Technische Universität München
TUM School of Engineering and Design

Gaussian Processes for Prognostics

Simon Bernhard Pfingstl, M.Sc.

Vollständiger Abdruck der von der TUM School of Engineering and Design der Technischen Universität München zur Erlangung eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

genehmigten Dissertation.

Vorsitz: Prof. Dr.-Ing. Wolfram Volk

Prüfer*innen der Dissertation:

1. Prof. Dr. Markus Zimmermann
2. Prof. Dr. Fu-Kuo Chang (Stanford University, Stanford, United States of America)
3. Prof. Dr.-Ing. habil. Fabian Duddeck

Die Dissertation wurde am 22.06.2022 bei der Technischen Universität München eingereicht und durch die TUM School of Engineering and Design am 23.01.2023 angenommen.

Deus satis est

Preface

Abstract

Gaussian process regression is a powerful tool for predicting variables with credible intervals. It has been used for various machine learning and uncertainty quantification applications as it provides estimates about the prediction's uncertainty by default. The thesis is concerned with how Gaussian processes can be utilized for prognostics, i.e., predicting the degradation of mechanical systems, particularly when several previous degradation trajectories are available.

Typically, predefined mean and covariance functions are employed to construct a Gaussian process model. However, these predefined functions without integrating prior information reduce the potential of Gaussian processes. This drawback is tackled by directly deriving the mean and covariance function from previous degradation trajectories. After specifying the Gaussian process, its prediction can be updated by computing the conditional distribution based on monitoring data. The approach facilitates using problem-specific governing equations, which opens the way to physics-informed Gaussian processes. An extension for treating non-normally distributed degradation trajectories is further presented.

The approach is applied to several degradation examples, such as fatigue crack growth, laser degradation, and milling machine wear. The predictive capabilities of the proposed approach are compared to the state-of-the-art Gaussian process regression and other machine learning methods. Lastly, the entire approach is applied to a real aerospace structure in order to predict future failure probabilities. The results show that the presented approach (1) outperforms the state-of-the-art Gaussian process method for prognostics applications, (2) leads to similar accuracies as recurrent neural networks while predicting the entire future degradation trajectory, and (3) needs only a fraction of training time.

Zusammenfassung

Gaußprozesse sind performante Werkzeuge für Regressionsprobleme. Sie wurden für unterschiedliche Anwendungen im Bereich des maschinellen Lernens und der Unsicherheitsquantifizierung verwendet, vor allem, weil sie zusätzlich zu Vorhersagen auch deren Unsicherheiten schätzen. Die vorliegende Dissertation beschäftigt sich mit der Implementierung von Gaußprozessen für die Vorhersage von Degradationsvorgängen, insbesondere für den Fall, bei dem mehrere Trajektorien, die den Versagensprozess widerspiegeln, verfügbar sind.

Typischerweise werden vordefinierte Mittelwerts- und Kovarianzfunktionen verwendet, um das zugrundeliegende Modell eines Gaußprozesses zu definieren. Diese vordefinierten Funktionen verringern jedoch das Potenzial von Gaußprozessen, wenn Vorinformationen nicht integriert werden. In der vorliegenden Arbeit wird deshalb die Mittelwerts- und Kovarianzfunktion direkt aus vorhandenen Trajektorien abgeleitet, um Vorinformationen in den Gaußprozess zu integrieren. Anschließend kann die Vorhersage des Gaußprozesses aktualisiert werden, indem ihre bedingte Wahrscheinlichkeitsverteilung basierend auf Strukturüberwachungsdaten berechnet wird. Der Ansatz ebnet zudem den Weg für physikalisch informierte Gaußprozesse. Zusätzlich wird eine Erweiterung für nicht-normalverteilter Daten vorgestellt.

Der Ansatz wird auf mehrere Ermüdungsbeispiele, wie z.B. Rissfortschritt in Luftfahrtstrukturen, Laserdegradation und Fräsmaschinenverschleiß, angewendet. Die neuartige Methode wird mit dem Stand der Technik und anderen Modellen des maschinellen Lernens verglichen. Schließlich wird der gesamte Ansatz auf eine reale Luftfahrtstruktur angewendet, um zukünftige Ausfallwahrscheinlichkeiten vorherzusagen. Die Ergebnisse zeigen, dass der vorgestellte Ansatz (1) den Stand der Technik für Prognoseanwendungen übertrifft, (2) zu ähnlichen Prognosegüten wie rekurrente neuronale Netze führt, während die neuartige Methode den gesamten zukünftigen Ermüdungsverlauf prognostiziert, und (3) der Ansatz dabei nur einen Bruchteil der Trainingszeit benötigt.

Journal publications

Authors:	**S. Pfingstl**, M. Steiner, O. Tusch, and M. Zimmermann
Title:	Crack detection zones: Computation and validation
Journal:	*Sensors* 20(9)
Year:	2020
Reference:	[106]
DOI:	10.3390/s20092568
Included in:	Section 3.3
Authors:	**S. Pfingstl** and M. Zimmermann
Title:	On integrating prior knowledge into Gaussian processes for prognostic health monitoring
Journal:	*Mechanical Systems and Signal Processing* 171
Year:	2022
Reference:	[105]
DOI:	10.1016/j.ymssp.2022.108917
Included in:	Section 4.1 and Sections 4.3–4.6
Authors:	**S. Pfingstl**, C. Braun, A. Nasrollahi, F.-K. Chang, and M. Zimmermann
Title:	Warped Gaussian processes for predicting the degradation of aerospace structures
Journal:	*Structural Health Monitoring* 0(0)
Year:	2022
Reference:	[108]
DOI:	10.1177/14759217221127252
Included in:	Sections 6.1–6.3 and Sections 6.5–6.6
Authors:	**S. Pfingstl***, D. Steinweg*, M. Zimmermann, and M. Hornung
Title:	On the potential of extending aircraft service time using load monitoring
Journal:	*Journal of Aircraft* 59(2)
Year:	2022
Reference:	[110]
DOI:	10.2514/1.C036569
Comment:	*shared first authorship
Authors:	J. Trauer*, **S. Pfingstl***, M. Finsterer, and M. Zimmermann
Title:	Improving production efficiency with a digital twin based on anomaly detection
Journal:	*Sustainability* 13(18)
Year:	2021
Reference:	[142]
DOI:	10.3390/su131810155
Comment:	*shared first authorship

Conference publications

Authors:	**S. Pfingstl** and M. Zimmermann
Title:	Strain-based structural health monitoring: Computing regions for critical crack detection
Conference:	*12th International Workshop on Structural Health Monitoring*
Year:	2019
Reference:	[104]
Authors:	**S. Pfingstl**, Y. Schoebel, and M. Zimmermann
Title:	Reinforcement learning for structural health monitoring based on inspection data
Conference:	*8th Asia-Pacific Workshop on Structural Health Monitoring*
Year:	2020
Reference:	[107]
Comment:	Conference canceled due to Covid-19. Proceedings only.
Authors:	**S. Pfingstl**, C. Braun, and M. Zimmermann
Title:	Warped Gaussian processes for prognostic health monitoring
Conference:	*13th International Workshop on Structural Health Monitoring*
Year:	2022
Reference:	[109]
Comment:	Conference canceled due to Covid-19. Proceedings only.
Authors:	**S. Pfingstl**, O. Tusch, M. Zimmermann
Title:	Comparison of error measures and machine learning methods for strain-based structural health monitoring
Conference:	*13th International Workshop on Structural Health Monitoring*
Year:	2022
Reference:	[111]
Comment:	Conference canceled due to Covid-19. Proceedings only.
Authors:	M. Franz, **S. Pfingstl**, M. Zimmermann, and S. Wartzack
Title:	Estimation of composite laminate ply angles using an inverse Bayesian approach based on surrogate models
Conference:	*Proceedings of the Design Society: DESIGN Conference*
Year:	2022
Reference:	[35]

Acknowledgments

It's been a ride! While looking back, I realize that there were a lot of ups and downs during my time as a Ph.D. student. I eventually made it through all the challenging and even devastating periods while also experiencing great times with even better people celebrating our victories, pushing us to our boundaries, and enjoying our time together. There is no doubt that I wouldn't have finished my thesis if there hadn't been amazing people by my side supporting and challenging me and cheering me up during times prevailed by concerns. Therefore, the time has finally come to say "thank you"!

One significant person from my time as a Ph.D. student is certainly Prof. Dr. Markus Zimmermann. Markus, thank you for accepting me as a Ph.D. student even though I did my undergraduate studies at a university of applied sciences. It meant a lot to me, and I tried my very best to fulfill the trust you placed in me. Your attitude of striving for the highest possible standards, constantly pushing your own and others' boundaries, and never stopping enhancing your profound knowledge has been an example to me. It might have been a coincidence but asking me in our first meeting, "What are the odds that a family with two kids has two girls if you know that they have at least one girl?" and whether my guess would change if I knew that the girl was born on a Tuesday, definitely paved my way to conditional probabilities. Thank you for your patience and guidance and for challenging me with plenty of questions!

I also owe deep gratitude to my second advisor Prof. Dr. Fu-Kuo Chang. Thank you, Prof. Chang, for hosting me at Stanford and giving me access to your lab facilities. During my time there, I gained valuable hands-on experience, allowing me to validate my theoretical methods. Your strong focus on applicability and real-world problem-solving has taught me that it is not enough to deepen my theoretical knowledge, but that I should also impact engineering by applying our novel methods to actual structures. It's been a fantastic time working with you and your students and utilizing my theoretical background for the experiments conducted at your lab.

Moreover, I would like to thank my third examiner Prof. Dr. Fabian Duddeck for his valuable feedback on my thesis and Prof. Dr. Wolfram Volk for chairing my Ph.D. committee.

There have been many other people who significantly contributed to my thesis. One of them is Dr. Mario Teixeira Parente. I would like to thank you, Mario, especially for your willingness to advise me and the countless hours you so patiently explained mathematical formulations and notations to me. Furthermore, I thank the entire team of the Laboratory of Product Development and Lightweight Design (LPL). In particular, I am appreciative of Lukas Krischer, Sebastian Rötzer, Julian Stumpf, Jasper Rieser, and Jakob Trauer for the great discussions of theoretical, practical, and philosophical nature. Without screaming, dancing, arguing, singing, explaining, and joking with y'all, a big part would be missing at LPL. You guys made my time at the lab an unforgettable one! I also owe Dr. Markus Kolb a big "thank you". As my Ph.D. mentor, your guidance in organizing my thesis was invaluable. Our highly intellectual meetings were indispensable, especially the first one at Schneider Bräuhaus, between deep AI discussions and analyzing different kinds of wheat beers. Moreover, I thank Dr. Martin Mahl for being gracious during my 2.5-hour interview before starting at LPL and for helping me out in the first few months of my research project. I also would like to thank my students, especially Martin Steiner, Jose Ignacio Rios, Christian Braun, and Moritz Fundel, for their great contributions and their patience with me as their supervisor.

Thanks to the entire Structures and Composites Laboratory team at Stanford University,

particularly to Dr. Amir Nasrollahi, Anthony Bombik, and Jeffrey Nolte, for the great discussions and making my stay a pleasant one. Thank you, Jeff, for proofreading my entire thesis. Your corrections improved the thesis significantly.

I am also surrounded by amazing people outside of work. I am so grateful to my parents, who have supported me my entire life. You always loved me unconditionally, even though I often followed my own ways and dreams. I am deeply impressed by your way of living, conveying virtues of diligence, trust in God, and humility. Furthermore, my sister Jana, her husband Andi, and my two little nephews, Timmy and Leon, are inspiring to me. They have taught me to find joy in the smallest incidences, living a life full of gratitude and charity. My deep appreciation also belongs to my sweetheart, Teresa. Even in turbulent times, you have been solid as a rock, keeping me grounded in glorious moments and cheering me up in devastating times.

Last and foremost, I thank God for the awesome adventure in which He has put me. Even though I worked really hard and put a lot of effort into this thesis, I know that *"it does not depend on human desire or effort but solely on God's mercy"*, as Paul writes in Romans 9:16. I could have never dreamt of being surrounded by so many dynamic people who help, comfort, and guide me. Thank you, God, for Your unending mercy. *Deus satis est!*

Simon Pfingstl

Munich, Germany
April 2023

Contents

Chapter 1

Introduction

"I never think of the future. It comes soon enough."

– Albert Einstein

Even though Albert Einstein stated that he never thinks of the future as it comes soon enough, estimating future conditions has never been more important than now in many emerging fields. This thesis is particularly devoted to Gaussian processes for prognostics — predicting the degradation of mechanical systems. Why this is beneficial, what topics still need to be explored, and which contributions are made throughout the thesis are addressed in this chapter. We start this chapter by motivating the use of prognostics and health management for aerospace structures. Moreover, we introduce prognostics and health management and explain frequently used terms in context. This is followed by a section that describes the current problems that come with applying Gaussian processes for prognostics and outlines the contributions of this work. Lastly, the structure of the presented studies is summarized.

1.1 Motivation of prognostics for aircraft

Severe fatigue problems should be avoided to prevent grounding the fleet, which implies loss of flying hours and poor publicity [126]. Yet, unexpected fatigue events have occurred repeatedly. According to Schijve [125], several aircraft fatigue accidents have led to rethinking aircraft design and understanding fatigue mechanisms better. In 1954, for example, two Comet 1 aircraft crashed after only 1290 and 900 pressurized flights [127] due to an unstable crack extension that caused an explosive fuselage decompression. It was concluded that the fuselage skin material should be resistant to fatigue crack growth, and local stress concentrations at cutouts should be avoided. Moreover, it became clear that a full-scale fatigue test should never be preloaded to a high level of stress before the intended fatigue life had been proven. Unfortunately, this was done on the front fuselage specimen to show the static strength capability of the Comet 1. In addition to some thirty earlier load cycles between one and two times the working pressure, the subsequent fatigue test revealed initial cracks after about 18,000 simulated flights [127] — after more than 13 times the service life. Today, we know that high preloads may cause local plastic deformation, introducing favorable compressive residual stresses [125].

Multiple other accidents led to a better understanding of the fatigue behavior in aerospace structures. During this learning process, one crucial mind shift was that aerospace structures

must be damage tolerant. Therefore, a structure must tolerate damages such as cracks, i.e., these damages must not lead to catastrophic failure. All damage tolerance concepts have to prove that the occurrence of any damage (fatigue initiated, foreign object induced, etc.) will not impair structural integrity [4]. On the one hand, this, together with detailed inspection and maintenance plans, leads to a safe aircraft operation. On the other hand, in the view of economic profit, aircraft of airlines should be in the air as much as possible [126], while maintenance is seen as a significant factor in aircraft operating costs. The average direct maintenance costs of a Boeing 757-200, for example, had a proportion of about 23% of the total flight operating costs in 2017 [1]. This is why current aircraft operation trends show an increasing demand for lower maintenance costs. This can be obtained by longer inspection intervals and shorter inspection downtimes [4].

Recent incidents have shown that fatigue is still an urgent challenge in the aviation industry and leads to unexpected events. In late 2019, for example, Boeing notified the Federal Aviation Administration of structural cracks in the pickle fork on a small number of their 737 Next Generation airplanes. Even though this structural part was designed to withstand more than 90,000 flights, the cracks occurred on some planes before 30,000 flights [47, 51, 87]. Moreover, after an uncontained engine failure on February 20, 2021, the U.S. airplane manufacturer recommended grounding all 777-model aircraft powered by a particular Pratt & Whitney engine model [26, 73]. If reliable predictions can solve fatigue problems, a significant amount of time and money can be saved [126]. Indeed, one way is to improve fatigue and fracture computations. However, since fatigue-induced damage is still one of the most uncertain failures in structures [43], it is challenging to compute the fatigue life precisely, which typically leads to a large margin of the computed fatigue life. Another way is to monitor mechanical systems with data gathered by attached sensors. This enables assessing mechanical parts' conditions during operation and avoids time-consuming inspections that involve disaggregating assemblies. One step beyond monitoring mechanical systems is to additionally predict the degradation and time to failure, which is referred to as prognostics [46]. If the system's degradation and the time to failure are known, its maintenance scheme can be adapted to be more favorable for airlines. Additionally, actions can be taken to diminish adverse consequences caused by a system failure, or the system's operation can be corrected to prolong its lifetime [44]. Even though it is currently common practice in the aviation industry to repair a crack once it is detected, in the future, this might be too expensive when having a fully operating monitoring system. Predicting the degradation of structures, e.g., the crack growth, makes it possible to wait until either multiple parts have to be repaired or structural safety cannot be assured anymore.

1.2 Prognostics and health management

In the literature, authors use many different terms in the context of monitoring mechanical systems. Various aspects such as conditioning monitoring, structural health monitoring, conditioned-based maintenance, predictive maintenance, and prognostics and health management are used interchangeably, confusing readers. Therefore, we give an overview of those terms by classifying them into three categories: monitoring, maintenance, and management.

Monitoring. According to [158], monitoring is concerned with five tasks, damage (1) detection, (2) localization, (3) classification, (4) assessment, and (5) prediction, where tasks 2–4 can

be regarded as diagnostics and the last one as prognostics [132]. Two major monitoring terms, namely structural health monitoring and condition monitoring, can be found in the literature. Both determine the internal condition of mechanical systems by analyzing data from attached sensors, see, e.g., [24] and [139]. Even though the terms are used ambiguously in literature [36], it seems to be that condition monitoring predominantly refers to the monitoring of rotating and reciprocating machinery. By contrast, structural health monitoring is mainly used for structures, such as bridges, buildings, or aerospace structures. Prognostics is often considered the last stage of structural health monitoring and condition monitoring and is thus a part of them [24, 158].

Maintenance. In contrast to monitoring, the term maintenance is used for all technical and managerial actions taken during the usage period to maintain or restore the required functionality of a product [129]. As every mechanical system is subject to loads during operation and deteriorates over time, maintenance should be exercised to ensure the system's reliability [65]. In order to retrieve maximum operating time between shutdowns, machines were originally "run to break" [115]. Nowadays, the so-called corrective maintenance policy is still used for large numbers of small machines, such as sewing machines, where catastrophic failure is unlikely, and the loss of one device is not critical to production. However, as corrective maintenance occasionally entailed catastrophic breakdowns with serious consequences for safety, production loss, and repair cost [115], the first response to it was time-based preventive maintenance, i.e., maintaining systems after specific intervals such that there is a small likelihood of failures between repairs. Preventive maintenance, however, leads to greater use of spare parts as well as more maintenance work than necessary [115] and thus results in high prevention costs. A new concept called condition-based maintenance was developed to minimize maintenance costs while preserving the desired level of reliability and safety [65]. It is sometimes further divided into proactive and predictive maintenance. Proactive maintenance focuses on determining the root causes of machine failure and dealing with those issues before significant problems occur, whereas predictive maintenance has its focus on failure prediction [29].

Management. After being successfully able to detect and predict the degradation of mechanical systems and applying a maintenance policy for them, one can focus on managing tasks such as adjusting and optimizing maintenance schedules and logistic supports [65] or changing envisaged operations [44]. The frequently used term "prognostics and health management" embraces all of the previously described tasks within the categories monitoring, maintenance, and management. Figure 1.1 shows the composition of them. As prognostics is seen as the key enabler of prognostics and health management [65], this dissertation is especially dedicated to prognostics, i.e., predicting the degradation of mechanical systems.

There are many fields where prognostics and health management have been applied, such as detecting and predicting damages in bearings, shafts, and gears. Prognostics and health management is particularly useful when catastrophic failures of these components lead to great downtimes associated with high economic losses [10]. Therefore, the concept has also been applied to civil engineering structures, such as bridges, buildings, underground tunnels, and aerospace structures. Gathering data from sensors attached to a structure opens the possibility to use machine learning models for prognostics and health management. For example, k-nearest neighbors, support vector machines, neural networks, and Gaussian process regression

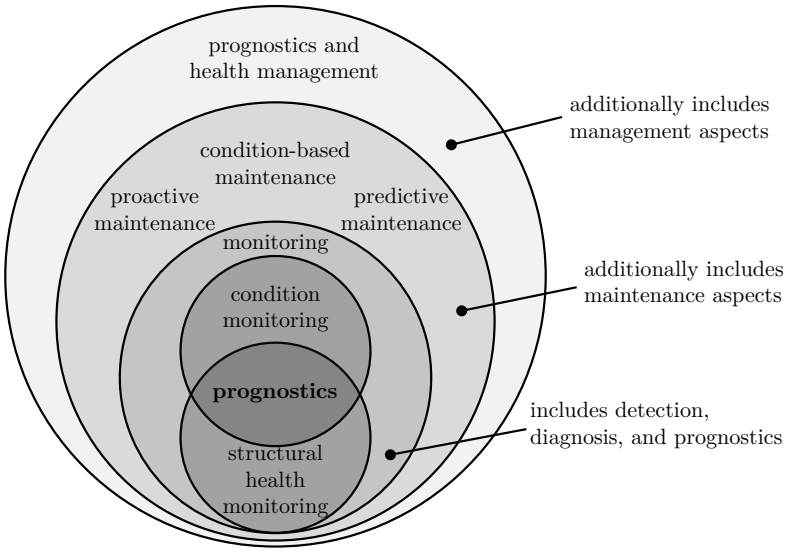


Figure 1.1: Composition of prognostics and health management

are four machine learning methods applied intensively to mechanical system prognostics and health management tasks in the last decade. As fatigue and fracture are subject to significant uncertainties, it is crucial to model and predict uncertainties associated with the problem. In contrast to k-nearest neighbors, support vector machines, and neural networks, Gaussian processes provide estimates about the output's uncertainty by default. Therefore, this work particularly focuses on Gaussian processes.

1.3 Problem statement and contributions

As previously shown, prognostics and health management is a promising field, also in the aviation industry. The main goal of this work is thus to establish a machine learning method for predicting the degradation of mechanical systems, particularly the crack growth in aerospace structures. Since fatigue and fracture scatters greatly, we also want to estimate the predictions' uncertainties. This allows us to predict future failure probabilities, which can be used to better schedule maintenance tasks. Another objective of this work is to create a method that leads to high accuracy. We not only want to predict the degradation precisely but also estimate valid credible regions. This means that if we, for example, predict a 95% credible interval, the data should, in fact, lie in 95% of the cases within this interval. The last goal of this thesis is that the established model is quickly trained in order to reduce computational effort. In summary, the objectives of this work are to

(1) predict the entire degradation trajectory of mechanical systems with valid credible regions,

(2) establish a prediction method with high accuracy, and

(3) enable quick training.

Gaussian process regression has shown to be a powerful tool for nonlinear regression problems that generalizes small data sets well [151]. It sometimes even outperforms other models, such as neural networks [55], and provides estimates of the output's uncertainty by default.

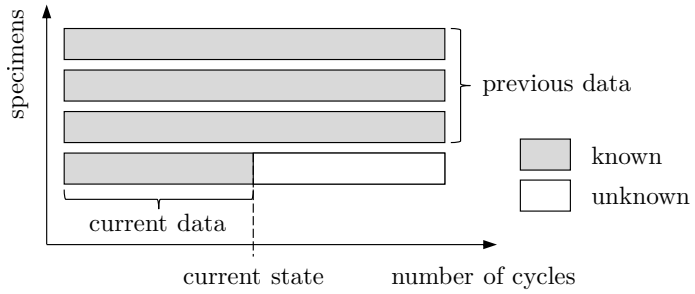


Figure 1.2: Definition of *previous* and *current* data.

Therefore, many researchers have been using them to predict future events in different application fields. For example, in [33, 94] the authors try to predict future stock prices and in [2, 19, 156] future CO_2 concentration in the air. The problem is that stock markets and the earth's environment are large systems with many influencing factors. In order to predict their future behaviors, one has to account for all these different factors. Predictions solely based on past observations might not be better than guessing since the influencing factors vary over time and are not the same as the condition for the recently observed data.

Yet, applying Gaussian processes to predict the degradation of mechanical systems seems to be promising. In contrast to predicting stocks or CO_2 concentrations, the system boundaries for mechanical experiments can often be precisely drawn, allowing us to control or be aware of different influencing factors regarding the experiment. Therefore, Gaussian processes have also been used for predicting the degradation of mechanical systems such as crack growth [5], fatigue of solder joints [71], and degrading bearings [52, 57]. Even though the underlying model can significantly influence prediction accuracy, Gaussian processes are typically used with default models that incorporate strong assumptions about the data's structure. The assumptions implicitly made by choosing a particular Gaussian process model are often incorrect and thus lead to low prediction accuracy and a short look-ahead time. As stated in [57, 63, 71, 112, 123], prior knowledge regarding the underlying model should improve the prediction accuracy of Gaussian processes. Therefore, the present work is particularly concerned with how prior knowledge can be integrated into Gaussian processes and whether it improves prediction accuracy.

One way of integrating prior knowledge is to incorporate data from previously executed experiments into a Gaussian process model. In this work, we thus distinguish between *previous* and *current* data, see Figure 1.2. The first-mentioned represents all data collected by previously conducted simulations or experiments of a similar mechanical system. Previous data comprises several trajectories, each belonging to one simulation or experiment executed until a specified end. Different trajectories can be achieved by varying uncertain parameters, such as material parameters and loads. In contrast to previous data, current data is collected from the actual monitored system with fixed, realized parameters that are generally unknown. Current data comprises the data points collected until the current state of the mechanical system.

In this thesis, we present an approach that infers a Gaussian process model from previous data and thus integrates prior knowledge into the model. As physical equations are often understood for mechanical systems, the work also addresses incorporating physical equations into Gaussian processes. The novel approach is compared to state-of-the-art Gaussian processes

as well as to three machine learning methods, namely k-nearest neighbors, support vector machines, and recurrent neural networks. Since observations of many engineering applications do not follow a normal distribution, the approach presented in this work is extended to handle non-normally distributed data by warping the Gaussian process. The different levels of the novel approach are applied to various problems such as crack growth, laser degradation, and milling machine wearing. The method is finally used to predict the crack growth in a real aerospace structure. The main contributions of this work are as follows:

(1) Inferring a Gaussian process model from previous data

(2) Incorporating physical equations into Gaussian processes

(3) Comparing the novel approach to other machine learning methods

(4) Warping Gaussian processes with integrated prior knowledge

1.4 Structure of the thesis

After this introductory chapter, the fundamentals of fatigue and fracture are introduced. The basics of computing the fatigue life of structures and how to consider cracks and their propagation in structures are explained. A short introduction to computing cracks with the finite element method (FEM) is also given. Furthermore, the normal distribution is introduced, and its generalization, the Gaussian process, is derived from it. The third chapter outlines an overview of the different problem types encountered in this thesis. Different data sets are described and linked to the, in this thesis, tackled challenges. The novel approach, which integrates prior knowledge into Gaussian processes, is described in Chapter 4. First, the underlying concept is explained and applied to a simple crack growth problem. Then, the approach is applied to three different data sets and compared to the state-of-the-art Gaussian process method. Third, the approach is extended by incorporating physical equations into Gaussian processes and applied to predict crack growth in an aluminum specimen. Chapter 5 introduces three machine learning models: k-nearest neighbors, support vector machines, and recurrent neural networks. The models are trained on three different data sets and compared to the approach proposed in Chapter 4. This is followed by Chapter 6, which introduces warped Gaussian processes and how to integrate prior knowledge into them. The extended approach, which is able to handle data that is not normally distributed, is applied to a set of simulated crack growth trajectories and compared to the approach without warping the data. Finally, the extended approach is used to predict the crack growth in a real aerospace structure. Lastly, Chapter 7 summarizes the outcomes of the research work and discusses them. The thesis ends with the conclusions drawn from the studies and an outlook of promising venues.

Chapter 2

Fundamentals

"The universe doesn't allow perfection."

– Stephen Hawking

Before diving into Gaussian processes towards the end of this chapter, it is essential to know the basic concepts of fatigue and fracture mechanics. Much research has already led to a good understanding of the fatigue behavior in structures. The involved computational methods help us to clarify our prior knowledge of structures' degradation and fatigue life. The focus of this chapter is to explain the fundamentals of fatigue and fracture mechanics as well as to introduce the, for this work, important implementations of fracture mechanics in the finite element analysis (FEA). As — the universe does not allow perfection — we will see that fatigue is subject to large uncertainty. Since the present work deals with them by using the normal distribution, an introduction of it and its generalization, the Gaussian process, are given afterwards.

2.1 Fatigue of structures

As early as 1858, August Wöhler found that steel can rupture even at stresses less than the elastic limit if they are sufficiently repeated. The reason for such a behavior is the repetitive, cyclic loading. Figure 2.1 shows an example for one cycle with its minimum, maximum, mean, and amplitude stress S_{min}, S_{max}, S_m, and S_a. Often cyclic loads are also defined by their

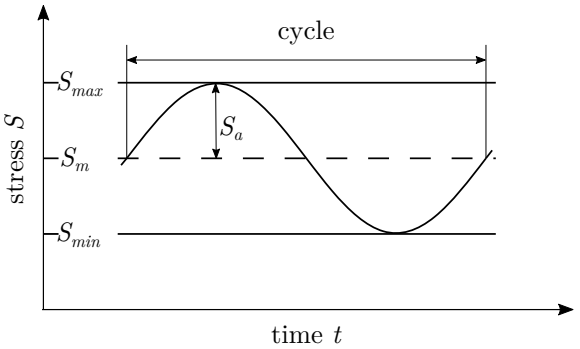


Figure 2.1: One cycle with its minimum, maximum, mean, and amplitude stress.

stress ratio

$$R = \frac{S_{min}}{S_{max}}. \tag{2.1}$$

These repetitive loads can cause accumulations of dislocations which lead to local stress concentrations. The high stresses may dissolve bonds along preferred lattice planes and thus result in cleavage [49]. Since the first discoveries of Wöhler, many researchers have established methods for predicting the structure's time to failure for a given cyclic loading program. The fundamentals of them are briefly explained in the following.

S-N curves. The S-N curve is an important tool for predicting fatigue life. As shown in Figure 2.2, it relates the applied nominal amplitude stress S_a to the number of cycles until fracture N. Therefore, the S-N curve quantifies how many cycles with a certain nominal amplitude stress are needed to fracture a specimen. An S-N curve can also be used to compute the fraction of the damage

$$D_i = \frac{n_i}{N_i} \tag{2.2}$$

for a given number of cycles n_i at the constant nominal stress amplitude $S_{a,i}$ and its corresponding number of cycles until fracture N_i.

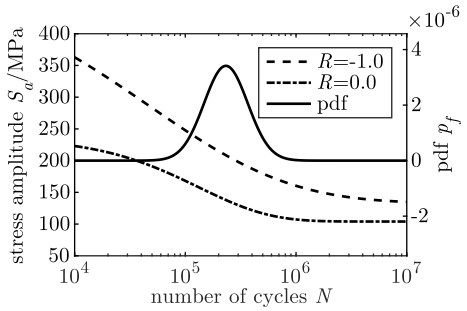


Figure 2.2: S-N curves and its probability density function (pdf) for an unnotched aluminum specimen.

S-N curves are generated by fatigue experiments where usually coupon specimens are cyclically loaded until fracture. The fatigue life is investigated depending on the geometry, loading type, and stress level. The geometry (shape and size) of a notch can be considered by the fatigue notch factor K_f, which is defined as the ratio of the fatigue limit of a smooth specimen to the fatigue limit of a notched one. It can be computed for a tension-compression loaded structure by

$$K_f = \frac{K_t}{n_\sigma}, \tag{2.3}$$

where K_t is the stress concentration factor and n_σ the support factor based on Stieler [130]. The support factor based on Stieler considers that only small regions are highly stressed for notches with large stress gradients, which has a positive effect on fatigue life. It can be estimated based on the stress gradient and corresponding material parameters, see p. 42 in [117]. The stress concentration factor K_t is defined by

$$K_t = \frac{\sigma_{max}}{S}, \tag{2.4}$$

where σ_{max} denotes the maximum local stress at the corresponding spot and S the nominal stress. There exist multiple definitions for the nominal stress. The FKM guideline [117] defines it for a uniaxial case by

$$S = \frac{F}{A_{net}}, \tag{2.5}$$

where F is the applied force and A_{net} is the net cross-section at the corresponding spot. The net cross-section for a plate with a centered hole, for example, is the difference between the cross-section of the plate and the hole's diameter times the thickness. Problems arise with this definition if structures become more complex and the net cross-section is not properly defined anymore. A simplified and conservative approach for computing the fatigue life of a complex structure is to compare the maximum principal stress in a notch with an S-N curve of $K_t = 1$ (no notch) as the stress gradient is neglected. For further details and more advanced methods for dealing with notched structures or multiaxial stress conditions see [117].

As fatigue life is subject to large uncertainty, a 50% S-N curve is usually depicted. Therefore, the curve denotes the number of cycles until 50% of all specimens are failed. Assuming a log_{10} distribution and the experience-based scatter parameter s, which can be found on p. 527 in [50] for different materials, the uncertainty of the time to failure can be estimated. Figure 2.2 shows the fatigue life distribution of an unnotched aluminum specimen for the constant amplitude stress $S_a = 200\,\text{MPa}$. The graph reveals the great uncertainty ($N_{P_f=2.5\%} \approx 117 \times 10^3$, $N_{P_f=97.5\%} \approx 694 \times 10^3$) associated with fatigue. Therefore, even if we know the material and load precisely, we can only roughly estimate when a specimen will fail. As an exact prediction is not possible, prognostics and health management become worthwhile.

Rainflow-counting. Typically structures are not loaded under a constant amplitude over their entire lives but rather experience a variety of different stress amplitudes. Rainflow-counting is a method to count similar cycles within a loading program such that it transforms a time-based loading program into a matrix. The method is based on counting closed hystereses that start from a starting level, return at the target level, and end up in the starting level again. The numbers in the rainflow matrix describe the occurrences of closed hystereses. The

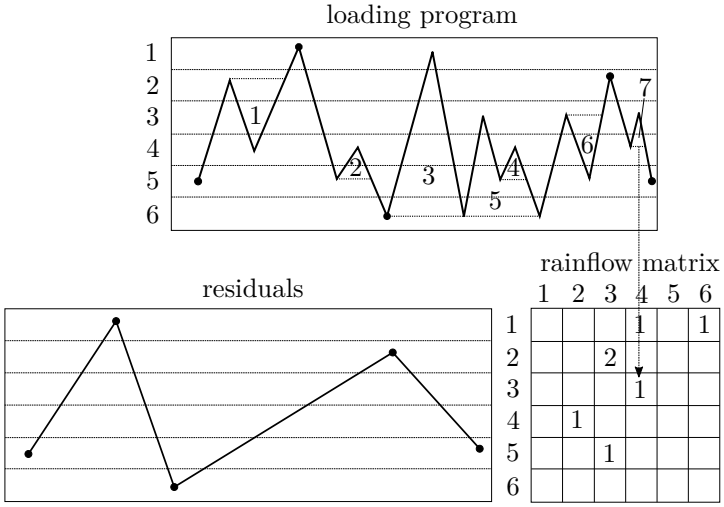


Figure 2.3: Concept of rainflow-counting, see [50].

counting method can be explained by imagining a raindrop floating down the 90° turned loading program. Whenever it hits another branch, the hysteresis loop is closed, i.e., the raindrops of this cycle are stopped, and a new raindrop is started afterwards. The closed hystereses in Figure 2.3 are indicated by numbered triangles. For each closed hysteresis, one cycle is added to the rainflow matrix at the corresponding row and column. The row and column numbers are the position of the starting and turning level, respectively. The rainflow matrix, therefore, relates the number of similar cycles to its corresponding minimum and maximum stress. The method also allows us to distinguish between hanging (triangle 1 and 6) and standing (triangle 2, 3, 4, 5, and 7) hystereses which lay left and right to the diagonal of the rainflow matrix, respectively (see Figure 2.3). However, as there is often no different computation scheme for these types, only the upper right corner of the rainflow matrix is filled out. By erasing the closed hystereses in Figure 2.3, the residuals are visible. There are different possibilities to deal with them. The ASTM standard on p. 287–293 in [8], for example, adds half a cycle for each residual to the rainflow matrix. After applying the rainflow-counting method to a time-based loading program, the cycle frequencies and corresponding stress levels are quantified.

Haigh diagram. S-N curves are usually generated with a constant stress ratio or mean stress. As loading programs can consist of cycles with various stress ratios, multiple S-N curves would be needed to compute the corresponding damages. A Haigh diagram as shown in Figure 2.4 can be used to map cycles with different stress ratios to a constant one. The resulting damage

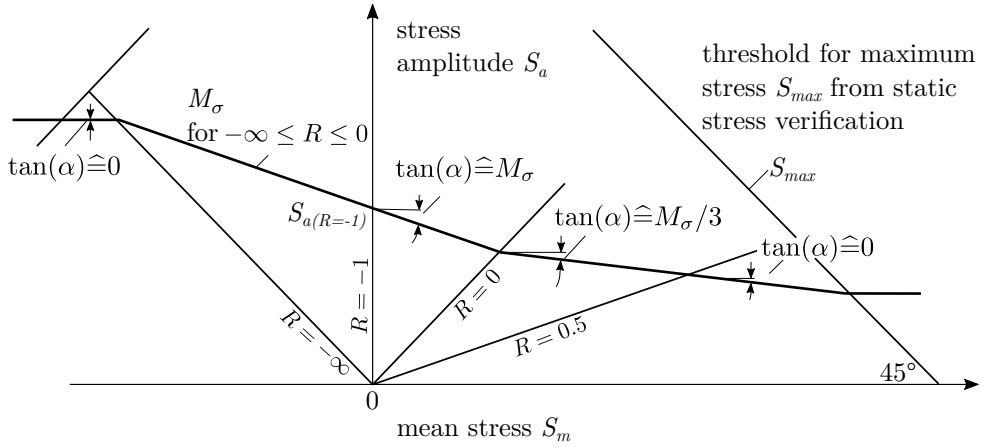


Figure 2.4: Haigh diagram, see [50].

equivalent stress amplitudes for usually $R = -1$ ($S_{a,R=-1}$) can then be compared to a single S-N curve. An equation-based transformation is given on p. 98 in [117], where the transformation is dependent on the mean stress sensitivity M_σ and can be computed as

$$\text{for } R > 1: \quad S_{a,R=-1} = S_a(1 - M_\sigma) \tag{2.6}$$

$$\text{for } -\infty \leq R \leq 0: \quad S_{a,R=-1} = S_a(1 + M_\sigma \tfrac{S_m}{S_a}) \tag{2.7}$$

$$\text{for } 0 < R < 0.5: \quad S_{a,R=-1} = S_a \tfrac{(1+M_\sigma)(3+M_\sigma \frac{S_m}{S_a})}{3+M_\sigma} \tag{2.8}$$

$$\begin{gathered}\text{for } R > 0.5\text{:} \\ S_{a,R=-1} = S_a \frac{3(1+M_\sigma)^2}{3+M_\sigma}\end{gathered} \tag{2.9}$$

for principal stresses. The mean stress sensitivity is a material parameter that can be found, for example, on p. 29 in [50].

Miner rule. When dealing with different stress amplitudes, a method for damage accumulation is needed to compute the total damage resulting from cycles with different stress amplitudes. The most popular and most often used method to quantify the fatigue life based on cycles with various amplitude stresses is the linear damage accumulation rule [50] based on Palmgren [100] and Miner [90]. The so-called Miner rule assumes that the damages of each cycle accumulate linearly to a total damage. All fractions of the damages D_i, that result from the number of loaded cycles n_i at the constant nominal stress amplitude $S_{a,i}$, are therefore summed up and lead to the total damage

$$D = \sum_i D_i = \sum_i \frac{n_i}{N_i}, \tag{2.10}$$

where N_i is the corresponding fatigue life at the nominal stress amplitude $S_{a,i}$. As displayed by Equation 2.10, the rule neglects the order of loads which is also lost by applying rainflow-counting to a time-based loading program. The critical damage value, where fracture is expected, is $D = 1$. There exist various versions of the Miner rule that consider the fatigue limit differently. The Miner original rule, for example, considers no damage for stresses below the fatigue limit, whereas the Miner elementary version assumes the slope of the S-N curve in a log-log diagram to be continued below the fatigue limit.

2.2 Fracture mechanics

Repetitive loads, as explained earlier, can cause accumulations of dislocations that eventually lead to cracks in structures. In contrast to the previously explained methods, fracture mechanics considers a slit or rather a crack in a body. Figure 2.5 shows the three distinguished deformation modes of a crack. As stated in [6], mode I is characterized by a principal load normal to the crack plane which tends to open the crack. An in-plane shear loading refers to mode II that slides one crack face with respect to the other, and mode III corresponds to an out-of-plane shear.

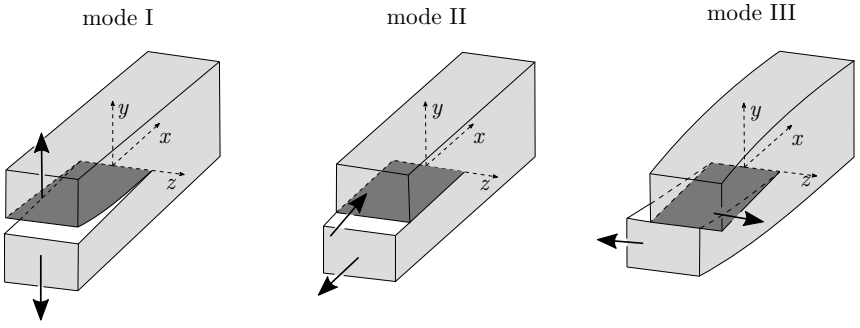


Figure 2.5: The three distinguished deformation modes of a crack, see [49].

A special area of fracture analysis is the linear-elastic fracture mechanics. It describes the behavior of cracks in bodies that are completely linear-elastic. Non-elastic mechanisms that might occur close to the crack tip must therefore be limited to a small region such that they can be neglected from a macroscopic perspective. This is the case in many cases, including typical metallic and brittle materials [49]. There are two approaches for fracture analysis: the energy criterion and the stress-intensity approach. They are briefly discussed in the following.

Griffith energy balance. In 1920, Griffith [48] applied the first law of thermodynamics to the formation of a crack. He postulated that a crack could form or propagate only if the total energy decreases or remain constant. The critical condition can be therefore defined as the point at which crack growth occurs without changing the total energy. The total energy E is composed of the potential energy Π and the work required to create new surfaces W_s. The potential energy is comprised of the internal strain energy and the work done by external forces. Under equilibrium conditions, the Griffith energy balance for an incremental increase of the crack area dA becomes

$$\frac{dE}{dA} = \frac{d\Pi}{dA} + \frac{dW_s}{dA} = 0 \tag{2.11}$$

$$-\frac{d\Pi}{dA} = \frac{dW_s}{dA}. \tag{2.12}$$

For an infinite plate with the thickness t, Young's modulus E, and a crack which lays perpendicular to the remotely applied stress σ_∞ with a length of $2a$, the potential energy is according to [6]

$$\Pi = \Pi_0 - \frac{\pi\sigma_\infty^2 a^2 t}{E}, \tag{2.13}$$

where Π_0 is the potential energy of the uncracked plate. Therefore, its derivative becomes

$$-\frac{d\Pi}{dA} = \frac{\pi\sigma_\infty^2 a}{E}. \tag{2.14}$$

Since the formation of a crack requires creating two surfaces, the work becomes $W_s = 4at\gamma_s$ with its derivative $\frac{dW_s}{dA} = 2\gamma_s$, where γ_s is the material's surface energy [6]. Equating both derivatives and solving for the remote stress leads to the fracture stress

$$\sigma_{\infty,f} = \left(\frac{2E\gamma_s}{\pi a}\right)^{1/2}. \tag{2.15}$$

Since this equation is valid only for ideally brittle solids such as glass, the expression can be generalized by

$$\sigma_{\infty,f} = \left(\frac{2Ew_f}{\pi a}\right)^{1/2}, \tag{2.16}$$

where w_f is the fracture energy that could include plastic, viscoelastic, or viscoplastic effects depending on the material. A closely related concept was proposed by Irwin in 1956 [59] where he defined the energy release rate G by

$$G = -\frac{d\Pi}{dA}, \tag{2.17}$$

which becomes for the previously investigated infinite plate

$$G = \frac{\pi \sigma_\infty^2 a}{E}, \tag{2.18}$$

as also stated in [6]. The energy release rate is a measure of the required energy for incremental crack growth. Fracture occurs when it reaches the fracture toughness G_c.

Stress intensity approach. A different approach to describe a crack in a body is based on the stress intensity factor (SIF) K_I. It fully characterizes the crack tip conditions in a linear-elastic material. There is an individual SIF for each fracture mode. For this work, it is sufficient to introduce the approach on mode I since only examples for this particular mode are examined. For the infinite plate considered previously, the SIF is

$$K_I = \sigma_\infty \sqrt{\pi a} \tag{2.19}$$

and the entire stress distribution at the crack tip can be computed by

$$\sigma_{xx} = \frac{K_I}{\sqrt{2\pi r}} \cos\left(\frac{\theta}{2}\right)\left(1 - \sin\left(\frac{\theta}{2}\right)\sin\left(\frac{3\theta}{2}\right)\right) \tag{2.20}$$

$$\sigma_{yy} = \frac{K_I}{\sqrt{2\pi r}} \cos\left(\frac{\theta}{2}\right)\left(1 + \sin\left(\frac{\theta}{2}\right)\sin\left(\frac{3\theta}{2}\right)\right) \tag{2.21}$$

$$\tau_{xy} = \frac{K_I}{\sqrt{2\pi r}} \cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\theta}{2}\right)\cos\left(\frac{3\theta}{2}\right), \tag{2.22}$$

where the coordinate system begins at the crack tip, and the position of the infinitely small element is described by the angle θ and distance r, see Figure 2.6. Note that all stress components

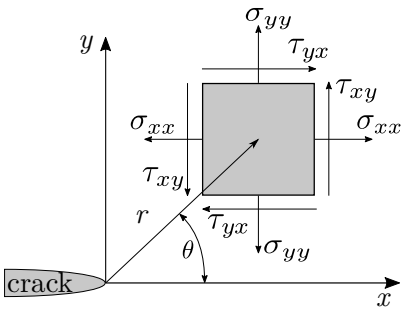


Figure 2.6: Coordinate system for SIF, see [6].

are proportional to $\frac{1}{\sqrt{r}}$. Therefore, the linear-elastic theory leads to infinitely high stresses for $r \to 0$. The Equations 2.20–2.22 for the stress components are only valid for stresses close to the crack tip. They can be derived from complex stress functions, see [49] (Chapter 1.5.2 and 4.2) for more details. To consider different geometries, a geometry factor γ is typically introduced to the SIF

$$K_I = \gamma(a)\sigma_\infty \sqrt{\pi a} \tag{2.23}$$

which is, in general, dependent on the crack length a. Several solutions for different configurations can be found in [70]. Failure occurs if $K_I = K_{I,c}$. Therefore, $K_{I,c}$ is another measure of the fracture toughness. As in linear-elastic mechanics, the principle of superposition can

be used in the linear-elastic fracture mechanics too. Therefore, complex configurations can be subdivided into simpler ones for which the SIF solutions are known. Applications of the so-called compounding method can be found in [23, 38, 39, 124] where the authors apply the method, for example, to a lap-joint configuration. Both approaches, the energy release rate and the SIF, are closely related by

$$G = \frac{K_I^2}{E'} \tag{2.24}$$

in the linear-elastic case, where $E' = E$ is valid for plane stress and $E' = \frac{E}{1-\nu^2}$ for plane strain.

Paris law. As stated by Gross and Seelig [49], a crack in a statically loaded structure will not grow if the crack length or the loading is below a certain value. However, when the structure is cyclically loaded, a crack will propagate even if loading is below a critical threshold. This is also called fatigue crack growth which is characterized by the crack growth rate $\frac{da}{dN}$. The reasons for such behavior are the inelastic processes in the crack tip zone. A metallic material, for example, experiences plastic hystereses in the crack tip zone due to tension and compression. The resulting changing residual stresses damage the material up to complete separation. The fatigue crack growth can be described with the SIF in the linear-elastic fracture mechanics. The cyclic loading corresponds to a cyclic SIF which is usually described by the SIF range ΔK, see Figure 2.7a. Measuring the crack growth rate $\frac{da}{dN}$ with respect to ΔK yields to the typical course shown in Figure 2.7b. A crack does not propagate below the threshold ΔK_0 and rapidly propagates if it reaches K_c. In between, i.e., region II, the crack growth rate can be approximated by a straight line with the slope α in a log-log diagram. Therefore, the crack growth rate can be modeled by

$$\frac{da}{dN} = C\Delta K^{\alpha}, \tag{2.25}$$

where C and α are material parameters. This is also called the Paris law [101]. Further extensions, such as the Foreman law [34] that considers the rising crack growth rate towards K_c exist in the literature. The Paris law, however, is one of the most frequently used ones.

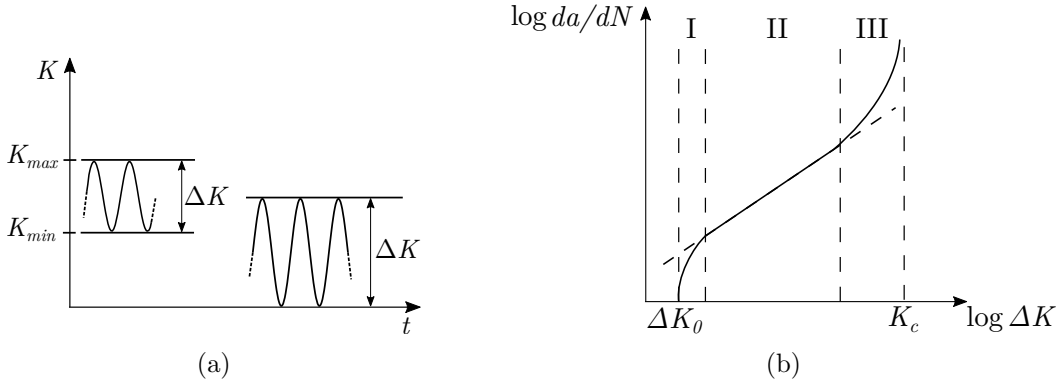


Figure 2.7: (a) SIF over time for cyclic loading and (b) Paris law, see [49].

Multiple crack growth experiments, such as those executed by Virkler et al. [148], reveal a huge scatter of the gathered crack growth trajectories, even though the loading and material are kept the same for all specimens. Consequently, fracture mechanics can often not be regarded as deterministic processes. They are rather probabilistic mechanisms whose uncertainties can be significant.

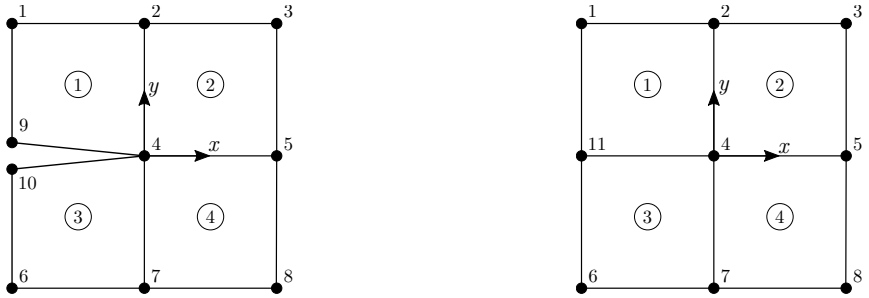


Figure 2.8: (a) FE model with crack and (b) its node replacement, see [92].

Extended finite element method. The SIF for arbitrary bodies and crack geometries can be determined by numerical methods such as FEM. FEM is based on dividing a body into a finite number of elements and on approximating the displacements of each element by displacement shape functions. A special type of FEM, which is used in this work, is the extended finite element method (XFEM). The advantage of XFEM in comparison with other methods is that cracks do not necessarily have to be in between elements but can also cut them. Its fundamentals were published by Belytschko and Black [11] and Moës et al. [92] in 1999 and are briefly introduced in the following. Figure 2.8a shows an FE model based on four elements with a crack in between element 1 and 3. The FE approximation for the displacement $\boldsymbol{u}^h = \left[u_x^h, u_y^h\right]^\top$ is

$$\boldsymbol{u}^h = \sum_{i=1}^{10} \boldsymbol{u}_i \phi_i, \tag{2.26}$$

where $\boldsymbol{u}_i$ is the displacement and ϕ_i the bilinear shape function at node i. By defining $\boldsymbol{a} = \frac{\boldsymbol{u}_9 + \boldsymbol{u}_{10}}{2}$ and $\boldsymbol{b} = \frac{\boldsymbol{u}_9 - \boldsymbol{u}_{10}}{2}$, and replacing $\boldsymbol{u}_9$ and $\boldsymbol{u}_{10}$ in Equation 2.26, the displacements can be expressed as

$$\boldsymbol{u}^h = \sum_{i=1}^{8} \boldsymbol{u}_i \phi_i + \boldsymbol{a}\left(\phi_9 + \phi_{10}\right) + \boldsymbol{b}\left(\phi_9 + \phi_{10}\right) H(\boldsymbol{x}), \tag{2.27}$$

where $H(\boldsymbol{x})$ is the jump function to consider the discontinuity of the crack. It is defined by

$$H(\boldsymbol{x}) = \begin{cases} +1, & \text{for } y > 0 \\ -1, & \text{for } y < 0\,. \end{cases} \tag{2.28}$$

If the nodes 9 and 10 are replaced by node 11, as shown in Figure 2.8b, $\phi_9 + \phi_{10}$ can be written as ϕ_{11} and $\boldsymbol{a}$ as $\boldsymbol{u}_{11}$ which leads to

$$\boldsymbol{u}^h = \sum_{i=1}^{8} \boldsymbol{u}_i \phi_i + \boldsymbol{u}_{11}\phi_{11} + \boldsymbol{b}\,\phi_{11} H(\boldsymbol{x}). \tag{2.29}$$

Equation 2.29 can be interpreted as the sum of the FE model without the crack $\boldsymbol{u}^h = \sum_{i=1}^{8} \boldsymbol{u}_i \phi_i + \boldsymbol{u}_{11}\phi_{11}$ and a discontinuous enrichment $\boldsymbol{b}\,\phi_{11} H(\boldsymbol{x})$.

The introduced concept can be transferred to a crack that does not align with the mesh. Every node is enriched if its support is cut by the crack into two disjoint pieces, see Figure 2.9a.

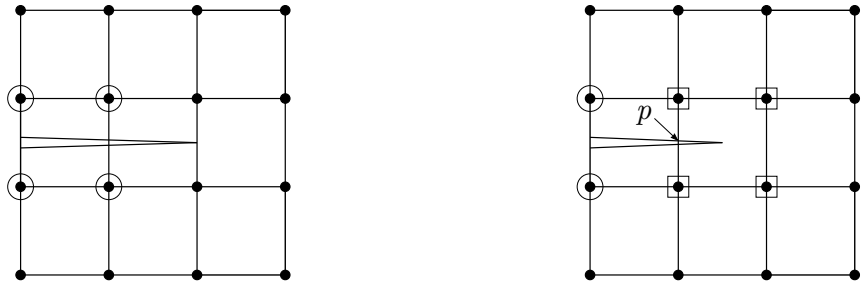


Figure 2.9: (a) FE model with a crack (a) aligned and (b) misaligned with the mesh, see [92].

This is in agreement with the previous example, in which only node 11 was enriched. If the crack tip does not coincide with an element edge, see Figure 2.9b, the discontinuity cannot be adequately described by using only the jump function $H(x)$. The enrichment of the circled nodes would consider the crack tip to be at point p. In order to seamlessly model the entire discontinuity along the crack, the squared nodes are enriched by the asymptotic crack tip functions $\boldsymbol{F}(r,\theta)$ by which a two-dimensional displacement field close to the crack tip can be described [6]. The approximation for an arbitrary crack with one crack tip becomes therefore

$$\boldsymbol{u}^h = \sum_{i\in I} \boldsymbol{u}_i \phi_i + \sum_{j\in J} \boldsymbol{b}_j \phi_j H(\boldsymbol{x}) + \sum_{k\in K} \phi_k \left(\sum_{l=1}^{4} \boldsymbol{c}_{k,l} F_l(\boldsymbol{x}) \right), \tag{2.30}$$

where J and K are the sets of the circled and squared nodes, respectively, and $\boldsymbol{c}_k^l$ are the enrichment coefficients associated with the crack tip functions

$$F_l(r,\theta) \equiv \left[\sqrt{r}\sin\left(\frac{\theta}{2}\right), \sqrt{r}\cos\left(\frac{\theta}{2}\right), \sqrt{r}\sin\left(\frac{\theta}{2}\right)\sin(\theta), \sqrt{r}\cos\left(\frac{\theta}{2}\right)\sin(\theta) \right], \tag{2.31}$$

where r and θ are the polar coordinates with the origin at the crack tip.

Virtual crack closure technique. The virtual crack closure technique is an efficient method for evaluating SIFs based on FEM. It leads also for coarse grids to good results and permits the evaluation of both mode I and mode II SIFs [120]. The technique is based on Irwin's contention that if a crack growths by a small amount Δa, the energy absorbed in the process is equal to the work required to close the crack to its original length [59]. For a polar coordinate system with its origin at the grown crack tip, the energy release rate can be formulated as

$$\begin{aligned} G =& G_I + G_{II} \\ =& \lim_{\Delta a\to 0} \frac{1}{2\Delta a} \int_0^{\Delta a} \sigma_{yy}(\Delta a – r, 0) v(r,\pi) dr \\ &+ \lim_{\Delta a\to 0} \frac{1}{2\Delta a} \int_0^{\Delta a} \tau_{xy}(\Delta a – r, 0) u(r,\pi) dr, \end{aligned} \tag{2.32}$$

where Δa is the crack growth, σ_{yy} and τ_{xy} are the stresses close to the crack tip, and u and v are the relative sliding and opening between the points on the crack faces. For the implementation

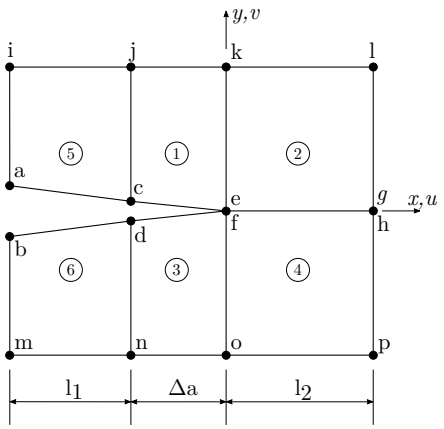


Figure 2.10: Virtual crack closure technique, see [120].

of this technique into FEM, where the displacements and forces at nodes can be computed, the energy release rates can be expressed as

$$G_I = \lim_{\Delta a \to 0} \frac{1}{2\Delta a} F_c(v_c - v_d) \tag{2.33}$$

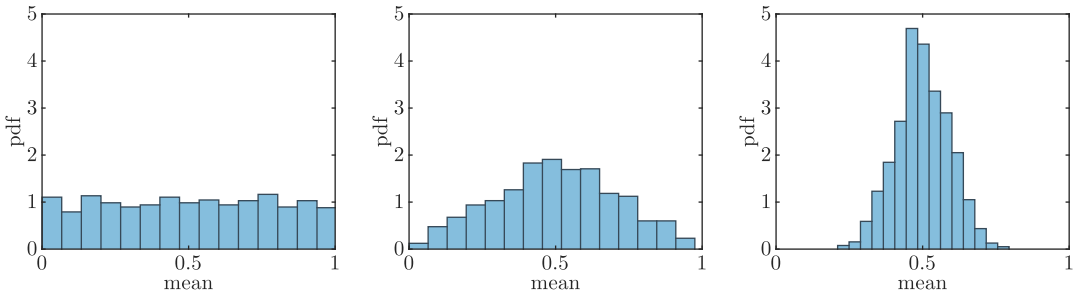
and

$$G_{II} = \lim_{\Delta a \to 0} \frac{1}{2\Delta a} T_c(u_c - u_d), \tag{2.34}$$

where T and F are the forces in x- and y-direction, respectively. The indices at the force and displacement denote the corresponding nodes. The equations represent the work required to close the crack by Δa, i.e., to hold the nodes c and d together, see Figure 2.10. In order to compute the energy release rates, the forces F_c and T_c must be determined. The forces F_e and T_e can be obtained by placing very stiff springs between nodes e and f and evaluating the forces in these springs. The forces F_c and T_c can then be approximated by $F_c = \sqrt{\frac{\Delta a}{l_2}} F_e$ and $T_c = \sqrt{\frac{\Delta a}{l_2}} T_e$. The introduction of this method describes the basic concept of the virtual crack closure technique. Further extensions and advances exist in the literature.

2.3 The normal distribution

The previous two sections indicate that fatigue and fracture are subject to uncertainty. Probability distributions are usually employed to model uncertainties. A widely used model for the distribution of continuous variables is the normal distribution which is also known as the Gaussian distribution. It plays a major role in many applications and can be motivated from different perspectives. One is the central limit theorem, which states that the sum of random variables increasingly follows a normal distribution as the number of terms in the sum increases too [149]. As many patterns — from fish to handwritten characters to some speech sounds — can be viewed as ideal patterns corrupted by many random processes, the normal distribution is often a good model for the probability distribution [16]. We can illustrate this behavior, as it is presented in [16], by computing the mean of n uniformly distributed random variables $x_1, \ldots, x_n$ over the interval $[0, 1]$ and plotting its histogram. Figure 2.11 shows this behavior for three different amounts of random variables, $n = 1$, $n = 2$, and $n = 10$. For each random variable, 1000 realizations are sampled.



(a) (b) (c)

Figure 2.11: Histogram of samples with (a) $n = 1$, (b) $n = 2$, and (c) $n = 10$, see also [16].

One-dimensional normal distribution. A single random variable X is normally distributed $X \sim \mathcal{N}(\mu, \sigma^2)$ if it follows the normal density distribution

$$p(x) = \mathcal{N}(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right). \tag{2.35}$$

The distribution is fully defined by its mean (or expected value)

$$\mu = \mathbb{E}[X] = \int_{-\infty}^{\infty} x\, p(x) dx \tag{2.36}$$

and variance (or expected squared deviation)

$$\sigma^2 = \mathbb{E}[(X - \mathbb{E}[X])^2] = \int_{-\infty}^{\infty} (x-\mu)^2 p(x) dx, \tag{2.37}$$

where σ is also known as the standard deviation. By conducting experiments, samples or rather realizations of the random variable can be drawn.

The mean and the variance can be inferred from realizations or observations of a random variable. If we have m observations $\boldsymbol{x} = [x_1, \ldots, x_m]^\top$ that are independently drawn from the same distribution, which is also denoted as independent and identically distributed (i.i.d.), the likelihood of all observations is the product of the likelihoods

$$p(\boldsymbol{x}|\mu, \sigma^2) = \prod_{j=1}^{m} \mathcal{N}(x_j|\mu, \sigma^2). \tag{2.38}$$

The term likelihood is used in this work to describe the probability density at a certain point x_j. A common criterion for determining the parameters of a probability distribution is to find the parameters that maximize the likelihood of all observations [16]. In practice, not the likelihood itself but rather the log of the likelihood is maximized. Since the logarithm is a monotonically increasing function, maximizing the log-likelihood leads to the same results as maximizing the likelihood. Using the logarithm not only simplifies the mathematical analysis but also reduces numerical precision errors. This is because the product of small likelihoods can easily underflow the computer's numerical precision, and the logarithm turns the product of likelihoods into a sum of log-likelihoods [16]. Therefore, Equation 2.38 turns into

$$\log p(\boldsymbol{x}|\mu, \sigma^2) = -\frac{m}{2} \log \sigma^2 - \frac{m}{2} \log(2\pi) - \frac{1}{2\sigma^2} \sum_{j=1}^{m} (x_j - \mu)^2. \tag{2.39}$$

Maximizing Equation 2.39 with respect to μ and σ^2 leads to the sample mean

$$\hat{\mu} = \frac{1}{m}\sum_{j=1}^{m} x_j \tag{2.40}$$

and to the sample variance

$$\tilde{\sigma}^2 = \frac{1}{m}\sum_{j=1}^{m}(x_j - \hat{\mu})^2. \tag{2.41}$$

Since the expectation of the sample variance is

$$\mathbb{E}[\tilde{\sigma}^2] = \frac{m-1}{m}\sigma^2, \tag{2.42}$$

the unbiased sample variance is given by

$$\hat{\sigma}^2 = \frac{1}{m-1}\sum_{j=1}^{m}(x_j - \hat{\mu})^2. \tag{2.43}$$

Multivariate normal distribution. In general, it is also possible that multiple random variables $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ follow a joint normal distribution, a so-called multivariate normal distribution

$$p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^n \det \boldsymbol{\Sigma}}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right), \tag{2.44}$$

where n is the dimensionality of the random variable vector, $\boldsymbol{\mu}$ is the mean vector, and $\boldsymbol{\Sigma}$ is the $n \times n$ covariance matrix. As for the one-dimensional normal distribution, the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ can be again inferred from observations by the sample mean vector

$$\hat{\boldsymbol{\mu}} = \frac{1}{m}\sum_{j=1}^{m} \boldsymbol{x}_j \tag{2.45}$$

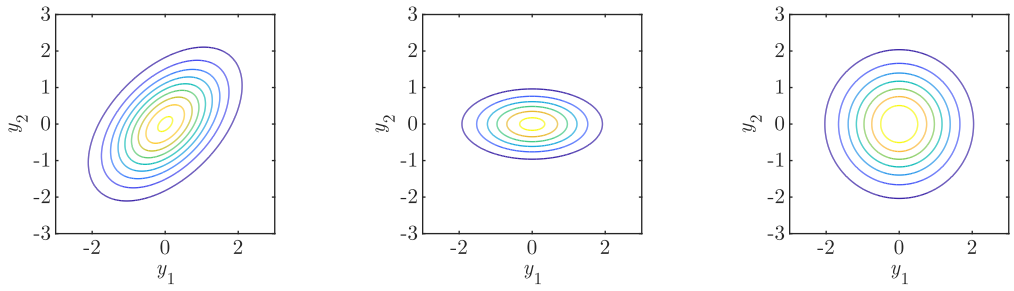
and the unbiased sample covariance matrix

$$\hat{\boldsymbol{\Sigma}} = \begin{bmatrix} \hat{\sigma}_{1,1} & \cdots & \hat{\sigma}_{1,n} \\ \vdots & \ddots & \vdots \\ \hat{\sigma}_{n,1} & \cdots & \hat{\sigma}_{n,n} \end{bmatrix}, \hat{\sigma}_{i,k} = \frac{1}{m-1}\sum_{j=1}^{m}(x_{i,j} - \hat{\mu}_i)(x_{k,j} - \hat{\mu}_k) \tag{2.46}$$

with $\hat{\sigma}_i^2 = \hat{\sigma}_{i,i}$. The main diagonal of the covariance describes the variances of each dimension whereas the non-diagonal elements are the covariances which also include information about the correlations between the dimensions. A two-dimensional covariance matrix can be expressed as

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1^2 & \rho_{1,2}\sigma_1\sigma_2 \\ \rho_{1,2}\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}, \tag{2.47}$$

where $\rho_{1,2}$ is the Pearson correlation coefficient which is a measure of linear correlation, in this case, between variable 1 and 2. The Pearson correlation coefficient is defined between -1 and $+1$ where -1 describes a perfect negative linear correlation, 0 no linear correlation, and $+1$ perfect positive linear correlation. Figure 2.12 shows the contour lines of three different bivariate normal distributions and their dependency on the variances and the Pearson correlation coefficient.



(a) (b) (c)

Figure 2.12: Contour lines of bivariate normal distribution with (a) positive correlation ($\rho_{1,2} > 0$), (b) no correlation ($\rho_{1,2} = 0$), and (c) no correlation ($\rho_{1,2} = 0$) and $\sigma_1^2 = \sigma_2^2$, see also [16].

Conditional normal distribution. An important property of a multivariate normal distribution is that its conditional is also a normal distribution. A detailed derivation is provided in Appendix A. If two sets of variables $\boldsymbol{x} = [\boldsymbol{x}_1, \boldsymbol{x}_2]^\top$ follow a multivariate normal distribution $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$, their marginal distributions as well as the conditional distribution of one set conditioned on the other are normal distributions too. For the two sets of variables, the mean vector and the covariance matrix can be split according to the two variable sets into $\boldsymbol{\mu} = [\boldsymbol{\mu}_1, \boldsymbol{\mu}_2]^\top$ and

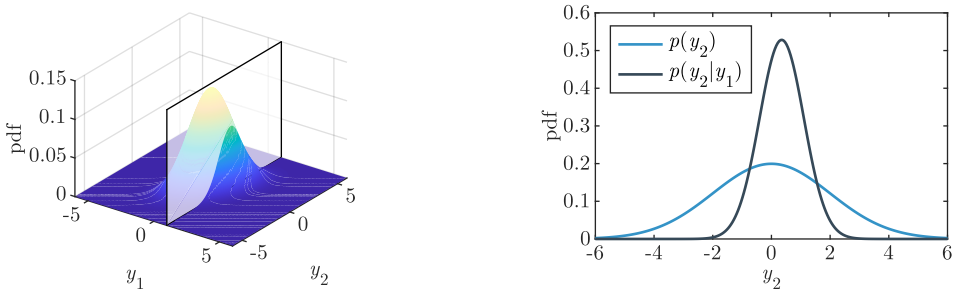
$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{1,1} & \boldsymbol{\Sigma}_{1,2} \\ \boldsymbol{\Sigma}_{2,1} & \boldsymbol{\Sigma}_{2,2} \end{bmatrix}, \tag{2.48}$$

where $\boldsymbol{\Sigma}_{1,1}$ and $\boldsymbol{\Sigma}_{2,2}$ are symmetric and $\boldsymbol{\Sigma}_{1,2} = \boldsymbol{\Sigma}_{2,1}^\top$. The marginal distributions for $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are $p(\boldsymbol{x}_1) = \mathcal{N}(\boldsymbol{x}_1|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_{1,1})$ and $p(\boldsymbol{x}_2) = \mathcal{N}(\boldsymbol{x}_2|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_{2,2})$, and the conditional distribution of $\boldsymbol{x}_1$ given $\boldsymbol{x}_2$ is $p(\boldsymbol{x}_1|\boldsymbol{x}_2) = \mathcal{N}(\boldsymbol{x}_1|\boldsymbol{\mu}_{1|2}, \boldsymbol{\Sigma}_{1|2})$, where

$$\boldsymbol{\mu}_{1|2} = \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{1,2}\boldsymbol{\Sigma}_{2,2}^{-1}(\boldsymbol{x}_2 - \boldsymbol{\mu}_2) \tag{2.49}$$

and

$$\boldsymbol{\Sigma}_{1|2} = \boldsymbol{\Sigma}_{1,1} - \boldsymbol{\Sigma}_{1,2}\boldsymbol{\Sigma}_{2,2}^{-1}\boldsymbol{\Sigma}_{2,1}. \tag{2.50}$$



(a) (b)

Figure 2.13: (a) Bivariate normal distribution with measured plane and (b) marginal and conditional normal distribution.

A simple example illustrates the behavior of conditional normal distributions. Figure 2.13a shows the bivariate normal distribution $\mathcal{N}(\boldsymbol{y}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with $\boldsymbol{\mu} = [0, 0]^\top$ and

$$\boldsymbol{\Sigma} = \begin{bmatrix} 1.0 & 0.7 \\ 0.7 & 2.0 \end{bmatrix}. \tag{2.51}$$

Now we assume that variable 1 is measured and realized by $y_1 = 1$, which is indicated by the transparent plane in Figure 2.13a. The conditional distribution of y_2 can be seen as the normalized distribution cut by the transparent plane in Figure 2.13a and can be computed with Equations 2.49–2.50. The marginal distribution of y_2 as well as the conditional distribution of y_2 given $y_1 = 1$ are shown in Figure 2.13b. We can see that the conditional distribution has a smaller variance than the marginal distribution, which indicates that y_2 is known more precisely if y_1 is given. This behavior, however, is only true if the covariances are non-zero.

Gaussian processes. A Gaussian process can be seen as a generalization of a multivariate normal distribution. In order to derive it graphically, we first consider a bivariate normal distribution where its 2×2 covariance matrix and its contour lines are shown in Figures 2.14a and 2.14b, respectively. Additionally, three realizations are illustrated in Figure 2.14b. The

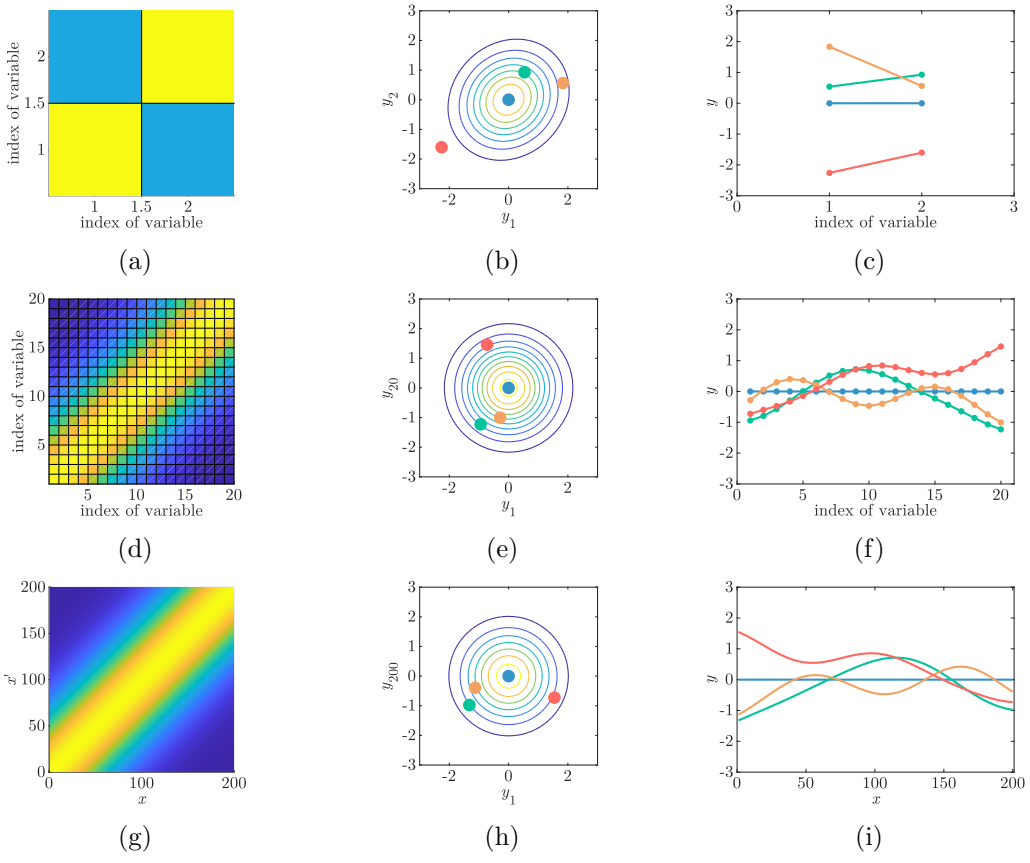


Figure 2.14: (a, d, g) Covariance matrices & function where yellow indicates high and blue low values, (b, e, h) contour lines for two variables, and (c, f, i) trajectories of the bivariate and the multivariate normal distribution, and of the Gaussian process.

realizations can also be displayed in a plot where both elements of the realized vector, y_1 and y_2, are plotted on the ordinate and their variable indices on the abscissa, see Figure 2.14c. For sampling different realizations, the line connecting the two points in Figure 2.14c changes its location. The mean vector of the distribution, which consists of two values $\boldsymbol{\mu} = [0, 0]^\top$, is displayed in blue. We now increase the dimensionality of the distribution. By considering, for example, a 20-dimensional normal distribution with a 20×20 covariance matrix (Figure 2.14d) and a 20×1 mean vector, we obtain 20 points for each realization, see Figures 2.14f. Due to space constraints, the marginal distribution's contour lines of only two variables, y_1 and y_{20} are shown in Figure 2.14b. The mean vector is again displayed in blue. Now, let us consider an infinite number of random variables that correspond to the position x rather than the variable indices. Then, we obtain a continuous line instead of a finite number of points, see Figure 2.14i. The mean vector and the covariance matrix turn, therefore, into a mean function $m(x)$ and a covariance function $k(x, x')$, respectively, and the multivariate normal distribution becomes a so-called Gaussian process. Figure 2.14g shows that the (co-)variance is now a function of x and x' indicated by the non-discrete color scheme. A Gaussian process is thus a collection of random variables where any finite number of them follows a joint normal distribution [116]. It is fully specified by its mean and covariance function and can be expressed as $\mathcal{N}(f(x)|m(x), k(x, x'))$. The shape of its realizations strongly depends on the chosen Gaussian process model, i.e., the mean and covariance function. Usually, predefined functions with free parameters $\boldsymbol{\theta}$ are used to specify a Gaussian process. Frequently used covariance and mean functions are, for example, the squared exponential covariance function

$$k(x, x') = \sigma_f^2 \exp\left(-\frac{(x - x')^2}{2\ell^2}\right), \tag{2.52}$$

where $\boldsymbol{\theta} = [\sigma_f, \ell]^\top$ are its free parameters, and the zero mean function $m(x) = 0$.

Conditional Gaussian processes. A Gaussian process can be conditioned on observed data similarly as previously shown for a multivariate normal distribution. Given the observed points $\boldsymbol{y}_+$ at $\boldsymbol{x}_+$, the conditional distribution of y_* at x_* can be expressed as $\mathcal{N}(y_*|m_{*|+}(x_*), k_{*|+}(x_*, x_*))$ with

$$m_{*|+}(x_*) = m(x_*) + k(x_*, \boldsymbol{x}_+)k(\boldsymbol{x}_+, \boldsymbol{x}_+)^{-1}(\boldsymbol{y}_+ - m(\boldsymbol{x}_+)) \tag{2.53}$$

and

$$k_{*|+}(x_*, x_*) = k(x_*, x_*) – k(x_*, \boldsymbol{x}_+)k(\boldsymbol{x}_+, \boldsymbol{x}_+)^{-1}k(x_*, \boldsymbol{x}_+)^\top, \tag{2.54}$$

where

$$m(\boldsymbol{x}_+) = [m(x_{+,1}), \ldots, m(x_{+,n})]^\top \tag{2.55}$$

and

$$k(\boldsymbol{x}_+, \boldsymbol{x}_+) = \begin{bmatrix} k(x_{+,1}, x_{+,1}) & \cdots & k(x_{+,1}, x_{+,n}) \\ \vdots & \ddots & \vdots \\ k(x_{+,n}, x_{+,1}) & \cdots & k(x_{+,n}, x_{+,n}) \end{bmatrix} \tag{2.56}$$

are the mean and covariance function evaluated at the observed locations $\boldsymbol{x}_+$, $m(x_*)$ and $k(x_*, x_*)$ are the mean and covariance function evaluated at the location to predict x_*, and $k(x_*, \boldsymbol{x}_+) = [k(x_*, x_{+,1}), \ldots, k(x_*, x_{+,n})]$ are the covariances between the values at the locations to predict and the observed locations.

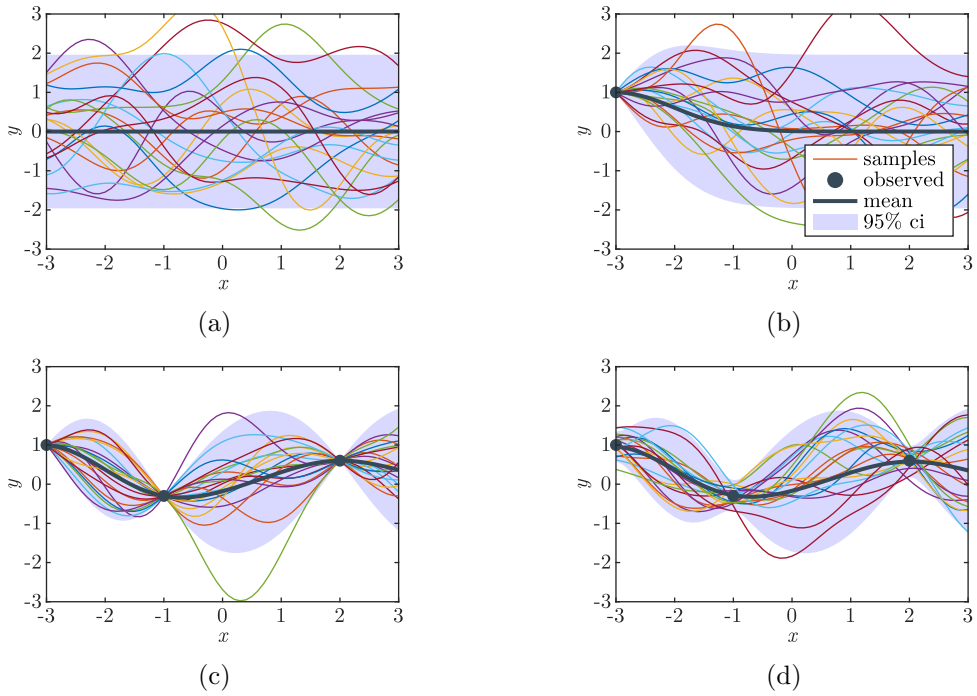


Figure 2.15: (a) Prior, (b) first update, (c) second and third update, (d) with observation error.

In the following, we consider a Gaussian process with a zero mean and a squared-exponential covariance function with $\sigma_f = 1.0$ and $\ell = 1.0$ defined over $x_* \in [-3, 3]$. Figure 2.15a displays the mean function in black, the symmetric 95% credible region in light blue, and 20 realizations, where each is now a continuous line rather than a finite number of points. Given an observed point $y_+ = 1.0$ at $x_+ = -3.0$, the conditional Gaussian process can be computed with Equations 2.53–2.54. The resulting Gaussian process is now conditioned on that observation, leading all realizations to intersect at the observed point, see Figure 2.15b. Additionally, the mean function passes this point, and the credible interval is narrowed down to zero as the value at this location is exactly known. The credible region close to the observed point is narrowed down too. By assuming a squared-exponential covariance function, close points are linearly correlated with each other. Therefore, knowing the value at $x_+ = -3.0$ results in knowing close points to it more precisely. This is based on the same behavior as previously seen for the two-dimensional example. Figure 2.15c shows an example where multiple values $\boldsymbol{y}_+ = [1.0, -0.3, 0.6]^\top$ at $\boldsymbol{x}_+ = [-3.0, -1.0, 2.0]^\top$ are observed, leading to three intersection points of the realizations. In reality, however, observing or measuring values exactly is not possible. This can be considered by adding the variance of the squared observation error σ_y^2 to the main diagonal of the covariance function evaluated at the observed location, i.e., $k(\boldsymbol{x}_+, \boldsymbol{x}_+) + \sigma_y^2 \boldsymbol{I}$. Figure 2.15d shows the same example as Figure 2.15c, but we consider an observation error with $\sigma_y = 0.2$. The figure illustrates that the realizations do not intersect at the three observed points anymore but are rather narrowed down to the standard deviation equals the one of the observation error.

The examples in this chapter let us anticipate that the normal distribution and Gaussian processes can be helpful in many engineering problems. Prognostics of aerospace structures can

benefit from those models since non-negligible uncertainties are involved in our observations and in fatigue and fracture mechanics.

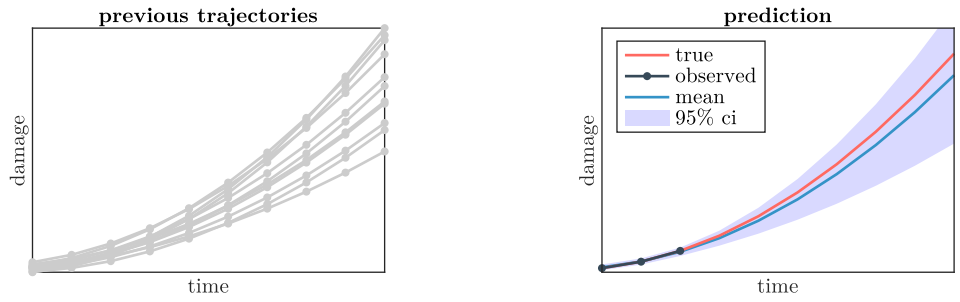
Chapter 3

Problem Types

"In God we trust, all others bring data."

– William Deming

In general, we want to leverage Gaussian processes for prognostics. Since prescribed Gaussian process models reduce the predictive capabilities of Gaussian processes, we use several previously collected trajectories, as shown in Figure 3.1a, to derive their underlying model. Then, we want to probabilistically predict future damage states based on observed conditions of our monitored mechanical system, see Figure 3.1b.



(a) (b)

Figure 3.1: (a) Previously collected trajectories from which we want to derive the underlying model and (b) prediction of future damage states based on observed conditions of our monitored mechanical system.

In the course of this work, we will encounter different challenges. The present chapter serves as an overview of the data sets used in this thesis, and the problems associated with them. The studied data sets entail different levels of difficulty. For example, one challenge occurs when the desired values cannot be measured precisely, and thus observation error is apparent. Previously collected trajectories can also consist of different numbers of data points, and statistical values like the sample mean vector cannot be computed anymore. Another challenge that arises in this thesis occurs when using simulation data as training trajectories. There, the observation error is absent but might be present in the data of the monitored

Table 3.1: Tackled challenges in the course of this thesis.

	Simulation	**Data from literature**				**Experiment**
Tackled challenges	crack growth simulation	FCG Hudak	FCG Virkler	Laser degradation	Milling machine wear	crack growth in aero. structure
present noise	(×)	×	×	×	×	×
diff. lengths of prev. data	(×)				×	×
non-normally distributed data	(×)					×
prev. data lacks noise	(×)					×
variable is hidden						×

system. We will also deal with non-normally distributed trajectories and with a data set in which the degradation variable, in this case, the crack length, is hidden. Table 3.1 links the tackled challenges to the data sets.

In this thesis, we will also use simulation data to showcase the proposed methods. The advantage of simulated data is that the underlying equations are known, and the challenges can be gradually added. This is denoted by (×) in Table 3.1. Apart from the simulation data, the difficulty level increases from left to right in the table. During this thesis, we will handle the more manageable data sets first and gradually increase the difficulty level. In total, six data sets regarding various fatigue crack growth problems, laser degradation, and milling machine wear are faced throughout the thesis. Each data set describes a particular system output/behavior. The term system output/behavior is used for the variable that indicates the deterioration of the mechanical system. The considered approaches deal with data sets where the system behavior is determined in the low-frequency domain.

In the present chapter, we first explain how to simulate fatigue crack growth in a pre-cracked plate. The results are used throughout this thesis and are therefore particularly important. In the second section, we describe the data sets found in literature. Those data sets are based on real-world applications and incorporate observation errors. The chapter ends by specifying the conducted experiments and presenting their results in the third section.

3.1 Crack growth simulation

As explained in the introduction chapter, aerospace structures are designed to tolerate damages, such as cracks. Many thin-walled aircraft structures follow this damage tolerant design guideline which is why we consider the crack growth in a plate as our first application. A rather simple example is a growing crack in a pre-cracked infinite plate with a remote stress range $\Delta\sigma_\infty$, see Figure 3.2. As introduced in the fundamentals chapter, the crack growth rate $\frac{da}{dN}$ can be modeled according to Paris and Erdogan [101] by

$$\frac{da}{dN} = C(\Delta K_I)^\alpha, \tag{3.1}$$

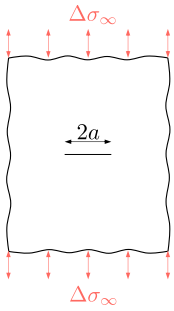


Figure 3.2: Pre-cracked infinite plate.

where C and α are material properties and ΔK_I is the range of the SIF in mode I. In general, the SIF range can be computed as $\Delta K_I = \gamma(a)\Delta\sigma_\infty\sqrt{\pi a}$, where a is half the crack length, see Figure 3.2, and $\gamma(a)$ is the geometry factor which is in general dependent on the crack length a. Assuming a constant remote stress range $\Delta\sigma_\infty$ and that C and α are independent of a, the differential equation can be solved with the initial condition $N_0 = 0$ for the number of cycles N dependent on the crack length a as

$$N(a) = \frac{1}{C\Delta\sigma_\infty^\alpha \pi^{\alpha/2}} \int_{a_0}^{a} \gamma(\bar{a})^{-\alpha}\, \bar{a}^{-\alpha/2}\, d\bar{a}. \tag{3.2}$$

For an infinite plate, the geometry factor becomes $\gamma = 1$ and the integral can be explicitly solved to

$$N(a) = \frac{2}{(2-\alpha)C\Delta\sigma_\infty^\alpha \pi^{\alpha/2}} \left(a^{\frac{2-\alpha}{2}} - a_0^{\frac{2-\alpha}{2}} \right) \tag{3.3}$$

or

$$a(N) = \left(\frac{(2-\alpha)C\Delta\sigma_\infty^\alpha \pi^{\alpha/2}}{2} N + a_0^{\frac{2-\alpha}{2}} \right)^{\frac{2}{2-\alpha}}. \tag{3.4}$$

Now, the crack growth in an infinite plate can be analytically computed. Different crack growth trajectories can be produced by varying the parameters in Equation 3.4. Since crack propagation is subject to uncertainties, one can consider some parameters in Equation 3.4 as random variables. By assuming a probability distribution for each parameter, a set of samples can be drawn, and the corresponding crack growth trajectory computed. For an initial crack length of $a_0 = 9 \times 10^{-3}\,\mathrm{m}$, a remote stress range of $\Delta\sigma_\infty = 48.26\,\mathrm{MPa}$, and the material parameters $\alpha = 2.9$ and $\mathcal{N}(C|\mu_C, \sigma_C^2)$ with $\mu_C = 8.7096 \times 10^{-11}$ and $\sigma_C = 1.519 \times 10^{-11}$, 50 sampled crack growth trajectories are shown in Figure 3.3a (C with $[da/dN] = \mathrm{m/cycle}$ and $[\Delta K_I] = \mathrm{MPa}\sqrt{\mathrm{m}}$). Each trajectory stops at approximately 0.05 m. Figure 3.3b shows the trajectories when additionally assuming an additive observation error of $\mathcal{N}(e|0, \sigma_y^2)$ with $\sigma_y = 0.4 \times 10^{-3}\,\mathrm{m}$.

When simulating crack growth trajectories, we can add an observation error, simulate trajectories with the same and different lengths, and deal with non-normally distributed data. Therefore, simulated data provides the great possibility to increase the difficulty level gradually. Since we also know the exact solution and the a priori assumed values, we are able to validate our methods.

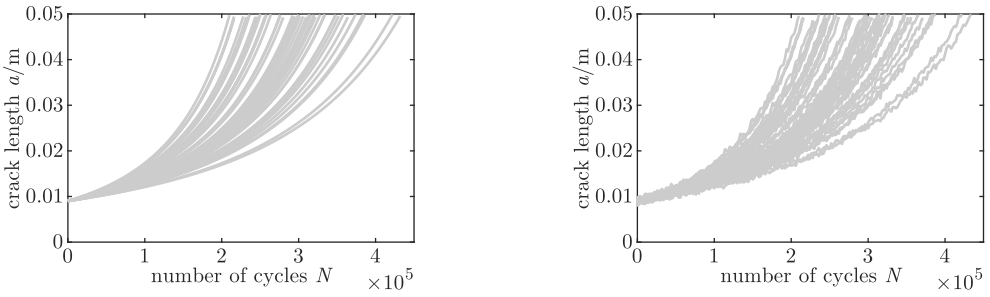


Figure 3.3: Crack growth trajectories (a) without and (b) with observation error.

3.2 Data sets from literature

In literature, various data sets regarding degrading mechanical systems can be found. Four data sets are used for multiple studies in this thesis, representing fatigue crack growth, laser degradation, and milling machine wear. The data sets shown in Figure 3.4 open the possibility to apply the methods presented in this work to different areas. All data sets are based on measurements and are subject to observation error.

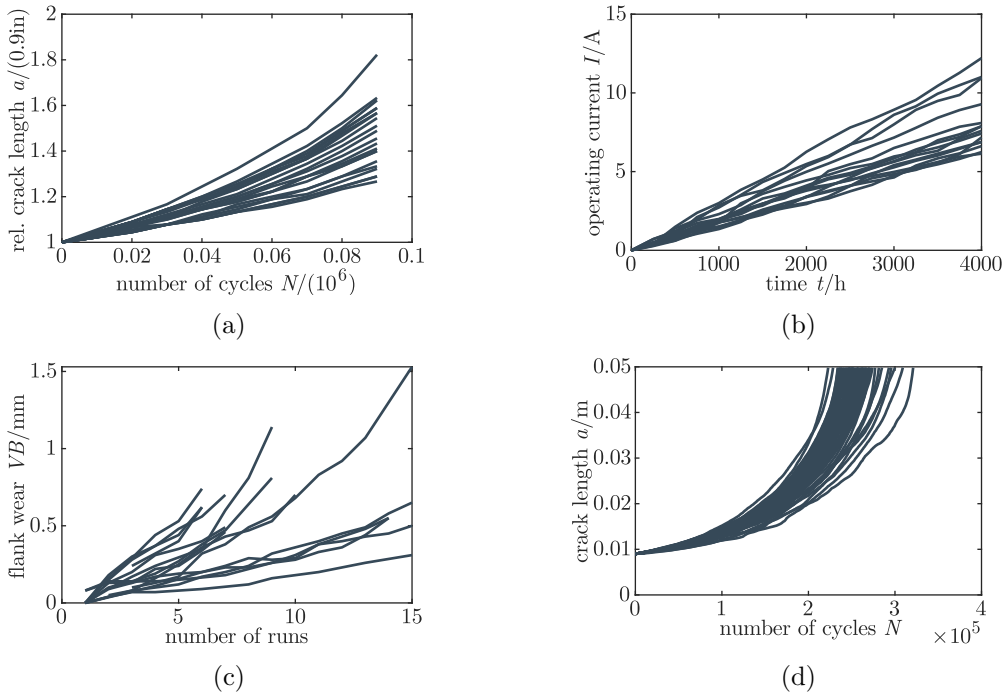


Figure 3.4: Data found in literature representing (a) fatigue crack growth in a compact tension specimen (FCG Hudak), (b) laser degradation, (c) milling machine wear, and (d) fatigue crack growth in an aluminum plate (FCG Virkler).

FCG Hudak data set. The fatigue crack growth data used in this work (Figure 3.4a) was originally gathered by Hudak et al. [58] and pictures crack growth in compact tension specimens

made of steel. The data set was visually obtained by Meeker and Escobar [89] from Figure 4.5.2 in [20]. The crack length a is given every $10,000$ cycles. We cut all trajectories at $90,000$ cycles such that each line consists of the same number of data points. Additionally, the crack lengths are normalized by the initial crack length of $a_0 = 0.9\,\text{in}$.

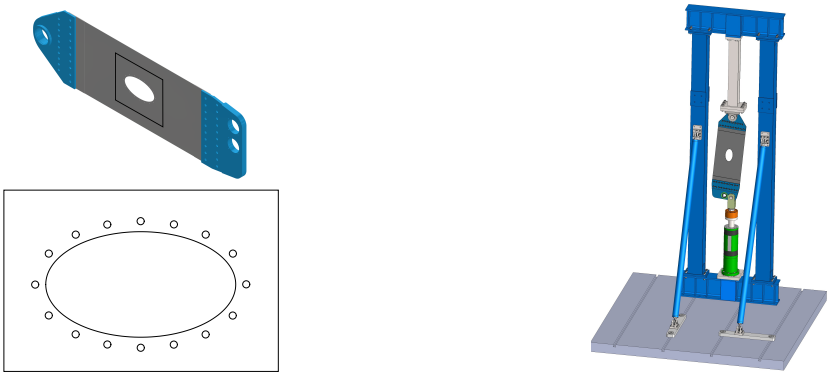
FCG Virkler data set. Another fatigue crack growth data set was published by Virkler et al. [148] which consists of 68 crack growth trajectories, see Figure 3.4d. The center-cracked aluminum 2024–T3 test specimens with a total width of $W = 152.4\,\text{mm}$ and an initial crack size of $a_0 = 9\,\text{mm}$ were tested under a constant amplitude loading with a stress range of $\Delta\sigma_\infty = 48.26\,\text{MPa}$ up to a final crack length of $a_c = 49.8\,\text{mm}$. During the test, the number of cycles was measured every $\Delta a = 0.2\,\text{mm}$ up to $a = 36.2\,\text{mm}$, every $\Delta a = 0.4\,\text{mm}$ up to $a = 44.2\,\text{mm}$, and every $\Delta a = 0.8\,\text{mm}$ up to $a = 49.8\,\text{mm}$.

Laser degradation data set. Meeker and Escobar [89] published the laser degradation data set consisting of 15 gallium arsenide (GaAs) laser device degradation trajectories. In general, the output light intensity decreases over time. However, due to a feedback mechanism in the experiment, the laser output was kept constant by increasing the operating current. The operating current I was measured every $250\,\text{h}$ until $4,000\,\text{h}$, showing an increasing trend. The trajectories that indicate the degrading lasers are shown in Figure 3.4b.

Milling machine wear data set. All the previously explained data sets from literature have in common that their trajectories are equally long and measured at the same locations. By contrast, the milling machine wear trajectories published by Agogino and Goebel [3] exhibit different numbers of data points per line. This is a challenge since the sample mean vector and covariance matrix cannot be computed. The entire data set consists of several sensor measurements. However, we only use the flank wear over the executed runs in this work. In total, 16 cases were considered by Agogino and Goebel. Since one case comprises only an initial value, we investigate 15 milling machine wear trajectories in our studies, see Figure 3.4c.

3.3 Conducted experiments

In contrast to the before-mentioned examples, usually, the degradation variable, such as the crack length, cannot be observed in real-world applications. For example, continuous visual

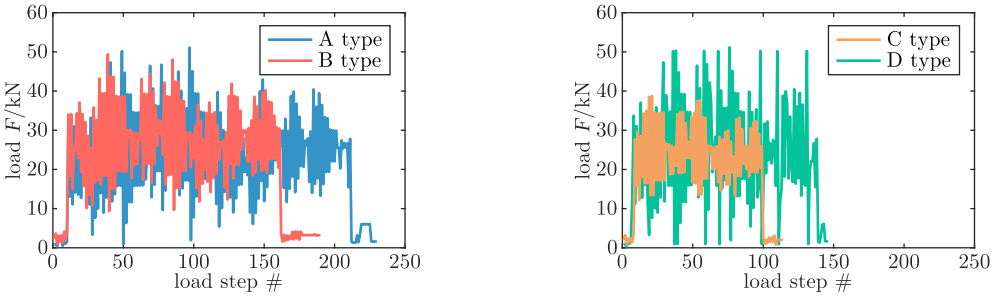


(a) (b)

Figure 3.5: (a) Specimen and (b) test rig.

monitoring for aircraft in flight is not applicable since cameras are heavy and difficult to mount.

Therefore, sensors are attached to structures to infer their conditions from the collected data. In order to generate data for such an application, we tested an aluminum panel that resembles a lower section of a civil aircraft wing. It is made of the aluminum alloy Al 2024–T351 with Young's modulus of $E = 72,000\,\text{MPa}$ and a Poisson's ratio of $\nu = 0.34$. The structure is $1,920\,\text{mm}$ long and $570\,\text{mm}$ wide and has an elliptical armhole with a length of $135\,\text{mm}$ and a width of $75\,\text{mm}$. The hole allows reaching into the wing to inspect the inner surfaces of the structure with an endoscope. Around the centered armhole, there are 16 small holes with a radius of $2\,\text{mm}$. The armhole is usually covered by a lid fixed on the small holes. The whole specimen is shown in Figure 3.5a.



(a) (b)

Figure 3.6: Loading program with flight types (a) A and B, and (b) C and D.

During the fatigue test, the load was applied by a hydraulic cylinder at an angle in order to represent shear stresses in a wing resulting from twisting, see Figure 3.5b. The loading program is based on four different flight types, A, B, C, and D (Figure 3.6), to realistically simulate the forces acting on an aircraft wing. They consist of 230, 190, 114, and 146 load steps, respectively. The flight types are ordered randomly for the first 100 flights. The order of the first 100 flights is repeated consecutively afterwards. The occurrences of the different flight types, A, B, C, and D, in the first 100 flights are 55, 15, 20, and 10 times, respectively.

(a) (b)

Figure 3.7: Sensors of (a) P02T01 and (b) P03T01.

In total, two equally manufactured specimens, P02T01 and P03T01, were tested with the same loading program. Several sensors were attached to the panel to monitor the structures. The sensors were predominantly positioned according to the crack detection zones method developed by Pfingstl and Zimmermann [104] and Pfingstl et al. [106] to ensure that the applied strain gauges detect the occurring cracks. In total, two single strain gauges and three

strain rosettes were attached to specimen P02T01 and two single strain gauges and four strain rosettes to specimen P03T01, see Figures 3.7a and 3.7b.

(a) (b) (c)

Figure 3.8: Fatigue process of aerospace structure with (a) intact structure, (b) crack growth, and (c) final fracture, see Pfingstl et al. [106].

At the beginning of each experiment, the specimen was completely sound, see Figure 3.8a. After loading the structure several hundreds of flights, a crack was initiated, as shown in Figure 3.8b. Afterwards, the crack propagated until the structure eventually failed, see Figure 3.8c. Test engineers regularly inspected the structure during the experiment to detect cracks and measure their lengths. The measured crack lengths of P02T01 and P03T01 are shown in Figure 3.9.

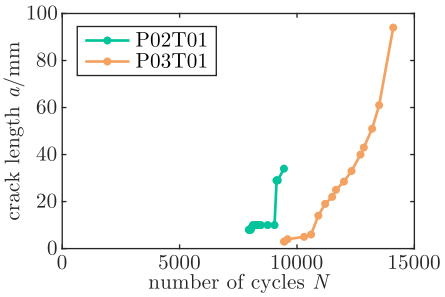


Figure 3.9: Crack lengths resulting from visual inspections for specimens P02T01 and P03T01.

The data sets presented in this chapter are used throughout the thesis. We start dealing with the simulated trajectories and continue with the data sets found in the literature. Finally, in the second-last chapter, we will propose methods to predict crack growth in the aerospace structure.

Chapter 4

Inferred Gaussian Processes

"Prognosen sind schwierig, besonders wenn sie die Zukunft betreffen."

– Karl Valentin

Predictions are difficult, especially when it comes to the future. This sentence, said by the German comedian Karl Valentin, points out amusingly that it is challenging to foresee events. As already explained in the introduction chapter, this is especially true for incidences in large systems such as future CO_2 concentrations in the air or stock prices where many influencing factors that might vary over time are apparent. By contrast, the system boundaries of experiments regarding mechanical systems can often be precisely drawn, allowing us to control or at least acknowledge influencing factors, leading to repetitive behaviors and similar results for the same experiments. As the results of a mechanical experiment are repetitive, we can tackle the challenge of predicting the conditions of mechanical systems by integrating prior information about the system behavior in form of previously executed experiments into our model.

Section 4.1 and Sections 4.3–4.6 of this chapter are fully based on the paper Pfingstl and Zimmermann [105]. The results were further discussed with Dr. Mario Teixeira Parente from the Jülich Centre for Neutron Science. His valuable advice and feedback are gratefully acknowledged.

Before introducing the approach of integrating prior information into Gaussian processes in Section 4.3, the work is motivated by recent publications discussed in Section 4.1. Furthermore, an introduction on how to use a multivariate normal distribution for prognostics is given in Section 4.2. The approach for inferring a Gaussian process model from training trajectories is applied to three different data sets in Section 4.4, on which the effect of integrating prior knowledge into Gaussian processes is studied. Section 4.5 details an extension of this approach by incorporating problem-specific governing equations into Gaussian processes and investigates its impact on the predictions. In the final section, we conclude from the studies conducted in this chapter.

4.1 Motivation

Using mathematical surrogate models is one way of predicting the condition of mechanical systems based on data. Researchers have proposed many different types of models for pre-

dicting fatigue life [131]. Some of them, e.g., recurrent neural networks [30, 85, 96], support vector regression [64, 134], and k-nearest neighbors [138, 150], have been applied successfully to prognostics and health management problems. As fatigue-induced damage is still one of the most uncertain failures in structures [43], the models should supply information about their prediction's uncertainty. However, by default, the before-mentioned surrogate models do not provide any.

Since Gaussian process regression provides credible intervals for its predicted states, it has frequently been used for probabilistic predictions, see, for example, [5, 9, 10, 25, 40, 52, 57, 66, 71, 76, 77, 79, 93, 137, 161, 163]. As already introduced in Section 2.3, a Gaussian process is entirely defined by its mean and covariance function, which we also refer to as the Gaussian process model. In order to apply Gaussian process regression, a mean and a covariance function have to be chosen a priori. Table 4.1 shows recently used functions in prognostic and health management.

Table 4.1: Recently used mean and covariance functions for prognostics and health management problems.

Topic	**Mean function**	**Covariance function**	**Year**	**Reference**
Crack growth prediction	*zero*	neural network (NN)	2011	[93]
Degradation of bearings	*zero*	*squared-exponential (SE) iso*, Matern, NN	2012	[57]
Battery degradation	linear, quadratic	*SE iso*, periodic, constant	2013	[79]
Crack growth prediction	polynomial	*SE iso*	2015	[5]
Degradation of solder joints	*zero*	*SE iso*	2015	[71]
Damage detection in bearings	*zero*, constant, linear, quadratic, cubic	*SE iso, SE ard*, linear, linear ard, Matern iso, noise, periodic, rational-quadratic (RQ) ard, RQ iso	2017	[10]
Battery degradation	*zero*	*SE ard*	2018	[161]
Tool wear prediction	linear	*SE iso*	2018	[66]
Crack growth prediction	*zero*	*SE iso*	2019	[76]
Degradation of bearings	linear	*SE iso*	2019	[52]
Crack growth evaluation	*zero*	SE iso + linear	2019	[163]
Crack growth evaluation	*zero*	SE ard + linear ard	2020	[25]
Seismic fragility	*zero*	*SE iso*	2020	[40]

iso: isotropic length scale

ard: automatic relevance determination

As illustrated in Table 4.1, a frequent choice is a zero mean and a squared-exponential covariance function. However, using such predefined functions can lead to poor predictions as they involve the assumption of a specific function family. As stated in [57, 63, 71, 112, 123], prior knowledge regarding the covariance function should improve the prediction of Gaussian processes. In practice, engineers usually have good a priori knowledge about the system behavior/output due to pre-executed simulations or tests. Nevertheless, defining a Gaussian process model might still be challenging resulting in choosing default priors such as a zero mean and a squared-exponential covariance function. Therefore, an approach for inferring a Gaussian process model from previous data seems worthwhile.

4.2 Prognostics with a multivariate normal distribution

A prerequisite for using Gaussian processes is assuming the system output y to be normally distributed. As explained in Section 2.3, the probability density function for a finite number of

data points $\boldsymbol{y}$ is $p(\boldsymbol{y}) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu}$ is the mean vector and $\boldsymbol{\Sigma}$ the covariance matrix of $\boldsymbol{y}$. If we have m previously gathered trajectories $\{y_i|i = 1, \ldots, n\}_j$ with $j = 1, \ldots, m$ of the system output (e.g., crack length) at the locations $\{x_i|i = 1, \ldots, n\}$ (e.g., number of cycles), we can build a model based on a multivariate normal distribution by simply computing the sample mean vector $\hat{\boldsymbol{\mu}}$ with $\hat{\mu}_i = \frac{1}{m}\sum_{j=1}^{m} x_{i,j}$ and the sample covariance matrix $\hat{\boldsymbol{\Sigma}}$ with $\hat{\sigma}_i^2 = \hat{\sigma}_{i,i}$ and $\hat{\sigma}_{i,k} = \frac{1}{m-1}\sum_{j=1}^{m}(x_{i,j} - \hat{\mu}_i)(x_{k,j} - \hat{\mu}_k)$ of the previously collected data. Without any current data of the monitored system, the prediction of this model is the sample mean. The symmetric 95% credible interval, i.e., the interval between the 2.5% and the 97.5% quantiles, is determined by $\hat{y}_{i,2.5\%} = \hat{\mu}_i - 1.96\hat{\sigma}_i$ and $\hat{y}_{i,97.5\%} = \hat{\mu}_i + 1.96\hat{\sigma}_i$. If now current data $\boldsymbol{y}_+$ at $\boldsymbol{x}_+$ of the monitored system is observed, the prediction can be updated by computing the conditional distribution as

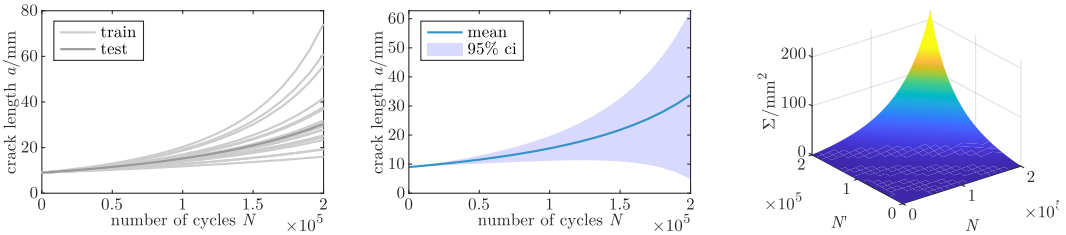
$$p(\boldsymbol{y}_*|\boldsymbol{y}_+) = \mathcal{N}\left(\boldsymbol{y}_*|\hat{\boldsymbol{\mu}}_* + \hat{\boldsymbol{\Sigma}}_{*,+}\hat{\boldsymbol{\Sigma}}_{+,+}^{-1}(\boldsymbol{y}_+ - \hat{\boldsymbol{\mu}}_+), \hat{\boldsymbol{\Sigma}}_{*,*} - \hat{\boldsymbol{\Sigma}}_{*,+}\hat{\boldsymbol{\Sigma}}_{+,+}^{-1}\hat{\boldsymbol{\Sigma}}_{*,+}^{\top}\right), \tag{4.1}$$

where $\hat{\boldsymbol{\mu}}_+$ is the sample mean vector and $\hat{\boldsymbol{\Sigma}}_{+,+}$ the sample covariance matrix at the the currently observed locations $\boldsymbol{x}_+$. $\hat{\boldsymbol{\mu}}_*$ is the sample mean vector and $\hat{\boldsymbol{\Sigma}}_{*,*}$ the sample covariance matrix at the prediction locations $\boldsymbol{x}_*$, and $\hat{\boldsymbol{\Sigma}}_{+,*}$ is the sample covariance matrix between the values at $\boldsymbol{x}_+$ and $\boldsymbol{x}_*$. Note that the currently observed locations $\boldsymbol{x}_+$ and the prediction locations $\boldsymbol{x}_*$ have to be subsets of the locations $\boldsymbol{x}$ where the previously collected trajectories were observed.

Based on this simple model, we can predict the crack growth in an infinite plate. As an example, let us assume to have 20 previously collected trajectories, as shown in bright grey in Figure 4.1a. The trajectories are determined by

$$a(N) = \left(\frac{(2-\alpha)C\Delta\sigma_\infty^\alpha \pi^{\alpha/2}}{2}N + a_0^{\frac{2-\alpha}{2}}\right)^{\frac{2}{2-\alpha}}, \tag{4.2}$$

which was derived in Section 3.1. The remote stress range is set to $\Delta\sigma_\infty = 48.26\,\mathrm{MPa}$, the material exponent to $\alpha = 2.9$, and the initial crack length to $a_0 = 9 \times 10^{-3}\,\mathrm{m}$. The different trajectories are generated by varying the parameter C according to a normal distribution $\mathcal{N}(C|\mu_C, \sigma_C^2)$ with $\mu_C = 8.7096 \times 10^{-11}$ and $\sigma_C = 1.5190 \times 10^{-11}$ (C with $[da/dN] = \mathrm{m/cycle}$ and $[\Delta K_I] = \mathrm{MPa}\sqrt{\mathrm{m}}$). In this example, the crack lengths are computed every 10,000 cycles until 200,000 cycles. With this previous data, we can determine the sample mean vector and the sample covariance matrix. They are shown in Figures 4.1b and 4.1c, respectively.

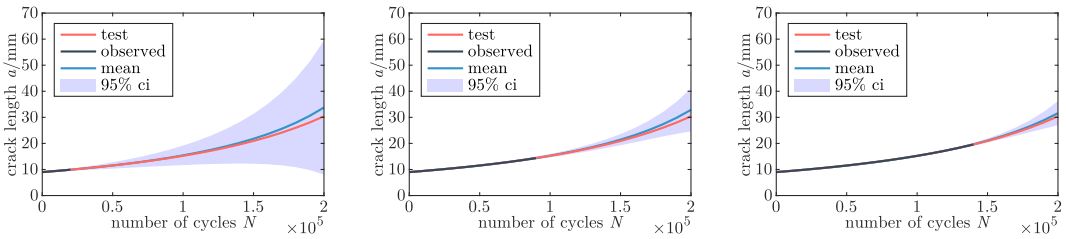


(a) (b) (c)

Figure 4.1: (a) Crack growth trajectories, (b) mean vector and credible intervals, and (c) sample covariance matrix.

Figure 4.1b shows the crack growth prediction with a symmetric 95% credible region for the initial state, i.e., no monitoring data is available. The prediction is thus completely based on

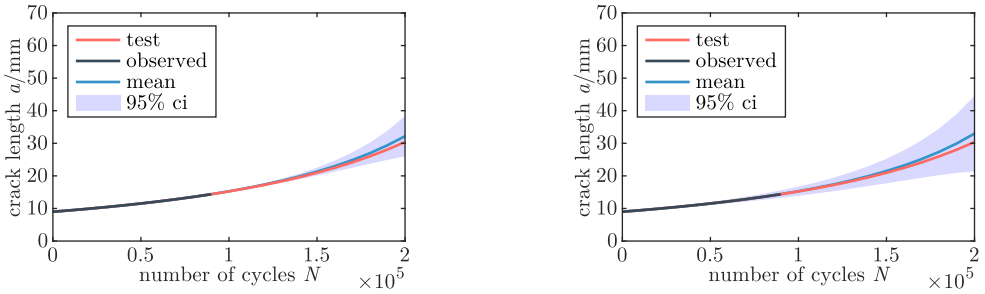
the computed sample mean vector and sample covariance matrix. Now, we suppose that crack growth in a new plate is gradually observed. For this, we use the test trajectory highlighted in Figure 4.1a. The crack growth prediction can be updated based on the currently measured crack lengths by computing the conditional distribution given in Equation 4.1. Figures 4.2a–4.2c show the updated predictions for an early, medium, and late observation state of the test trajectory. We can see that the more data we observe, the narrower the credible region of the prediction becomes.



(a) (b) (c)

Figure 4.2: Crack growth prediction at an (a) early stage, (b) medium stage, and (c) late stage.

Considering imprecision due to measurement, we add the variance of the observation error σ_y^2 to the principal diagonal of the covariance matrix at the observed locations $\tilde{\boldsymbol{\Sigma}}_{+,+} = \hat{\boldsymbol{\Sigma}}_{+,+} + \sigma_y^2 \boldsymbol{I}$. In doing so, the predicted mean does not follow the measured data exactly anymore. The higher the observation error's variance is, the less precisely the predicted mean follows the currently observed data, and the less the predicted variance is reduced, see Figures 4.3a and 4.3b.



(a) (b)

Figure 4.3: Crack growth predictions with a (a) small and (b) large observation error.

A drawback of using a multivariate normal distribution is that states can only be predicted at locations where the previously collected trajectories are measured. One could train a regression model on the sample mean vector and the sample covariance matrix mapping $\boldsymbol{x} \to \hat{\boldsymbol{\mu}}$ and $(\boldsymbol{x}, \boldsymbol{x}') \to \hat{\boldsymbol{\Sigma}}$ to bypass this restriction. Yet, a problem arises for fitting a regression model to the sample covariance matrix: One must ensure that the model leads to a symmetric and positive semidefinite matrix for every possible input (x, x') [128]. Another problem of this approach occurs when the previous trajectories are not equally long, or data points within trajectories are missing. Then, the sample mean vector and the sample covariance matrix cannot be determined with the standard equations anymore. As Gaussian processes describe

an infinite number of random variables, they could solve these problems. This is why we try to integrate the information of the previously collected trajectories into Gaussian processes in the next section.

4.3 Integrating prior knowledge into Gaussian processes

As already introduced in Chapter 2, a Gaussian process is defined by its mean $m_\theta(x)$ and covariance function $k_\theta(x, x')$ which typically have some free parameters $\boldsymbol{\theta}$. Table 4.1 shows that a frequent choice is the squared-exponential covariance function that additionally considers noise

$$k_\theta(x, x') = \sigma_f^2 \exp\left(-\frac{(x - x')^2}{2\ell^2}\right) + \delta\, \sigma_y^2 \tag{4.3}$$

and a zero mean function, where δ is the Kronecker delta with $\delta = 0$ for $x \neq x'$ and $\delta = 1$ for $x = x'$, and $\boldsymbol{\theta} = [\sigma_f, \ell, \sigma_y]^\top$ are the free parameters. Typically, these parameters are trained by maximizing the log-likelihood

$$\begin{aligned} \log p(\boldsymbol{y}_+|\boldsymbol{\theta}) = &- \frac{n_+}{2}\log(2\pi) - \frac{1}{2}\log\left(\det k_\theta(\boldsymbol{x}_+, \boldsymbol{x}_+)\right) \\ &- \frac{1}{2}\left(\boldsymbol{y}_+ - m_\theta(\boldsymbol{x}_+)\right)^\top k_\theta(\boldsymbol{x}_+, \boldsymbol{x}_+)^{-1}\left(\boldsymbol{y}_+ - m_\theta(\boldsymbol{x}_+)\right), \end{aligned} \tag{4.4}$$

with n_+ as the dimensionality of the observed data, e.g., crack length,

$$\boldsymbol{y}_+ = \left[y_{+,1}, \ldots, y_{+,n_+}\right]^\top \tag{4.5}$$

at the locations, e.g., number of cycles,

$$\boldsymbol{x}_+ = \left[x_{+,1}, \ldots, x_{+,n_+}\right]^\top, \tag{4.6}$$

the mean vector

$$m_\theta(\boldsymbol{x}_+) = \left[m_\theta(x_{+,1}), \ldots, m_\theta(x_{+,n_+})\right]^\top, \tag{4.7}$$

and the covariance matrix

$$k_\theta(\boldsymbol{x}_+, \boldsymbol{x}'_+) = \begin{bmatrix} k_\theta(x_{+,1}, x_{+,1}) & \ldots & k_\theta(x_{+,1}, x_{+,n_+}) \\ \vdots & \ddots & \vdots \\ k_\theta(x_{+,n_+}, x_{+,1}) & \ldots & k_\theta(x_{+,n}, x_{+,n_+}) \end{bmatrix}, \tag{4.8}$$

leading to the optimized parameters $\hat{\boldsymbol{\theta}}$. Then, the conditional distribution at a location x is given as

$$\begin{aligned} f(x) \mid \boldsymbol{x}_+, \boldsymbol{y}_+, \hat{\boldsymbol{\theta}} \sim \mathcal{N}(&m_\theta(x) + k_\theta(x, \boldsymbol{x}_+)k_\theta(\boldsymbol{x}_+, \boldsymbol{x}_+)^{-1}(\boldsymbol{y}_+ - m_\theta(\boldsymbol{x}_+)), \\ &k_\theta(x, x) - k_\theta(x, \boldsymbol{x}_+)k_\theta(\boldsymbol{x}_+, \boldsymbol{x}_+)^{-1}k_\theta(x, \boldsymbol{x}_+)^\top). \end{aligned} \tag{4.9}$$

This algorithm is especially powerful for regression problems, see [116], and is referred to as Gaussian process regression. The algorithm was used, for example in [5, 52, 57, 71], to predict damage states: Each time a new currently observed data point is available, the Gaussian process model is optimized, and its conditional distribution computed, leading to a new prediction for upcoming states. However, no prior information is integrated into the Gaussian process

model besides the a priori chosen mean and covariance function, which often leads to poor predictions and a short look-ahead time. Therefore, we present how to integrate prior knowledge into a predefined Gaussian process model in the following. In the presented state-of-the-art algorithm, the mean and covariance functions have to be chosen a priori, involving the assumption of a specific function family. As this can also be a reason for poor predictions, we further propose an approach that infers an appropriate Gaussian process model from previous data and thus weakens the assumption about a certain function family made by choosing prescribed models.

Training of a prescribed Gaussian process model with previous data

Instead of optimizing a prescribed Gaussian process model only on currently observed data, we can also train the parameters of a prescribed mean and covariance function on previous data. As previously, we assume a Gaussian process model chosen a priori which is dependent on a set of parameters $\boldsymbol{\theta}$. The standard deviation of the observation error can be seen as one parameter of this set. With Bayes' rule, the posterior distribution based on a given trajectory $\left(\boldsymbol{x}_j, \boldsymbol{y}_j = f_j(\boldsymbol{x_j})\right)$ is given by

$$p(\boldsymbol{\theta} \mid \boldsymbol{y}_j) = \frac{p(\boldsymbol{y}_j \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta})}{\int p(\boldsymbol{y}_j \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta})\, d\boldsymbol{\theta}} \propto p(\boldsymbol{y}_j \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta}), \tag{4.10}$$

where $p(\boldsymbol{\theta})$ and $p(\boldsymbol{y}_j \mid \boldsymbol{\theta})$ are the prior and the likelihood distribution, respectively. Assuming an uninformative prior and that all m trajectories are independent and identically distributed, the posterior distribution reads

$$p(\boldsymbol{\theta} \mid \boldsymbol{y}_1, ..., \boldsymbol{y}_m) = \prod_{j=1}^{m} \frac{p(\boldsymbol{y}_j \mid \boldsymbol{\theta})}{\int p(\boldsymbol{y}_j \mid \boldsymbol{\theta})\, d\boldsymbol{\theta}} \propto \prod_{j=1}^{m} p(\boldsymbol{y}_j \mid \boldsymbol{\theta}) \tag{4.11}$$

and its log-likelihood is

$$\log p(\boldsymbol{\theta} \mid \boldsymbol{y}_1, ..., \boldsymbol{y}_m) \propto \sum_{j=1}^{m} \log p(\boldsymbol{y}_j \mid \boldsymbol{\theta}) \tag{4.12}$$

with

$$\begin{aligned} \log p(\boldsymbol{y}_j|\boldsymbol{\theta}) = &- \frac{n}{2} \log(2\pi) - \frac{1}{2} \log\left(\det k_\theta(\boldsymbol{x}_j, \boldsymbol{x}_j)\right) \\ &- \frac{1}{2} \left(\boldsymbol{y}_j - m_\theta(\boldsymbol{x}_j)\right)^\top k_\theta(\boldsymbol{x}_j, \boldsymbol{x}_j)^{-1} \left(\boldsymbol{y}_j - m_\theta(\boldsymbol{x}_j)\right), \end{aligned} \tag{4.13}$$

Therefore, the sum of the trajectories' log-likelihoods must be maximized, as also stated in [91]. By optimizing the parameters such that the sum of the trajectories' log-likelihoods is maximized, prior knowledge is integrated into the Gaussian process model. However, we still have to choose a mean and a covariance function and thus assume a specific function family. Therefore, we propose an approach for inferring the mean function and the covariance function from previous data in the following.

Inferring Gaussian process models from previous data

In contrast to the previously explained models, no Gaussian process model is chosen a priori. The following approach aims to weaken the assumption about the function family by inferring an appropriate Gaussian process model from previous data. Instead of implicitly making assumptions about the system output by choosing a mean and a covariance function, we explicitly specify which (basis) functions the system output consists of. In doing so, we assume that each realization $f_j(x)$ can be approximated using a linear combination of a finite set of basis functions $\phi_k(x)$ with $k = 1, ..., p$ so that

$$f_j(x) = \sum_{k=1}^{p} \phi_k(x)\beta_{k,j} = \boldsymbol{\phi}(x)^\top \boldsymbol{\beta}_j, \tag{4.14}$$

where $\beta_{k,j}$ is the coefficient of realization j belonging to the basis function $\phi_k(x)$. Then, the mean function reads

$$\begin{aligned} m(x) =& \mathbb{E}[f(x)] \\ =& \mathbb{E}[\boldsymbol{\phi}(x)^\top \boldsymbol{\beta}] \\ =& \boldsymbol{\phi}(x)^\top \mathbb{E}[\boldsymbol{\beta}] \\ =& \boldsymbol{\phi}(x)^\top \boldsymbol{\mu}_\beta \\ \approx& \boldsymbol{\phi}(x)^\top \hat{\boldsymbol{\mu}}_\beta. \end{aligned} \tag{4.15}$$

and the covariance function is

$$\begin{aligned} k(x, x') =& \mathrm{cov}[f(x), f(x')] = \mathbb{E}[(f(x) - \mathbb{E}[f(x)])(f(x') - \mathbb{E}[f(x')])] \\ =& \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))] \\ =& \mathbb{E}[f(x)f(x') - f(x)m(x') - m(x)f(x') + m(x)m(x')] \\ =& \mathbb{E}[f(x)f(x')] - \mathbb{E}[f(x)m(x')] - \mathbb{E}[m(x)f(x')] + \mathbb{E}[m(x)m(x')] \\ =& \mathbb{E}[\boldsymbol{\phi}(x)^\top \boldsymbol{\beta}\, \boldsymbol{\phi}(x')^\top \boldsymbol{\beta}] - \mathbb{E}[\boldsymbol{\phi}(x)^\top \boldsymbol{\beta}\, \boldsymbol{\phi}(x')^\top \mathbb{E}[\boldsymbol{\beta}]] - \mathbb{E}[\boldsymbol{\phi}(x)^\top \mathbb{E}[\boldsymbol{\beta}]\, \boldsymbol{\phi}(x')^\top \boldsymbol{\beta}] \\ & + \mathbb{E}[\boldsymbol{\phi}(x)^\top \mathbb{E}[\boldsymbol{\beta}]\, \boldsymbol{\phi}(x')^\top \mathbb{E}[\boldsymbol{\beta}]] \\ =& \mathbb{E}[\boldsymbol{\phi}(x)^\top \boldsymbol{\beta}\, \boldsymbol{\beta}^\top \boldsymbol{\phi}(x')] - \mathbb{E}[\boldsymbol{\phi}(x)^\top \boldsymbol{\beta}\, \mathbb{E}[\boldsymbol{\beta}^\top] \boldsymbol{\phi}(x')] - \mathbb{E}[\boldsymbol{\phi}(x)^\top \mathbb{E}[\boldsymbol{\beta}]\, \boldsymbol{\beta}^\top \boldsymbol{\phi}(x')] \\ & + \mathbb{E}[\boldsymbol{\phi}(x)^\top \mathbb{E}[\boldsymbol{\beta}]\, \mathbb{E}[\boldsymbol{\beta}^\top] \boldsymbol{\phi}(x')] \\ =& \boldsymbol{\phi}(x)^\top \left(\mathbb{E}[\boldsymbol{\beta}\, \boldsymbol{\beta}^\top] - \mathbb{E}[\boldsymbol{\beta}]\, \mathbb{E}[\boldsymbol{\beta}^\top] - \mathbb{E}[\boldsymbol{\beta}]\, \mathbb{E}[\boldsymbol{\beta}^\top] + \mathbb{E}[\boldsymbol{\beta}]\, \mathbb{E}[\boldsymbol{\beta}^\top] \right) \boldsymbol{\phi}(x') \\ =& \boldsymbol{\phi}(x)^\top \left(\mathbb{E}[\boldsymbol{\beta}\, \boldsymbol{\beta}^\top] - \mathbb{E}[\boldsymbol{\beta}]\, \mathbb{E}[\boldsymbol{\beta}^\top] \right) \boldsymbol{\phi}(x') \\ =& \boldsymbol{\phi}(x)^\top \mathrm{cov}[\boldsymbol{\beta}] \boldsymbol{\phi}(x') \\ =& \boldsymbol{\phi}(x)^\top \boldsymbol{\Sigma}_\beta \boldsymbol{\phi}(x') \\ \approx& \boldsymbol{\phi}(x)^\top \hat{\boldsymbol{\Sigma}}_\beta \boldsymbol{\phi}(x'), \end{aligned} \tag{4.16}$$

where $\boldsymbol{\phi}(x)$ is a vector consisting of all basis functions

$$\boldsymbol{\phi}(x) = \left[\phi_1(x), \ldots, \phi_p(x)\right]^\top, \tag{4.17}$$

$\hat{\boldsymbol{\mu}}_\beta$ is the sample mean vector

$$\hat{\boldsymbol{\mu}}_\beta = \frac{1}{m} \sum_{j=1}^{m} \boldsymbol{\beta}_j = \left[\hat{\mu}_{\beta_1}, \ldots, \hat{\mu}_{\beta_p}\right]^\top, \tag{4.18}$$

and $\hat{\boldsymbol{\Sigma}}_\beta$ is the sample covariance matrix

$$\hat{\boldsymbol{\Sigma}}_\beta = \begin{bmatrix} \hat{\sigma}_{1,1} & \cdots & \hat{\sigma}_{1,p} \\ \vdots & \ddots & \vdots \\ \hat{\sigma}_{p,1} & \cdots & \hat{\sigma}_{p,p} \end{bmatrix}, \hat{\sigma}_{k,l} = \frac{1}{m-1} \sum_{j=1}^{m} (\beta_{k,j} - \hat{\mu}_{\beta_k})(\beta_{l,j} - \hat{\mu}_{\beta_l}) \tag{4.19}$$

of the coefficients $\boldsymbol{B} = [\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_m]$. The approach is similar to assuming a Bayesian linear regression model, the weights' distribution to be multivariate normal (see [116]), and approximating the distribution by the sample mean vector and the sample covariance matrix. Furthermore, this idea is similar to modeling the residuals of a global linear model by a Gaussian process, which was explored as early as 1975 for polynomials by Blight and Ott [18] and for a linear combination of basis functions by O'Hagan [99].

To apply the presented approach, let $\boldsymbol{x}_j \in \mathbb{R}^{n_j}$ be the previously gathered inputs and $\boldsymbol{y}_j \in \mathbb{R}^{n_j}$ the corresponding function values, where trajectory j consists of n_j data points. We first approximate each trajectory $(\boldsymbol{x}_j, \boldsymbol{y}_j)$ using a linear regression with p basis functions, where $\hat{\boldsymbol{\beta}}_j$ are the estimated coefficients of trajectory j with

$$\hat{\boldsymbol{\beta}}_j = (\boldsymbol{\Phi}_j^\top \boldsymbol{\Phi}_j)^{-1} \boldsymbol{\Phi}_j^\top \boldsymbol{y}_j \tag{4.20}$$

and

$$\boldsymbol{\Phi}_j = \boldsymbol{\Phi}(\boldsymbol{x}_j) = \begin{bmatrix} \phi_1(x_{1,j}) & \cdots & \phi_p(x_{1,j}) \\ \vdots & \ddots & \vdots \\ \phi_1(x_{n_j,j}) & \cdots & \phi_p(x_{n_j,j}) \end{bmatrix}, \tag{4.21}$$

where $n_j \geq p$ and $m > p$. If $n_j < p$, $(\boldsymbol{\Phi}_j^\top \boldsymbol{\Phi}_j)$ becomes rank deficient and therefore has no inverse, which causes problems in Equation 4.20. One could use the Moore-Penrose pseudoinverse instead. If $m \leq p$, $\operatorname{rank}(\hat{\boldsymbol{\Sigma}}_{\boldsymbol{a}}) = m - 1$ and therefore the sample covariance matrix has no inverse. This problem could be solved by adding a small perturbation to the main diagonal of the sample covariance matrix so that the sample covariance matrix has full rank.

After fitting every trajectory by a set of basis functions, we are able to compute the sample mean vector $\hat{\boldsymbol{\mu}}_{\hat{\beta}}$ and the sample covariance matrix $\hat{\boldsymbol{\Sigma}}_{\hat{\beta}}$ of the estimated coefficients $\hat{\boldsymbol{B}}$, leading to an approximate mean (Equation 4.15) and covariance function (Equation 4.16). Additionally, the observation error's variance σ_y^2 can be estimated using the mean square of the residuals $\hat{r}$

$$\hat{\boldsymbol{r}}_j = \boldsymbol{\Phi}_j \hat{\boldsymbol{\beta}}_j - \boldsymbol{y}_j \tag{4.22}$$

$$\hat{\sigma}_y^2 = \frac{1}{m} \sum_{j=1}^{m} \frac{1}{n_j - p} \sum_{i=1}^{n_j} \hat{r}_{i,j}^2 \tag{4.23}$$

or by maximizing the sum of the trajectories' log-likelihoods in Equation 4.12 with its derivative

$$\begin{aligned} &\frac{\partial \log p(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m | \sigma_y^2)}{\partial \sigma_y^2} = \\ &\sum_{j=1}^{m} \left(-\frac{1}{2} \operatorname{tr}(\mathbf{K}^{-1}) + \frac{1}{2} (\boldsymbol{y}_j - m(\boldsymbol{x}_j))^\top \mathbf{K}^{-2} (\boldsymbol{y}_j - m(\boldsymbol{x}_j)) \right), \end{aligned} \tag{4.24}$$

where the covariance matrix $\mathbf{K} = k_{IGP}(\boldsymbol{x}_j, \boldsymbol{x}_j) + \sigma_y^2 \boldsymbol{I}$. The denominator $n_j - p$ in Equation 4.23 results from the reduced degrees of freedom due to fitting p basis functions to the trajectories.

See [7] for a derivation. The entire inferred model reads

$$\begin{aligned} m(x) &= \boldsymbol{\phi}(x)^\top \hat{\boldsymbol{\mu}}_{\hat{\beta}} \\ k(x,x') &= k_{IGP}(x,x') + \delta\,\hat{\sigma}_y^2 = \boldsymbol{\phi}(x)^\top \hat{\boldsymbol{\Sigma}}_{\hat{\beta}} \boldsymbol{\phi}(x') + \delta\,\hat{\sigma}_y^2. \end{aligned} \tag{4.25}$$

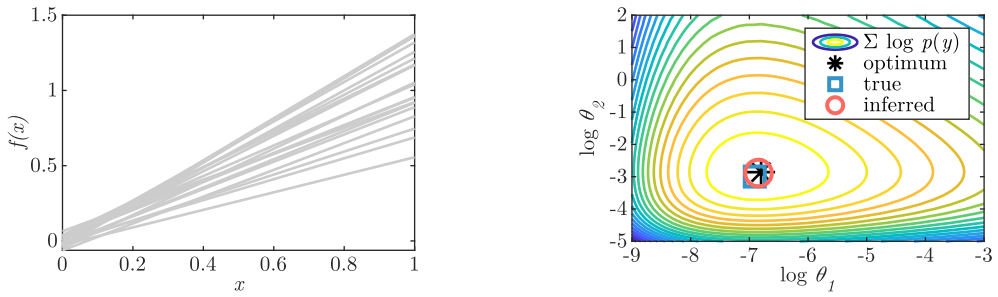
As an example, let us consider the 20 trajectories shown in Figure 4.4a that are realized by the Gaussian process $f(x) \sim \mathcal{N}(m(x), k(x,x'))$ with

$$m(x) = x \tag{4.26}$$

and

$$k(x,x') = \theta_1 + \theta_2 x x', \tag{4.27}$$

where the parameters are set to $\theta_1 = 0.001$ and $\theta_2 = 0.05$ (true values). Based on Equation 4.12, the negative sum of the trajectories' log-likelihoods is minimized to find the parameters of a prescribed Gaussian process model. For this example, we use the same Gaussian



(a) (b)

Figure 4.4: (a) Linear training trajectories and (b) objective function with determined values.

process model defined in Equations 4.26–4.27 with the free parameters θ_1 and θ_2. The contour lines of the objective function are displayed in Figure 4.4b. Furthermore, the objective function's optimum is shown as a black star and the true values $\theta_1 = 0.001$ and $\theta_2 = 0.05$ as a blue square. The true parameter values are off the optimum as the objective function is based on the limited 20 true trajectories. If the number of trajectories goes towards infinity, the realized and optimum parameter values become the same. We can also apply the method of inferring a Gaussian process model from the previous data to the 20 trajectories. For this, we assume a constant and linear basis function, leading to the values shown as a red circle in Figure 4.4b. We can see that this method leads to values close to the optimum of the objective function. The small difference between them results from the mean function, which is also determined by the inferring method but fixed to $m(x) = x$ for the objective function.

The inferring method can also be applied to the crack growth trajectories in Figure 4.1a, for which the multivariate normal distribution was introduced. The inferred mean function with the polynomial basis functions of order four is shown in blue in Figure 4.5. It well matches the computed sample mean of the trajectories displayed as a dashed black line. Figure 4.5 also illustrates that the symmetric 95% credible regions shown in blue for the inferring method and framed in black for the multivariate normal distribution are very similar. In contrast to the multivariate normal distribution approach, the proposed inferring method can deal with previous trajectories that are not equally long or miss data points in between.

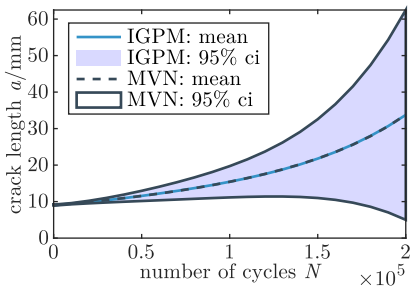


Figure 4.5: Comparison of inferred Gaussian process model (IGPM) with multivariate normal distribution (MVN) for the crack growth problem.

In the following, we apply the proposed Gaussian process methods, i.e., Gaussian processes with (1) current data & a prescribed Gaussian process model, (2) previous data & a prescribed Gaussian process model, and (3) inferred model from previous data, to four data sets in order to show the impact of the different assumptions.

4.4 Effect of integrating prior knowledge

In this section, we investigate the effect of integrating prior knowledge into Gaussian processes. In order to compare the three methods proposed in the previous section, we apply them to published data sets, which represent fatigue crack growth, laser degradation, and milling machine wear. The three data sets are shown in Figure 3.4 and were previously introduced in Section 3.2. The results presented in this section are based on a leave-one-out validation scheme, i.e., each trajectory is used as test data while the other ones are used for training. The highlighted test trajectory in Figure 4.6 is used to visualize the results.

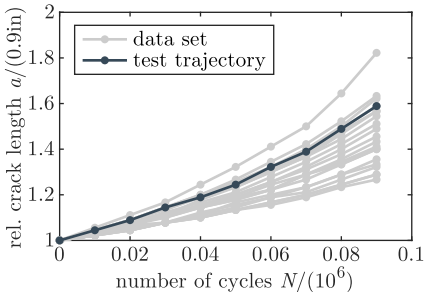


Figure 4.6: Fatigue crack growth trajectories of a compact tension specimen (FCG Hudak). The highlighted line is used to visualize the results.

For comparing the methods proposed, underlying models have to be assumed. As the data sets of Figure 3.4 predominantly follow a polynomial behavior, we assume polynomial mean and covariance functions in the following. Additionally, a zero mean and a squared-exponential covariance function are considered as they have been frequently used by other researchers. In order to find the maximum polynomial order, we split each previous data trajectory into a train ($\approx 70\%$) and a test set ($\approx 30\%$) and evaluate the mean squared error for the test set. The smallest test set error is obtained with a maximum order of $q = 1$ for the laser degradation

and the milling machine wear data sets, and an order of $q = 2$ for the FCG Hudak data set. Table 4.2 summarizes the models explained in the following.

Table 4.2: Investigated Gaussian process models (GPM).

Abbreviation	**Model**	**Mean function**	**Covariance function**	**Data for parameter estimation**	**Data for conditional distribution**
GPM curr.	predefined GPM	zero	polynomial	current data	current data
GPM prev. ZM & SE	predefined GPM	zero	squared-exponential	previous data	current data
GPM prev. POLY	predefined GPM	polynomial	polynomial	previous data	current data
IGPM	inferred GPM	polynomial	polynomial	previous data	current data

For *training Gaussian process models with current data* (GPM curr.), we use only the currently observed data points and no previous trajectories, i.e., each trajectory is predicted without knowing the other ones. For this method, we assume a zero mean and a polynomial covariance function with the above determined maximum polynomial order q

$$k_\theta(x, x') = \sigma_f^2(xx' + b)^q + \delta\,\sigma_y^2, \tag{4.28}$$

where σ_f, b, and σ_y are the parameters $\boldsymbol{\theta}$ of the Gaussian process model. These parameters $\boldsymbol{\theta}$ are trained for every new data point by maximizing the log-likelihood. We found that using a polynomial mean function in this particular setup decreases the accuracy, which results from too many parameters for a relatively small quantity of data. Moreover, the optimization starting point strongly influences the result. Therefore, it is useful to start the optimization with a set of different starting points. To be able to compare the results to the other methods, we use the parameters optimized to the previous data as the starting point for this method.

For *training Gaussian process models with previous data* (GPM prev.), we assume all trajectories but the currently investigated one to be previously observed. Due to the larger amount of data, the assumed Gaussian process model consists of a polynomial covariance function, see Equation 4.28, and a polynomial mean function

$$m_\theta(x) = \sum_{k=1}^{q+1} c_k x^{k-1}, \tag{4.29}$$

where σ_f, b, and c_k are the parameters $\boldsymbol{\theta}$ of the Gaussian process model (GPM prev. POLY). Additionally, a Gaussian process model with a zero mean and a squared-exponential function, see Equation 4.3, is trained with previous data (GPM prev. ZM & SE) in order to investigate the influence of the prescribed model. As explained in Section 4.3, we maximize the sum of the trajectories' log-likelihood for optimizing the parameters $\boldsymbol{\theta}$. These parameters are fixed for the entire prediction series. For this method, the currently observed data acts as a conditional on the Gaussian process, leading to an updated prediction.

For *inferring Gaussian process models* (IGPM), we again assume all trajectories but the currently investigated one to be previously observed. In order to determine the Gaussian process model, first, each previous trajectory is approximated by a linear regression of polynomial basis functions with the maximum order q, see Equation 4.20. Second, the estimated coefficients are used to determine the mean and the covariance function, see Equations 4.15–4.16. Additionally, the observation error's standard deviation is estimated with Equation 4.23. As for the Gaussian process model trained with previous data, the resulting mean function,

covariance function, and the observation error's standard deviation are fixed for the entire prediction series. After observing a current data point, the conditional distribution of the Gaussian process is computed, leading to an updated prediction, see Equation 4.9.

4.4.1 Results

The probabilistic predictions for the FCG Hudak data are shown in Figure 4.7. The figure visualizes the true trajectory (red), the mean prediction (blue), and the symmetric 95% credible region (light blue area), i.e., the region between the 2.5% and the 97.5% quantiles. Figure 4.7 compares the prediction of the Gaussian process model, which relies only on current data, with the Gaussian process models trained with previous data (GPM prev. ZM & SE and GPM prev. POLY), and with the inferred model. Two different time states of the currently observed trajectory are visualized.

Table 4.3: Errors for different methods with respect to the corresponding data set.

Data set	Model	MAE		MAPE		MAE half		MAPE half		Prediction time		Training time	
FCG Hudak predict: $a(N_{end})$	IGPM	0.04		0.03		0.02		0.01		0.003 s		0.16 s	
	GPM curr.	0.13	+208.4%	0.09	+207.2%	0.05	+163.6%	0.03	+162.1%	0.91 s	+26,797.1%	0.00 s	−100.0%
	GPM prev. ZM & SE	0.20	+381.9%	0.13	+383.8%	0.07	+237.9%	0.04	+237.1%	0.01 s	+261.8%	1.46 s	+786.7%
	GPM prev. POLY	0.05	+14.2%	0.03	+15.9%	0.02	+8.1%	0.01	+9.1%	0.02 s	+352.9%	1.78 s	+982.8%
Laser degradation predict: $I(t_{end})$	IGPM	0.67 A		0.09		0.38 A		0.05		0.004 s		0.18 s	
	GPM curr.	1.15 A	+71.4%	0.15	+68.2%	0.38 A	−1.6%	0.05	−1.2%	2.45 s	+59,624.4%	0.00 s	−100.0%
	GPM prev. ZM & SE	1.38 A	+105.4%	0.17	+96.9%	0.49 A	+28.3%	0.06	+20.4%	0.02 s	+373.2%	1.00 s	+465.1%
	GPM prev. POLY	0.68 A	+0.5%	0.09	+0.1%	0.38 A	−1.2%	0.05	−1.0%	0.02 s	+490.2%	2.72 s	+1,441.1%
Milling machine wear predict: $VB(N_{end,j})$	IGPM	0.19 mm		0.27		0.13 mm		0.17		0.0009 s		0.17 s	
	GPM curr.	0.26 mm	+35.3%	0.35	+28.8%	0.14 mm	+7.8%	0.18	+11.7%	4.07 s	+465,892.9%	0.00 s	−100.0%
	GPM prev. ZM & SE	0.29 mm	+53.6%	0.41	+53.6%	0.15 mm	+12.8%	0.20	+20.9%	0.01 s	+1,422.4%	1.04 s	+496.0%
	GPM prev. POLY	0.19 mm	0.0%	0.27	+1.9%	0.13 mm	−3.1%	0.16	−5.0%	0.02 s	+1,708.5%	1.57 s	+796.5%

Common to all methods is that we obtain an updated prediction after observing a new current data point, leading to n_j predictions for each trajectory. The models are evaluated in a leave-one-out validation scheme, i.e., each trajectory is used as test data while the other ones are used for training. In this study, we evaluate the mean absolute error (MAE) and the mean absolute percentage error (MAPE) of the last predicted point for one trajectory from prediction 1 to prediction $n_j - 1$ and compute the average of all investigated trajectories. In order to get an idea of how the accuracy changes with an increasing number of current data points, we also evaluate the MAE and MAPE starting from prediction $\lceil n_j/2 \rceil$. These are referred to as MAE *half* and MAPE *half*. The evaluated errors for the three data sets are presented in Table 4.3. Moreover, the average prediction time for one series, i.e., the total time for all $n_j - 1$ predictions of one trajectory including parameter optimization on current data, as well as the training time for the Gaussian process model, i.e., the time for inferring or optimizing the used Gaussian process model on previous data, are listed in Table 4.3.

Additionally, we evaluate how often the last measurement value lies within a certain predicted credible interval. This opens up the possibility of quantifying the predicted uncertainty since the relative frequency of the real value lying within the credible interval should correspond to the assumed credible interval itself. The symmetric 95% credible interval, for example, is the interval between the 2.5% and the 97.5% quantile and should correspond to a relative frequency of 95%. The results are plotted in Figures 4.8a and 4.8b for four symmetric credible intervals $\{50\%, 90\%, 95\%, 99\%\}$ where the inferred Gaussian process model is compared to the model trained with current data and to the polynomial model trained with previous data, respectively. A value above the black dashed line indicates an overly wide credible interval,

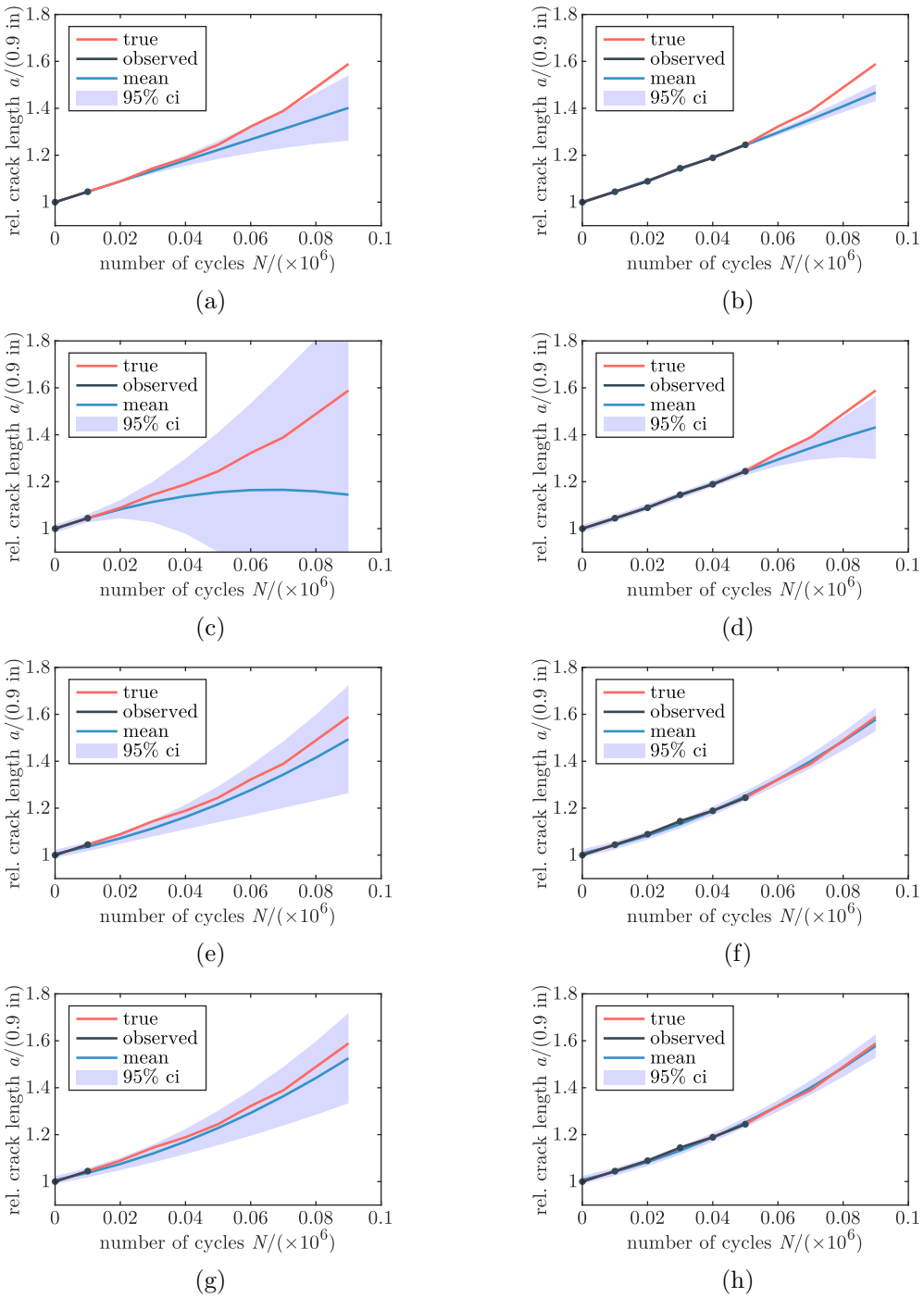
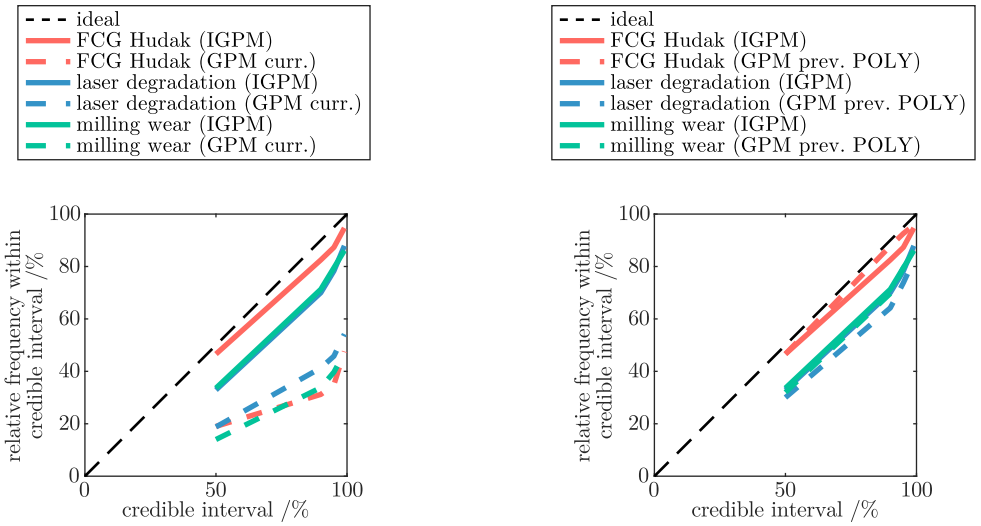


Figure 4.7: Probabilistic predictions of crack growth (FCG Hudak) at two different time states. (a) GPM curr.: Gaussian process model trained with current data for an early and (b) a late time state. (c) GPM prev. ZM & SE: Zero mean and squared-exponential function trained with previous data for an early and (d) a late time state. (e) GPM prev. POLY: Polynomial Gaussian process model trained with previous data for an early and (f) a late time state. (g) IGPM: Inferred Gaussian process model for an early and (h) a late time state.



(a) (b)

Figure 4.8: Credible interval comparison of inferred Gaussian process model (IGPM) with (a) the Gaussian process model trained with current data (GPM curr.), and with (b) the polynomial Gaussian process model trained with previous data (GPM prev. POLY).

which can be viewed as a too-conservative prediction. By contrast, a value below the dashed line results from a too-small credible interval, which can be viewed as an overly optimistic prediction.

4.4.2 Discussion

By comparing the inferred model to the Gaussian process model trained with current data, we see that the inferring method results in higher accuracy, in particular at the beginning (see Table 4.3). This results from using previous data since the inferred Gaussian process model gains information about the shape of typical trajectories. This information is embedded in the mean function. Furthermore, the information about how different locations are correlated and distributed is embedded in the covariance function. The error margin decreases towards the end of a prediction series (see MAE half and MAPE half) because of the larger amount of current data, enabling better parameter training. At the beginning of the prediction series, the ratio of the number of parameters to current data is high, and the optimization problem is prone to overfit, whereas the ratio decreases towards the end. Moreover, Figures 4.7a and 4.7b show that training a Gaussian process model only with current data leads to a shorter look-ahead time, whereas the inferred model is able to accurately predict even points that are far away, see Figures 4.7g and 4.7h. On the other hand, for the laser degradation data set, the MAE half and MAPE half of the inferred model are larger than the ones based on the Gaussian process model trained with current data (see Table 4.3). This is a result of the estimated standard deviation of the observation error, which is flexible within a prediction series of one trajectory for the model trained with current data. Another reason for this behavior is that the laser degradation data set represents a relatively simple degradation behavior that is predominantly linear. For more complex data sets, e.g., FCG Hudak and milling machine wear, the MAE half

and MAPE half are larger.

Furthermore, the Gaussian process model trained with current data tends to underestimate uncertainties. As Figure 4.8a shows, the colored dashed lines (GPM curr.) are further away from the black dashed line than the solid lines (IGPM), resulting from a variance that was underestimated, a wrong mean, or both. Again, this is due to the lack of previous data. There is no prior information about how large the scatter is, resulting in wrong assumptions/estimations of the Gaussian process model's parameters. By contrast, this information is integrated into the inferred covariance function.

By comparing the prediction times, we see that the inferring method obtains results more quickly. This is because only the conditional distribution is computed for new current observations. By contrast, for the Gaussian process model trained with current data, an optimization problem has to be solved every time a new current data point is observed. The training time for the inferred model is larger because no previous selection process is carried out for the Gaussian process model trained with current data.

The results of the inferred Gaussian process model and the polynomial model trained with previous data are very similar since their determined mean functions are identical, and their covariance functions are almost the same. The inferred model tends to have slightly higher accuracy, see Figures 4.7e–4.7f and Table 4.3, because the model captures the correlation between the coefficients. This improves accuracy if, for example, the coefficient of order 1 is correlated with the coefficient of order 2. By writing the covariance functions for $q = 2$ explicitly, the difference can be identified:

$$
\begin{aligned}
k_{prev}(x, x') = \quad & \sigma_f^2\, b^2 && +2\,\sigma_f^2\, b\, x x' && +\sigma_f^2\, (x x')^2 \\
k_{IGPM}(x, x') = \quad & (\mathbb{E}[\beta_0^2] - \mathbb{E}[\beta_0]^2) && +(\mathbb{E}[\beta_1^2] - \mathbb{E}[\beta_1]^2)\, x x' && +(\mathbb{E}[\beta_2^2] - \mathbb{E}[\beta_2]^2)\, (x x')^2 \\
& +(\mathbb{E}[\beta_1\, \beta_0] - \mathbb{E}[\beta_1]\, \mathbb{E}[\beta_0])\, (x + x') \\
& +(\mathbb{E}[\beta_2\, \beta_0] - \mathbb{E}[\beta_2]\, \mathbb{E}[\beta_0])\, (x^2 + x'^2) \\
& +(\mathbb{E}[\beta_2\, \beta_1] - \mathbb{E}[\beta_2]\, \mathbb{E}[\beta_1])\, (x^2 x' + x'^2 x)
\end{aligned}
$$

Of course, parameters could be added to the modeled covariance function in order to capture the correlation. This, however, would lead to a more complex optimization problem due to the possibility of including other local optima. Ultimately, the predicted mean and variance are also almost the same, see Figure 4.8b, as the mean and the covariance functions are similar.

Additionally, a prescribed zero mean and a squared-exponential covariance function are trained with previous data. Due to the poorly chosen Gaussian process model, the accuracy is significantly worse than the accuracy of the inferred model. As the mean function is set to zero, the mean prediction converges to zero for points that are far away from the currently measured data, see Figure 4.7c. For close points, the error is smaller, see Table 4.3 and Figure 4.7d. Consequently, it can be derived that the chosen prescribed Gaussian process model has a significant influence on the accuracy.

We see that the inferring method is faster than training a predefined model by looking at the training time. This is due to the fact that, in general, optimizing the parameters of a Gaussian process model is a non-convex optimization problem [116]. By contrast, the coefficients of a linear combination of basis functions can be estimated by solving a linear system of equations for the inferring method. Furthermore, we found that the starting point of the optimization problem associated with choosing a prescribed Gaussian process model significantly influences the results. For a couple of runs, the optimized parameters lead to a

completely different covariance function, resulting in poor predictions. Since this optimization problem is, in general, non-convex, a gradient-based optimizer can lead to local optima.

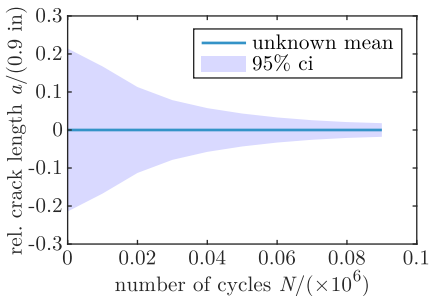


Figure 4.9: Predicted symmetric 95% credible interval of the crack length at 90,000 cycles. The x-axis indicates at which time the prediction is made. The shown credible interval is known before any current data point is observed. Yet, the predicted mean of the crack length is unknown as it depends on the observed function values.

Moreover, we recognized an interesting property of Gaussian processes with fixed parameters, i.e., all free parameters including the standard deviation of the observation error are determined: the variance of the conditional distribution is known over the entire range of x once the Gaussian process model, the observation error's standard deviation, and the measurement locations are determined. This is due to the fact that the (co-)variance of a conditioned Gaussian distribution is independent of the conditioning values, see Equation 4.9. Therefore, we know for every point in time with what certainty we will be able to predict the system output before we have observed any current data point. This is not possible if the Gaussian process model is re-optimized for every new current data point because this changes the model's parameters. Figure 4.9 shows the symmetric 95% credible interval of the crack length at 90,000 cycles predicted at different times. The x-axis shows the point in time up to which current data is measured. The credible interval at 90,000 cycles shrinks as more and more data is collected. This figure can be created before any data point is measured since the variance of the conditional distribution depends only on the measurement locations, i.e., the number of cycles at which new data is observed. Yet, the mean depends also on the measured function values, which is why it is not known beforehand.

4.5 Physics-informed basis functions

For the inferring method, see Equations 4.15 and 4.16, we can also choose other types of basis functions. In this section, we apply physics-informed basis functions to a set of published fatigue crack growth data and examine the effect of using such basis functions. The measurements of the investigated data set were performed by Virkler in 1977 [148] and consists of 68 crack growth experiments. A detailed description of the experiment was given in Chapter 3.2. The first 47 trajectories of this data set are shown in Figure 4.10 and used for inferring the Gaussian process model. The highlighted test trajectory in Figure 4.10 is used to visualize the results.

The crack growth rate can be modeled by Paris' law [101] $\frac{da}{dN} = C(\Delta K_I)^\alpha$, and according to [8], the SIF range ΔK_I for a center-cracked metal sheet with a finite width W can be

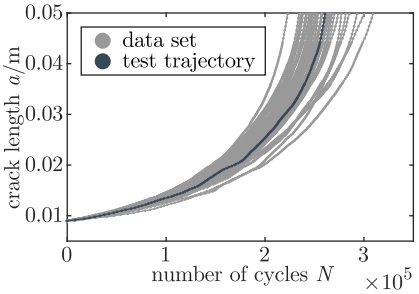


Figure 4.10: Utilized fatigue crack growth data published by Virkler. The highlighted line is used to visualize the results.

computed as

$$\Delta K_I = \frac{\Delta\sigma_\infty\sqrt{\pi a}}{\sqrt{\cos\left(\frac{\pi a}{W}\right)}}. \tag{4.30}$$

Assuming that C and α are independent of a, with the initial condition $N_0 = 0$, the differential equation can be solved for the number of cycles N as

$$N(a) = \frac{1}{C\,\Delta\sigma_\infty^\alpha\,\pi^{\alpha/2}} \int_{a_0}^{a} \left(\frac{\cos\left(\frac{\pi\bar{a}}{W}\right)}{\bar{a}}\right)^{\alpha/2} d\bar{a}. \tag{4.31}$$

We can now use $\phi \equiv N$ as our basis function. The total width $W = 152.4 \times 10^{-3}\,\mathrm{m}$ and the initial crack size of $a_0 = 9 \times 10^{-3}\,\mathrm{m}$ are set according to the experiment, and the exponent is set to $\alpha = 2.9$ according to [153]. The parameter C is not important since we compute a multiplicative coefficient for each trajectory, see Equation 4.20. However, in order to avoid small values for the coefficients, we set the material parameter to $C = 8.7096 \times 10^{-11}$. The trajectories 1-47 are used to infer the Gaussian process model. The factor σ_x of the observation error's standard deviation

$$\sigma_y(x) = \sigma_x \left|\frac{dm(x)}{dx}\right| \tag{4.32}$$

is optimized by maximizing the log-likelihood of the last measured point.

In general, the exponent α might not be the same for all trajectories. To take this into account, we can set several values for α, which results in multiple basis functions

$$\boldsymbol{\phi}(x) = \left[\phi_1(x, \alpha = \alpha_1), \ldots, \phi_p(x, \alpha = \alpha_p)\right]^\top. \tag{4.33}$$

In this study, we apply the proposed method with $\alpha = 2.9$ and compare it to a polynomial basis function of order $q = 4$ and to the physics-informed basis functions with $\alpha = \{2.6, 2.8, 3.0, 3.2\}$. Actually, from a mathematical point of perspective, it would be necessary to prove linear independence such that the corresponding regression system matrix $\boldsymbol{\Phi}_j^\top \boldsymbol{\Phi}_j$ is invertible, see Equation 4.20.

4.5.1 Results

As in Section 4.3, we evaluate the MAE and the MAPE of the last predicted point. The results are listed in Table 4.4.

Table 4.4: Errors for different basis functions with respect to Virkler's crack growth data set.

Data set	Model	MAE		MAPE		MAE half		MAPE half	
FCG Virkler predict: $N(a_{end})$	Physics-informed $\alpha = 2.9$	7525		0.03		3128		0.01	
	Polynomial $p = 4$	7448	-1.0%	0.03	-2.1%	4578	$+46.3\%$	0.02	$+50.4\%$
	Physics-informed $\alpha = \{2.6, 2.8, 3.0, 3.2\}$	5393	-28.3%	0.02	-28.3%	2737	-12.5%	0.01	-10.9%

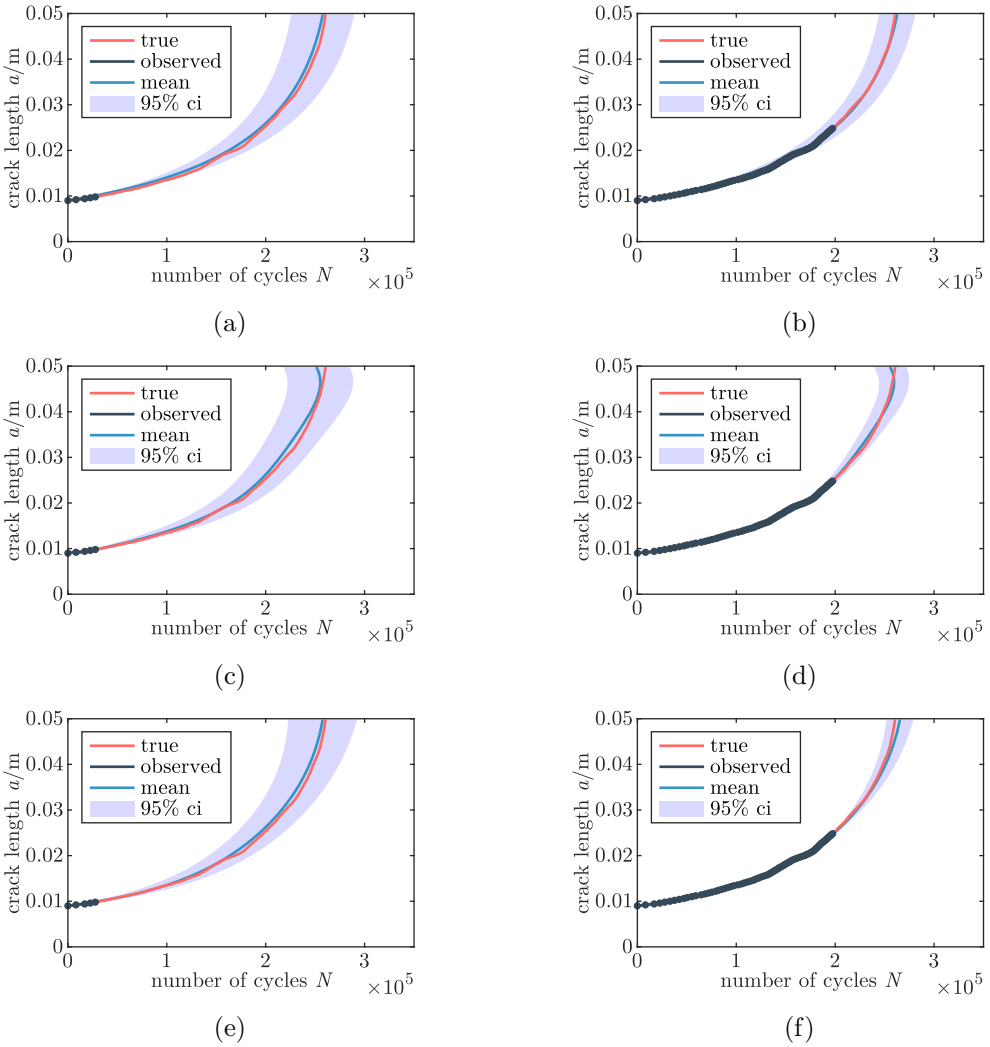


Figure 4.11: Probabilistic predictions of crack growth with different basis functions at two different time states. Note that $x = a$ and $f \equiv N$ are flipped. (a) Physics-informed mean and covariance function with $\alpha = 2.9$ for an early and (b) a late time state. (c) Polynomial mean and covariance function with $q = 4$ for an early and (d) a late time state. (e) Physics-informed mean and covariance function with $\alpha = \{2.6, 2.8, 3.0, 3.2\}$ for an early and (f) a late time state.

Additionally, the probabilistic predictions for the three different basis functions are shown in Figure 4.11. The figure depicts the true trajectory (red), the mean prediction (blue), and the symmetric 95% credible region (light blue area). Two different time states of the currently

observed trajectory are shown.

4.5.2 Discussion

Figure 4.11 shows that using different basis functions influences the predictions, see the curvature of the mean towards the end in Figures 4.11c and 4.11d. The figure depicts that the polynomial basis functions do not represent the crack growth trajectory as closely as the physics-informed basis functions. This effect is particularly apparent towards the end of a prediction series, see MAE half and MAPE half in Table 4.4. The reasons for this are that the polynomial covariance function (Figure 4.12b) is significantly different from the computed covariance matrix (Figure 4.12a) in the regions of larger crack lengths and the polynomial mean function does not represent the data well.

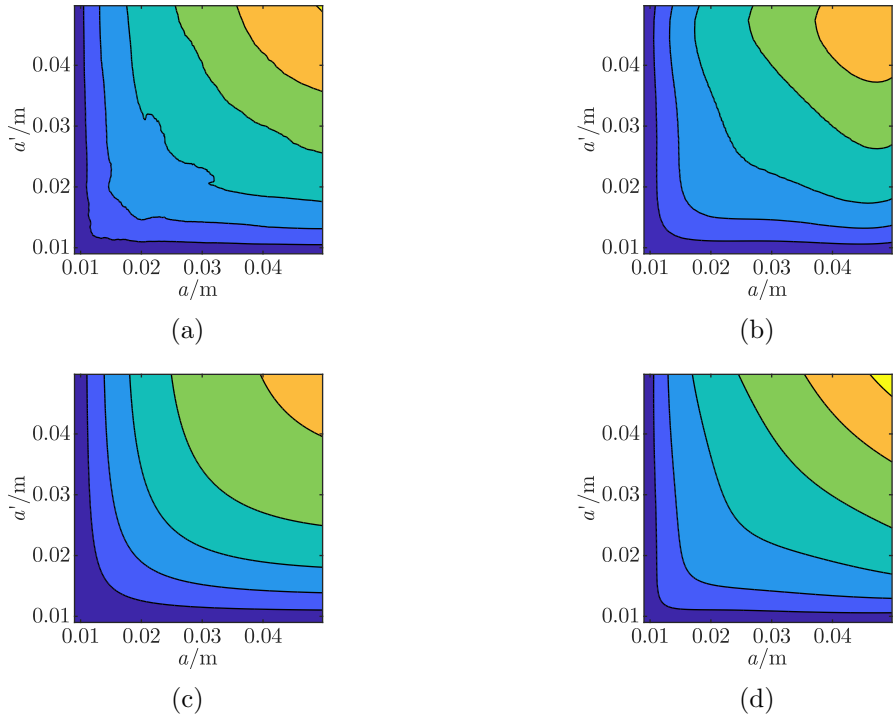


Figure 4.12: Comparison of (a) computed covariance matrix, (b) polynomial covariance function with $q = 4$, (c) physics-informed covariance function with $\alpha = 2.9$, and (d) physics-informed covariance function with $\alpha = \{2.6, 2.8, 3.0, 3.2\}$.

The effect of using several exponents can be observed by comparing Figures 4.11a and 4.11b to Figures 4.11e and 4.11f. Towards the end, the credible region predicted with several basis functions $\alpha = \{2.6, 2.8, 3.0, 3.2\}$ becomes wider than the credible region predicted with only one $\alpha = 2.9$. Moreover, the optimized standard error of the observation error is smaller for $\alpha = \{2.6, 2.8, 3.0, 3.2\}$. Using additional exponents further improves prediction accuracy, see Table 4.4. The reason for this might be that not only C but also the material parameter α scatters over different trajectories. The effect of using several values for α can also be seen in Figure 4.12: the covariance function shown in Figure 4.12d is closer to the computed covariance matrix (Figure 4.12a) than the covariance function for $\alpha = 2.9$ in Figure 4.12c.

Other physics-based models can also be used for predicting crack growth. However, these are restricted to data sets that have problem-specific governing equations. The method presented in this chapter can treat data sets with and without problem-specific equations.

4.6 Summary

In this chapter, we first introduced how multivariate normal distributions can be used for prognostics. By first computing the sample mean vector and the sample covariance matrix and then conditioning the normal distribution on current data, we can predict the entire degradation trajectory and update the prediction based on currently observed data as shown on a crack growth example. The model not only outputs the mean degradation behavior but also credible intervals, estimating the prediction's uncertainty. However, to use this method, all previously observed trajectories must be equally long and measured at the same locations to compute the sample mean vector and the sample covariance matrix. Moreover, predictions can only be made at locations where previous data points were collected.

In order to solve these drawbacks, the chapter further presents two methods (GPM prev. & IGPM) to integrate previous data into Gaussian processes. The methods are applied to three data sets representing fatigue crack growth, laser degradation, and milling machine wear. The results show that integrating prior knowledge into Gaussian processes improves prediction accuracy significantly.

One way of incorporating previous data into Gaussian processes is to train a predefined Gaussian process model by maximizing the sum of the trajectory's log-likelihoods. However, optimization tends to be numerically expensive, and due to generally missing convexity, the result may not be globally optimal. By contrast, following the inferring method, only a linear equation system has to be solved which reduces the computation effort significantly. Additionally, the latter method allows the user to specify which basis functions best represent the system output rather than implicitly making assumptions about it by choosing a mean and a covariance function. Inferring Gaussian process models thus bypasses the challenge of selecting predefined mean and covariance functions and opens the way to physics-informed Gaussian processes, which further increases accuracy, see Section 4.5.

Additionally, we recognized a useful property for using models with fixed parameters in the context of prognostics and health management: the variance of the conditional distribution is known over the entire range of x once the Gaussian process model, the observation error's standard deviation, and the measurement locations, e.g., number of cycles, are determined, see Section 4.4.

In summary, the results of this chapter show that (1) taking previous data into account increases accuracy and look-ahead time significantly, (2) deriving the mean and the covariance function by using basis functions decreases computational effort remarkably, and (3) using physics-informed basis functions further increases accuracy.

Chapter 5

Comparison with other Machine Learning Methods

"All models are wrong, but some are useful."

– George Box

The previous chapter introduced a new approach for deriving a Gaussian process from previous data in order to predict the degradation of mechanical systems. As indicated in the introduction of this thesis, there exist various other machine learning methods that can be useful to predict structural degradation. Most of the time, however, they do not provide any information about their predictions' uncertainties. Therefore, this chapter presents different machine learning methods and how to estimate their prediction's uncertainty. This study was conducted together with my master's student Jose Ignacio Rios, who coded the first attempts of the recurrent neural network and the support vector machine methods. The work was further supervised and discussed with Prof. Dr. Horst Baier, the former director of the Institute of Lightweight Design at TUM, and my supervisor Prof. Dr. Markus Zimmermann.

The chapter begins by motivating the study, giving an overview of different machine learning methods, and introducing the data sets used. Afterwards, we present the k-nearest neighbors algorithm and explain how it can be applied to predict the degradation of mechanical systems. Section 5.3 explains support vector machines and an extension to estimate their prediction's uncertainty. The use of recurrent neural networks for prognostics and their implementation for probabilistic predictions are presented in Section 5.4. This is followed by a summary of the applied multivariate normal distribution and the inferred Gaussian process methods. Section 5.6 summarizes the results of the conducted study and discusses them. Lastly, conclusions are drawn from the presented work.

5.1 Motivation

Many researchers have recently proposed new types of models for predicting fatigue life due to the recent advances in machine learning, surrogate modeling, and uncertainty quantification research [131]. Research particularly focused on using machine learning methods, such as k-nearest neighbors [28], support vector machines [27, 145, 146, 147] and artificial neural networks, which have been used frequently in the context of prognostics and health management.

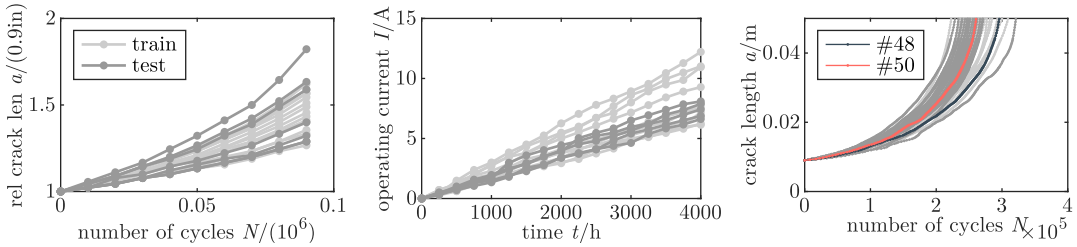
Some of the publications related to prognostics and health management use similarity approaches, such as k-nearest neighbors, to diagnose and predict the health states of mechanical systems. The k-nearest neighbors algorithm, which is based on a distance measure between existing data points, can be used for classification and regression problems. The method has been especially popular for damage detection and classification. For example, He et al. [53] used k-nearest neighbors to identify different types of faults in plastic bearings. Also, Safizadeh and Latifi [122] and Tian et al. [138] utilized k-nearest neighbors to identify the condition of ball bearings based on vibration signals. In [60, 61], a k-nearest neighbors classifier was used to characterize various fault conditions of rolling bearings after obtaining low-dimensional features based on Marginal Fisher analysis. Van and Kang [143] also present a method on feature selection before they diagnose faults in rolling element bearings based on a k-nearest neighbors classifier. In order to take into account correlations between features, the Mahalanobis distance can be used as shown in [160] for rolling element bearing fault diagnosis. The k-nearest neighbors algorithm is also used for other applications. For example, Gharavian et al. [41] used it to detect faults in an automotive multi-speed gearbox and Li et al. [78], Wang [150] and Vanraj et al. [144] to identify fault patterns in experimental gearboxes. Moreover, Park et al. [102] classified spall and crack faults in gear teeth with k-nearest neighbors and Glowacz and Glowacz [42] different stator faults of a single-phase induction motor. In our work, however, we predict the degradation of mechanical systems with k-nearest neighbors and compute credible intervals for the predicted states based on the variance of the k closest neighbors.

Since recurrent neural networks are explicitly suitable for dealing with time-series data, they are widely used for prognostics [74]. For example, Zemouri et al. [164] and Malhi et al. [86] used recurrent neural networks for predicting sensor signal evolution and the degradation process of bearings, respectively. Heimes [54] and Peng et al. [103] ranked second and 22$^{\text{nd}}$ in the PHM08 challenge [114] utilized recurrent neural networks to forecast the time to failure of turbines. Additionally, Narendhar Gugulothu et al. [96] and Yuan et al. [162] tried to predict turbine degradation with recurrent neural networks. In [30], the authors predict the fatigue crack growth in structures with recurrent networks. Zhao et al. [165] further used recurrent neural networks for monitoring the tool wear of a milling machine and Liu et al. [80] to detect faults in rolling bearings. However, a drawback of recurrent neural networks is that they do not provide any information about the prediction's uncertainty by default. Therefore, Xie et al. [159] and Ma and Mao [85] combined them with a particle filter to predict the remaining useful life of fuel cells and the crack growth in structures, respectively. This enables predictions with credible intervals, yet, a particle filter requires physical knowledge about the structural system and can thus only be utilized if the physical behavior is known. In the present work, we combine recurrent neural networks with a mixture density network [15] in order to obtain probabilistic predictions without the knowledge of physical behavior.

Moreover, support vector machines have been used in the context of prognostics and health management. According to Lei et al. [75], support vector machines have been extensively used for fault diagnosis of rolling element bearings, gears, engines, and rotor systems. Support vector machines have also been utilized to predict the degradation of mechanical systems. For example, the authors of [12, 22, 31, 81, 135, 155] utilized support vector machines to predict the degradation of bearings and Fumeo et al. [37] to estimate the remaining useful life of railway transportation systems. Benkedjouh et al. [13] utilized support vector machines to assess the wear of machining tools and predict their remaining useful lives. Moreover, Khelif et al. [64] predict the remaining useful life of an aero-propulsion system with support

vector machines, and in [134], the authors use XFEM and support vector machines to predict the fatigue crack growth. One of the major disadvantages of support vector machines is that they provide only point predictions rather than predictive distributions [17]. Therefore, Tipping [140] introduced relevance vector machines, a probabilistic model whose function form is identical to support vector machines. Widodo and Yang [154] utilized relevance vector machines, for example, to assess the degradation of bearings and Wang et al. [152] to predict the degradation of turbine engines and cooling fans. There exist also other publications and methods that enable support vector regression to predict states with credible intervals. Authors introduced, for example, probabilistic predictions for the degradation of a methane compressor [98], bearings [84], and reactor coolant pumps [82, 83]. In the present study, we also apply support vector machines to predict the degradation of mechanical systems. In order to estimate the prediction's uncertainty, we utilize a k-nearest neighbors estimator as suggested by Quicit [113]. Moreover, we compare the results to those determined with relevant vector machines.

The machine learning approaches based on k-nearest neighbors, recurrent neural networks, and support vector machines are compared to the multivariate normal distribution and Gaussian process approaches proposed in the previous chapter. The presented study is carried out in order to quantify and compare the models' accuracies and to find the advantages and disadvantages of each method. In particular, we are interested in the performance of the approaches presented in the previous chapter compared to other machine learning methods. All approaches



(a) (b) (c)

Figure 5.1: Training and test trajectories of the investigated (a) FCG Hudak, (b) laser degradation, and (b) FCG Virkler data set.

are applied to the three data sets shown in Figure 5.1, representing fatigue crack growth and laser degradation, which were introduced in Section 3.2. These data sets are chosen because their trajectories end at the same last state. The data sets are split into training ($\approx 70\%$) and test trajectories ($\approx 30\%$), as shown in Figure 5.1. The models are trained to predict the last damage state with credible intervals. The inputs of the models are the damage states and their corresponding measurement locations. We train the models such that they map the damage states of each trajectory to the corresponding last damage state. This means that we predict the last damage state based on the first, the first two, the first three, and so on measured damage states. The models should therefore be able to update their predictions over time. Note that the numbers of cycles are measured according to certain crack length intervals for the data set shown in Figure 5.1c. Therefore, the number of cycles and not the crack length is predicted for this particular data set. Figure 5.1c also shows two highlighted trajectories on which we illustrate our predictions.

5.2 K-nearest neighbors

The k-nearest neighbors algorithm is a rather simple approach that first computes the distances in the input space between all existing (training) data points and the new input. Then, the output is classified by the majority class of the k closest neighbors, i.e., the k training data points with the least distances compared to the input. For regression problems, the output can be determined by taking the mean of the k nearest neighbors. Computing the sample variance of the k nearest neighbors further enables predicting the output with credible intervals and thus estimating the prediction's uncertainty. An advantage of k-nearest neighbors is that the algorithm does not assume any specific property of the underlying function. However, a disadvantage is that the distances between the input data point and all training data points must be computed. If the training set is large, the computational time for predicting a new state can be significant. Moreover, the results strongly depend on the training data set as no underlying function is assumed. If the training set is small, the predictions might lack variety since they are based on the sparse diversity of the training data points.

We denote $(\boldsymbol{x}_j, \boldsymbol{y}_j) \in \mathbb{R}^n$ with $j = 1, ..., m$ as our training data and $(\boldsymbol{x}_{1:i,+}, \boldsymbol{y}_{1:i,+}) \in \mathbb{R}^i$ as the currently observed input data, where $y_{i,j}$ is measured at $x_{i,j}$ and $y_{i,+}$ at $x_{i,+}$. In this study, we want to predict the last damage state of a degradation trajectory based on the already measured states. Therefore, we want to predict $y_{n,+}$ based on $(\boldsymbol{x}_{1:i,+}, \boldsymbol{y}_{1:i,+})$. There exist multiple options to quantify the distance between the input and the training data. We can, for example, evaluate the Euclidean distance for all measured points

$$d_{i,j}^{(all)} = \sqrt{\sum_{l=1}^{i} (y_{l,+} - y_{l,j})^2}, \tag{5.1}$$

for only the last point

$$d_{i,j}^{(1)} = \sqrt{(y_{i,+} - y_{i,j})^2}, \tag{5.2}$$

or, in general, for the last s points

$$d_{i,j}^{(s)} = \sqrt{\sum_{l=i-s+1}^{i} (y_{l,+} - y_{l,j})^2}, \tag{5.3}$$

where $j \in 1, ..., m$ for m training trajectories. Since all trajectories of our data sets are measured at the same locations, we quantify only the distance of the damage states $\boldsymbol{y}$ observed at $\boldsymbol{x}$. Otherwise, we could also take the measurement locations $\boldsymbol{x}$ for the distance measure into account. For dealing with multiple input variables, one can normalize them to avoid overestimating the effect of the variable with the lowest scale. After computing the distances to the m training trajectories, we sort the trajectories according to their distance measures. Then, we take the outputs $y_{n,1}, ..., y_{n,k}$ corresponding to the k smallest distances $d_{i,1}, ..., d_{i,k}$ in order to compute the sample mean

$$\hat{\mu}_{i,+} = \frac{1}{k} \sum_{l=1}^{k} y_{n,l} \tag{5.4}$$

and the sample variance

$$\hat{\sigma}_{i,+}^2 = \frac{1}{k-1} \sum_{l=1}^{k} (y_{n,l} - \hat{\mu}_{i,+})^2 \tag{5.5}$$

for our prediction. Moreover, we can apply different distance measures such as the Manhattan or the Minkowski distance. This study, however, is restricted to the Euclidean distance as we have shown for other cases that simple distance measures are often sufficient, see Pfingstl et al. [111]. Figure 5.2 visualizes the 13 nearest neighbors for two different input sizes.

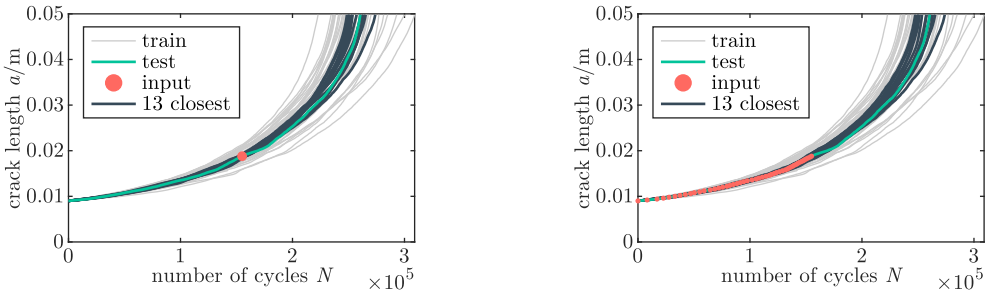


Figure 5.2: 13 nearest neighbors for (a) the last data point and (b) all data points as input.

In order to determine the hyperparameter k, we utilize a leave-one-out validation scheme. This means we predict the last damage state of each trajectory by using the remaining trajectories as the training set. This is repeated for every training trajectory being the one to be predicted. The objective function for the corresponding optimization problem is the average of all predictions' negative log-likelihoods

$$\bar{\mathcal{L}} = -\frac{1}{m}\sum_{j=1}^{m}\frac{1}{n-1}\sum_{i=2}^{n}\log p(y_{n,j}|\boldsymbol{x}_{1:i,j},\boldsymbol{y}_{1:i,j}) \tag{5.6}$$

with

$$p(y_{n,j}|\boldsymbol{x}_{1:i,j},\boldsymbol{y}_{1:i,j}) = \frac{1}{\sqrt{2\pi\hat{\sigma}_{i,j}^2}}\exp\left(-\frac{(y_{n,j}-\hat{\mu}_{i,j})^2}{2\hat{\sigma}_{i,j}^2}\right) \tag{5.7}$$

beginning from two data points since the first data points of all trajectories are the same in our data sets. Computing the distance measure for the first data point would lead for all training trajectories to zero, and the k-nearest neighbors algorithm would fail. Minimizing the negative log-likelihood leads to the optimized design variable $\hat{k}$. Table 5.1 summarizes the results of $\hat{k}$ and $\bar{\mathcal{L}}$ for three different distances considering only the latest, the latest two, and all observed data points.

Table 5.1: Optimized hyperparameter $\hat{k}$ and mean negative log-likelihood $\bar{\mathcal{L}}$ for considering different numbers of input data points. The minimum negative log-likelihood of each data set is written in italics.

	FCG Hudak			**Laser degradation**			**FCG Virkler**		
	last 1	**last 2**	**all**	**last 1**	**last 2**	**all**	**last 1**	**last 2**	**all**
$\hat{k}$	4	5	4	5	4	4	13	13	14
$\bar{\mathcal{L}}$	*−1.3544*	−1.3541	−1.3469	*1.6780*	1.7091	1.7144	*10.5739*	10.5798	10.7605

Table 5.1 shows that the distance based on the latest measured damage state outperforms the other distance measures. One rationale is that the latest measured damage state incorporates the best information about the last damage state. Taking previously measured data points also into account decreases the influence of the latest measured data point, leading to a worse prediction. This is especially true for late predictions. The difference is particularly present for trajectories with many data points like the Virkler data set and almost negligible for trajectories with only a few data points like the Hudak data set.

5.3 Support vector machines

The support vector algorithm is a nonlinear generalization of the generalized portrait algorithm [145] developed in the sixties and improved by Vapnik and Chervonenkis [147]. Support vector machines can be used for both classification and regression analyses. For a classification problem and given the training data $\{(\boldsymbol{x}_i, y_i)|i = 1, \ldots, n; y_i \in \{-1, 1\}\}$, a support vector machine determines a hyperplane such that it maximizes the distances between the different classes. A hyperplane can be defined as the set of points $\boldsymbol{x}$ satisfying

$$\boldsymbol{w}^\top \boldsymbol{x} + b = 0, \tag{5.8}$$

where $\boldsymbol{w}$ is the normal vector to the hyperplane and $\frac{b}{||\boldsymbol{w}||}$ the offset of the hyperplane from the origin along the opposite direction of the normal vector $\boldsymbol{w}$. If the training data set is normalized or standardized and linearly separable, we can define two parallel hyperplanes the following way: The first one

$$\boldsymbol{w}^\top \boldsymbol{x} + b = 1, \tag{5.9}$$

which classifies anything on or above this plane as the category $y = 1$, and the second one

$$\boldsymbol{w}^\top \boldsymbol{x} + b = -1, \tag{5.10}$$

which classifies anything on or below this plane as the category $y = -1$. Therefore, the distance between those two hyperplanes is $\frac{2}{||\boldsymbol{w}||}$. In order to maximize this distance, we have to choose $\boldsymbol{w}$ and b such that they minimize $||\boldsymbol{w}||$ while ensuring that every data point is classified correctly as

$$\boldsymbol{w}^\top \boldsymbol{x}_i + b \geq 1 \text{ for } y_i = 1 \tag{5.11}$$

and

$$\boldsymbol{w}^\top \boldsymbol{x}_i + b \leq -1 \text{ for } y_i = -1 \tag{5.12}$$

The underlying optimization problem can be written as

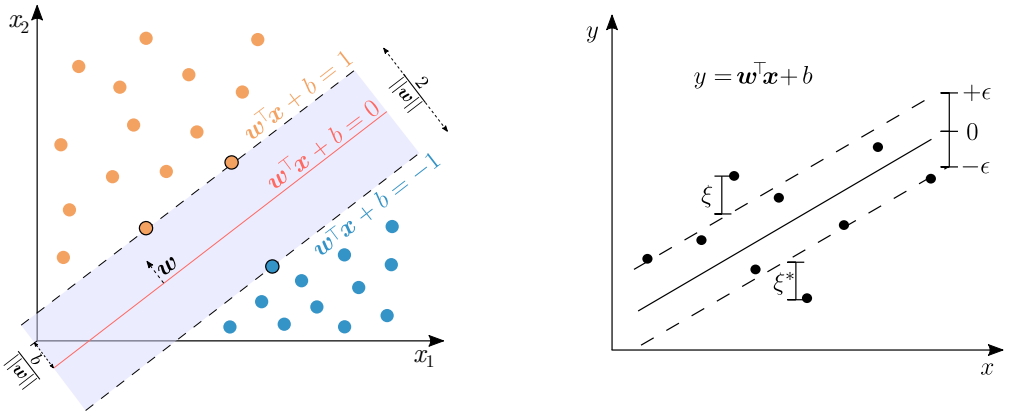
$$\min_{\boldsymbol{w},b} \frac{1}{2} \boldsymbol{w}^\top \boldsymbol{w} \tag{5.13}$$

subject to

$$y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1 \text{ for } i = 1, \ldots, n. \tag{5.14}$$

The algorithm can be extended to cases in which the training data cannot be linearly separated. For these cases, we allow some samples to be at a distance ζ from their correct margin boundary, leading to

$$\min_{\boldsymbol{w},b,\zeta} \frac{1}{2} \boldsymbol{w}^\top \boldsymbol{w} + C \sum_{i=1}^{n} \zeta_i \tag{5.15}$$



(a) (b)

Figure 5.3: Support vector machines for (a) classification and (b) regression.

subject to

$$\begin{aligned} y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) &\geq 1 - \zeta_i, \\ \zeta_i &\geq 0, i = 1, \ldots, n, \end{aligned} \tag{5.16}$$

where C is the penalty factor that controls the tradeoff between increasing the margin size and ensuring that $\boldsymbol{x}_i$ lies on the correct side of the margin boundary. Moreover, we can use this approach to learn a nonlinear classification rule by transforming our data with $\boldsymbol{\phi}(\boldsymbol{x}_i)$. This corresponds to a linear classification rule of the transformed data points $\boldsymbol{\phi}(\boldsymbol{x}_i)$. The optimization problem in Equations 5.15–5.16 becomes

$$\min_{\boldsymbol{w},b} \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} + C \sum_{i=1}^{n} \max(0, 1 - y_i(\boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}_i) + b)). \tag{5.17}$$

The concept can also be applied for regression problems. In the case of regression analyses, we penalize all points that are further away than ϵ. The optimization problem becomes therefore

$$\min_{\boldsymbol{w},b} \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} + C \sum_{i=1}^{n} \max(0, |y_i - \boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}_i) + b| - \epsilon). \tag{5.18}$$

Like most regression algorithms, support vector machines require a fixed number of inputs. The previous section shows that the latest monitored data point includes the most important information. Therefore, we use the most recent data point as the input, similar to the k-nearest neighbors approach. The input vector of the support vector machine model consists of both the most recent damage state and its measurement location in order to predict the last damage state. Furthermore, the popular radial basis function is used to transform the data.

Error estimator. Since support vector machines do not provide probabilistic outputs, we apply an error estimator that estimates the squared error made by the trained support vector machine model. The value predicted by the error estimator serves as the prediction's variance. We choose a k-nearest neighbors model to predict the squared error as it was shown in other cases to work better than alternatives [113]. For this approach, we split the training set further

into a training ($\approx 70\%$) and a validation ($\approx 30\%$) data set. We first train the support vector machine model on the training set to predict the last damage state. As the input, we use the latest measurement location and the latest damage state. Second, the trained model is applied to predict the last damage state of the validation set, and the squared error between the model's prediction and the true value is computed. Third, the k-nearest neighbors model is trained on the validation set to predict the squared error. It is then applied to predict the variance for the training set. The parameter k is chosen such that the negative log-likelihood is minimized.

Relevance vector machines. Another possibility to quantify the prediction's uncertainty is using relevance vector machines. They were first introduced by Tipping [140] and generate predictive distributions rather than point predictions. Relevance vector machines are a probabilistic model whose functional form is equivalent to support vector machines [17]. Like support vector machines, relevance vector machines assume a linear model

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} w_i k(\boldsymbol{x}, \boldsymbol{x}_i) + w_0, \tag{5.19}$$

where $\boldsymbol{w}$ are the weights and $k(\boldsymbol{x}, \boldsymbol{x}_i)$ is the kernel function to transform the data. The model tries to predict the output y based on the input $\boldsymbol{x}$ according to $y = f(\boldsymbol{x}) + \epsilon$, where ϵ are independent noise samples from a normal distribution with a zero mean and a variance of σ_y^2. The likelihood of the data set can be written as

$$p(\boldsymbol{y}|\boldsymbol{w}, \sigma_y^2) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{1}{2\sigma_y^2}||\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{w}||^2\right), \tag{5.20}$$

where $\boldsymbol{y} = [y_1, ..., y_n]^\top$, $\boldsymbol{w} = [w_0, ..., w_n]^\top$, and $\boldsymbol{\Phi}$ is the $n \times (n+1)$ design matrix with $\boldsymbol{\Phi} = [\boldsymbol{\phi}(\boldsymbol{x}_1), ..., \boldsymbol{\phi}(\boldsymbol{x}_n)]^\top$, wherein $\boldsymbol{\phi}(\boldsymbol{x}_i) = [1, k(\boldsymbol{x}_i, \boldsymbol{x}_1), ..., k(\boldsymbol{x}_i, \boldsymbol{x}_n)]^\top$. As maximum likelihood estimation of $\boldsymbol{w}$ and σ_y would generally lead to overfitting, Tipping [140] encoded a preference for smoother functions by defining an automatic relevance determination Gaussian prior over the weights:

$$p(\boldsymbol{w}|\boldsymbol{\alpha}) = \prod_{i=0}^{n} \mathcal{N}(w_i|0, \alpha_i^{-1}) \tag{5.21}$$

According to Tipping [140], the posterior over the weights is obtained from Bayes' rule by

$$p(\boldsymbol{w}|\boldsymbol{y}, \boldsymbol{\alpha}, \sigma_y^2) = \frac{1}{\sqrt{(2\pi)^{n+1} \det \boldsymbol{\Sigma}}} \exp\left(-\frac{1}{2}(\boldsymbol{w} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{w} - \boldsymbol{\mu})\right), \tag{5.22}$$

with

$$\boldsymbol{\Sigma} = (\boldsymbol{\Phi}^\top \boldsymbol{B} \boldsymbol{\Phi} + \boldsymbol{A})^{-1} \tag{5.23}$$

and

$$\boldsymbol{\mu} = \boldsymbol{\Sigma} \boldsymbol{\Phi}^\top \boldsymbol{B} \boldsymbol{y}, \tag{5.24}$$

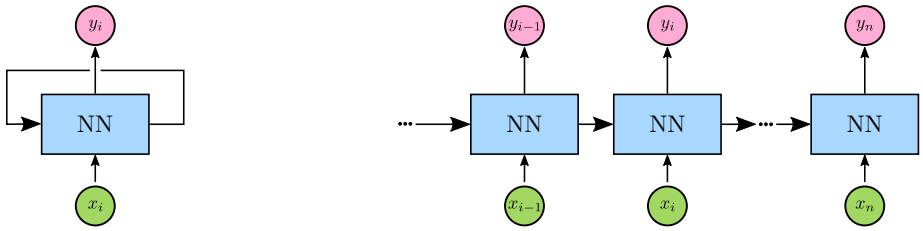
where $\boldsymbol{A} = \mathrm{diag}(\alpha_0, \ldots, \alpha_n)$ and $\boldsymbol{B} = \sigma_y^{-2} \boldsymbol{I}$. The likelihood for the parameters is obtained by integrating out the weights

$$p(\boldsymbol{y}|\boldsymbol{\alpha}, \sigma_y^2) = \frac{1}{\sqrt{(2\pi)^n \det(\boldsymbol{B}^{-1} + \boldsymbol{\Phi} \boldsymbol{A}^{-1} \boldsymbol{\Phi}^\top)}} \exp\left(-\frac{1}{2} \boldsymbol{y}^\top (\boldsymbol{B}^{-1} + \boldsymbol{\Phi} \boldsymbol{A}^{-1} \boldsymbol{\Phi}^\top)^{-1} \boldsymbol{y}\right). \tag{5.25}$$

For more information, see [141]. In contrast to the approach with the error estimator, we do not have to split our training set for the relevance vector machine approach. We can directly train the model on the training set and obtain a probabilistic model. As for the previous support vector machines model, we use a radial basis function as our kernel. The results of predicting the last damage states of our three data sets are presented in Section 5.6.

5.4 Recurrent neural networks

Recurrent neural networks, developed in the 1980s, are a class of neural networks that captures time dynamics. They constitute a set of traditional neural networks in order to process information across time steps. Due to their internal state memory, they can treat a sequence of inputs [97]. This type of algorithm is an effective method for addressing time series [131]. Figure 5.4a shows a compressed version of a recurrent neural network. Like regular neural networks, it predicts the output y_i based on the input x_i. Moreover, it outputs another variable that serves as an additional input for the next prediction. Recurrent neural networks are thus able to inhere information from previous data. Figure 5.4b depicts an unfolded recurrent neural network and shows how the additional output is used as the input for the next prediction. In the figure, the recurrent neural network seems to have several layers. However, these are different time steps of the same neural network. Due to the vanishing gradient problem that recurrent neural networks face during backpropagation, long short-term memory (LSTM) [56] were developed. Therefore, an LSTM is used in this study.



(a) (b)

Figure 5.4: Recurrent neural network (a) compressed and (b) unfolded.

In our case, we would like to predict the last damage state $y_{n,j}$ of each trajectory $j = 1, .., m$ based on the already observed damage states $\boldsymbol{y}_{1:i,j}$ collected at $\boldsymbol{x}_{1:i,j}$. Therefore, we use $\boldsymbol{x}_{1:i,j}$ and $\boldsymbol{y}_{1:i,j}$ as the input and $y_{n,j}$ as the targets of our model. Moreover, the model should predict not only the last damage state but also provide information about the prediction's uncertainty. In order to obtain probabilistic predictions, we combine LSTM with a mixture density network and assume the output to be normally distributed. Therefore, the model predicts two outputs, the mean $\hat{\mu}$ and the standard deviation $\hat{\sigma}$ of the last damage state. We denote $\hat{\mu}_{i,j}$ and $\hat{\sigma}_{i,j}$ as the i-th prediction for the last damage state of trajectory j. The model is illustrated in Figure 5.5.

The objective function is based on the probability density

$$p(y_{n,j}|\boldsymbol{x}_{1:i,j}, \boldsymbol{y}_{1:i,j}) = \mathcal{N}(y_{n,j}|\hat{\mu}_{i,j}, \sigma^2_{i,j}). \tag{5.26}$$

For n data points per trajectory and m trajectories, we can maximize the product of all

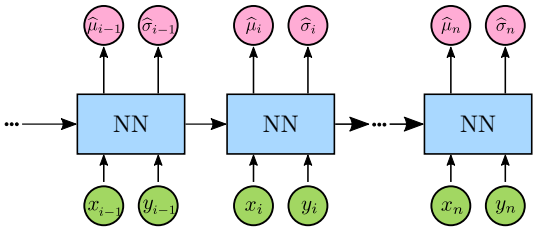


Figure 5.5: Illustration of the recurrent neural network used in this study.

likelihoods

$$\prod_{j=1}^{m}\prod_{i=1}^{n} p(y_{n,j}|\boldsymbol{x}_{1:i,j}, \boldsymbol{y}_{1:i,j}). \tag{5.27}$$

Instead of maximizing the product of all likelihoods, we can minimize the sum of all negative log-likelihoods

$$\sum_{j=1}^{m}\sum_{i=1}^{n} -\log p(y_{n,j}|\boldsymbol{x}_{1:i,j}, \boldsymbol{y}_{1:i,j}). \tag{5.28}$$

In order to avoid dependencies on the trajectory size, we use the mean of all negative log-likelihoods

$$\bar{\mathcal{L}} = -\frac{1}{m}\sum_{j=1}^{m}\frac{1}{n}\sum_{i=1}^{n}\log p(y_{n,j}|\boldsymbol{x}_{1:i,j}, \boldsymbol{y}_{1:i,j}) \tag{5.29}$$

as our objective function.

Before training the model, we have to choose the number of neurons in our recurrent neural network. In order to find this number, we split our training set into a training ($\approx 70\%$) and validation set ($\approx 30\%$), train different models on the training set, and evaluate their performance on the validation set. We increase the number of neurons in 2^u steps with $u \in \mathbb{N}$ starting from $u = 1$ until $\bar{\mathcal{L}}$ rises. This approach results in a neural network with 8, 4, and 8 neurons for the Virkler data set, the Hudak data set, and the laser degradation data set, respectively. We present the results together with the outcomes from the other models in Section 5.6.

5.5 Multivariate normal distributions and inferred Gaussian processes

Using a multivariate normal distribution is a simple approach to update predictions based on currently measured data. As explained in Section 4.2, we can compute the sample mean vector and the sample covariance matrix from the training trajectories to estimate the normal distribution parameters. Since we compare the negative log-likelihood of the last damage state and the other models are trained to minimize this measure, we estimate the observation error by minimizing Equation 5.29. As the observation error is already incorporated in the sample covariance matrix, an attempt to estimate it by minimizing the negative log-likelihood on the training data set leads to very small values and thus to overfitting. Therefore, the observation error is determined in a leave-one-out scheme, i.e., each training trajectory is used for evaluating the mean negative log-likelihood of all predicted last damage states while the other ones are used to determine the sample mean vector and the sample covariance matrix. This is repeated for every training trajectory being the one to predict.

An advantage of the multivariate normal distribution approach is that we do not need to assume any underlying function. For the presented Gaussian process method, however, the basis functions have to be specified in advance. As in the previous chapter, we use polynomial basis functions for the FCG Hudak and the laser degradation data. In order to determine the polynomial order q, we split the training set into training ($\approx 70\%$) and validation ($\approx 30\%$) trajectories as we did for finding the number of neurons in the recurrent neural network. To find the best underlying basis functions, we

(1) choose a maximum polynomial order q,
(2) infer the mean function $m_{IGP}(x)$ and covariance function $k_{IGP}(x, x')$ from the training trajectories,
(3) estimate the observation error's standard deviation σ_y
    (a) by computing the square root of the mean squared errors between the fitted basis functions and the data regarding all training trajectories, see Equation 4.23, or
    (b) by minimizing the mean negative log-likelihood of all training trajectories with the covariance function $k(x, x') = k_{IGP}(x, x') + \delta\, \sigma_y^2$, and
(4) evaluate the mean negative log-likelihood of all validation trajectories based on the resulting mean function, covariance function, and observation error.

The four steps are repeated for different polynomial orders. Eventually, the model that minimizes the mean negative log-likelihood of all validation trajectories is used. Compared to 3b, step 3a is faster since it removes solving an optimization problem. In cases where the observation error is not the same over the entire trajectory, step 3b with $\sigma_y^2(x)$ can be used. After the optimized polynomial order $\hat{q}$ is determined, a Gaussian process model can be inferred from the training and validation trajectories.

The explained procedure executed with $q \in \{1, 2, \ldots, 6\}$ leads to $\hat{q}_{Hudak} = 5$ for the FCG Hudak data set and to $\hat{q}_{laser} = 4$ for the laser degradation data set. The same optimized polynomial order is found using either step (3 a) or (3 b). The resulting model chosen by minimizing the mean negative log-likelihood is not the one with the highest polynomial degree for both data sets, even though a higher polynomial degree leads to a better fit of each trajectory. Therefore, the procedure does not tend to overfit. This is what we strive for, as overfitting should be avoided.

The procedure based on minimizing the negative log-likelihood is also used for the Virkler data. As in the previous chapter, we use physics-informed basis functions. We saw that the exponent α influences the results strongly. Therefore, we train four different models with the basis functions $\alpha = [2.6, 2.8, 3.0, 3.2]$ (from previous chapter), $\alpha = [2.6, 2.75, 2.9, 3.05, 3.2]$, $\alpha = [2.3, 2.6, 2.9, 3.2, 3.5]$, and $\alpha = [2.0, 2.45, 2.9, 3.35, 3.8]$, and follow the steps 1-4 (with step 3b since the observation error is not constant over the trajectories, see Section 4.5). $\alpha = [2.6, 2.75, 2.9, 3.05, 3.2]$ leads to the lowest mean of the negative log-likelihoods of all validation trajectories and is therefore used. Introducing one more coefficient reduced the negative log-likelihood compared to the ones used in the previous chapter. Broadening the range of α did not improve the results.

The results differ from the ones determined in the last chapter, where we estimated the polynomial order by splitting each training trajectory into training and validation data points and minimizing the average mean squared error ($q_{Hudak} = 2, q_{laser} = 1$). The mean negative log-likelihoods of all validation trajectories for the models used in the previous chapter are compared to those determined in this chapter in Table 5.2. The procedure based on the negative log-likelihood leads to a lower negative log-likelihood than the one based on the mean

Table 5.2: Mean negative log-likelihoods of the validation trajectories for the FCG Hudak, laser degradation, and FCG Virkler data set.

	FCG Hudak	**Laser degradation**	**FCG Virkler**
$\bar{\mathcal{L}}$ (prev. chapter)	-21.7	-3.33	1405
$\bar{\mathcal{L}}$ (this chapter)	-34.4	-5.70	1400

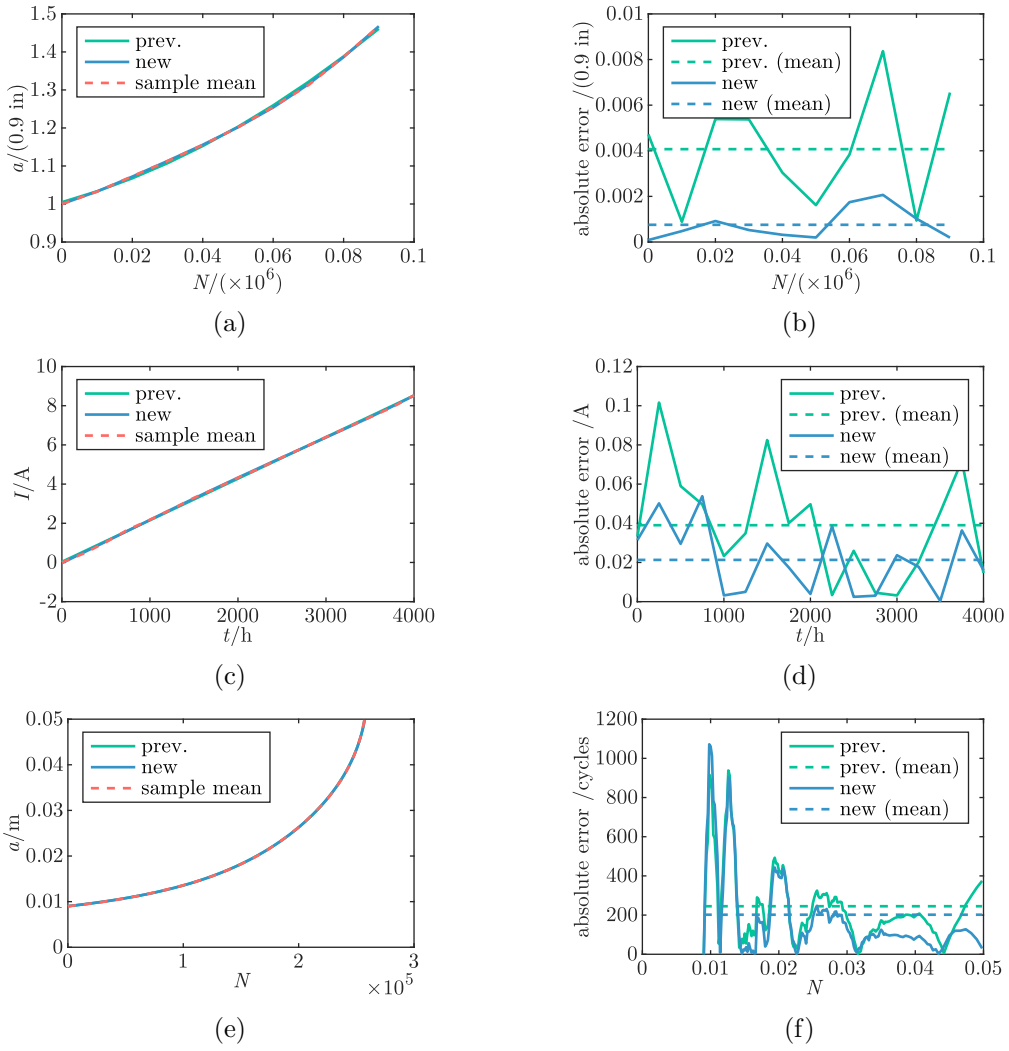


Figure 5.6: Mean comparisons (left) and errors of mean functions (right) for the (a & b) FCG Hudak, (c & d) laser degradation, and (e & f) FCG Virkler data set.

squared error. This is because not only the observation error but also the polynomial order is chosen such that it minimizes the negative log-likelihood.

As a Gaussian process is fully defined by its mean and covariance function, we additionally compare them to the ones determined in the previous chapter. Figure 5.6 compares the

resulting mean functions for each data set to the sample mean vectors and the mean functions derived in the last chapter. Compared to the sample mean vector, the mean functions deter-

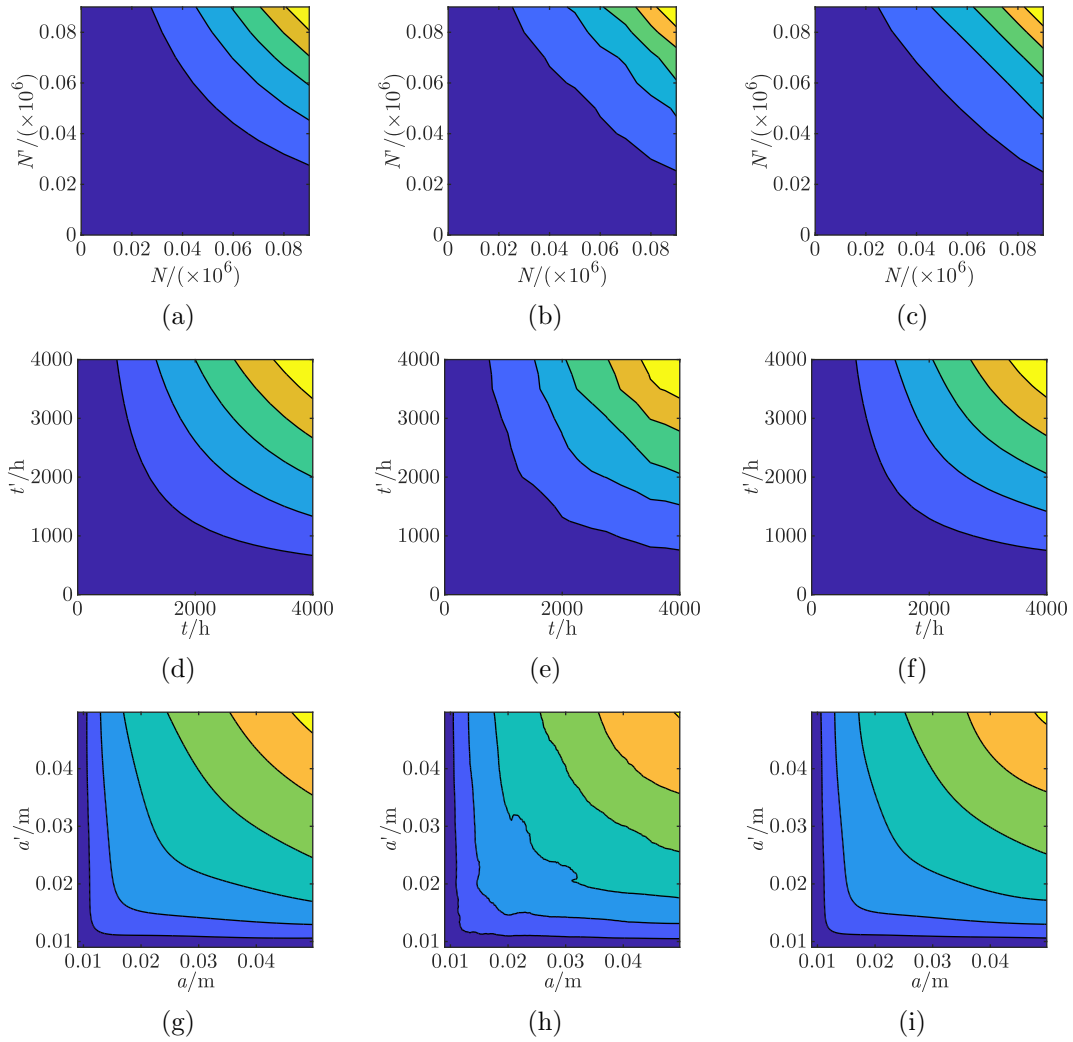


Figure 5.7: Previous covariance functions (left), sample covariance matrices (center), and new covariance functions (right) for the (a–c) FCG Hudak, (d–f) laser degradation, and (g–i) FCG Virkler data set.

mined by the procedure based on the negative log-likelihood lead to smaller MAEs than those used in the previous chapter. This is due to the higher polynomial order and the additional basis function for the Virkler data set, leading to a better fit for each trajectory.

The covariance functions are shown in Figure 5.7. We can see that the higher polynomial order for the FCG Hudak data set enables the method to better approximate the sample covariance matrix. The light blue areas in Figure 5.7b are non-convex, which can be modeled with a higher polynomial degree, see Figure 5.7c, but not with the polynomial order used in the previous chapter, see Figure 5.7a. By contrast, the covariance functions for the laser degradation data set are very similar (Figures 5.7d–5.7f). The covariance function determined in this chapter represents the covariance matrix only slightly better. For the Virkler data

set, however, we can see a larger difference towards higher crack lengths again. The orange area of the sample covariance matrix in Figure 5.7h can be better approximated by the model determined with the procedure based on the negative log-likelihood (Figure 5.7i) than the model used in the previous chapter (Figure 5.7g).

After selecting the polynomial degree and the basis functions, we infer the Gaussian process from the entire training data set (training and validation trajectories). As the other models are trained to minimize the mean of the negative log-likelihoods of all predicted last damage states, we estimate the observation error likewise. Table 5.3 summarizes the estimated observation errors' standard deviations of the multivariate normal distribution and the inferred Gaussian process approaches. The results show that the observation errors' standard deviations of the models determined by the presented procedure are smaller than the previous ones. Again, this indicates that the Gaussian processes derived in this chapter better capture the underlying structure of the data and are not forced to do that by increasing the observation error's standard deviation.

Table 5.3: Estimated observation errors' standard deviations of the multivariate normal distribution (MVN) and the inferred Gaussian process model (IGPM).

Model	**FCG Hudak** $\sigma_y/(0.9\,\text{in})$	**Laser degradation** σ_y/A	**FCG Virkler** σ_y/mm
MVN	0.0054	0.4585	0.5561
IGPM (prev. chapter)	0.0136	0.6153	0.9572
IGPM (this chapter)	0.0037	0.1613	0.7446

5.6 Results and discussion

After training all models explained in the previous sections, we can evaluate their predictive capabilities over time. Figure 5.8 shows the probabilistic predictions of all trained models for the Virkler set trajectory #50, which is a line close to the data set's mean. In each figure, the abscissa represents the last measured crack length a_i at which the fatigue life is predicted. The blue line displays the mean value of the predicted fatigue life (i.e., the last damage state), and the light blue area indicates the symmetric 95% credible region. The red straight line determines the actual fatigue life to be predicted.

We see in Figure 5.8a that the predicted credible region of the support vector machines model with the error estimator narrows towards longer crack lengths. This follows our intuition that the more data we have and the closer we are to the fatigue life, the better we can predict the final number of cycles. By contrast, the model based on relevance vector machines (Figure 5.8b) is not able to capture this behavior. Since a relevance vector machine is equivalent to a Gaussian process with a zero mean and the covariance function

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma_y^2 \boldsymbol{I} + \sum_{i=0}^{n} \frac{1}{\alpha_i} \boldsymbol{v}_i \boldsymbol{v}_i^\top \tag{5.30}$$

with $\boldsymbol{v}_i = [\phi_i(\boldsymbol{x}_1), ..., \phi_i(\boldsymbol{x}_n)]^\top$ (see Section 5.2 in [141]), the underlying model with the assumed radial basis kernel constrains the predictive capabilities.

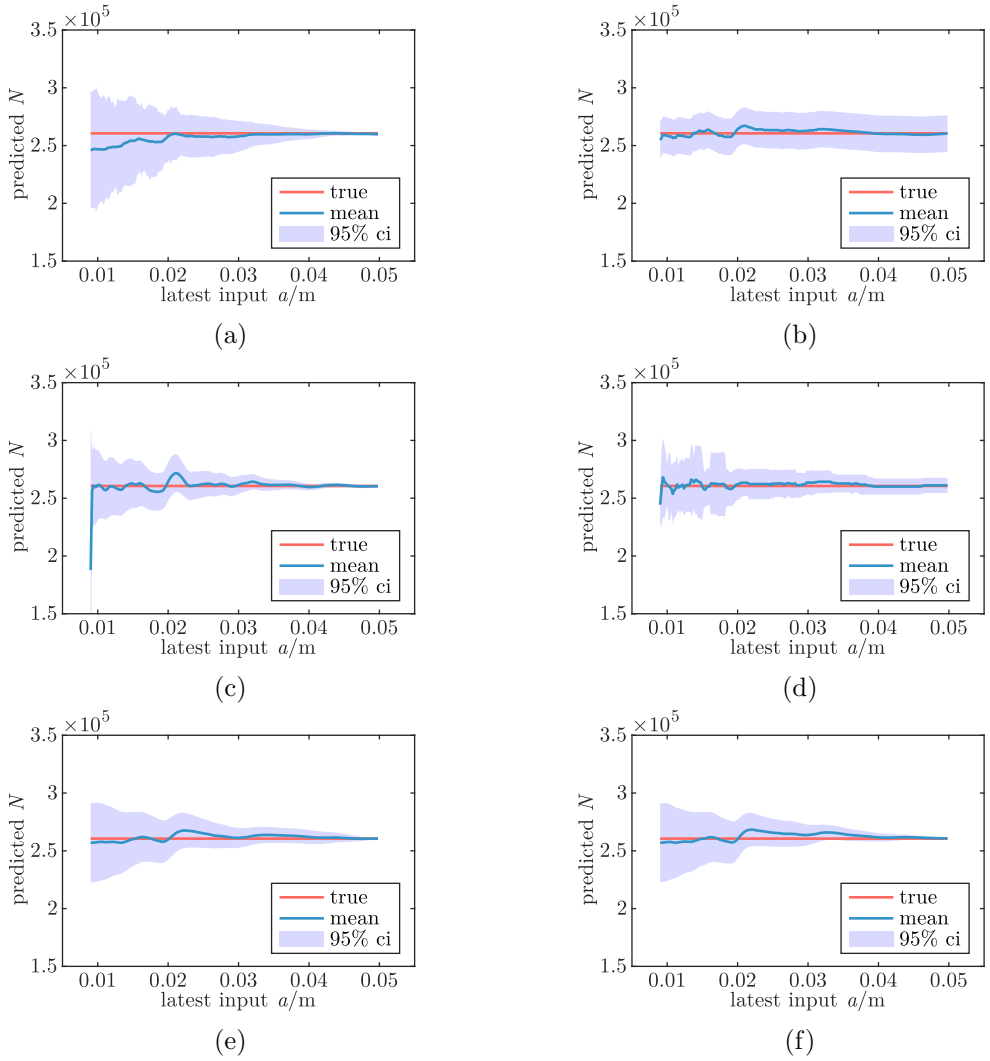


Figure 5.8: Predicted fatigue life for trajectory #50 of the model based on (a) support vector machines with error estimator, (b) relevance vector machines, (c) recurrent neural networks, (d) k-nearest neighbors, (e) inferred Gaussian process, and (f) multivariate normal distribution.

Furthermore, the k-nearest neighbors' probabilistic predictions in Figure 5.8d narrow down towards larger crack lengths since the model is able to choose lines closer to the trajectory to be predicted. Whenever the set of the 13 closest trajectories changes, the credible interval adjusts accordingly, leading to a stepwise evolution of the credible interval. The predicted mean value and credible interval do not alter after a crack length of approximately 0.04 m as the set of the 13 closest trajectories does not change either.

The recurrent neural network predicts a tight credible region around the true fatigue life that narrows towards larger crack lengths, see Figure 5.8c. The credible intervals for large crack lengths are particularly tight around the true value, leading to precise predictions towards the end. The first mean prediction is too small but adjusts after the second one, indicating that the first prediction hardly influences the objective function, i.e., the mean negative log-likelihood of all predictions. This is because predictions at larger crack lengths lead to lower negative

log-likelihoods as the predicted credible intervals are smaller, and the mean estimations are closer to the true values.

Moreover, Figures 5.8e and 5.8f show that the predictions of the inferred Gaussian process and the multivariate normal distribution are very similar. Both predicted credible regions have similar sizes and narrow towards larger crack lengths. A slightly narrower credible interval for the multivariate normal distribution can be noticed towards the end. Also, the predicted mean values of both approaches follow a similar path, indicating that the inferred Gaussian process is akin to the derived multivariate normal distribution.

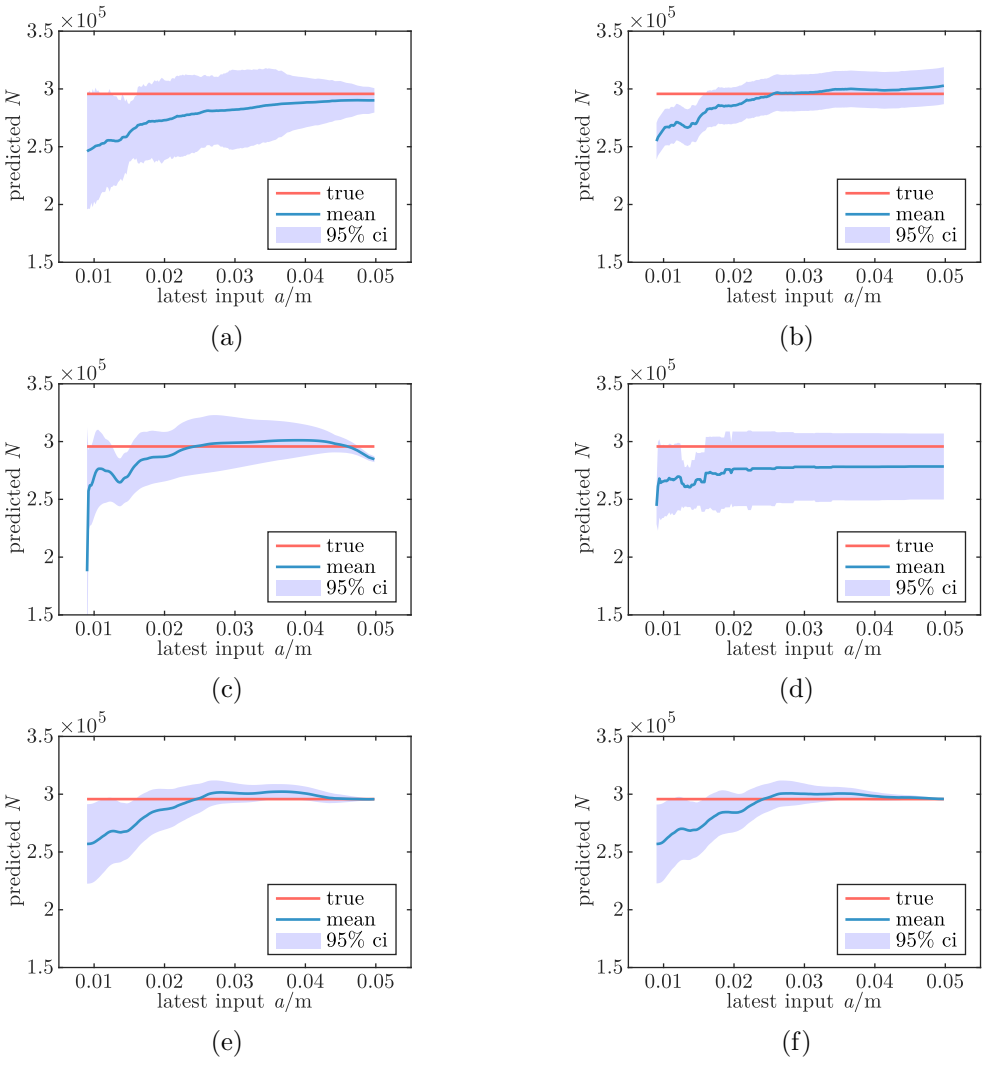


Figure 5.9: Predicted fatigue life for trajectory #48 of the model based on (a) support vector machines with error estimator, (b) relevance vector machines, (c) recurrent neural networks, (d) k-nearest neighbors, (e) inferred Gaussian process, and (f) multivariate normal distribution.

In order to investigate the predictions for a trajectory that represents a rather unusual behavior, Figure 5.9 shows the probabilistic predictions for trajectory #48, a line that resembles a specimen with a rather long fatigue life. For this trajectory, the credible region of the support vector machines model (Figure 5.9a) first narrows down but broadens again after a current crack

length of about $a = 0.015\,\mathrm{m}$. A similar behavior can be seen for the recurrent neural network in Figure 5.9c and the k-nearest neighbors model in Figure 5.9d. This is because trajectory #48 follows a similar trend as the mean until a crack length of approximately $0.015\,\mathrm{m}$, leading to overly confident predictions. However, the crack grows slower than the mean trajectory towards larger crack lengths, resulting in broadened predicted credible intervals. Since trajectory #48 is close to the boundary of the training trajectories, the k-nearest neighbors model cannot change its set of the 13 closest trajectories towards larger crack lengths and thus ends up in predicting the same large credible intervals, see Figure 5.9d. Since the trajectory resembles a rather unusual behavior, the recurrent neural network struggles to predict the true fatigue life. This is especially apparent towards the end, see Figure 5.9c, where the prediction seems to be unreasonable. This indicates that the trained recurrent neural network is not able to treat data at the training boundaries well and that its predictions are difficult to interpret. The relevance vector machines model is again unable to narrow its credible region for larger crack lengths, see Figure 5.9b.

In contrast to the before-mentioned models, the credible regions predicted by the inferred Gaussian process in Figure 5.9e and the multivariate normal distribution in Figure 5.9f gradually narrow towards larger crack lengths. Due to the equations for a conditional normal distribution, the updating scheme of the variance is fixed, leading to a credible region that narrows down with more observed data points. Furthermore, the mean prediction gradually shifts towards the true value for more input data, leading to accurate predictions especially towards the end. The probabilistic predictions of both approaches are very similar again.

In order to quantify and compare the prediction accuracy of all models, we evaluate the mean negative log-likelihood and the MAE of all predicted last damage states for the test trajectories. Since the first data points of each data set are the same, we start evaluating from prediction 2. Therefore, the predicted spike of the recurrent neural network at the beginning of each trajectory is excluded from the evaluation. As the last point n of a trajectory is the damage state to be predicted, prediction $n - 1$ is the last one considered for our evaluation. Table 5.4 shows the evaluated measures and ranks the models according to each data set's lowest mean negative log-likelihood.

The lowest mean negative log-likelihood for the Virkler data set is achieved by the recurrent neural network followed by the multivariate normal distribution and inferred Gaussian process models. The predictions of those three models are based on all observed data points. By contrast, the support vector machines, relevance vector machines, and k-nearest neighbors models are only based on the most recently observed data point. As a recurrent neural network is a highly flexible model, it is able to predict small credible intervals and accurate mean values especially towards the end, leading to very low negative log-likelihoods. In contrast to the recurrent neural network, the updating scheme of the Gaussian process and the multivariate normal distribution is based on the equation of the conditional normal distribution. Since the standard deviation of the observation error is fixed, these models lead to accurate predictions, however, not to very low negative log-likelihoods. By looking further in detail, we recognized that the recurrent neural network has lower negative log-likelihoods for trajectories close to the mean and greater ones for trajectories further off the mean. This agrees with the previously described pattern, where we stated that the recurrent neural network lacks predicting data on or slightly outside the training boundary well. The normal distribution approaches, however, also predict those data points rather accurately. Yet, due to the prescribed updating scheme, they have greater negative log-likelihoods for data close to the mean. Even though the inferred

Gaussian process and the multivariate normal distribution models have slightly greater mean negative log-likelihoods, their predictions lead to lower MAEs. This reveals their good predictive capabilities and indicates that their greater mean negative log-likelihoods result from too broad credible intervals.

Table 5.4: Mean negative log-likelihood $\bar{\mathcal{L}}$, MAE, and training time for k-nearest neighbors (KNN), support vector machines with error estimator (SVM), relevance vector machines (RVM), recurrent neural networks (RNN), multivariate normal distribution (MVN), and inferred Gaussian process model (IGPM) for the FCG Hudak, laser degradation, and FCG Virkler data set.

Data set	**Model**	$\bar{\mathcal{L}}$		**MAE**		**Training time** in min	
FCG Hudak predict: $a(N_{end})$	MVN	−1.85		0.0334		0.0166	
	IGPM	−1.72	+6.9%	0.0308	−7.8%	0.0910	+448.2%
	RVM	−1.19	+35.6%	0.0446	+33.5%	0.00164	−90.1%
	RNN	−0.846	+54.2%	0.0571	+71.0%	8.15	+49010.8%
	KNN	3.29	+277.9%	0.0718	+115.0%	0.00110	−93.4%
	SVM	42.4	+2397.3%	0.0559	+67.4%	0.0376	+126.7%
Laser degradation predict: $I(t_{end})$	RNN	0.663		0.426 A		19.6	
	IGPM	0.820	+23.7%	0.473 A	+11.1%	0.109	−99.4%
	MVN	0.942	+42.1%	0.446 A	+4.6%	0.0172	−99.9%
	SVM	0.957	+44.3%	0.402 A	−5.7%	0.0920	−99.5%
	KNN	1.07	+61.3%	0.490 A	+14.9%	0.00130	−100.0%
	RVM	1.12	+69.6%	0.397 A	−6.9%	0.00424	−100.0%
FCG Virkler predict: $N(a_{end})$	RNN	9.90		5.38×10^3		976	
	MVN	9.96	+0.6%	5.14×10^3	−4.4%	0.712	−99.9%
	IGPM	10.0	+1.2%	5.23×10^3	−2.7%	1.51	−99.8%
	SVM	10.2	+3.2%	6.27×10^3	+16.7%	11.0	−98.9%
	KNN	10.4	+4.8%	7.36×10^3	+36.8%	0.0369	−100.0%
	RVM	10.4	+5.1%	5.22×10^3	−3.0%	6.93	−99.3%

Similar good performances of the multivariate normal distribution and inferred Gaussian process models are determined for the FCG Hudak data set. Both models have the lowest mean negative log-likelihoods and MAEs. The results differ between those models due to the different observation errors obtained from optimization. For the FCG Hudak data set, the recurrent neural network performs significantly worse than the previously mentioned models since one test trajectory lies outside the training trajectories, see Figure 5.1c. As explained earlier, the recurrent neural network struggles to predict data that is not included in the training set. Table 5.4 shows that this is also the case for the models based on relevance vector machines, support vector machines, and k-nearest neighbors. In contrast to merely mapping the input data to the last damage state, the inferred Gaussian process and the multivariate normal distribution model the distribution of entire trajectories. Therefore, they are less prone to overfitting than the other models, which is essential especially for real-life applications, as outliers can always occur in an uncertain environment.

Based on the mean negative log-likelihood for the laser degradation data set, the best model is again the recurrent neural network, followed by the inferred Gaussian process and the derived

multivariate normal distribution. This time, the predictions of the recurrent neural network result in a much lower mean negative log-likelihood than the inferred Gaussian process and the multivariate normal distribution models since all test trajectories are surrounded by the training trajectories and follow their predominant behavior. Moreover, the inferred Gaussian process and the multivariate normal distribution models struggle to predict the last damage state as accurately as the recurrent neural network since they rely on normally distributed data. As explained earlier, they model the distribution of trajectories and thus need the trajectories to be normally distributed. Yet, Figure 3.4b clearly shows that the laser degradation trajectories do not follow a normal distribution which is why their performances are worse than the recurrent neural network.

Table 5.4 also shows the training time of each model, including the time for selecting the model, i.e., choosing the number of neurons or the polynomial order and optimizing the standard deviation of the observation error. The k-nearest neighbors model is trained in the shortest time for all data sets. Support vector machines and relevance vector machines are trained quickly for small data sets like the FCG Hudak and laser degradation data sets. Yet, they need substantial training time for larger data sets like the Virkler data. Moreover, we can see that the recurrent neural network needs the longest training time. Compared to the inferred Gaussian process method, the recurrent neural network takes about 100 times longer to be trained. The multivariate normal distribution is derived in about 1/10 of the Gaussian process training time. Even though both normal distribution models determine the distribution of trajectories, they need significantly less training time than recurrent neural networks.

The models based on multivariate normal distributions and Gaussian processes can predict not only the last damage state but also the entire degradation trajectory. Thus, they enable computing the probability of failure at every time in the future, which greatly benefits prognostics and health management applications. Predicting only the last damage state is, in fact, not very helpful since merely one state in time is estimated. No maintenance schedule can be adjusted based on predicting the last damage condition as the stages in between are unknown. Of course, the other machine learning models might be trained such that they are able to predict multiple damage states. However, the training process of recurrent neural networks is even for predicting only the last state 100 times longer than the one of Gaussian processes. Therefore, Gaussian processes reveal excellent predictive capabilities since they are able to predict all future damage states without any adjustments and are trained quickly.

5.7 Summary

This chapter presents six different models for predicting damage states with credible intervals. The results show that the best models are the ones based on recurrent neural networks, inferred Gaussian processes, and multivariate normal distributions. In general, inferred Gaussian processes lead to similar prediction accuracies as recurrent neural networks. While recurrent neural networks lead to lower negative log-likelihoods for non-normally distributed degradation trajectories, Gaussian processes are able to treat data that is close to or slightly outside the training boundaries better. Furthermore, inferred Gaussian processes lead to more intuitive predictions than recurrent neural networks and need only a fraction of training time ($\approx 1/100$). Gaussian processes can further predict the entire degradation trajectory, which enables computing the probability of failure for every future time. This is essential for prognostics applications. By introducing a selection procedure for inferring Gaussian processes, the

polynomial order can be chosen to minimize the negative log-likelihood, which further improves their predictive capabilities.

In summary, this chapter reveals that Gaussian processes are very suitable for prognostics and health management applications as they are able to predict the entire degradation trajectory and have great predictive performance measures compared to other models. However, we see that Gaussian processes struggle to accurately predict non-normally distributed trajectories. Therefore, the next chapter is concerned with the question of how to improve their predictions for data that does not follow a normal distribution.

Chapter 6

Warped Gaussian Processes

"...all models are wrong, but yours are stupid too."

– ML Hipster

We know that Gaussian processes rely on normally distributed functions and that crack lengths cannot be subject to a normal distribution since they assume only positive values. Therefore, modeling crack lengths with a Gaussian process might not only be wrong but *"stupid too"*. ML Hipster ascribes the posted quote on Twitter to "George Box in a less magnanimous mood" and amusingly clarifies that we should think before simply applying machine learning models to data sets. In this chapter, we would like to revisit the approach for inferring Gaussian processes from previous degradation trajectories and adjust the method such that it is able to treat non-normally distributed data.

The work presented in Sections 6.1–6.3 and Sections 6.5–6.6 of this chapter is fully based on the publication Pfingstl et al. [108] and resulted from a strong collaboration with my master's student Christian Braun. The results were further discussed with Dr. Amir Nasrollahi, PostDoc at Stanford University, Prof. Dr. Fu-Kuo Chang, director of the Structures and Composites Laboratory at Stanford University, and my supervisor Prof. Dr. Markus Zimmermann.

The chapter's first section motivates the conducted study and gives an overview of related works. Section 6.2 presents an extension to the approach for inferring Gaussian processes from previous data in order to treat non-normally distributed trajectories, which is founded on the concept of warped Gaussian processes. In the third section, the extension is applied to an academic example, an infinite pre-cracked plate, revealing the advantages and drawbacks of the advanced method. Since collecting several experimental degradation trajectories is cumbersome and sometimes infeasible, integrating simulation data into Gaussian processes is considered in Section 6.4. As simulation data lacks noise, we present a method for estimating the standard deviation of the observation error from currently monitored data. The section also explains how to compute the probability of failure for future time steps. This is followed by applying warped Gaussian processes to a real aerospace structure, a wing section of an aircraft, in order to predict crack growth. The results of all sub-studies are summarized and concluded in the last section.

6.1 Motivation

As shown in the two previous chapters, Gaussian processes have excellent predictive capabilities, estimating outputs with credible intervals. Many researchers have demonstrated their usefulness for various tasks. Yet, Gaussian processes are constrained by relying on a normal distribution. There are many examples where data is non-normally distributed, and Gaussian processes would lead to non-physical predictions. Especially in prognostics and health management applications, where we deal with crack lengths and damage indices, the data cannot be normally distributed since the variables are only defined on the positive domain.

For a better treatment of data that is not normally distributed, Snelson et al. [133] introduced warped Gaussian processes. The idea is to transform the observed data y into a latent space z by a so-called warping function $\psi_\theta(y) = z$. The warping function usually has some free parameters $\boldsymbol{\theta}$, which are determined by optimization. The Gaussian process is then modeled in the latent space $f_z(x) = z$ as

$$\mathcal{N}(f_z(x) \mid m_{z,\theta}(x), k_{z,\theta}(x, x')), \tag{6.1}$$

where $m_{z,\theta}(x)$ and $k_{z,\theta}(x, x')$ are the mean and covariance function in the latent space. Note that we write an index to the functions to indicate which space they are related to. For estimating future values in the observed space, the predictions are transformed by the inverse warping function $\psi_\theta^{-1}(z) = y$.

Some researchers have already used warped Gaussian processes, for example, to predict power supplies of wind turbines. In contrast to wind speeds, power supplies cannot be assumed to be normally distributed due to the nonlinear correlation between wind speed and power [72]. Therefore, the authors of [67, 68, 69] utilized warped Gaussian processes to predict the power supplies of wind turbines. They used a sum of *tanh* as their warping function and proved the approach's usefulness on real data. Moreover, Mateo-Sanchis et al. [88] applied warped Gaussian processes to oceanic content data. They predicted the oceanic chlorophyll content from multispectral data and concluded that warped Gaussian processes outperform standard Gaussian processes. Again, a sum of *tanh* as the warping function was used.

If the inverse warping function is not available in closed form, which is the case in the before-mentioned papers, additional complexity arises from numerical approximations [118]. One can use, for example, the Box-Cox transformation [21]

$$\psi_\theta(y) = \begin{cases} \frac{y^{\theta_1} - 1}{\theta_1}, & \text{if } \theta_1 \neq 0 \\ \log y, & \text{if } \theta_1 = 0\,. \end{cases} \tag{6.2}$$

as the warping function to bypass this problem. Rios and Tobar [118] utilized a type of Box-Cox transformation function and showed its effectiveness on real data, enforcing their predicted yearly sunspot numbers to be strictly positive. A similar application of warped Gaussian processes was presented by Gonçalves et al. [45]. The authors estimated future sunspot numbers and enforced their predictions to be positive by using an integrated softplus function in their warping function.

Even though these studies show that warped Gaussian processes are able to deal with non-normally distributed data, researchers tend to use them rarely. This might be because the warping function introduces additional parameters to the modeling task. These parameters must be optimized in addition to the mean and covariance function parameters. Therefore,

one entailed problem is the arising computational complexity by determining not only the mean and covariance function parameters but also the ones of the warping function. Another problem is that after introducing a warping function, an optimizer might find, in fact, a different solution. But the result is not necessarily better, as minimizing the negative log-likelihood is, in general, a non-convex optimization problem. In order to remove the mean and covariance function parameters, we can derive these functions from previously gathered trajectories. In this way, we significantly reduce the computational complexity and additionally integrate prior knowledge into warped Gaussian processes.

6.2 Inferred warped Gaussian processes

Both approaches are combined in the following to exploit the advantages of warped Gaussian processes and the use of basis functions for inferring the Gaussian process model quickly from previously collected data. For doing so, we assume that the warped realizations $f_{z;j}(x)$ can be approximated by a linear combination of p linearly independent basis functions $\boldsymbol{\phi}_z(x) = [\phi_{z;1}, ..., \phi_{z;p}]^\top$ in the latent space z. Therefore, the warped trajectories $\{\boldsymbol{x}_j, \psi_\theta(\boldsymbol{y}_j) = \boldsymbol{z}_j\}$ can be represented with $j = 1, ..., m$ as

$$f_{z;j}(\boldsymbol{x}_j) = \boldsymbol{z}_j = \psi_\theta(\boldsymbol{y}_j) = \boldsymbol{\phi}_z(\boldsymbol{x}_j)^\top \boldsymbol{\beta}_j \tag{6.3}$$

and the Gaussian process in the latent space as

$$m_z(x) = \boldsymbol{\phi}_z(x)^\top \hat{\boldsymbol{\mu}}_\beta \tag{6.4}$$

$$k_z(x, x') = \boldsymbol{\phi}_z(x)^\top \hat{\boldsymbol{\Sigma}}_\beta \, \boldsymbol{\phi}_z(x') \tag{6.5}$$

$$p_z(z) = \mathcal{N}(f_z(x) \mid m_z(x), k_z(x, x')). \tag{6.6}$$

Note that the Gaussian process model is independent of any free parameters $\boldsymbol{\theta}$. Therefore, no optimization problem regarding the mean and covariance function parameters has to be solved. However, by introducing a warping function, free parameters are integrated into the formulation. In order to determine the warping function parameters, the negative log-likelihood in the observed space p_y is minimized. Since the probability distribution of p_y is unknown, we have to find a different way to represent it. In this case, we can take advantage of the change of variables. For a one-dimensional random variable Y that is transformed by a strictly increasing differentiable function $Z = \psi(Y)$, we can write

$$\begin{aligned} P_y(y) &= \Pr(Y \leq y) = \Pr(\psi^{-1}(Z) \leq y) = \Pr(Z \leq \psi(y)) = P_z(\psi(y)) \\ \frac{dP_y(y)}{dy} &= \frac{dP_z(\psi(y))}{dy} \\ p_y(y) &= p_z(\psi(y)) \, \frac{d\psi}{dy}(y) = p_z(z) \, \frac{d\psi}{dy}(y). \end{aligned} \tag{6.7}$$

In the case of a strictly decreasing function, the inequality sign flips by putting the warping function to the other side, which leads to

$$\begin{aligned} P_y(y) &= \Pr(\psi^{-1}(Z) \leq y) = \Pr(Z \geq \psi(y)) = 1 - P_z(\psi(y)) \\ \frac{dP_y(y)}{dy} &= \frac{d(1 - P_z(\psi(y)))}{dy} \end{aligned}$$

$$p_y(y) = -p_z(\psi(y))\,\frac{d\psi}{dy}(y) = -p_z(z)\,\frac{d\psi}{dy}(y). \tag{6.8}$$

Therefore, we can write for the general case of a monotonic function

$$p_y(y) = p_z(z)\left|\frac{d\psi}{dy}(y)\right|. \tag{6.9}$$

Since we know that the distribution in the latent space p_z is Gaussian, we found an analytical equation for the distribution in the observed space p_y. If $\boldsymbol{Y}$ is a vector-valued random variable in $\mathbb{R}^n$, we get an $n \times n$ Jacobian matrix

$$J(\boldsymbol{y}) = \left[\frac{\partial z_{i_1}}{\partial y_{i_2}}\right]_{i_1,i_2} = \left[\frac{\partial \psi(y_{i_1})}{\partial y_{i_2}}\right]_{i_1,i_2} \tag{6.10}$$

with $i_1 = 1, ..., n$ and $i_2 = 1, ..., n$ instead of $\frac{d\psi}{dy}(y)$, and thus

$$p_y(\boldsymbol{y}) = p_z(\boldsymbol{z})\left|\det J(\boldsymbol{y})\right|, \tag{6.11}$$

see [157]. For the same mapping in every dimension, we can write

$$J(\boldsymbol{y}) = \operatorname{diag}\frac{d\psi}{dy}(\boldsymbol{y}). \tag{6.12}$$

Now, if our warping function is monotonic, the optimization problem for determining the parameters of the warping function can be stated as

$$\begin{aligned}
\boldsymbol{\theta}^* &= \arg\min_{\boldsymbol{\theta}} -\sum_{j=1}^{m} \log p_y(\boldsymbol{y}_j) \\
&= \arg\min_{\boldsymbol{\theta}} -\sum_{j=1}^{m} \log\left(p_z(\boldsymbol{z}_j)\left|\det\left(\operatorname{diag}\frac{d\psi_\theta}{dy}(\boldsymbol{y}_j)\right)\right|\right) \\
&= \arg\min_{\boldsymbol{\theta}} -\sum_{j=1}^{m}\left(\log p_z(\psi_\theta(\boldsymbol{y}_j)) + \sum_{i=1}^{n_j}\log\left|\frac{d\psi_\theta}{dy}(y_{i,j})\right|\right)
\end{aligned} \tag{6.13}$$

with the log-likelihood in the latent space

$$\begin{aligned}
\log p_z(\psi_\theta(\boldsymbol{y}_j)) = &-\frac{n_j}{2}\log(2\pi) - \frac{1}{2}\log\left(\det k_z(\boldsymbol{x}_j, \boldsymbol{x}_j)\right) \\
&-\frac{1}{2}\left(\psi_\theta(\boldsymbol{y}_j) - m_z(\boldsymbol{x}_j)\right)^\top k_z(\boldsymbol{x}_j, \boldsymbol{x}_j)^{-1}\left(\psi_\theta(\boldsymbol{y}_j) - m_z(\boldsymbol{x}_j)\right)
\end{aligned} \tag{6.14}$$

and its gradient

$$\begin{aligned}
-\frac{\partial(\log p_y(\boldsymbol{y}_j))}{\partial\theta_k} = &\frac{1}{2}\left(\left(\frac{\partial\psi_\theta}{\partial\theta_k}(\boldsymbol{y}_j)\right)^\top k_z(\boldsymbol{x}_j, \boldsymbol{x}_j)^{-1}\left(\psi_\theta(\boldsymbol{y}_j) - m_z(\boldsymbol{x}_j)\right)\right. \\
&\left. + \left(\psi_\theta(\boldsymbol{y}_j) - m_z(\boldsymbol{x}_j)\right)^\top k_z(\boldsymbol{x}_j, \boldsymbol{x}_j)^{-1}\left(\frac{\partial\psi_\theta}{\partial\theta_k}(\boldsymbol{y}_j)\right)\right) \\
&- \sum_{i=1}^{n_j}\left(\left(\frac{d\psi_\theta}{dy}(y_{i,j})\right)^{-1}\frac{\partial^2\psi_\theta}{\partial\theta_k\partial y}(y_{i,j})\right)
\end{aligned} \tag{6.15}$$

with n_j being the number of data points of trajectory j and $k = 1, ..., l$ for l warping function parameters.

By transforming the observed data into the latent space, the observation error is warped too. The standard deviation of a Gaussian observation error in the observed space σ_y can be approximated in the latent space with

$$\sigma_z(y) = \sigma_y \left| \frac{d\psi_\theta}{dy}(y) \right|. \tag{6.16}$$

This approximation is particularly accurate if the standard deviation σ_y and the second derivative of the warping function at y are rather small. Considering this non-constant noise term in the latent space, a weighted least squares regression with the diagonal weight matrix

$$\boldsymbol{W}_j = \sigma_y^{-2} \boldsymbol{V}_j, \tag{6.17}$$

where

$$\boldsymbol{V}_j = \text{diag}(V_{1,j}, ..., V_{n_j,j}) \text{ with } V_{i,j} = \left(\frac{d\psi_\theta}{dy}(y_{i,j}) \right)^{-2}, \tag{6.18}$$

can be applied to determine the weights of the basis functions

$$\begin{aligned} \hat{\boldsymbol{\beta}}_j &= \left(\boldsymbol{\Phi}_{z;j}^\top \boldsymbol{W}_j \boldsymbol{\Phi}_{z;j} \right)^{-1} \boldsymbol{\Phi}_{z;j}^\top \boldsymbol{W}_j \boldsymbol{z}_j \\ &= \left(\boldsymbol{\Phi}_{z;j}^\top \frac{1}{\sigma_y^2} \boldsymbol{V}_j \boldsymbol{\Phi}_{z;j} \right)^{-1} \boldsymbol{\Phi}_{z;j}^\top \frac{1}{\sigma_y^2} \boldsymbol{V}_j \boldsymbol{z}_j \\ &= \left(\boldsymbol{\Phi}_{z;j}^\top \boldsymbol{V}_j \boldsymbol{\Phi}_{z;j} \right)^{-1} \boldsymbol{\Phi}_{z;j}^\top \boldsymbol{V}_j \psi_\theta(\boldsymbol{y}_j) \end{aligned} \tag{6.19}$$

with

$$\boldsymbol{\Phi}_{z;j} = \boldsymbol{\Phi}_z(\boldsymbol{x}_j) = \left[\phi_{z;1}(\boldsymbol{x}_j), ..., \phi_{z;p}(\boldsymbol{x}_j) \right]. \tag{6.20}$$

Note that the weights can be determined without knowing σ_y. After estimating the weights, the sample mean $\hat{\boldsymbol{\mu}}_{\hat{\beta}}$ and the sample covariance matrix $\hat{\boldsymbol{\Sigma}}_{\hat{\beta}}$ of the estimated weights $\hat{\boldsymbol{B}} = [\hat{\boldsymbol{\beta}}_1, ..., \hat{\boldsymbol{\beta}}_m]$ can be computed. The standard deviation of the observation error σ_y can be approximated by

$$\hat{\sigma}_y = \sqrt{\frac{1}{m} \sum_{j=1}^{m} \frac{1}{n_j - p} \sum_{i=1}^{n_j} \left(\psi_\theta^{-1}(\boldsymbol{\Phi}_{z;j} \hat{\boldsymbol{\beta}}_j) - \boldsymbol{y}_j \right)^2}, \tag{6.21}$$

where p is the number of basis functions. This removes all parameters from the mean and covariance function and thus reduces the computational complexity significantly. Now, the mean function in Equation 6.4 and the covariance function in Equation 6.5 are completely specified.

In order to determine the maximum polynomial order of the basis functions, we can use the main parts of the procedure presented in the previous chapter. The training trajectories are again split into training ($\approx 70\%$) and validation ($\approx 30\%$) trajectories. Now, we have to solve an optimization task for every chosen polynomial order. Therefore, the steps are as follows:

(1) Choose maximum polynomial order q.
(2) Determine the optimized free parameters $\hat{\boldsymbol{\theta}}$ of the warping function by minimizing the sum of all training trajectories' negative log-likelihoods in the observed space, see Equation 6.13:

(a) Choose the free parameters $\boldsymbol{\theta}$.
(b) Infer the mean function $m_z(x)$ and covariance function $k_{z,IGP}(x, x')$ from the warped training trajectories based on the chosen free parameters.
(c) Estimate the standard deviation of the observation error σ_y by Equation 6.21.
(d) Compute the sum of all training trajectories' negative log-likelihoods with $m_z(x)$ and $k_z(x, x') = k_{z,IGP}(x, x') + \delta\, \sigma_y^2 \left(\frac{d\psi_\theta}{dy}(y)\right)^2$.

(3) Infer the mean function $m_z(x)$ and covariance function $k_{z,IGP}(x, x')$ from the warped training trajectories based on the optimized free parameters $\hat{\boldsymbol{\theta}}$.
(4) Estimate the standard deviation of the observation error σ_y by Equation 6.21 based on the optimized free parameters $\hat{\boldsymbol{\theta}}$.
(5) Evaluate the negative log-likelihood sum of all validation trajectories in the observed space by using $m_z(x)$ and $k_z(x, x') = k_{z,IGP}(x, x') + \delta\, \sigma_y^2 \left(\frac{d\psi_\theta}{dy}(y)\right)^2$ as the mean function and covariance function in the latent space.

After repeating the five steps for different polynomial orders, the order that minimizes the summed negative log-likelihoods of all validation trajectories in the observed space is eventually used. The design variable that minimizes our objective function is the same for using the mean or the sum of all negative log-likelihoods as our objective.

After the basis functions are chosen, the warping function's parameters are optimized, and the mean and covariance functions are derived from the entire training set (training and validation trajectories), we can compute the latent mean by $m_z(x)$ and, for example, the symmetric 95% credible region in the latent space with

$$z^{m\pm 1.96\sigma}(x) = m_z(x) \pm 1.96\sqrt{k_z(x, x)}. \tag{6.22}$$

The corresponding median $\tilde{m}_y(x)$ and the 95% credible region in the observed space are determined by

$$\tilde{m}_y(x) = \psi_\theta^{-1}(m_z(x)) \tag{6.23}$$

and

$$y^{\tilde{m}\pm 1.96\sigma}(x) = \psi_\theta^{-1}\left(m_z(x) \pm 1.96\sqrt{k_z(x, x)}\right), \tag{6.24}$$

respectively [133].

When we collect the data $\boldsymbol{y}_+$ observed at $\boldsymbol{x}_+$ from the monitored system, we can compute the conditional distribution in the latent space in order to update the Gaussian process' prediction by

$$\begin{aligned} f_z(x) \mid \boldsymbol{x}_+, \boldsymbol{y}_+, \hat{\boldsymbol{\theta}} \sim \mathcal{N}(&m_z(x) + k_z(x, \boldsymbol{x}_+) k_z(\boldsymbol{x}_+, \boldsymbol{x}_+)^{-1} \left(\psi_\theta(\boldsymbol{y}_+) - m_z(\boldsymbol{x}_+)\right), \\ &k_z(x, x) - k_z(x, \boldsymbol{x}_+) k_z(\boldsymbol{x}_+, \boldsymbol{x}_+)^{-1} k_z(x, \boldsymbol{x}_+)^\top). \end{aligned} \tag{6.25}$$

6.3 Predicting crack growth in an infinite plate

As a first example, we apply the proposed method to an academic problem, a pre-cracked infinite plate, for which the governing equations are known. We consider the infinite plate data set which we described in Section 3.1. As a recap: The governing equation is explicitly given by

$$a(N) = \left(\frac{(2-\alpha) C \Delta\sigma_\infty^\alpha \pi^{\alpha/2}}{2} N + a_0^{\frac{2-\alpha}{2}}\right)^{\frac{2}{2-\alpha}}, \tag{6.26}$$

where a is the crack length, N the number of cycles, $\Delta\sigma_\infty$ the remotely applied stress range, a_0 the initial crack length, and C and α two material parameters. We generated 50 crack growth trajectories by setting $\Delta\sigma_\infty = 48.26\,\text{MPa}$, $a_0 = 9 \times 10^{-3}\,\text{m}$, $\alpha = 2.9$, and sampling C from a normal distribution with $\mu_C = 8.7096 \times 10^{-11}$ and $\sigma_C = 1.519 \times 10^{-11}$ (C with $[da/dN] = \text{m/cycle}$ and $[\Delta K_I] = \text{MPa}\sqrt{\text{m}}$) for each trajectory. In order to represent the measurement noise, an observation error with a variance of $\sigma_y^2 = 0.16 \times 10^{-6}\,\text{m}^2$ was added to the crack lengths. The data points of each trajectory were computed every $2,000$ cycles and stop before a crack length longer than $0.05\,\text{m}$ is reached. In order to infer a Gaussian process model from this data set, we split it into training, validation, and test trajectories. Figure 6.1a displays the subsets.

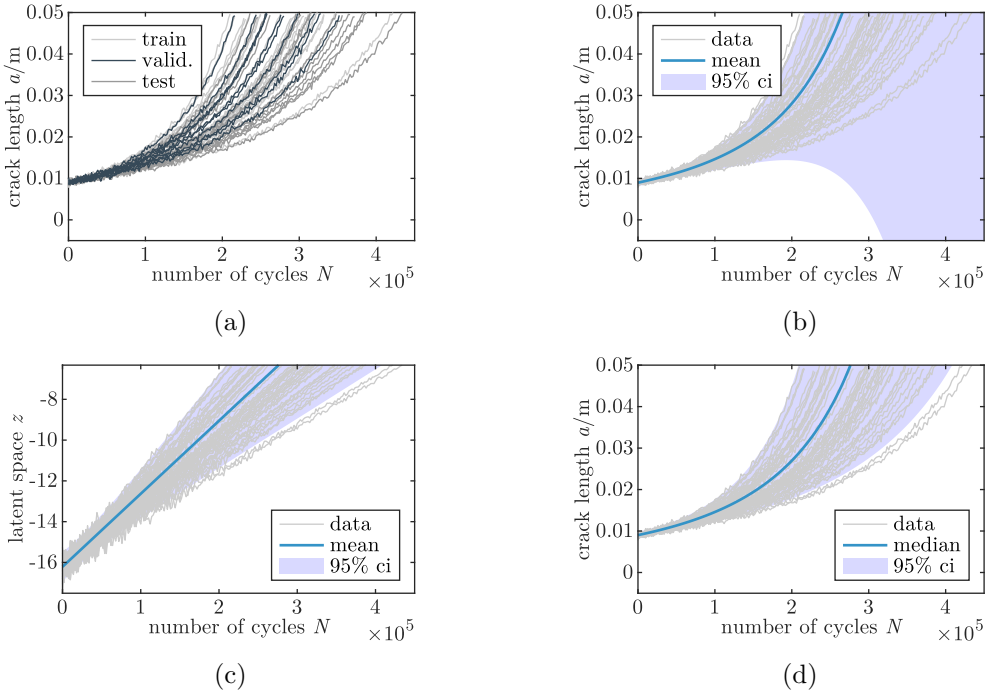


Figure 6.1: (a) Crack growth trajectories, (b) GP in the observed space, (c) warped GP in the latent space, and (d) in the observed space

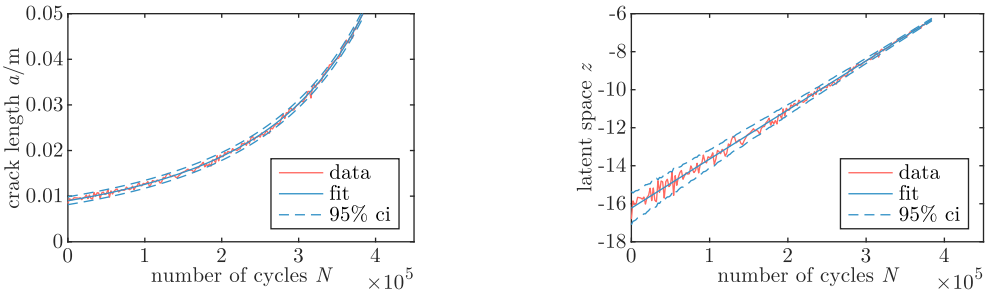
First, we infer a Gaussian process model without a warping function from the data set using polynomial basis functions. In contrast to the Virkler data set, where the numbers of cycles are measured for a fixed incremental crack length, we now are concerned with crack lengths measured for a fixed step size of the number of cycles. Therefore, the physics-informed basis functions used in the two previous chapters are not applied in this example. In order to select the maximum polynomial order, we follow the four steps of the procedure presented in Section 5.5, which results in $\hat{q}_{stand} = 5$. The determined Gaussian process with its mean function and symmetric 95% credible region is shown in Figure 6.1b. It can be seen that the credible region reaches negative values, which is non-physical since crack lengths can only assume non-negative numbers.

In the second case, we consider the warping function of Equation 6.2. The procedure for determining the maximum polynomial order presented in Section 6.2 is followed. We optimize the free parameter θ_1 with respect to Equation 6.13 for the maximum polynomial

orders $q = \{1, 2, 3\}$. For the infinite plate example, the resulting polynomial order should be $q^* = 1$ since the inner part of Equation 6.26 represents a Gaussian process with straight lines. This is the case because C is normally distributed, and the formulation between the brackets in Equation 6.26 describes straights. Furthermore, the optimization of the warping function should lead to the inverse of the outer exponent $\theta_1^* = \frac{2-\alpha}{2} = -0.45$.

In our example, the procedure leads to a maximum polynomial order of $\hat{q}_{warp} = 1$ and an optimized value of $\hat{\theta}_1 = -0.4489$ with a relative error of 0.25 % compared to the exact value. The difference compared to the analytical solution results from approximating the observation error's standard deviation in the latent space with Equation 6.16. By contrast, if we remove the added noise from the data, the optimization leads to the analytical solution $\hat{\theta}_1 = \theta_1^* = -0.45$. The resulting Gaussian process in the latent space and the corresponding warped Gaussian process, which is mapped to the observed space by the inverse warping function, are shown in Figures 6.1c and 6.1d, respectively. Figure 6.1d shows that the credible region of the warped Gaussian process assumes only positive values and surrounds the training data also on the right-hand side. Figure 6.1c additionally shows that the combination of the warping function and the choice of the polynomial basis functions with orders 0 and 1 leads to the desired solution. This is because, on the one hand, the optimizer tries to warp the data such that the trajectories follow a normal distribution. On the other hand, the simulated data are straightened to fit the trajectories by the given basis functions in the latent space.

Figure 6.2 shows the effect on the observation error when a trajectory is warped. The observation error (Figure 6.2a) is warped too, leading to a non-constant standard deviation of the observation error over x in the latent space, see Figure 6.2b. For higher cycle numbers, the standard deviation in the latent space σ_z decreases. The proposed approach takes this into account by applying weighted least squares regression and approximating σ_z according to Equation 6.16. Since Equation 6.16 is dependent on the observed data points $\boldsymbol{y}_j$, which are subject to observation error, σ_z varies according to the observed data, see Figure 6.2b.



(a) (b)

Figure 6.2: Observation error in (a) observed space and (b) latent space.

Conditional Gaussian process based on crack length data

In the next step, we gradually condition the Gaussian process on currently observed data, updating the prediction for an unseen trajectory according to Equation 6.25. In order to compare the two models, the negative log-likelihood and the absolute error between the crack length predicted at the trajectory's last number of cycles and the latest realized crack length are computed for every prediction step. The results are summarized in Table 6.1 by evaluating

the mean of all negative log-likelihoods and the mean of all absolute errors. The warped Gaussian process performs better than the standard one for both metrics.

Table 6.1: Mean negative log-likelihood $\bar{\mathcal{L}}$, MAE, and training time of the standard and warped Gaussian process for predicting the crack growth in an infinite plate.

Data set	**Model**	$\bar{\mathcal{L}}$		**MAE**		**Training time** in s	
crack growth simulation predict: $a(N_{end,j})$	standard IGPM	−3.6		14.7 mm		4.1	
	warped IGPM	−4.7	−32.2%	6.8 mm	−53.6%	39.9	+866%

Figure 6.7 shows the Gaussian processes' predictions for the shortest and the longest test trajectories conditioned on current data observed up to 40,000 cycles. The test trajectories are plotted in red, and the currently observed data is depicted in black. Additionally, the predicted mean and median are displayed in blue, and the symmetric 95% credible regions in light blue. Regarding the long trajectory, the updated mean function of the standard Gaussian

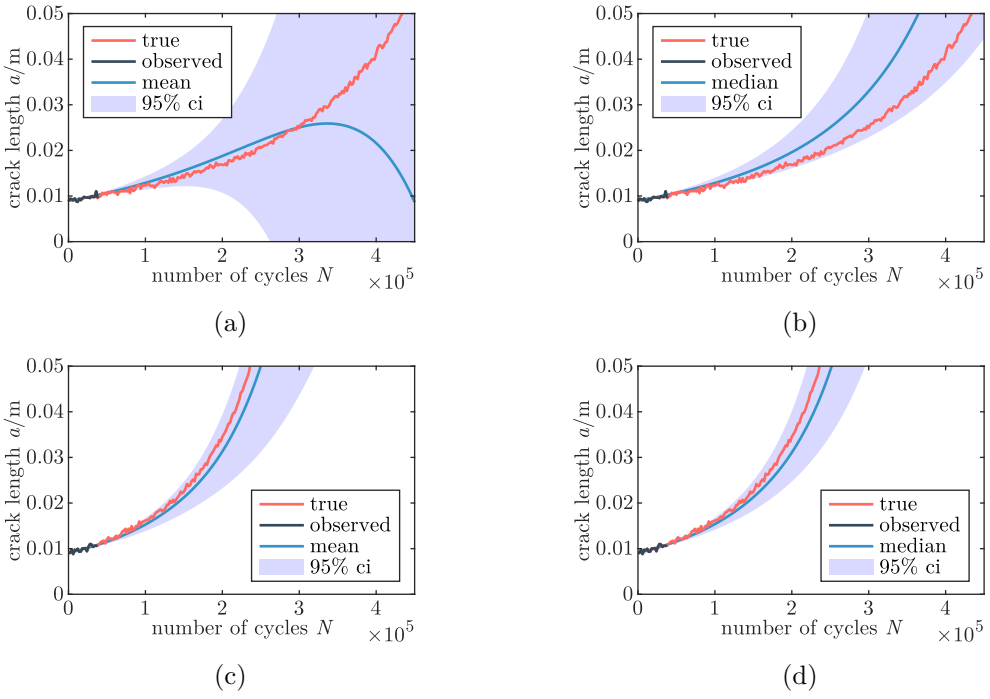


Figure 6.3: Crack growth prediction of the standard (left) and warped Gaussian process (right) for the longest (a & b) and shortest (c & d) trajectories at 40,000 cycles.

process becomes negative for larger numbers of cycles since the current data has a relatively flat trend which in combination with the assumed basis functions results in such predictions. By comparison, the warped Gaussian process in Figure 6.3b shows strictly positive predictions. Furthermore, we can see that the credible region of the standard Gaussian process for the long test trajectory still includes negative values even though it is conditioned on observed data, see Figure 6.3a. By contrast, Figure 6.3b shows that the warped Gaussian process' credible region embraces only positive values. With more available data, both credible regions narrow

down. However, the standard Gaussian process considers negative crack lengths within its 95% credible region also for predictions later in time. By contrast, the warped Gaussian process predicts strictly positive values, and its credible region narrows quicker.

In contrast to the results for the long trajectory, the predictions of the standard and the warped Gaussian processes for the short trajectory show similar quantities, see Figures 6.3c and 6.3d. In this case, the standard Gaussian process' symmetric 95% credible region excludes negative values. However, since the prediction relies on a normal distribution, the distribution related to the ordinate axis is symmetric around the mean. This is not the case for the warped Gaussian process. While the warped Gaussian process' median is similar to the mean of the standard Gaussian process, its 95% credible region, which is transformed by the inverse warping function to the observed space, is smaller.

Additionally, the times for training a standard Gaussian process with a given polynomial order of $\hat{q}_{stand} = 5$ and a warped Gaussian process with $\hat{q}_{warp} = 1$ are evaluated. A starting point of $\theta_0 = -0.1$ is chosen for the training process of the warped Gaussian process. Table 6.1 shows that training a warped Gaussian process needs longer than inferring a standard one since the parameter of the warping functions needs to be additionally optimized besides inferring the mean and covariance function. Its training time is about ten times larger than using a standard Gaussian process. The time depends, of course, on the starting point. It decreases if the starting point is chosen closer to the optimum.

Using the Box-Cox transformation for warping our data leads to approximations close to the analytical solution for the infinite plate example. However, using this transformation results in imaginary numbers if θ_1 is not an integer and y is negative as indicated by the missing lines for $\theta_1 = -0.5$ and $\theta_1 = 0.2$ in Figure 6.4a. Additionally, if θ_1 is an even number, the warping function becomes non-monotonic over $y \in \mathbb{R}$, violating our assumption. As presented earlier, we first compute our predictions in the latent space and second map them by the inverse warping function to the observed space. Since the inverse of the Box-Cox transformation reads

$$\psi_\theta^{-1}(z) = y = (\theta_1 z + 1)^{\frac{1}{\theta_1}} = \sqrt[\theta_1]{\theta_1 z + 1}, \tag{6.27}$$

we must ensure that $z \geq -\frac{1}{\theta_1}$ for $\theta_1 > 0$ and $z \leq -\frac{1}{\theta_1}$ for $\theta_1 < 0$ (both with $\theta_1 \neq 0$ and $\theta_1 \neq 1$). In the infinite plate example, we do not encounter any of the before-mentioned problems as our data is strictly positive ($y > 0$) and our predictions in the latent space comply with the requirement $z \leq -\frac{1}{\theta_1}$ ($\hat{\theta}_1 = -0.4489$).

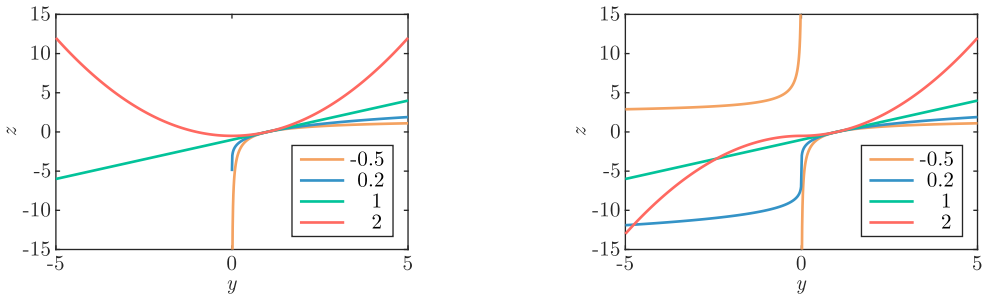


Figure 6.4: Box-Cox transformation functions for different θ_1 values based on (a) the original and (b) the modified Box-Cox transformation.

Although the Box-Cox transformation can be useful even in situations where no power transformation can produce normality exactly [32], these constraints limit the applicability of the Box-Cox transformation. Therefore, Bickel and Doksum [14] presented the modified Box-Cox transformation function

$$\psi(y) = z = \frac{\operatorname{sgn}(y)\,|y|^{\theta_1} - 1}{\theta_1} \text{ for } \theta_1 > 0, \tag{6.28}$$

which is also used in [118]. This modification leads to strictly monotonic warping functions over $y \in \mathbb{R}$ also if θ_1 becomes an even number, as shown in Figure 6.4b. Yet, the one-sided limits become

$$\lim_{y \to 0^+} \overset{+1}{\operatorname{sgn}(y)} \frac{1}{\theta_1} \left(\frac{1}{|y|^{-\theta_1}} - 1 \right) \overset{\infty}{\to} -\infty \tag{6.29}$$

and

$$\lim_{y \to 0^-} \overset{-1}{\operatorname{sgn}(y)} \frac{1}{\theta_1} \left(\frac{1}{|y|^{-\theta_1}} - 1 \right) \overset{\infty}{\to} +\infty \tag{6.30}$$

for $\theta_1 < 0$, resulting in a discontinuity at $y = 0$ as also shown in Figure 6.4b. Therefore, we must constrain θ_1 to be positive, ensuring our warping function complies with the assumptions made for our approach.

The infinite plate example shows that the advantage of warped Gaussian processes is particularly apparent for long trajectories. One possible reason for this is that long trajectories have a relatively small slope at the beginning. Since the standard Gaussian process relies on polynomials of order 5, the possible functions of the conditional distribution can still become negative. Yet, one drawback of warped Gaussian processes is that the parameters of the warping function need to be optimized, leading to more computational effort. Even though the Box-Cox transformation function leads to the analytical solution and physical predictions for the presented infinite plate example, only its modified version stated in Equation 6.27 complies with our assumptions when $y \in \mathbb{R}$.

6.4 Estimating observation errors and future probabilities of failure

Sometimes gathering degradation trajectories is tedious, including high expenses for experiments and equipment. Collecting several degradation lines can even be impossible if, for example, the test rig is massive, as is the case for full-scale fatigue tests. Simulations often provide a good understanding of the mechanical system's degradation and yield, therefore, valuable information. By considering the parameters' uncertainties, simulations can be used to quantify the scatter of the quantity of interest, such as the fatigue behavior. In these cases, we can extract valuable information from simulation data and integrate it into Gaussian processes. More specifically, after conducting Monte-Carlo computations to simulate various ways of how the mechanical system degrades, the simulated degradation trajectories can be used to derive a Gaussian process model. A drawback of doing so is that the Gaussian process entirely relies on the simulated data. Therefore, the simulation must have shown that it accurately describes the real mechanical system's behavior and includes all uncertainty sources associated with the problem. Luckily, the fatigue and fracture mechanics equations explained in the fundamentals chapter are well established. Researchers have shown that they are suitable for describing the

crack growth in structures. Moreover, values for estimating the parameters' uncertainties are found in the literature and can be used to simulate various degradation trajectories. Yet, the observation error is not included in simulated data and differs from use case to use case. This is why the observation error should be estimated from experimentally gathered data directly.

If the degradation behavior can be observed, the standard deviation of the observation error can be estimated in the following way. In the case of a standard Gaussian process, we can add the noise term $\delta\,\sigma_y^2$ to the main diagonal of the covariance function inferred from simulated trajectories $k_{y,IGP}(x,x')$ so that the resulting covariance function becomes $k_y(x,x') = k_{y,IGP}(x,x') + \delta\,\sigma_y^2$. After observing a part of the trajectory to predict $(\boldsymbol{x}_+, \boldsymbol{y}_+) \in \mathbb{R}^{n_+}$, we can minimize the negative log-likelihood

$$\begin{aligned} -\log p_y(\boldsymbol{y}_+) =& \frac{n_+}{2}\log(2\pi) + \frac{1}{2}\log(\det(k_{y,IGP}(\boldsymbol{x}_+,\boldsymbol{x}_+) + \sigma_y^2\boldsymbol{I})) \\ &+ \frac{1}{2}\left(\boldsymbol{y}_+ - m_y(\boldsymbol{x}_+)\right)^\top \left(k_{y,IGP}(\boldsymbol{x}_+,\boldsymbol{x}_+) + \sigma_y^2\boldsymbol{I}\right)^{-1}\left(\boldsymbol{y}_+ - m_y(\boldsymbol{x}_+)\right) \end{aligned} \tag{6.31}$$

to determine the observation error's variance σ_y^2. The gradient with respect to σ_y^2 is given by

$$\begin{aligned} -\frac{\partial(\log p_y(\boldsymbol{y}_+))}{\partial\sigma_y^2} =& \frac{1}{2}\mathrm{tr}\left((k_{y,IGP}(\boldsymbol{x}_+,\boldsymbol{x}_+) + \sigma_y^2\boldsymbol{I})^{-1}\right) \\ &- \frac{1}{2}\left(\boldsymbol{y}_+ - m_y(\boldsymbol{x}_+)\right)^\top \left(k_{y,IGP}(\boldsymbol{x}_+,\boldsymbol{x}_+) + \sigma_y^2\boldsymbol{I}\right)^{-2} \\ &\left(\boldsymbol{y}_+ - m_y(\boldsymbol{x}_+)\right). \end{aligned} \tag{6.32}$$

Alternatively, we can determine the observation error's variance by

$$\hat{\sigma}_y^2 = \frac{1}{n_+ - p}\left(\boldsymbol{\Phi}(\boldsymbol{x}_+)\hat{\boldsymbol{\beta}}_+ - \boldsymbol{y}_+\right)^2, \tag{6.33}$$

where

$$\hat{\boldsymbol{\beta}}_+ = \left(\boldsymbol{\Phi}(\boldsymbol{x}_+)^\top\boldsymbol{\Phi}(\boldsymbol{x}_+)\right)^{-1}\boldsymbol{\Phi}(\boldsymbol{x}_+)^\top\boldsymbol{y}_+, \tag{6.34}$$

which is only meaningful when the number of gathered data points n_+ is larger than the number of basis functions p.

We can apply the same concept to warped Gaussian processes by adding a noise term to the main diagonal of the derived covariance function in the latent space

$$k_z(x,x') = k_{z,IGP}(x,x') + \delta\,\sigma_y^2\left(\frac{d\psi_\theta}{dy}(y_{i,j})\right)^2. \tag{6.35}$$

The observation error's variance σ_y^2 can be optimized by minimizing the negative log-likelihood of the currently observed data

$$-\log p_y(\boldsymbol{y}_+) = -\log p_z(\psi_\theta(\boldsymbol{y}_+)) - \sum_{i=1}^{n_+}\log\left|\frac{d\psi_\theta}{dy}(y_{i,+})\right| \tag{6.36}$$

with

$$
\begin{aligned}
-\log p_z(\psi_\theta(\boldsymbol{y}_+)) =& \frac{n_+}{2}\log(2\pi) \\
&+ \frac{1}{2}\log\left(\det\left(k_{z,IGP}(\boldsymbol{x}_+,\boldsymbol{x}_+) + \sigma_y^2\,\mathrm{diag}\left(\left(\frac{d\psi_\theta}{dy}(\boldsymbol{y}_+)\right)^2\right)\right)\right) \\
&+ \frac{1}{2}\left(\psi_\theta(\boldsymbol{y}_+) - m_z(\boldsymbol{x}_+)\right)^\top \\
&\left(k_{z,IGP}(\boldsymbol{x}_+,\boldsymbol{x}_+) + \sigma_y^2\,\mathrm{diag}\left(\left(\frac{d\psi_\theta}{dy}(\boldsymbol{y}_+)\right)^2\right)\right)^{-1} \\
&\left(\psi_\theta(\boldsymbol{y}_+) - m_z(\boldsymbol{x}_+)\right)
\end{aligned}
\tag{6.37}
$$

and its gradient with respect to σ_y^2

$$
\begin{aligned}
&-\frac{\partial(\log p_y(\boldsymbol{y}_+))}{\partial\sigma_y^2} = \\
&\frac{1}{2}\mathrm{tr}\left(\left(k_{z,IGP}(\boldsymbol{x}_+,\boldsymbol{x}_+) + \sigma_y^2\,\mathrm{diag}\left(\left(\frac{d\psi_\theta}{dy}(\boldsymbol{y}_+)\right)^2\right)\right)^{-1}\mathrm{diag}\left(\left(\frac{d\psi_\theta}{dy}(\boldsymbol{y}_+)\right)^2\right)\right) \\
&-\frac{1}{2}\left(\psi_\theta(\boldsymbol{y}_+) - m_z(\boldsymbol{x}_+)\right)^\top\left(k_{z,IGP}(\boldsymbol{x}_+,\boldsymbol{x}_+) + \sigma_y^2\,\mathrm{diag}\left(\left(\frac{d\psi_\theta}{dy}(\boldsymbol{y}_+)\right)^2\right)\right)^{-2} \\
&\left(\psi_\theta(\boldsymbol{y}_+) - m_z(\boldsymbol{x}_+)\right).
\end{aligned}
\tag{6.38}
$$

The observation error's standard deviation can also be estimated with Equation 6.21. With the estimated standard deviation of the observation error, we are now able to compute the conditional distribution.

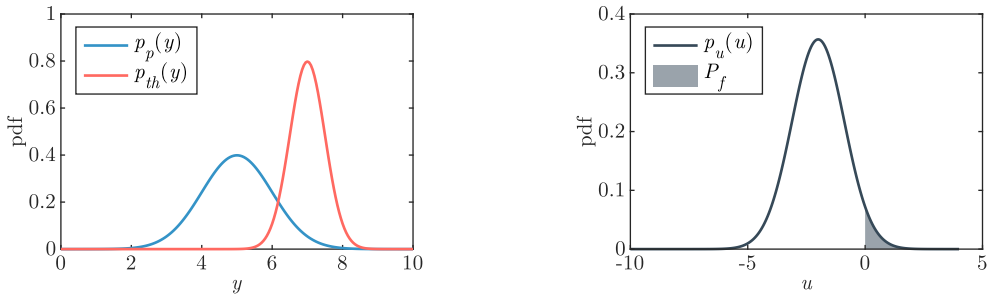
In many cases, operators base their decision on probability quantities rather than likelihoods or probability density values. Often the probability of failure is used to decide when mechanical systems shall be maintained. If the probability density function of the damage threshold $p_{th}(y)$, e.g., the critical crack length, and the predictive probability density function $p_p(y)$ follow normal distributions ($p_{th}(y) = \mathcal{N}(y|\mu_{th},\sigma_{th}^2)$ and $p_p(y) = \mathcal{N}(y|\mu_p,\sigma_p^2)$), see Figure 6.5a, the probability of failure can be analytically computed. We denote Y_p as the random variable of the prediction and Y_{th} as the random variable of the threshold. In order to determine the probability of the prediction being larger than the threshold, we introduce the auxiliary random variable $U = Y_p - Y_{th}$ with $p_u(u) = \mathcal{N}(u|\mu_p - \mu_{th}, \sigma_p^2 + \sigma_{th}^2)$, which is shown in Figure 6.5b. The probability of failure, i.e., the probability that the predicted value is larger than the threshold or that $U > 0$, is

$$
P_f = \Pr(U > 0) = \int_0^\infty p_u(u)du = 1 - \int_{-\infty}^0 p_u(u)du = 1 - P_u(0) \tag{6.39}
$$

with

$$
P_u(0) = \frac{1}{\sqrt{2\pi(\sigma_p^2 + \sigma_{th}^2)}}\int_{-\infty}^0 \exp\left(-\frac{(u - \mu_p + \mu_{th})^2}{2(\sigma_p^2 + \sigma_{th}^2)}\right)du, \tag{6.40}
$$

see Figure 6.5b.



(a) (b)

Figure 6.5: (a) Probability density function of the prediction $p_p(y)$ and the threshold $p_{th}(y)$, and (b) the probability density function of the auxiliary variable $p_u(u)$ and the probability of failure P_f.

In the case of warped Gaussian processes with strictly increasing warping functions, the same Equations are valid in the latent space z. If the warping function is strictly decreasing, the probability of failure is computed by $P_f = P_u(0)$.

The methods for estimating the observation error's standard deviation and computing the probability of failure are applied to the already investigated infinite plate. As in the previous section, we model the latent space by the inner part of Equation 6.26 and thus by a polynomial of order $q = 1$, and define the distribution of the coefficients by $\mathcal{N}(\boldsymbol{\beta}|\boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_\beta)$ with

$$\boldsymbol{\mu}_\beta = \left[a_0^{\frac{2-\alpha}{2}}, \quad \frac{(2-\alpha)\mu_C \sigma_\infty^\alpha \pi^{\frac{\alpha}{2}}}{2}\right]^\top \tag{6.41}$$

and

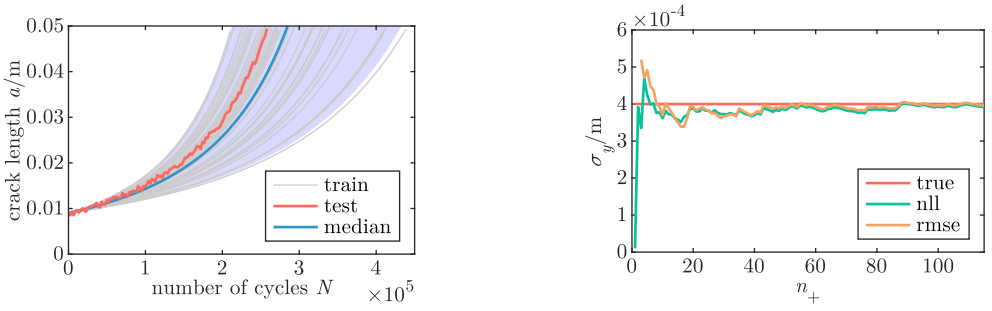
$$\boldsymbol{\Sigma}_\beta = \begin{bmatrix} 0 & 0 \\ 0 & \left(\frac{(2-\alpha)\sigma_C \sigma_\infty^\alpha \pi^{\frac{\alpha}{2}}}{2}\right)^2 \end{bmatrix}. \tag{6.42}$$

Therefore, the underlying Gaussian process is defined by

$$\mathcal{N}(f_z(x) \mid \boldsymbol{\phi}_z(x)^\top \boldsymbol{\mu}_\beta, \boldsymbol{\phi}_z(x)^\top \boldsymbol{\Sigma}_\beta \boldsymbol{\phi}_z(x)) \tag{6.43}$$

with $\boldsymbol{\phi}_z(x) = [x^0, x^1]^\top$.

The grey lines in Figure 6.6a are produced by sampling the coefficients from the normal distribution and transforming the trajectories from the latent space to the observed space with the inverse warping function $\psi^{-1}(z) = z^{\frac{2}{2-\alpha}}$. Additionally, the median function is shown in blue, and the test trajectory, which is to be predicted, in red. Note that in contrast to the training trajectories, the test trajectory embeds observation error with a standard deviation of $\sigma_y = 0.4 \times 10^{-3}\,\mathrm{m}$, which is why we need to add the before-mentioned noise term to the covariance function. We gradually observe the test data points and optimize the observation error's standard deviation after every additional data point according to the two previously described methods. Figure 6.6b summarizes the results. The red line illustrates the true standard deviation, the green line the optimized standard deviation based on minimizing the negative log-likelihood, and the orange line the estimates based on Equation 6.21. The green and orange lines approach the true value for more and more data points and eventually converge. The outputs settle around the correct solution after about 20 data points. The method based

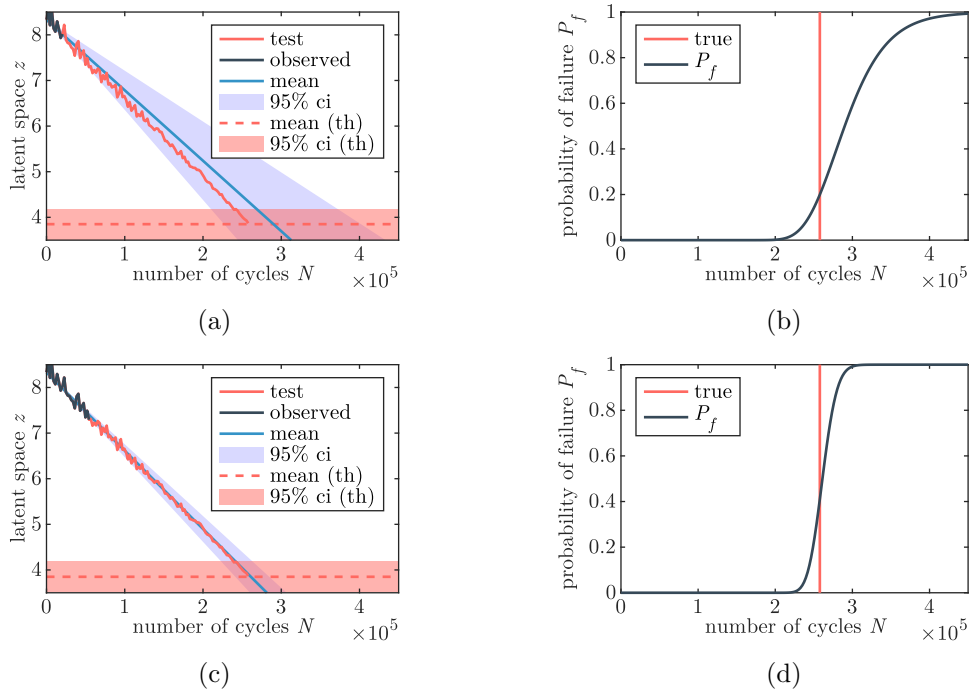


(a) (b)

Figure 6.6: (a) Data of the infinite plate example and (b) approximated standard deviation of the observation error.

on minimizing the negative log-likelihood outputs zero for one data point and estimates the standard deviation fairly well ($< 20\%$ error) after two data points. By contrast, the estimator based on Equation 6.21 produces similar estimates ($< 30\%$ error) after three data points since the number of data points has to be larger than the number of basis functions, which is in this case two.

Moreover, we can predict the probability of failure: For each future time step that we are interested in, we can compute the probability for which the predicted crack length is larger than the critical one. Since the warping function is strictly decreasing, we use $P_f = P_u(0)$.



(a) (b)

(c) (d)

Figure 6.7: Predictive mean, credible region, and threshold distribution for the prediction at (a) $N \approx 20,000$ and (c) $N \approx 54,000$. Last point of trajectory and predicted probability of failure at (b) $N \approx 20,000$ and (d) $N \approx 54,000$.

We further assume the critical crack length to be normally distributed in the latent space with $\mathcal{N}(z_{th}|\mu_{z,th}, \sigma^2_{z,th})$, where $\mu_{z,th} = \psi^{-1}(a_c)$ and $\sigma_{z,th} = \sigma_{a_c}\frac{d\psi}{dy}(a_c)$, with $a_c = 0.05\,\text{m}$ and $\sigma_{a_c} = 0.005\,\text{m}$. Figures 6.7a and 6.7c show the predictions and the threshold distribution in the latent space for two different time states. The corresponding failure probabilities over the number of cycles are depicted in Figures 6.7b and 6.7d, where the red line indicates the last data point of the test trajectory. The prediction allows the operator to know the probability of failure at every future time step based on the currently observed data. Again, we can see that the credible region of the predictive distribution narrows with more observed data points. At the same time, the s-shaped failure probability function over the number of cycles narrows down, indicating that the time to failure is known more precisely, see Figures 6.7b and 6.7d.

6.5 Predicting crack growth in an aerospace structure

The approach of warped Gaussian processes with integrated prior knowledge proposed in Section 6.2 is also applied to an aluminum panel that resembles a lower section of a civil aircraft wing. The investigated structure is $1,920\,\text{mm}$ long and $570\,\text{mm}$ wide. The centered elliptical armhole, shown in Figure 6.8, with a length of $135\,\text{mm}$ and width of $75\,\text{mm}$ is usually used to inspect the inner surfaces of the wing. Sixteen holes surround the armhole to attach a lid. In total, two equally manufactured specimens were loaded with the same sequence of four different flight types that were consecutively repeated until the structures eventually failed. The detailed procedure of the experiment can be found in Section 3.3.

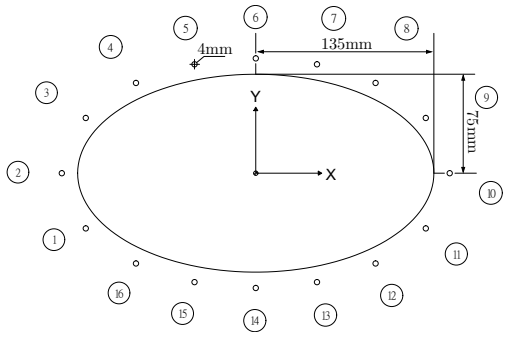


Figure 6.8: Elliptical armhole surrounded by 16 small holes.

The approach proposed in Section 6.2 infers the Gaussian process model from previously collected degradation trajectories. As large structural fatigue tests are usually carried out only once, previously measured degradation trajectories are missing. However, due to analytical equations and FEAs, much of the degradation behavior is understood a priori. In order to integrate this knowledge into Gaussian processes, we produce degradation trajectories by conducting virtual simulations and use them for inferring our Gaussian process. After explaining the simulation procedure in the following, we propose a method to infer the crack length present in the structure from strain gauge data. The values are finally used to compute the conditional Gaussian process, updating our predictions.

6.5.1 Simulating crack growth trajectories

In contrast to the infinite plate example, the specimens regarded in this section have no initial cracks. Therefore, we have to consider in our simulations the time of the structure being sound until a crack initiates. To simulate the structure's degradation, we aggregate the time-based loading program by rainflow-counting. Then, we first compute the fatigue life and second the crack growth. The procedure of simulating the degradation trajectories is schematically shown in Figure 6.9 and explained in detail in the following.

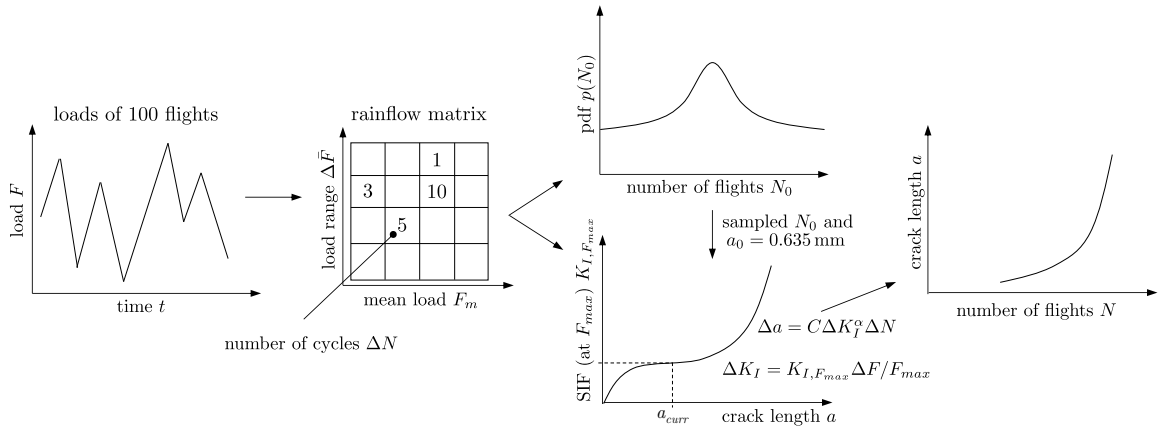


Figure 6.9: Schematic representation of how the crack growth in the aerospace structure is simulated, where F_{max} is the maximum force in the loading program.

Fatigue life. First, an FEA is executed to quantify the stresses in the structure. Two local hot spots, at the small holes 5 and 6 on the side towards the armhole, are found. In the following, we assume the crack to start at hole 6. However, the methodology can also be extended to consider cracks forming from multiple spots. By applying rainflow-counting and the Haigh diagram to the computed stresses at the hot spot of hole 6 corresponding to the various load levels, the stresses for the entire loading program can be mapped to amplitude stress blocks with a constant stress ratio of $R = -1.0$. The median number of cycles after a crack will occur can be determined using Miner's linear damage accumulation rule and a 50% S-N curve, which corresponds to our used material Al 2024–T351. In order to quantify

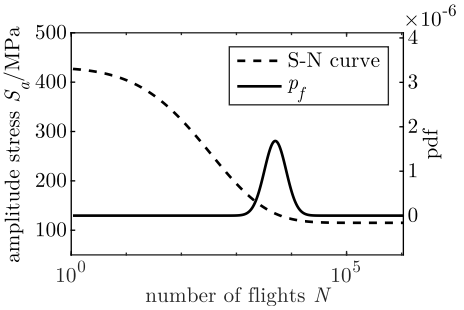


Figure 6.10: S-N curve and log_{10} normal distribution for the present loading program.

the uncertainty of the crack initiation, a material corresponding scatter parameter $s = 0.197$

of the S-N curve found in [50] is used. The log_{10} normal distribution and the S-N curve are shown in Figure 6.10. Now, the number of cycles for an initial crack N_0 can be sampled from the defined distribution. For detailed descriptions of the methods used in this paragraph, see Section 2.1.

Crack growth. Second, the crack growth in the structure is computed. Multiple crack computations are evaluated using XFEM to quantify the relationship between the SIF and the crack length a. The concept of XFEM is described in depth in Section 2.2. For our structure, we assume the crack to first propagate towards the armhole (crack length a_1) and then towards the edge of the structure (crack length a_2), see Figure 6.11. In total, 382 static XFEM analyses with

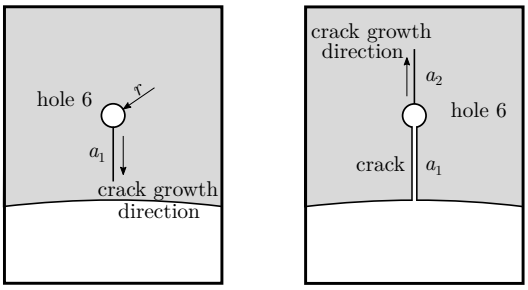
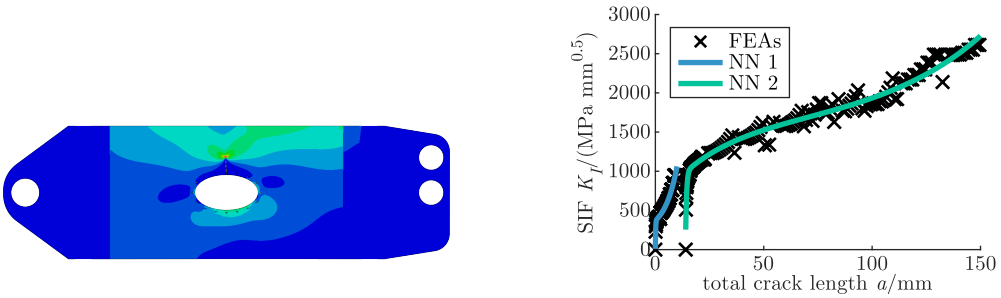


Figure 6.11: Assumption for crack propagation where crack 1 first propagates towards the armhole (left) and crack 2 to the edge of the specimen afterwards (right).

different crack lengths and the maximum load F_{max} are evaluated to quantify the relationship between the crack length a and the SIF at the maximum load $K_{I,F_{max}}$. Figure 6.12a shows one of the 382 XFEM computations. Two separate neural networks, for a_1 and a_2, are trained to map the crack length onto the SIF, see Figure 6.12b.



(a) (b)

Figure 6.12: (a) XFEM computation and (b) trained neural networks (NN).

In this study, we define the total crack length a as

$$a = \begin{cases} a_1, & \text{if } a \leq 10\,\text{mm} \\ a_1 + 2r + a_2, & \text{if } a > 10\,\text{mm}\,, \end{cases} \tag{6.44}$$

where r is the radius of the small hole where the crack initiates. An initial crack length of $a_0 = 0.635\,\text{mm}$ which is, according to [121], the smallest crack length detectable by eddy current

testing, is assumed. In order to reduce computational time, the crack growth computations are simplified by applying rainflow-counting to the load steps of 100 flights in order to compute the load ranges $\Delta\bar{\boldsymbol{F}}$ and using 1/100 of their frequencies for each flight. In doing so, we average the severity of the flights and assume every flight to be equally destructive. We also set

$$\Delta K_I = K_{I,F_{max}} \frac{\Delta F}{F_{max}}. \tag{6.45}$$

The crack growth can then be computed by the well-known Paris law.

According to [148], crack growth is subject to great uncertainties. Therefore, C is assumed to be a random variable that is normally distributed with $\mu_C = 8.7096 \times 10^{-11}$ and $\sigma_C = 6.5680 \times 10^{-12}$ (determined from the crack growth data published by Virkler et al. [148] with $[da/dN]$ = m/cycle and $[\Delta K_I]$ = MPa$\sqrt{\text{m}}$). The material parameter α is set to $\alpha = 2.9$ according to Spencer et al. [136]. Moreover, the load ranges are varied according to a normal distribution with $\mathcal{N}(\epsilon_{\Delta F}|0, 0.05^2)$ and $\boldsymbol{\Delta F} = \Delta\bar{\boldsymbol{F}} \times (1 + \epsilon_{\Delta F})$. For computing different crack growth trajectories, a set of parameters (N_0, C, $\epsilon_{\Delta F}$) is sampled for each trajectory. Figure 6.13 shows the computed degradation trajectories. Note that the step in each trajectory results from the two different cracks, a_1 and a_2, and the added diameter of the small hole once crack 1 reaches the armhole. Now, a Gaussian process model can be inferred from those crack growth trajectories.

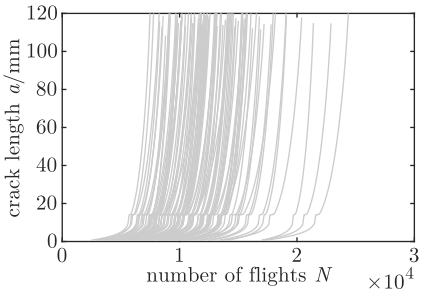


Figure 6.13: Computed degradation trajectories.

6.5.2 Inferring crack lengths from strain data

In order to compute the conditional distribution of a trained Gaussian process, the current state of the mechanical system and its uncertainty has to be known. In this application, the current crack length is not directly observed, as was the case for the infinite plate example. Therefore, we determine the state of the mechanical system by using the data gathered from the attached strain gauges. Since Bayesian inference offers a neat way to solve inverse problems which also considers uncertainties, we exploit it for our purposes in the following. With Bayes law, the crack length a can be inferred from the measured strains $\boldsymbol{\varepsilon}_{SG}$ by

$$p(a|\boldsymbol{\varepsilon}_{SG}) = \frac{p(\boldsymbol{\varepsilon}_{SG}|a)p(a)}{\int_{a_0}^{a_c} p(\boldsymbol{\varepsilon}_{SG}|a)p(a)da}, \tag{6.46}$$

where $p(a)$ is the prior distribution and $p(\boldsymbol{\varepsilon}_{SG}|a)$ the likelihood. For measuring multiple strains and assuming them to be measured independently, the probability density function of $\boldsymbol{\varepsilon}_{SG}$ for

s applied strain gauges given the crack length a becomes

$$p(\boldsymbol{\varepsilon}_{SG}|a) = p(\varepsilon_{SG;1}, ..., \varepsilon_{SG;s}|a) = \prod_{l=1}^{s} p(\varepsilon_{SG;l}|a). \tag{6.47}$$

By also assuming the prior $p(a)$ to be uniformly distributed within the bounds $a \in [a_0, a_c]$, the probability density function of a given all strain measurements becomes

$$p(a|\boldsymbol{\varepsilon}_{SG}) = \frac{\prod_{l=1}^{s} p(\varepsilon_{SG;l}|a)}{\int_{a_0}^{a_c} \prod_{l=1}^{s} p(\varepsilon_{SG;l}|a) da}. \tag{6.48}$$

In order to be able to use Equation 6.48, the likelihoods $p(\varepsilon_{SG;l}|a)$ with $l \in 1, ..., s$ must be known as they incorporate how the measured data $\boldsymbol{\varepsilon}_{SG}$ varies with the crack length a. Therefore, the strains of all XFEM computations (see Figure 6.12) are evaluated for each sensor position to quantify the relationship between the crack lengths and the strains. Then, the results are used to fit a neural network for each strain gauge position. Since we only consider strictly monotonic strain gauges, in total, seven neural networks $\varepsilon_{NN;l}(a)$ are trained. By assuming a normally distributed measurement error with a standard deviation of $\sigma_\varepsilon = 200\,\mu$m/m, the likelihoods $p(\varepsilon_{SG;l}|a)$ are completely defined and can be evaluated. However, in order to cancel out the bias term which might emerge from the difference between the FEA and the real measurement, only the relative changes of the strains due to a crack

$$\Delta\varepsilon_{NN,rel;l}(a) = \frac{\varepsilon_{NN;l}(a) - \varepsilon_{NN;l}(a_0 = 0)}{\varepsilon_{NN;l}(a_0 = 0)} \tag{6.49}$$

and

$$\Delta\varepsilon_{SG,rel;l} = \frac{\varepsilon_{SG;l} - \varepsilon_{SG;l}(a_0 = 0)}{\varepsilon_{SG;l}(a_0 = 0)} \tag{6.50}$$

are considered. Since at the beginning of each test settlement effects happen, the measurement of flight 500 is used as $\varepsilon_{SG;l}(a_0 = 0)$. The likelihood becomes

$$p(\Delta\varepsilon_{SG,rel;l}|a) = \frac{1}{\sqrt{2\pi \left(\frac{\sigma_\varepsilon}{\varepsilon_{NN,l}(a_0)}\right)^2}} \exp\left(-\frac{\left(\Delta\varepsilon_{SG,rel;l} - \Delta\varepsilon_{NN,rel;l}(a)\right)^2}{2\left(\frac{\sigma_\varepsilon}{\varepsilon_{NN;l}(a_0)}\right)^2}\right), \tag{6.51}$$

which we use instead of $p(\varepsilon_{SG;l}|a)$. The current crack length is determined by

$$\hat{a} = \arg\max p(a|\boldsymbol{\varepsilon}_{SG}) \tag{6.52}$$

and its variance by

$$\sigma_a^2 = \int_{a_0}^{a_c} (a - \mu_a)^2 p(a|\boldsymbol{\varepsilon}_{SG}) da. \tag{6.53}$$

with

$$\mu_a = \int_{a_0}^{a_c} a\, p(a|\boldsymbol{\varepsilon}_{SG}) da \tag{6.54}$$

Figures 6.14a and 6.14b show the crack lengths inferred from the measured strains and the corresponding crack lengths that test engineers visually obtained for both specimens. The inferred crack lengths for P03T01 closely match the inspected ones ($R^2 = 0.926$), whereas

the match of the P02T01 trajectory is further off ($R^2 = 0.656$). Figure 6.14a reveals a big step in the inspected data of specimen P02T01 at about 9,000 flights. It is likely that the test engineers did not detect the crack at the other side of the small hole (a_2) right when it started to grow. As the P03T01 specimen was tested after P02T01, the test engineers were already familiar with the type of structure and could therefore measure the crack lengths more accurately. Both crack growth behaviors closely resemble the simulations regarding locations, numbers, and crack growth rate.

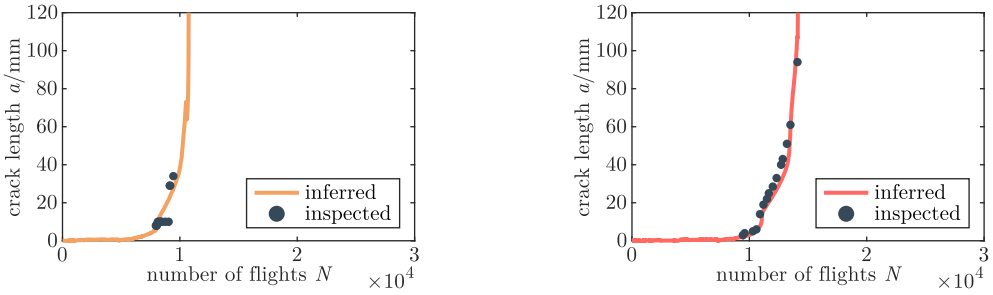


Figure 6.14: Inferred and visually observed crack lengths for specimen (a) P02T01 and (b) P03T01.

6.5.3 Enabling Gaussian processes for crack growth prediction

Training of the Gaussian process. Section 6.3 revealed that the Box-Cox transformation leads to straight lines in the latent space for the pre-cracked infinite plate. Straight lines are strictly monotonic, and thus their inverse Box-Cox transformations are as well. By choosing polynomials of order 0 and 1 (intersection and slope) in the latent space, we ensure that the crack growth is strictly monotonic, which agrees with the physics since cracks cannot become shorter while loading the structure. Therefore, we choose the basis functions accordingly. Additionally, we utilize the modified Box-Cox transformation function stated in Equation 6.27 and infer the Gaussian process model from the simulated training trajectories. Since no observation error is present in the simulation data, a non-weighted least squares regression is applied to determine the basis functions' weights. Due to the chosen basis functions, the applied opti-

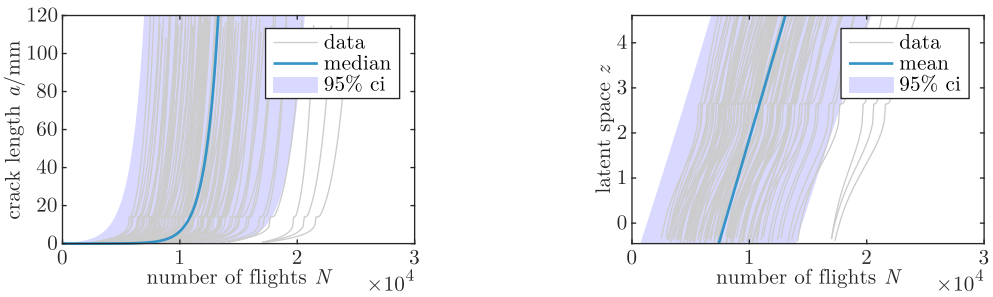


Figure 6.15: Trained Gaussian process in the (a) observed space and (b) latent space.

mizer tries not only to achieve a normal distribution in the latent space but also straight lines. The optimized solution is $\hat{\theta}_1 = 4.46 \times 10^{-8}$ which is close to a *log* transformation. For a *log* transformation, we know that the predicted crack lengths will solely assume positive values after transforming the predictions from the latent space to the observed space. The modeling error in the latent space $\hat{\sigma}_{z,m}$ approximated by the residuals is assumed to be constant in the latent space. Figures 6.15a and 6.15b show the training data and the determined Gaussian process in the observed and the latent space, respectively. The figures reveal that the trajectories in the latent space are nearly linear. Moreover, the mean function and the credible region in the observed space assume only positive values, which agrees with the physics. Figure 6.15a depicts the prediction before any data of the monitored structure is available.

Updating GP predictions. Although Figures 6.14a and 6.14b show the entire trajectories of the inferred crack lengths, during the test, the crack length is gradually determined. After each new flight, the current crack length and its uncertainty can be estimated based on Bayesian inference and used to compute the conditional Gaussian process, leading to an updated prediction. For doing so, all inferred crack lengths $\hat{\boldsymbol{a}}$ up to the current flight cycle and their estimated standard deviations $\hat{\boldsymbol{\sigma}}_a$ are transformed to the latent space using Equations 6.2 and 6.16, respectively. In this case, the variance of the total observation error in the latent space σ_z^2 is assumed to be the sum of the modeling error's variance $\sigma_{z,m}^2$ and the transformed variance of the observation error $\sigma_{z,a}^2$, i.e., $\sigma_z^2 = \sigma_{z,a}^2 + \sigma_{z,m}^2$. Furthermore, crack sizes smaller than 0.635 mm are ignored since the Gaussian process is trained on trajectories starting from that crack length. Then, the conditional Gaussian process is computed in the latent space using Equation 6.25. The updated prediction can be transformed into the observed space using the inverse warping function.

Figure 6.16 show the updated predictions for specimens P02T01 and P03T01 at different time states. Initially, the Gaussian process prediction is entirely based on the knowledge gained from analytical equations and FEAs. Once a crack length greater than 0.635 mm is inferred from the strain data, the median starts to change, and the credible region narrows down, leading to a more accurate prediction. Since the Gaussian process model is defined by a set of polynomial basis functions with orders 0 and 1, the step due to the two different crack regimes is not apparent in the prediction. Figures 6.16a and 6.16b show that the uncertainties up to the current state (black point) are relatively large. This is because we neglect crack lengths whose most likely length is below 0.635 mm. Moreover, the variances estimated based on Bayesian inference become in the approximately logarithmic latent space at small crack lengths rather large which is why the prediction is hardly affected. Therefore, actions based on crack lengths whose estimated mean is below 0.635 mm should be decided regarding the results of Bayesian inference. Figure 6.16 also shows that the predictions are strictly positive, which complies with the physics.

Figure 6.17 displays all predicted median values and credible intervals for the critical crack length $a_c \approx 120\,\text{mm}$. The x-axis resembles the current state up to which we have conducted the experiment. The y-axis shows the predicted and the realized number of cycles. The realized last numbers of cycles corresponding to the two specimens are colored in orange and red. The credible interval colored in light blue surrounds the realized value, indicating that the prediction does not lead to overly confident estimations. Once a crack length greater than 0.635 mm is determined, the credible interval narrows quickly. After about $8,000$ and $11,000$ cycles, see Figures 6.17a and 6.17b, the Gaussian process leads to predictions that indicate an earlier final

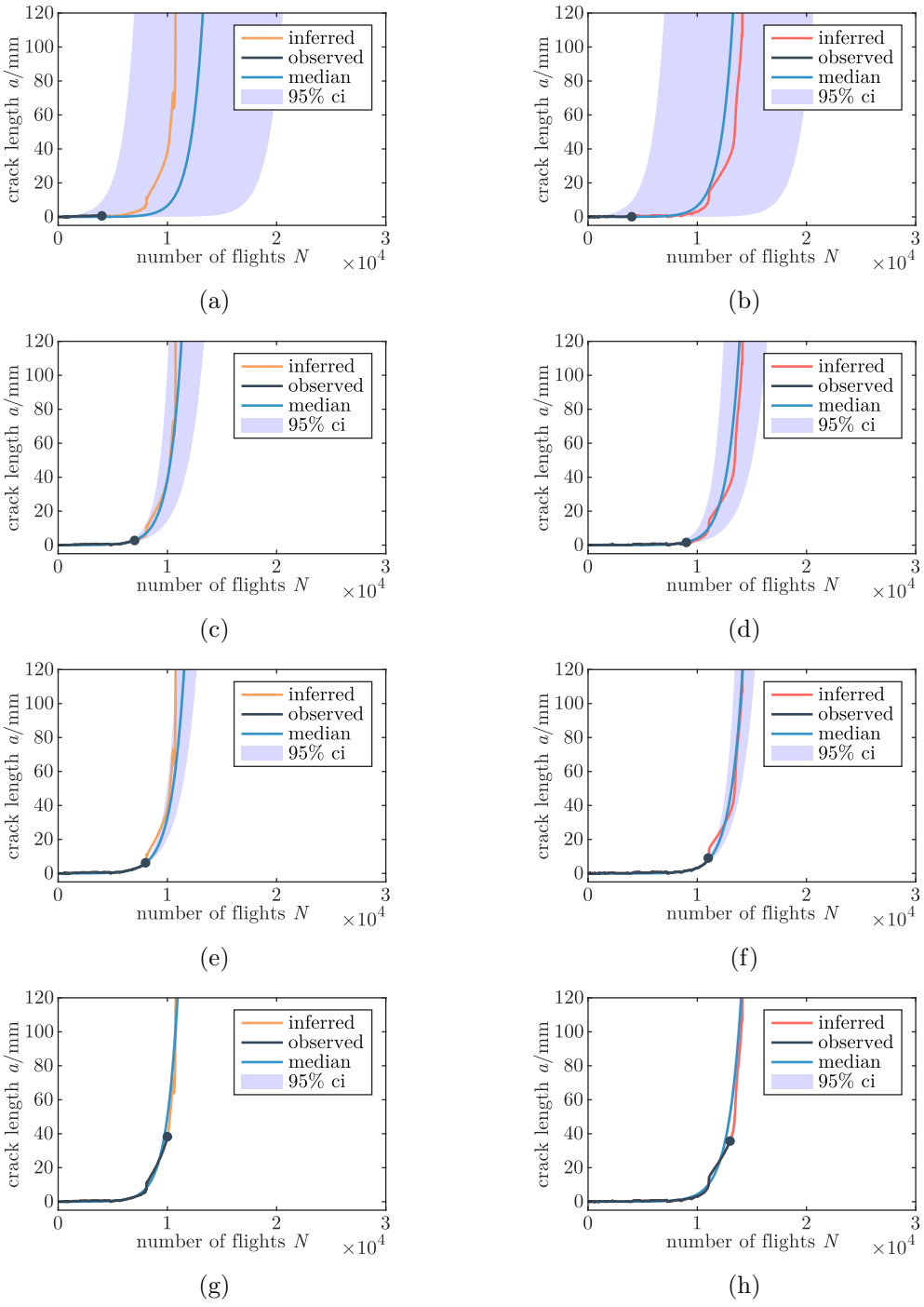


Figure 6.16: Predictions for specimen P02T01 at number of flights equal to (a) 4,000, (c) 7,000, (e) 8,000, and (g) 10,000, and for specimen P03T01 at (b) 4,000, (d) 9,000, (f) 11,000, and (h) 13,000 flights.

fracture than was realized. This is because the two crack regimes are not distinguished in the model, and a_1 exhibits a rapid crack growth before a_2 initiates and starts propagating slowly.

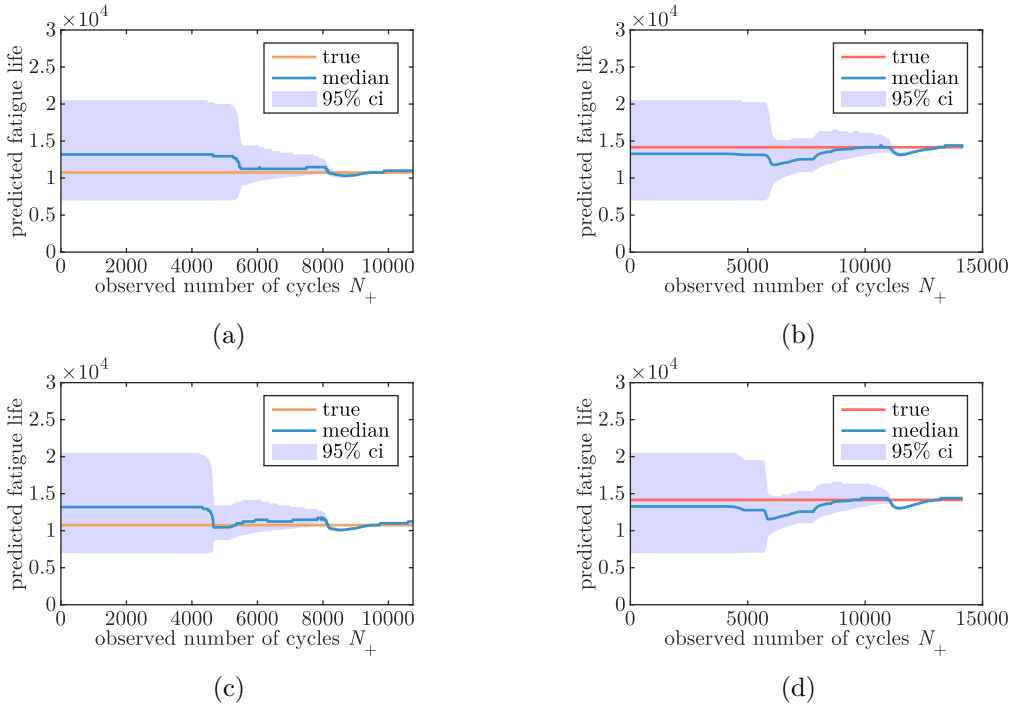


Figure 6.17: Fatigue life predictions with $a_c = 120\,\text{mm}$ for specimen (a) P02T01 and (b) P03T01 with $\sigma_\varepsilon = 200\,\mu\text{m/m}$ and (c) P02T01 and (d) P03T01 with $\sigma_\varepsilon = 100\,\mu\text{m/m}$.

Therefore, the model leads to predictions that are slightly conservative.

Figures 6.17c and 6.17d show the predicted fatigue lives for an assumed standard deviation of the measurement error of $\sigma_\varepsilon = 100\,\mu\text{m/m}$. Compared to the predictions with $\sigma_\varepsilon = 200\,\mu\text{m/m}$, the credible interval based on $\sigma_\varepsilon = 100\,\mu\text{m/m}$ narrows earlier and slightly quicker since the standard deviation of the crack length determined with Bayesian inference is smaller. Especially towards larger observed number of cycles, the credible intervals of the predictions based on $\sigma_\varepsilon = 100\,\mu\text{m/m}$ become very small, leading to overconfident predictions. Table 6.2 lists the evaluated MAEs between the predicted and true fatigue life for the two different standard deviations. The errors for $\sigma_\varepsilon = 100\,\mu\text{m/m}$ and $\sigma_\varepsilon = 200\,\mu\text{m/m}$ are very similar. While the predictions with $\sigma_\varepsilon = 100\,\mu\text{m/m}$ lead to a lower MAE for specimen P02T01, the error for specimen P03T01 is larger than the one based on $\sigma_\varepsilon = 200\,\mu\text{m/m}$. Table 6.2 also shows how often the true fatigue life lies within the predicted 95% credible interval. For specimen P03T01, the relative frequencies for both standard deviations are almost identical and close to the targeted 95%, indicating that the predicted credible intervals are valid. Similar results are achieved with $\sigma_\varepsilon = 200\,\mu\text{m/m}$ for specimen P02T01. By contrast, the predictions for specimen P02T01 based on $\sigma_\varepsilon = 100\,\mu\text{m/m}$ result in a relative frequency of 82.2%, indicating too narrow predicted credible intervals.

Figures 6.17a–6.17d show large credible intervals for the initial prediction, indicating that the fatigue computation exhibits significant uncertainties. Without the proposed method, early and many inspections would be necessary to ensure structural integrity. Since the approach presented in this chapter estimates the current crack length, unnecessary examinations can be avoided. Moreover, not only the current but also future crack lengths can be estimated with credible regions. Therefore, we can predict future failure probabilities and schedule the

maintenance tasks accordingly. Once the method determines a crack length, predictions with narrow credible intervals are possible. This allows us to wait until multiple parts need to be maintained or the entire fatigue life of a structure is exploited. Thus, unnecessary inspections can be skipped, and structures are only maintained when they have to be.

In total, the method leads to low MAEs and relative frequencies close to the targeted ones for both standard deviations. Yet, the standard deviation of the measurement error should not be underestimated to avoid too narrow predicted credible intervals.

Table 6.2: MAE and relative frequencies within the predicted 95% credible interval for two different standard deviations of the measurement error.

		P02T01			**P03T01**		
Data set	σ_ε	**MAE**		**rel. frequency within 95% ci**	**MAE**		**rel. frequency within 95% ci**
aerospace structure	200 μm/m	1680		95.5%	836		93.0%
predict: $N(a_{end})$	100 μm/m	1645	-2.1%	82.2%	874	+4.6%	92.9%

6.6 Summary

As demonstrated for an infinite plate and an aerospace structure, the proposed approach of warped Gaussian processes can handle data that is non-normally distributed. In both cases, introducing a warping function leads to predictions that are solely defined on the positive domain. This agrees with the physics since crack lengths can only be positive. The approach reproduces the analytical solution for problems without an observation error and leads to a close approximation ($< 0.25\%$) for the case where an observation error is present. By using warped Gaussian processes, free parameters $\boldsymbol{\theta}$ are introduced. These need to be determined by minimizing the negative log-likelihood which is, in general, a non-convex optimization problem, requiring increased computational effort. As the Gaussian process is derived by solving a linear regression problem, no additional model parameters must be determined, enabling quick training. Moreover, using warped Gaussian processes reduced the MAE by 53.6% and the negative log-likelihood by 32.2% for the infinite plate example investigated in this chapter. The established model not only predicts the crack length for every future time step but also their credible intervals. The predictions can be used to compute future failure probabilities, which we showcased on the infinite plate example. The information can be used to schedule maintenance tasks according to a failure probability threshold.

Prior knowledge in the form of degradation trajectories is often rare, especially if the mechanical system is large and expensive. Using analytical and FEA-based simulations can produce valuable information that can be incorporated into Gaussian processes by the approach proposed. Two methods for estimating the observation error of the currently observed data are shown. Both methods lead to similar results and converge to the correct solution. Nevertheless, incorrect simulations resulting, for example, from using wrong parameters or assumptions can lead to weak predictions. Therefore, the simulation, parameters, and uncertainties should be well known. As there might be sources of uncertainties that are not known in advance, it is better to assume overly large variances than variances that are too small.

After a Gaussian process is defined by its mean and covariance function, it can be conditioned on current data. The current state of the system, however, is often hidden. Therefore,

the chapter presents how to apply Bayesian inference to infer the crack length from strain data. Based on the coefficient of determination, the resulting crack lengths using Bayesian inference match the crack lengths observed during inspection with $R^2 = 0.656$ and $R^2 = 0.926$ for the first and second aerospace specimen, respectively. The approach enables continuous monitoring of the crack length and its uncertainty. By using this information, the Gaussian process predictions can be continuously updated.

In summary, the results of this chapter show that (1) simulation data can be used to infer Gaussian process models but must be carefully evaluated beforehand, (2) crack lengths can be inferred from strain gauges based on Bayesian inference, and (3) warped Gaussian processes lead to physical results and increase prediction accuracy for crack growth problems.

Chapter 7

Concluding Remarks

"The illusion that we understand the past fosters overconfidence in our ability to predict the future."

– Daniel Kahneman

7.1 Discussion

In this thesis, we deal with the question of how to utilize Gaussian processes for prognostics. In contrast to the existing Gaussian process regression approach, where an underlying model is assumed and optimized, we first describe all possible degradation trajectories by a Gaussian process and second update our predictions by computing the conditional distribution based on monitoring data. The approach considers several previously gathered trajectories from which we derive the Gaussian process model without using a prescribed mean and covariance function. The approach proposed in Chapter 4 is based on a weighted sum of basis functions that are used to fit all previously gathered trajectories. The mean and covariance function, which define the Gaussian process, are directly derived from the resulting weights. Since the derived model is based on a Gaussian process, it is able to predict future damage states probabilistically, i.e., we obtain estimates of our predictions' uncertainties. Therefore, we are able to compute the future failure probabilities of mechanical systems. The approach is applied to several examples, including fatigue crack growth, laser degradation, and milling machine wear. The results show that the prediction accuracy is significantly increased compared to the usually assumed Gaussian process models. The proposed approach also enables integrating physical knowledge into Gaussian processes. By using governing equations as basis functions, their predictive capabilities are further increased. Additionally, the approach requires only a fraction ($\approx 1/10$) of training time compared to the standard method.

As previously described, the proposed method for deriving Gaussian process models from previous data is based on a set of basis functions. In Section 4.4.2, we show that assuming, for example, polynomial basis functions leads to a polynomial mean and covariance function that additionally considers correlations between the weights. Therefore, the a priori defined basis functions constrain the underlying model in a similar fashion to choosing a Gaussian process model. Yet, the basis functions are explicitly defined, which might be easier than constructing a mean and covariance function. Furthermore, we present a methodology for selecting the basis

functions to tackle the difficulty of selecting a specific set. Another aspect of the proposed approach is that the training process becomes much quicker when using basis functions, and it also opens the way for physics-informed Gaussian processes. By utilizing problem-specific governing equations as basis functions, physical knowledge can directly be integrated into Gaussian processes, further increasing accuracy. The entire method was showcased on several examples for prognostics which revealed an ever-decreasing uncertainty prediction for more and more monitored data. This agrees with our intuition that the more data we have observed, the more precise we are able to predict the future degradation of mechanical systems.

However, one must be aware that the selected basis functions are also assumed for the region after the training trajectory ends. The basis functions fitted solely for the region where data points exist might exhibit non-physical behavior outside the training trajectories, leading to poor predictions after the shortest line. For fatigue crack growth problems, where trajectories stop at a critical crack length, the x-y data might be flipped to assume a Gaussian process over the number of cycles instead of the crack length in order to bypass this problem. Furthermore, until now, the approach has only been applied to one-dimensional inputs, assuming the same operation for each specimen. Yet, the method can be extended for handling multiple input variables.

The approach for deriving Gaussian processes relies on multiple trajectories. This data set must be gathered by executing experiments or simulations and should be collected on the low-frequency domain. Solving the equation system for estimating the basis functions' weights might be intractable if the trajectories consist of too many data points. A solution to this is, for example, to compress the data points. Another problem that might be more severe appears when the derived model does not represent the monitored mechanical system well. Since we tend to overestimate our ability to understand the past, we are prone to underestimate the existing uncertainties. This is followed by overconfident predictions or, as the Nobel prize winner Daniel Kahneman puts it in his award-winning book "Thinking, fast and slow" [62]: *"The illusion that we understand the past fosters overconfidence in our ability to predict the future."* Similar overconfident predictions are obtained from inferred Gaussian processes if our simulations or experiments do not inherit all uncertainty sources present in the monitored device or if the uncertainties are underestimated. If, for example, the standard deviation of the measured strain data gathered from the strain gauges attached to the aerospace structure is underestimated, the predicted credible intervals are too small, and the predictions become wobbly. Therefore, the degradation process must be understood fully, and simulations should be validated before using them as training data. Moreover, uncertainty sources must be carefully determined, and scatter parameters might rather be assumed to be too big than too small.

Moreover, three other machine learning methods, namely recurrent neural networks, support vector machines, and k-nearest neighbors are utilized for prognostics. The models are constructed such that they predict the last damage state with credible intervals. They are compared to the proposed Gaussian process method, revealing that recurrent neural networks and the presented Gaussian process approach show the highest prediction accuracies. While recurrent neural networks lead to better results for non-normally distributed data, Gaussian processes are superior for trajectories that are close to or slightly outside the training boundaries. The training process of the presented Gaussian process method is much quicker than the one based on recurrent neural networks ($\approx 1/100$). Furthermore, the Gaussian process is able to predict the entire future degradation trajectory instead of only the last point. Even though

the machine learning models that we constructed for comparison might be designed such that they are also able to predict multiple future damage states, the results reveal that the training process and time would explode. Even for predicting one future damage state, the training time of recurrent neural networks is two orders of magnitude higher than the proposed Gaussian process methodology. Additionally, the results of the machine learning methods tend to be difficult to interpret, while the predictions of the inferred Gaussian process can be explained by the assumed basis functions and the updating scheme based on normality assumptions.

Yet, the results of the comparison chapter also reveal that Gaussian processes are inferior to recurrent neural networks for predicting non-normally distributed trajectories. This is because the updating scheme is based on normality assumptions that do not hold in such cases. By contrast, the process of updating the predictions of recurrent neural networks is not defined beforehand but rather learned during the training phase. In order to tackle this drawback of Gaussian processes, we present in Chapter 6 inferred warped Gaussian processes, which are showcased on two fatigue crack growth problems. The extension treats non-normally distributed trajectories by transforming them into a latent space where the Gaussian process is modeled based on the method proposed in Chapter 4. The results of predicting fatigue crack growth in an infinite plate show that the extension can significantly increase the predictive capabilities of Gaussian processes. Moreover, the advanced method leads to strictly positive predictions for the investigated crack growth examples, which agrees with the physics since crack lengths are only defined on the positive domain. However, free parameters are introduced to the training process by transforming the Gaussian process with a warping function. The free parameters need to be optimized, which increases training time. Additionally, the basis functions might be more difficult to choose since the degradation trajectories are warped to a latent space where the trajectories' shape is not known a priori. This challenge is tackled by selecting multiple different basis functions, evaluating the log-likelihood of each, and choosing the ones that maximize this measure. Many applications inherit log-normally distributed variables and are defined solely on the positive domain. In those cases, taking the logarithm as the warping function (without any free parameters) can be utilized. A warping function with more free parameters can be chosen for more sophisticated scenarios. The extended approach was also applied to predict the crack growth in a real aerospace structure. The defined Gaussian process relies again on crack growth trajectories, which is why we need current crack lengths to compute the conditional distribution for updating our predictions. In a real-world scenario, however, crack lengths are not directly monitored. Therefore, we further present a method for inferring crack lengths from strain gauge data in Chapter 6. The determined crack lengths agree well with the conducted experiments and are used to compute the conditional distribution, updating the prediction.

7.2 Conclusion

Concluding this work, the proposed approach for inferring Gaussian process models from several trajectories and computing the conditional distribution based on monitoring data leverage Gaussian processes for prognostics. The results show that taking previous data into account increases prediction accuracy and look-ahead time significantly and leads to valid credible regions for fatigue crack growth, laser degradation, and milling machine wear applications. The approach also facilitates physics-informed Gaussian processes by using problem-specific governing equations, which further improves accuracy. Moreover, deriving the Gaussian process

model from previously collected trajectories instead of training prescribed mean and covariance functions decreases computational effort remarkably.

Compared to other machine learning methods, such as k-nearest neighbors, support vector machines, and recurrent neural networks, Gaussian processes exhibit excellent predictive capabilities for normally distributed data since they predict entire trajectories without excessive training. Their training process is much quicker ($\approx 1/100$) than the one of recurrent neural networks. Yet, Gaussian processes struggle with predicting non-normally distributed trajectories as their underlying updating scheme is based on normality assumptions.

The extended approach for deriving warped Gaussian processes from several trajectories proposed in Chapter 6 tackles the challenge of applying Gaussian processes to non-normally distributed data. For the fatigue crack growth problems presented in this thesis, the extension forces the predictions to be strictly positive, thus leading to physical results. The advanced method increases prediction accuracy even further and facilitates Gaussian processes for prognostics of non-normally distributed trajectories.

In cases where previous trajectories are missing, simulation data can be used to derive the underlying Gaussian process model. However, the utilized simulation must be evaluated beforehand, uncertainty sources should be carefully determined, and involved uncertainties must not be underestimated. In cases where the degradation variable cannot be directly observed, Bayesian inference can be used to determine its current state from collected data, as we present for inferring crack lengths from strain gauge data. Then, continuous updates of the degradation prediction with credible intervals are obtained by computing the conditional distribution. The entire approach is successfully applied to a real aerospace structure, leading to accurate fatigue life estimations (MAPE of 15.6% for P02T01 and 5.9% for P03T01) and valid credible regions. It can be used to schedule better maintenance plans based on failure probabilities in place of the current but outdated manual crack inspection. In summary, we have met our objectives and established a method that

(1) predicts entire degradation trajectories with valid credible regions,

(2) leads to high accuracy for normally and non-normally distributed data, and

(3) quickly infers the model from training data.

7.3 Outlook

As mentioned in the discussion section, one problem arises if the training trajectories are available in different regions, which is the case for crack growth trajectories. One possibility to solve this challenge is to flip the x-y data such that the derived Gaussian process describes the x variable. This, however, would mean that the measured current damage state and its observation error must be transformed too. Another possibility to gain higher prediction accuracy is to not only warp the y variable but also the input x. Doing so would add more flexibility to the model, and a set of basis functions could better fit complicated trajectories. Also, more sophisticated warping functions, such as the compositional warping functions presented in [119], which have multiple free parameters, can be applied. An automatic way to select the best option based on minimizing the negative log-likelihood would ensure usability.

A severe problem might occur if the training trajectories do not resemble the monitored system. In order to loosen the dependency on previous trajectories, we can combine the derived covariance function with state-of-the-art ones and update their free parameters based

on minimizing the negative log-likelihood. Broadening the research field on kernel (covariance function) construction with the proposed method promises physics-informed Gaussian processes with integrated prior knowledge while preserving the flexibility for novel data. A drawback of doing so might be that the predicted credible intervals do not necessarily narrow down the more data points are observed since the Gaussian process model is updated each time anew when data is gathered.

A promising future venue is also extending the presented approach to multiple inputs and conditions. Several inputs can be considered by adjusting the set of basis functions. Additionally, mixtures of multiple Gaussian processes can easily be formulated, leveraging the proposed methodologies for applications with various scenarios where, for example, cracks can occur and propagate from multiple locations. Moreover, different operational conditions can then be considered. In total, the approach proposed in this thesis opens the way for various advancements that can leverage Gaussian processes to be an indispensable tool for future predictive maintenance scheduling.

Appendix A

Conditional Multivariate Normal Distribution

For deriving the analytical equation for a conditional multivariate normal distribution, we follow the derivation of Murphy [95] in Section 4.3.4.3. The joint distribution of two normally distributed random vectors can be written as

$$p(\boldsymbol{x}_1, \boldsymbol{x}_2) = p(\boldsymbol{x}_1 \mid \boldsymbol{x}_2)\, p(\boldsymbol{x}_2) \tag{A.1}$$

with the two marginal distributions

$$p(\boldsymbol{x}_1) = \mathcal{N}\left(\boldsymbol{x}_1 \mid \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_{1,1}\right) \tag{A.2}$$

and

$$p(\boldsymbol{x}_2) = \mathcal{N}\left(\boldsymbol{x}_2 \mid \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_{2,2}\right), \tag{A.3}$$

and the joint normal distribution

$$p(\boldsymbol{x}_1, \boldsymbol{x}_2) = \mathcal{N}\left(\begin{bmatrix}\boldsymbol{x}_1 \\ \boldsymbol{x}_2\end{bmatrix} \middle| \boldsymbol{\mu} = \begin{bmatrix}\boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2\end{bmatrix}, \boldsymbol{\Sigma} = \begin{bmatrix}\boldsymbol{\Sigma}_{1,1} & \boldsymbol{\Sigma}_{1,2} \\ \boldsymbol{\Sigma}_{2,1} & \boldsymbol{\Sigma}_{2,2}\end{bmatrix}\right). \tag{A.4}$$

In the following, we regard only the proportional exponent

$$p(\boldsymbol{x}_1, \boldsymbol{x}_2) \propto \exp\left\{-\frac{1}{2}\begin{bmatrix}\boldsymbol{x}_1 - \boldsymbol{\mu}_1 \\ \boldsymbol{x}_2 - \boldsymbol{\mu}_2\end{bmatrix}^\top \begin{bmatrix}\boldsymbol{\Sigma}_{1,1} & \boldsymbol{\Sigma}_{1,2} \\ \boldsymbol{\Sigma}_{2,1} & \boldsymbol{\Sigma}_{2,2}\end{bmatrix}^{-1} \begin{bmatrix}\boldsymbol{x}_1 - \boldsymbol{\mu}_1 \\ \boldsymbol{x}_2 - \boldsymbol{\mu}_2\end{bmatrix}\right\}. \tag{A.5}$$

The key to this derivation is to split the exponent such that we have a term that describes the marginal distribution of $\boldsymbol{x}_2$ and another one for the conditional distribution of $\boldsymbol{x}_1$ given $\boldsymbol{x}_2$. For doing so, we need to represent the inverse of the partitioned covariance matrix $\boldsymbol{\Sigma}$ in a different way. One way is to utilize the Schur complement, which arises when performing a block Gaussian elimination. Then, the inverse of the covariance matrix can be written as

$$\boldsymbol{\Sigma}^{-1} = \begin{bmatrix}\boldsymbol{\Sigma}_{1,1} & \boldsymbol{\Sigma}_{1,2} \\ \boldsymbol{\Sigma}_{2,1} & \boldsymbol{\Sigma}_{2,2}\end{bmatrix}^{-1} = \begin{bmatrix}\boldsymbol{I} & 0 \\ -\boldsymbol{\Sigma}_{2,2}^{-1}\boldsymbol{\Sigma}_{2,1} & \boldsymbol{I}\end{bmatrix} \begin{bmatrix}(\boldsymbol{\Sigma}/\boldsymbol{\Sigma}_{2,2})^{-1} & 0 \\ 0 & \boldsymbol{\Sigma}_{2,2}^{-1}\end{bmatrix} \begin{bmatrix}\boldsymbol{I} & -\boldsymbol{\Sigma}_{1,2}\boldsymbol{\Sigma}_{2,2}^{-1} \\ 0 & \boldsymbol{I}\end{bmatrix} \tag{A.6}$$

with

$$\boldsymbol{\Sigma}/\boldsymbol{\Sigma}_{2,2} = \boldsymbol{\Sigma}_{1,1} - \boldsymbol{\Sigma}_{1,2}\boldsymbol{\Sigma}_{2,2}^{-1}\boldsymbol{\Sigma}_{2,1} \tag{A.7}$$

being the Schur complement of $\boldsymbol{\Sigma}$ with respect to $\boldsymbol{\Sigma}_{2,2}$. The exponent in Equation A.5 can be substituted as

$$p(\boldsymbol{x}_1, \boldsymbol{x}_2) \propto \exp\left\{ -\frac{1}{2} \begin{bmatrix} \boldsymbol{x}_1 - \boldsymbol{\mu}_1 \\ \boldsymbol{x}_2 - \boldsymbol{\mu}_2 \end{bmatrix}^\top \begin{bmatrix} \boldsymbol{I} & 0 \\ -\boldsymbol{\Sigma}_{2,2}^{-1}\boldsymbol{\Sigma}_{2,1} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} (\boldsymbol{\Sigma}/\boldsymbol{\Sigma}_{2,2})^{-1} & 0 \\ 0 & \boldsymbol{\Sigma}_{2,2}^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{\Sigma}_{1,2}\boldsymbol{\Sigma}_{2,2}^{-1} \\ 0 & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 - \boldsymbol{\mu}_1 \\ \boldsymbol{x}_2 - \boldsymbol{\mu}_2 \end{bmatrix} \right\}. \tag{A.8}$$

The vectors and matrices can be multiplied out such that

$$\begin{aligned}
p(\boldsymbol{x}_1, \boldsymbol{x}_2) \propto & \exp\left\{ -\frac{1}{2} \left(\left((\boldsymbol{x}_1 - \boldsymbol{\mu}_1)^\top - (\boldsymbol{x}_2 - \boldsymbol{\mu}_2)^\top \boldsymbol{\Sigma}_{2,2}^{-1}\boldsymbol{\Sigma}_{2,1} \right) (\boldsymbol{\Sigma}/\boldsymbol{\Sigma}_{2,2})^{-1} \right. \right. \\
& \left. \left. \left(\boldsymbol{x}_1 - \boldsymbol{\mu}_1 - \boldsymbol{\Sigma}_{1,2}\boldsymbol{\Sigma}_{2,2}^{-1} (\boldsymbol{x}_2 - \boldsymbol{\mu}_2) \right) + (\boldsymbol{x}_2 - \boldsymbol{\mu}_2)^\top \boldsymbol{\Sigma}_{2,2}^{-1} (\boldsymbol{x}_2 - \boldsymbol{\mu}_2) \right) \right\} \\
p(\boldsymbol{x}_1, \boldsymbol{x}_2) \propto & \exp\left\{ -\frac{1}{2} \left(\boldsymbol{x}_1 - \boldsymbol{\mu}_1 - \boldsymbol{\Sigma}_{1,2}\boldsymbol{\Sigma}_{2,2}^{-1} (\boldsymbol{x}_2 - \boldsymbol{\mu}_2) \right)^\top (\boldsymbol{\Sigma}/\boldsymbol{\Sigma}_{2,2})^{-1} \right. \\
& \left. \left(\boldsymbol{x}_1 - \boldsymbol{\mu}_1 - \boldsymbol{\Sigma}_{1,2}\boldsymbol{\Sigma}_{2,2}^{-1} (\boldsymbol{x}_2 - \boldsymbol{\mu}_2) \right) \right\} \times \\
& \exp\left\{ -\frac{1}{2} (\boldsymbol{x}_2 - \boldsymbol{\mu}_2)^\top \boldsymbol{\Sigma}_{2,2}^{-1} (\boldsymbol{x}_2 - \boldsymbol{\mu}_2) \right\} \\
p(\boldsymbol{x}_1, \boldsymbol{x}_2) \propto & \exp\left\{ -\frac{1}{2} \left(\boldsymbol{x}_1 - \boldsymbol{\mu}_{1|2} \right)^\top \boldsymbol{\Sigma}_{1|2}^{-1} \left(\boldsymbol{x}_1 - \boldsymbol{\mu}_{1|2} \right) \right\} \times \\
& \exp\left\{ -\frac{1}{2} (\boldsymbol{x}_2 - \boldsymbol{\mu}_2)^\top \boldsymbol{\Sigma}_{2,2}^{-1} (\boldsymbol{x}_2 - \boldsymbol{\mu}_2) \right\}
\end{aligned} \tag{A.9}$$

with

$$\boldsymbol{\mu}_{1|2} = \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{1,2}\boldsymbol{\Sigma}_{2,2}^{-1} (\boldsymbol{x}_2 - \boldsymbol{\mu}_2) \tag{A.10}$$

and

$$\boldsymbol{\Sigma}_{1|2} = \boldsymbol{\Sigma}/\boldsymbol{\Sigma}_{2,2} = \boldsymbol{\Sigma}_{1,1} - \boldsymbol{\Sigma}_{1,2}\boldsymbol{\Sigma}_{2,2}^{-1}\boldsymbol{\Sigma}_{2,1}. \tag{A.11}$$

Since the second term of Equation A.9 describes the exponent of the marginal distribution of $\boldsymbol{x}_2$, the first term must be the exponent of the conditional distribution of $\boldsymbol{x}_1$ given $\boldsymbol{x}_2$. Therefore, we have successfully split the joint normal distribution as

$$\begin{aligned}
p(\boldsymbol{x}_1, \boldsymbol{x}_2) &= p(\boldsymbol{x}_1 \mid \boldsymbol{x}_2)\, p(\boldsymbol{x}_2) \\
&= \mathcal{N}\left(\boldsymbol{x}_1 \mid \boldsymbol{\mu}_{1|2}, \boldsymbol{\Sigma}_{1|2} \right) \times \mathcal{N}\left(\boldsymbol{x}_2 \mid \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_{2,2} \right).
\end{aligned} \tag{A.12}$$

Appendix B

Directories

B.1 List of Figures

B.2 List of Tables

References

[1] Airline operating costs and productivity. https://www.icao.int/mid/documents/2017/aviation%20data%20and%20analysis%20seminar/ppt3%20-%20airlines%20operating%20costs%20and%20productivity.pdf. (accessed May 2, 2022).

[2] A. B. Abdessalem, N. Dervilis, D. J. Wagg, and K. Worden. Automatic kernel selection for Gaussian processes regression with approximate Bayesian computation and sequential Monte Carlo. *Frontiers in Built Environment*, 3:52, 2017. DOI: 10.3389/fbuil.2017.00052.

[3] A. Agogino and K. Goebel. Milling data set. *NASA Ames Prognostics Data Repository, BEST Lab: Berkeley, CA, USA*, 2007.

[4] R. Alderliesten. Damage tolerance of bonded aircraft structures. *International Journal of Fatigue*, 31(6): 1024–1030, 2009. DOI: 10.1016/j.ijfatigue.2008.05.001.

[5] D. An, N. H. Kim, and J.-H. Choi. Practical options for selecting data-driven or physics-based prognostics algorithms with reviews. *Reliability Engineering & System Safety*, 133:223–236, 2015. DOI: 10.1016/j.ress.2014.09.014.

[6] T. L. Anderson. *Fracture mechanics: Fundamentals and applications*. Taylor & Francis Group, 2005.

[7] A. Ang. Estimation of the noise variance in OLS. https://angms.science/doc/Regression/Regression_4_OLSEstimateNoiseVariance.pdf, 2017. (accessed May 2, 2022).

[8] ASTM International. Test method for measurement of fatigue crack growth rates. Technical report, 1981.

[9] L. D. Avendano-Valencia, E. N. Chatzi, and D. Tcherniak. Gaussian process models for mitigation of operational variability in the structural health monitoring of wind turbines. *Mechanical Systems and Signal Processing*, 142:106686, 2020. DOI: 10.1016/j.ymssp.2020.106686.

[10] S. Aye and P. Heyns. An integrated Gaussian process regression for prediction of remaining useful life of slow speed bearings based on acoustic emission. *Mechanical Systems and Signal Processing*, 84:485–498, 2017. DOI: 10.1016/j.ymssp.2016.07.039.

[11] T. Belytschko and T. Black. Elastic crack growth in finite elements with minimal remeshing. *International Journal for Numerical Methods in Engineering*, 45(5):601–620, 1999. DOI: 10.1002/(SICI)1097-0207(19990620)45:5<601::AID-NME598>3.0.CO;2-S.

[12] T. Benkedjouh, K. Medjaher, N. Zerhouni, and S. Rechak. Remaining useful life estimation based on nonlinear feature reduction and support vector regression. *Engineering Applications of Artificial Intelligence*, 26(7):1751–1760, 2013. DOI: 10.1016/j.engappai.2013.02.006.

[13] T. Benkedjouh, K. Medjaher, N. Zerhouni, and S. Rechak. Health assessment and life prediction of cutting tools based on support vector regression. *Journal of Intelligent Manufacturing*, 26(2):213–223, 2015. DOI: 10.1007/s10845-013-0774-6.

[14] P. J. Bickel and K. A. Doksum. An analysis of transformations revisited. *Journal of the American Statistical Association*, 76(374):296–311, 1981. DOI: 10.2307/2287831.

[15] C. M. Bishop. *Mixture density networks.* Aston University, 1994.

[16] C. M. Bishop. *Pattern recognition and machine learning.* Springer, 2006.

[17] C. M. Bishop and M. E. Tipping. Variational relevance vector machines. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, pages 46–53, 2000.

[18] B. Blight and L. Ott. A Bayesian approach to model inadequacy for polynomial regression. *Biometrika*, 62(1):79–88, 1975. DOI: 10.1093/biomet/62.1.79.

[19] M. Blum and M. A. Riedmiller. Optimization of Gaussian process hyperparameters using Rprop. In *European Symposium on Artificial Neural Networks*, pages 339–344, 2013.

[20] J. L. Bogdanoff. *Probabilistic models of cumulative damage.* John Wiley & Sons Inc, 1985.

[21] G. E. Box and D. R. Cox. An analysis of transformations. *Journal of the Royal Statistical Society: Series B (Methodological)*, 26(2):211–243, 1964. DOI: 10.1111/j.2517-6161.1964.tb00553.x.

[22] J. A. Carino, D. Zurita, M. Delgado, J. Ortega, and R. Romero-Troncoso. Remaining useful life estimation of ball bearings by means of monotonic score calibration. In *IEEE International Conference on Industrial Technology*, pages 1752–1758. IEEE, 2015.

[23] G. Cavallini and R. Lazzeri. A probabilistic approach to fatigue design of aerospace components by using the risk assessment evaluation. In *Recent Advances in Aircraft Technology*, chapter 2. INTECH Open Access Publisher, 2012.

[24] B. Chapuis and E. Sjerve. *Sensors, algorithms and applications for structural health monitoring.* Springer, 2018.

[25] J. Chen, S. Yuan, and H. Wang. On-line updating Gaussian process measurement model for crack prognosis using the particle filter. *Mechanical Systems and Signal Processing*, 140:106646, 2020. DOI: 10.1016/j.ymssp.2020.106646.

[26] N. Chokshi and B. Dooley. Boeing calls for global grounding of 777s equipped with one engine model. `https://www.nytimes.com/2021/02/21/business/faa-boeing-777-engines.html`, 2021. (accessed January 18, 2022).

[27] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995. DOI: 10.1007/BF00994018.

[28] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967. DOI: 10.1109/TIT.1967.1053964.

[29] H. de Faria Jr, J. G. S. Costa, and J. L. M. Olivas. A review of monitoring methods for predictive maintenance of electric power transformers based on dissolved gas analysis. *Renewable and Sustainable Energy Reviews*, 46:201–209, 2015. DOI: 10.1016/j.rser.2015.02.052.

[30] D. T. Do, J. Lee, and H. Nguyen-Xuan. Fast evaluation of crack growth path using time series forecasting. *Engineering Fracture Mechanics*, 218:106567, 2019. DOI: 10.1016/j.engfracmech.2019.106567.

[31] S. Dong and T. Luo. Bearing degradation process prediction based on the PCA and optimized LS-SVM model. *Measurement*, 46(9):3143–3152, 2013. DOI: 10.1016/j.measurement.2013.06.038.

[32] N. R. Draper and D. R. Cox. On distributions and their transformation to normality. *Journal of the Royal Statistical Society: Series B (Methodological)*, 31(3):472–476, 1969. DOI: 10.1111/j.2517-6161.1969.tb00806.x.

[33] M. T. Farrell and A. Correa. Gaussian process regression models for predicting stock trends. *Relation*, 10:3414, 2007.

[34] R. G. Forman, V. Kearney, and R. Engle. Numerical analysis of crack propagation in cyclic-loaded structures. *Journal of Basic Engineering*, 89(3):459–463, 1967. DOI: 10.1115/1.3609637.

[35] M. Franz, S. Pfingstl, M. Zimmermann, and S. Wartzack. Estimation of composite laminate ply angles using an inverse Bayesian approach based on surrogate models. In *Proceedings of the Design Society: DESIGN Conference*. Cambridge University Press, 2022.

[36] H. Friedmann and P. Kraemer. Vibration-based condition monitoring, structural health monitoring, population monitoring–approach to a definition of the different concepts by means of practical examples from the field of wind energy. In *8th European Workshop on Structural Health Monitoring*, 2016.

[37] E. Fumeo, L. Oneto, and D. Anguita. Condition based maintenance in railway transportation systems based on big data streaming analysis. *Procedia Computer Science*, 53:437–446, 2015. DOI: 10.1016/j.procs.2015.07.321.

[38] R. Galatolo and R. Lazzeri. Fatigue crack growth of multiple interacting cracks: Analytical models and experimental validation. *Fatigue & Fracture of Engineering Materials & Structures*, 41(1):183–196, 2018. DOI: 10.1111/ffe.12671.

[39] R. Galatolo and R. Lazzeri. Experiments and model predictions for fatigue crack propagation in riveted lap-joints with multiple site damage. *Fatigue & Fracture of Engineering Materials & Structures*, 39(3): 307–319, 2016. DOI: 10.1111/ffe.12354.

[40] R. Gentile and C. Galasso. Gaussian process regression for seismic fragility assessment of building portfolios. *Structural Safety*, 87:101980, 2020. DOI: 10.1016/j.strusafe.2020.101980.

[41] M. H. Gharavian, F. A. Ganj, A. Ohadi, and H. H. Bafroui. Comparison of FDA-based and PCA-based features in fault diagnosis of automobile gearboxes. *Neurocomputing*, 121:150–159, 2013. DOI: 10.1016/j.neucom.2013.04.033.

[42] A. Glowacz and Z. Glowacz. Diagnosis of stator faults of the single-phase induction motor using acoustic signals. *Applied Acoustics*, 117:20–27, 2017. DOI: 10.1016/j.apacoust.2016.10.012.

[43] M. Gobbato, J. B. Kosmatka, and J. P. Conte. A recursive Bayesian approach for fatigue damage prognosis: An experimental validation at the reliability component level. *Mechanical Systems and Signal Processing*, 45(2):448–467, 2014. DOI: 10.1016/j.ymssp.2013.10.014.

[44] K. Goebel, M. Daigle, A. Saxena, S. Sankararaman, I. Roychoudhury, and J. Celaya. *Prognostics: The science of predictions*. CreateSpace Independent Publishing Platform, 2017.

[45] Í. G. Gonçalves, E. Echer, and E. Frigo. Sunspot cycle prediction using warped Gaussian process regression. *Advances in Space Research*, 65(1):677–683, 2020. DOI: 10.1016/j.asr.2019.11.011.

[46] D. Goodman, J. P. Hofmeister, and F. Szidarovszky. *Prognostics and health management: A practical approach to improving system reliability using condition-based data*. John Wiley & Sons, 2019.

[47] M. Greenwood. Cracks found on Boeing's 737 NG pickle forks. `https://www.engineering.com/story/cracks-found-on-boeings-737-ng-pickle-forks`, 2019. (accessed May 2, 2022).

[48] A. A. Griffith. The phenomena of rupture and flow in solids. *Philosophical Transactions*, 221:163–198, 1920. DOI: 10.1098/rsta.1921.0006.

[49] D. Gross and T. Seelig. *Bruchmechanik: Mit einer Einführung in die Mikromechanik*. Springer, 2016.

[50] E. Haibach. *Betriebsfestigkeit*. Springer, 2006.

[51] C. Harger. Unexpected cracking found on critical Boeing 737 NG equipment. `https://komonews.com/news/local/exclusive-unexpected-cracking-found-on-critical-boeing-737ng-equipment`, 2019. (accessed January 18, 2022).

[52] S. M. M. Hassani N., X. Jin, and J. Ni. Physics-based Gaussian process for the health monitoring for a rolling bearing. *Acta Astronautica*, 154:133–139, 2019. DOI: 10.1016/j.actaastro.2018.10.029.

[53] D. He, R. Li, and J. Zhu. Plastic bearing fault diagnosis based on a two-step data mining approach. *IEEE Transactions on Industrial Electronics*, 60(8):3429–3440, 2012. DOI: 10.1109/TIE.2012.2192894.

[54] F. O. Heimes. Recurrent neural networks for remaining useful life estimation. In *International Conference on Prognostics and Health Management*, pages 59–64. IEEE, 2008.

[55] T. Heyns, J. P. De Villiers, and P. S. Heyns. Consistent haul road condition monitoring by means of vehicle response normalisation with Gaussian processes. *Engineering Applications of Artificial Intelligence*, 25 (8):1752–1760, 2012. DOI: 10.1016/j.engappai.2012.01.009.

[56] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. DOI: 10.1162/neco.1997.9.8.1735.

[57] S. Hong and Z. Zhou. Remaining useful life prognosis of bearing based on Gauss process regression. In *5th International Conference on BioMedical Engineering and Informatics*, pages 1575–1579. IEEE, 2012.

[58] S. Hudak, A. Saxena, R. Bucci, and R. Malcolm. Development of standard methods of testing and analyzing fatigue crack growth rate data. Technical Report AFML-TR-78-40, 1978.

[59] G. R. Irwin. Onset of fast crack propagation in high strength steel and aluminum alloys. Technical Report NRL Report 4763, 1956.

[60] L. Jiang, J. Xuan, and T. Shi. Feature extraction based on semi-supervised kernel marginal Fisher analysis and its application in bearing fault diagnosis. *Mechanical Systems and Signal Processing*, 41(1-2):113–126, 2013. DOI: 10.1016/j.ymssp.2013.05.017.

[61] L. Jiang, T. Shi, and J. Xuan. Fault diagnosis of rolling bearings based on marginal Fisher analysis. *Journal of Vibration and Control*, 20(3):470–480, 2014. DOI: 10.1177/1077546312463747.

[62] D. Kahneman. *Thinking, fast and slow.* Macmillan, 2011.

[63] M. S. Kan, A. C. Tan, and J. Mathew. A review on prognostic techniques for non-stationary and non-linear rotating systems. *Mechanical Systems and Signal Processing*, 62:1–20, 2015. DOI: 10.1016/j.ymssp.2015.02.016.

[64] R. Khelif, B. Chebel-Morello, S. Malinowski, E. Laajili, F. Fnaiech, and N. Zerhouni. Direct remaining useful life estimation based on support vector regression. *IEEE Transactions on Industrial Electronics*, 64(3):2276–2285, 2016. DOI: 10.1109/TIE.2016.2623260.

[65] N.-H. Kim, D. An, and J.-H. Choi. *Prognostics and health management of engineering systems: An introduction.* Springer, 2017.

[66] D. Kong, Y. Chen, and N. Li. Gaussian process regression for tool wear prediction. *Mechanical Systems and Signal Processing*, 104:556–574, 2018. DOI: 10.1016/j.ymssp.2017.11.021.

[67] P. Kou, F. Gao, X. Guan, and J. Wu. Prediction intervals for wind power forecasting: Using sparse warped Gaussian process. In *IEEE Power and Energy Society General Meeting*. IEEE, 2012.

[68] P. Kou, F. Gao, and X. Guan. Sparse online warped Gaussian process for wind power probabilistic forecasting. *Applied Energy*, 108:410–428, 2013. DOI: 10.1016/j.apenergy.2013.03.038.

[69] P. Kou, D. Liang, F. Gao, and L. Gao. Probabilistic wind power forecasting with online model selection and warped Gaussian process. *Energy Conversion and Management*, 84:649–663, 2014. DOI: 10.1016/j.enconman.2014.04.051.

[70] A. Kuo, D. Yasgur, and M. Levy. Assessment of damage tolerance requirements and analysis – task I report. Technical Report AFWAL-TR-86-3003, 1986.

[71] D. Kwon, M. H. Azarian, and M. Pecht. Remaining-life prediction of solder joints using RF impedance analysis and Gaussian process regression. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, 5(11):1602–1609, 2015. DOI: 10.1109/TCPMT.2015.2477098.

[72] M. Lange. On the uncertainty of wind power predictions – analysis of the forecast accuracy and statistical distribution of errors. *Journal of Solar Energy Engineering*, 127(2):177–184, 2005. DOI: 10.1115/1.1862266.

[73] T. Leggett. Boeing 777: Dozens grounded after Denver engine failure. `https://www.nytimes.com/2021/02/21/business/faa-boeing-777-engines.html`, 2021. (accessed January 18, 2022).

[74] Y. Lei, N. Li, L. Guo, N. Li, T. Yan, and J. Lin. Machinery health prognostics: A systematic review from data acquisition to RUL prediction. *Mechanical Systems and Signal Processing*, 104:799–834, 2018. DOI: 10.1016/j.ymssp.2017.11.016.

[75] Y. Lei, B. Yang, X. Jiang, F. Jia, N. Li, and A. K. Nandi. Applications of machine learning to machine fault diagnosis: A review and roadmap. *Mechanical Systems and Signal Processing*, 138:106587, 2020. DOI: 10.1016/j.ymssp.2019.106587.

[76] G. Li, S. Datta, A. Chattopadhyay, N. Iyyer, and N. Phan. An online-offline prognosis model for fatigue life prediction under biaxial cyclic loading with overloads. *Fatigue & Fracture of Engineering Materials & Structures*, 42(5):1175–1190, 2019. DOI: 10.1111/ffe.12983.

[77] M. Li, M. Sadoughi, Z. Hu, and C. Hu. A hybrid Gaussian process model for system reliability analysis. *Reliability Engineering & System Safety*, 197:106816, 2020. DOI: 10.1016/j.ress.2020.106816.

[78] Z. Li, X. Yan, Z. Tian, C. Yuan, Z. Peng, and L. Li. Blind vibration component separation and nonlinear feature extraction applied to the nonstationary vibration signals for the gearbox multi-fault diagnosis. *Measurement*, 46(1):259–271, 2013. DOI: 10.1016/j.measurement.2012.06.013.

[79] D. Liu, J. Pang, J. Zhou, Y. Peng, and M. Pecht. Prognostics for state of health estimation of lithium-ion batteries based on combination Gaussian process functional regression. *Microelectronics Reliability*, 53 (6):832–839, 2013. DOI: 10.1016/j.microrel.2013.03.010.

[80] H. Liu, J. Zhou, Y. Zheng, W. Jiang, and Y. Zhang. Fault diagnosis of rolling bearings with recurrent neural network-based autoencoders. *ISA Transactions*, 77:167–178, 2018. DOI: 10.1016/j.isatra.2018.04.005.

[81] J. Liu and E. Zio. An adaptive online learning approach for support vector regression: Online-SVR-FID. *Mechanical Systems and Signal Processing*, 76:796–809, 2016. DOI: 10.1016/j.ymssp.2016.02.056.

[82] J. Liu, R. Seraoui, V. Vitelli, and E. Zio. Nuclear power plant components condition monitoring by probabilistic support vector machine. *Annals of Nuclear Energy*, 56:23–33, 2013. DOI: 10.1016/j.anucene.2013.01.005.

[83] J. Liu, V. Vitelli, E. Zio, and R. Seraoui. A novel dynamic-weighted probabilistic support vector regression-based ensemble for prognostics of time series data. *IEEE Transactions on Reliability*, 64(4):1203–1213, 2015. DOI: 10.1109/TR.2015.2427156.

[84] T. H. Loutas, D. Roulias, and G. Georgoulas. Remaining useful life estimation in rolling bearings utilizing data-driven probabilistic E-support vectors regression. *IEEE Transactions on Reliability*, 62(4):821–832, 2013. DOI: 10.1109/TR.2013.2285318.

[85] M. Ma and Z. Mao. Rotating machinery prognostics via the fusion of particle filter and deep learning. In *12th International Workshop on Structural Health Monitoring*, 2019.

[86] A. Malhi, R. Yan, and R. X. Gao. Prognosis of defect propagation based on recurrent neural networks. *IEEE Transactions on Instrumentation and Measurement*, 60(3):703–711, 2011. DOI: 10.1109/TIM.2010.2078296.

[87] R. Marsh. 'Cracking issue' discovered on some of Boeing's 737 NG planes, airline says. `https://edition.cnn.com/2019/09/27/business/boeing-737-ng-cracks/index.html`, 2019. (accessed January 18, 2022).

[88] A. Mateo-Sanchis, J. Muñoz-Marí, A. Pérez-Suay, and G. Camps-Valls. Warped Gaussian processes in remote sensing parameter estimation and causal inference. *IEEE Geoscience and Remote Sensing Letters*, 15(11):1647–1651, 2018. DOI: 10.1109/LGRS.2018.2853760.

[89] W. Q. Meeker and L. A. Escobar. *Statistical methods for reliability data*. John Wiley & Sons, 2014.

[90] M. A. Miner. Cumulative damage in fatigue. *Journal of Applied Mechanics*, (3):159–164, 1945. DOI: 10.1115/1.4009458.

[91] T. P. Minka and R. W. Picard. Learning how to learn is learning with point sets. `https://www.microsoft.com/en-us/research/publication/learning-learn-learning-point-sets`, 1999. (accessed December 8, 2022).

[92] N. Moës, J. Dolbow, and T. Belytschko. A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering*, 46(1):131–150, 1999. DOI: 10.1002/(SICI)1097-0207(19990910)46:1<131::AID-NME726>3.0.CO;2-J.

[93] S. Mohanty, A. Chattopadhyay, P. Peralta, and S. Das. Bayesian statistic based multivariate Gaussian process approach for offline/online fatigue crack growth prediction. *Experimental Mechanics*, 51(6):833–843, 2011. DOI: 10.1007/s11340-010-9394-7.

[94] M. Mojaddady, M. Nabi, and S. Khadivi. Stock market prediction using twin Gaussian process regression. `http://disi.unitn.it/~nabi/files/stock.pdf`, 2011. (accessed March 30, 2023).

[95] K. P. Murphy. *Machine learning: A probabilistic perspective*. MIT Press, 2012.

[96] V. T. Narendhar Gugulothu, P. Malhotra, L. Vig, P. Agarwal, and G. Shroff. Predicting remaining useful life using time series embeddings based on recurrent neural networks. In *International Journal on Prognostics and Health Management*, 2017.

[97] V. Nguyen, M. Kefalas, K. Yang, A. Apostolidis, M. Olhofer, S. Limmer, and T. Bäck. A review: Prognostics and health management in automotive and aerospace. *International Journal of Prognostics and Health Management*, 10(2), 2019. DOI: 10.36001/ijphm.2019.v10i2.2730.

[98] G. Niu and B.-S. Yang. Intelligent condition monitoring and prognostics system based on data-fusion strategy. *Expert Systems with Applications*, 37(12):8831–8840, 2010. DOI: 10.1016/j.eswa.2010.06.014.

[99] A. O'Hagan. Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society: Series B (Methodological)*, 40(1):1–24, 1978. DOI: 10.1111/j.2517-6161.1978.tb01643.x.

[100] A. Palmgren. Die Lebensdauer von Kugellagern. *Zeitschrift des Vereines Deutscher Ingenieure*, 14:339–341, 1924.

[101] P. Paris and F. Erdogan. A critical analysis of crack propagation laws. *Journal of Basic Engineering*, 85 (4):528–533, 1963. DOI: 10.1115/1.3656900.

[102] S. Park, S. Kim, and J.-H. Choi. Gear fault diagnosis using transmission error and ensemble empirical mode decomposition. *Mechanical Systems and Signal Processing*, 108:262–275, 2018. DOI: 10.1016/j.ymssp.2018.02.028.

[103] Y. Peng, H. Wang, J. Wang, D. Liu, and X. Peng. A modified echo state network based remaining useful life estimation approach. In *IEEE Conference on Prognostics and Health Management*, pages 1–7. IEEE, 2012.

[104] S. Pfingstl and M. Zimmermann. Strain-based structural health monitoring: Computing regions for critical crack detection. In *12th International Workshop on Structural Health Monitoring*, 2019.

[105] S. Pfingstl and M. Zimmermann. On integrating prior knowledge into Gaussian processes for prognostic health monitoring. *Mechanical Systems and Signal Processing*, 171:108917, 2022. DOI: 10.1016/j.ymssp.2022.108917.

[106] S. Pfingstl, M. Steiner, O. Tusch, and M. Zimmermann. Crack detection zones: Computation and validation. *Sensors*, 20(9):2568, 2020. DOI: 10.3390/s20092568.

[107] S. Pfingstl, Y. N. Schoebel, and M. Zimmermann. Reinforcement learning for structural health monitoring based on inspection data. In *Materials Research Proceedings*, volume 18. Materials Research Forum LLC, 2021.

[108] S. Pfingstl, C. Braun, A. Nasrollahi, F.-K. Chang, and M. Zimmermann. Warped Gaussian processes for predicting the degradation of aerospace structures. *Structural Health Monitoring*, 0(0), 2022. DOI: 10.1177/14759217221127252.

[109] S. Pfingstl, C. Braun, and M. Zimmermann. Warped Gaussian processes for prognostic health monitoring. In *13th International Workshop on Structural Health Monitoring*, 2022.

[110] S. Pfingstl, D. Steinweg, M. Zimmermann, and M. Hornung. On the potential of extending aircraft service time using load monitoring. *Journal of Aircraft*, 59(2):377–385, 2022. DOI: 10.2514/1.C036569.

[111] S. Pfingstl, O. Tusch, and M. Zimmermann. Comparison of error measures and machine learning methods for strain-based structural health monitoring. In *13th International Workshop on Structural Health Monitoring*, 2022.

[112] C. Plagemann, K. Kersting, and W. Burgard. Nonstationary Gaussian process regression using point estimates of local smoothness. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 204–219. Springer, 2008.

[113] Quicit. A simple technique to estimate prediction intervals for any regression model. `https://medium.com/@qucit/a-simple-technique-to-estimate-prediction-intervals-for-any-regression-model-2dd73f630bcb`, 2018. (accessed December 14, 2019).

[114] E. Ramasso and A. Saxena. Performance benchmarking and analysis of prognostic methods for CMAPSS datasets. *International Journal of Prognostics and Health Management*, 5(2):1–15, 2014. DOI: 10.36001/ijphm.2014.v5i2.2236.

[115] R. Randall. *Vibration-based condition monitoring: Industrial, aerospace and automotive applications*. John Wiley & Sons, 2011.

[116] C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. MIT Press, 2006.

[117] R. Rennert and F. F. Maschinenbau. *Rechnerischer Festigkeitsnachweis für Maschinenbauteile aus Stahl, Eisenguss-und Aluminiumwerkstoffen*. VDMA-Verlag, 2012.

[118] G. Rios and F. Tobar. Learning non-Gaussian time series using the Box-Cox Gaussian process. In *International Joint Conference on Neural Networks*, pages 1–8. IEEE, 2018.

[119] G. Rios and F. Tobar. Compositionally-warped Gaussian processes. *Neural Networks*, 118:235–246, 2019. DOI: 10.1016/j.neunet.2019.06.012.

[120] E. F. Rybicki and M. F. Kanninen. A finite element calculation of stress intensity factors by a modified crack closure integral. *Engineering Fracture Mechanics*, 9(4):931–938, 1977. DOI: 10.1016/0013-7944(77)90013-3.

[121] M. Ryschkewitsch. Nondestructive evaluation requirements for fracture critical metallic components. *NASA Technical Standard NASASTD-5009*, 2008.

[122] M. Safizadeh and S. Latifi. Using multi-sensor data fusion for vibration fault diagnosis of rolling element bearings by accelerometer and load cell. *Information Fusion*, 18:1–8, 2014. DOI: 10.1016/j.inffus.2013.10.002.

[123] S. Saha, B. Saha, A. Saxena, and K. Goebel. Distributed prognostic health management with Gaussian process regression. In *IEEE Aerospace Conference*, pages 1–8. IEEE, 2010.

[124] S. Sampath and D. Broek. Estimation of requirements of inspection intervals for panels susceptible to multiple site damage. In *Structural Integrity of Aging Airplanes*, pages 339–389. Springer, 1991.

[125] J. Schijve. Fatigue of aircraft materials and structures. *International Journal of Fatigue*, 16(1):21–32, 1994. DOI: 10.1016/0142-1123(94)90442-1.

[126] J. Schijve. Fatigue damage in aircraft structures, not wanted, but tolerated? *International Journal of Fatigue*, 31(6):998–1011, 2009. DOI: 10.1016/j.ijfatigue.2008.05.016.

[127] C. M. Sharp. *DH: An outline of de Havilland history.* Faber et Faber, 1960.

[128] J. Shawe-Taylor and N. Cristianini. *Kernel methods for pattern analysis.* Cambridge University Press, 2004.

[129] J.-H. Shin and H.-B. Jun. On condition based maintenance policy. *Journal of Computational Design and Engineering*, 2(2):119–127, 2015. DOI: 10.1016/j.jcde.2014.12.006.

[130] E. Siebel and M. Stieler. Ungleichförmige Spannungsverteilung bei schwingender Beanspruchung. *Zeitschrift des Vereines Deutscher Ingenieure*, (5):121–126, 1955.

[131] J. Sikorska, M. Hodkiewicz, and L. Ma. Prognostic modelling options for remaining useful life estimation by industry. *Mechanical Systems and Signal Processing*, 25(5):1803–1836, 2011. DOI: 10.1016/j.ymssp.2010.11.018.

[132] M. Silva, A. Santos, and E. Figueiredo. Damage detection for structural health monitoring of bridges as a knowledge discovery in databases process. In *Data Mining in Structural Dynamic Analysis*, pages 1–24. Springer, 2019.

[133] E. Snelson, Z. Ghahramani, and C. Rasmussen. Warped Gaussian processes. In *Advances in Neural Information Processing Systems 16 (NIPS 2003)*. MIT Press, 2004.

[134] W. Song, Z. Jiang, and H. Jiang. Predict the fatigue life of crack based on extended finite element method and SVR. In *AIP Conference Proceedings*, number 1. AIP Publishing LLC, 2018.

[135] A. Soualhi, K. Medjaher, and N. Zerhouni. Bearing health monitoring based on Hilbert–Huang transform, support vector machine, and regression. *IEEE Transactions on Instrumentation and Measurement*, 64(1): 52–62, 2014. DOI: 10.1109/TIM.2014.2330494.

[136] B. Spencer, J. Tang, and M. Artley. Stochastic approach to modeling fatigue crack growth. *AIAA Journal*, 27(11):1628–1635, 1989. DOI: 10.2514/3.10311.

[137] G. Su, L. Peng, and L. Hu. A Gaussian process-based dynamic surrogate model for complex engineering structural reliability analysis. *Structural Safety*, 68:97–109, 2017. DOI: 10.1016/j.strusafe.2017.06.003.

[138] J. Tian, M. H. Azarian, and M. Pecht. Anomaly detection using self-organizing maps-based k-nearest neighbor algorithm. In *Proceedings of the European Conference of the Prognostics and Health Management Society*, 2014.

[139] T. Tinga and R. Loendersloot. Aligning PHM, SHM and CBM by understanding the physical system failure behaviour. In *Proceedings of the European Conference of the Prognostics and Health Management Society*, pages 162–171, 2014.

[140] M. E. Tipping. The relevance vector machine. In *Advances in Neural Information Processing Systems 12 (NIPS 1999)*, 1999.

[141] M. E. Tipping. Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1(Jun):211–244, 2001.

[142] J. Trauer, S. Pfingstl, M. Finsterer, and M. Zimmermann. Improving production efficiency with a digital twin based on anomaly detection. *Sustainability*, 13(18):10155, 2021. DOI: 10.3390/su131810155.

[143] M. Van and H.-J. Kang. Two-stage feature selection for bearing fault diagnosis based on dual-tree complex wavelet transform and empirical mode decomposition. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 230(2):291–302, 2016. DOI: 10.1177/0954406215573976.

[144] Vanraj, S. Dhami, and B. Pabla. Hybrid data fusion approach for fault diagnosis of fixed-axis gearbox. *Structural Health Monitoring*, 17(4):936–945, 2018. DOI: 10.1177/1475921717727700.

[145] V. Vapnik. Pattern recognition using generalized portrait method. *Automation and Remote Control*, 24: 774–780, 1963.

[146] V. Vapnik and A. Chervonenkis. *Theory of pattern recognition*. Nauka, 1974.

[147] V. Vapnik and A. Chervonenkis. A note on one class of perceptrons. *Automation and Remote Control*, 25, 1964.

[148] D. A. Virkler, B. Hillberry, and P. Goel. The statistical nature of fatigue crack propagation. *Journal of Engineering Materials and Technology*, 101(2):148–153, 1979. DOI: 10.1115/1.3443666.

[149] A. M. Walker. On the asymptotic behaviour of posterior distributions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 31(1):80–88, 1969. DOI: 10.1111/j.2517-6161.1969.tb00767.x.

[150] D. Wang. K-nearest neighbors based methods for identification of different gear crack levels under different motor speeds and loads: Revisited. *Mechanical Systems and Signal Processing*, 70:201–208, 2016. DOI: 10.1016/j.ymssp.2015.10.007.

[151] J. M. Wang, D. J. Fleet, and A. Hertzmann. Gaussian process dynamical models for human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):283–298, 2007. DOI: 10.1109/TPAMI.2007.1167.

[152] P. Wang, B. D. Youn, and C. Hu. A generic probabilistic framework for structural health prognostics and uncertainty management. *Mechanical Systems and Signal Processing*, 28:622–637, 2012. DOI: 10.1016/j.ymssp.2011.10.019.

[153] W. Wang, W. Hu, and N. Armstrong. Fatigue crack prognosis using Bayesian probabilistic modelling. *Mechanical Engineering Journal*, 4(5):16–00702, 2017. DOI: 10.1299/mej.16-00702.

[154] A. Widodo and B.-S. Yang. Application of relevance vector machine and survival probability to machine degradation assessment. *Expert Systems with Applications*, 38(3):2592–2599, 2011. DOI: 10.1016/j.eswa.2010.08.049.

[155] A. Widodo and B.-S. Yang. Machine health prognostics using survival probability and support vector machine. *Expert Systems with Applications*, 38(7):8430–8437, 2011. DOI: 10.1016/j.eswa.2011.01.038.

[156] A. Wilson and R. Adams. Gaussian process kernels for pattern discovery and extrapolation. In *International Conference on Machine Learning*, pages 1067–1075. PMLR, 2013.

[157] R. L. Wolpert. Change of variables. `https://www2.stat.duke.edu/courses/Spring15/sta250/lec/250mvnorm.pdf`. (accessed May 2, 2022).

[158] K. Worden. Structural health monitoring using pattern recognition. In *New Trends in Vibration Based Structural Health Monitoring*, pages 183–246. Springer, 2010.

[159] R. Xie, R. Ma, S. Pu, L. Xu, D. Zhao, and Y. Huangfu. Prognostic for fuel cell based on particle filter and recurrent neural network fusion structure. *Energy and AI*, 2:100017, 2020. DOI: 10.1016/j.egyai.2020.100017.

[160] B. Yao, P. Zhen, L. Wu, and Y. Guan. Rolling element bearing fault diagnosis using improved manifold learning. *IEEE Access*, 5:6027–6035, 2017. DOI: 10.1109/ACCESS.2017.2693379.

[161] J. Yu. State of health prediction of lithium-ion batteries: Multiscale logic regression and Gaussian process regression ensemble. *Reliability Engineering & System Safety*, 174:82–95, 2018. DOI: 10.1016/j.ress.2018.02.022.

[162] M. Yuan, Y. Wu, and L. Lin. Fault diagnosis and remaining useful life estimation of aero engine using LSTM neural network. In *IEEE International Conference on Aircraft Utility Systems*, pages 135–140. IEEE, 2016.

[163] S. Yuan, H. Wang, and J. Chen. A PZT based on-line updated guided wave-Gaussian process method for crack evaluation. *IEEE Sensors Journal*, 20(15):8204–8212, 2019. DOI: 10.1109/JSEN.2019.2960408.

[164] R. Zemouri, D. Racoceanu, and N. Zerhouni. Recurrent radial basis function network for time-series prediction. *Engineering Applications of Artificial Intelligence*, 16(5-6):453–463, 2003. DOI: 10.1016/S0952-1976(03)00063-0.

[165] R. Zhao, J. Wang, R. Yan, and K. Mao. Machine health monitoring with LSTM networks. In *10th International Conference on Sensing Technology*, pages 1–6. IEEE, 2016.