $t_P = 80$ are shown. In the background the reference data is printed that was used for the training of the KNN classifier.

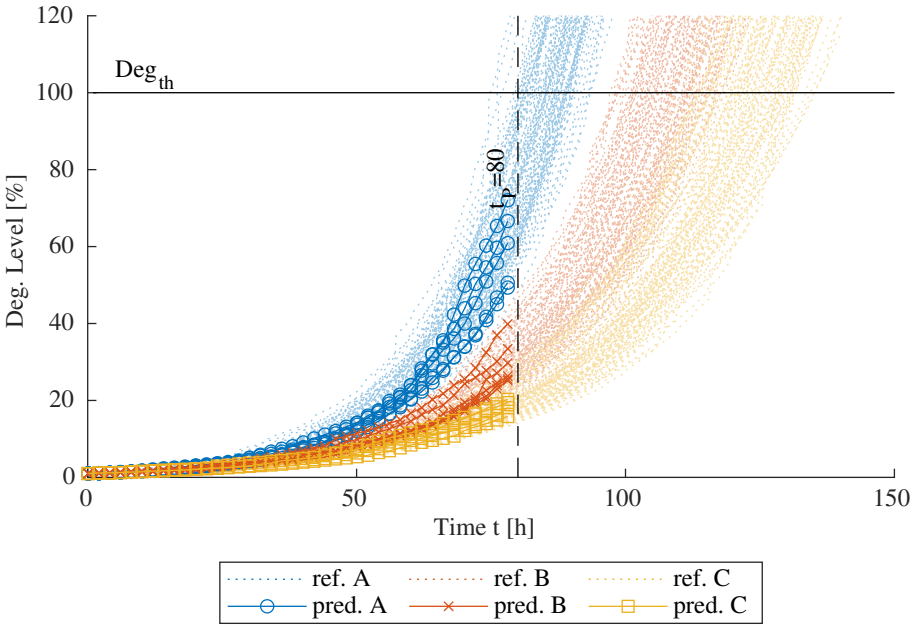


Figure C.2.: Output of the KNN classifier of the improved PHM algorithm with assignments to classes A, B and C for observed degradation data up to $t_P = 80$

D. Appendix Chapter 5 (evaluation)

D.1. Definition of the used confusion matrix

The performance and economic metrics introduced in Chapter 5 are largely based on a confusion matrix, a general concept for the evaluation of classification problems. In the following, the definition and used notation for this work is outlined, based on the example of a classification problem with three classes, respectively states:

Table D.1.: Structure of the confusion matrix for a three class problem

		Ground Truth		
		S_1	S_2	S_3
Prediction	$S_{1,est}$	$c_{1,1}$	$c_{1,2}$	$c_{1,3}$
	$S_{2,est}$	$c_{2,1}$	$c_{2,2}$	$c_{2,3}$
	$S_{3,est}$	$c_{3,1}$	$c_{3,2}$	$c_{3,3}$

With the above defined confusion matrix the basic metrics TP, FP, FN and TN can be defined. However, it is important to notice that for a multi-class problem each metric has to be defined for a specific class i of the problem. For class $i = 1$ the metrics are defined as follows:

The resulting TP, FP, FN and TN metric can then be put into the overall ratio, representing the probability of a (mis-)classification. The following table shows the relevant combinations, their definition and meaning in correspondence to the presented DHRM state prediction.

Table D.2.: Defintions of metrics TP, FP, FN and TN for state $i = 1$

	TP_i	$TP_1 = c_{1,1}$
	FP_i	$FP_1 = c_{1,2} + c_{1,3}$
	FN_i	$FN_1 = c_{2,1} + c_{3,1}$
	TN_i	$TN_1 = c_{2,2} + c_{2,3} + c_{3,2} + c_{3,3}$

Table D.3.: Probability metrics of the confusion matrix

Metric	Meaning
True Positive Rate $TPR_i = \frac{TP_i}{TP_i+FN_i}$	Probability that state i is predicted, when the true state is i.
False Negative Rate $FNR_i = \frac{FN_i}{TP_i+FN_i} = 1 - TPR_i$	Probability that state i is not predicted, when the true state is i. (Inverse of TPR_i)
False Positive Rate $FPR_i = \frac{FP_i}{TN_i+FP_i}$	Probability that state i is predicted, but the true state is different.
True Negative Rate $TNR_i = \frac{TN_i}{TN_i+FP_i} = 1 - FPR_i$	Probability that other states are predicted correctly. (Inverse of FPR_i)