

A Computational Framework for Efficient Reliability Analysis of Complex Networks

Von der Fakultät für Bauingenieurwesen und Geodäsie
der Gottfried Wilhelm Leibniz Universität Hannover
zur Erlangung des Grades

**Doktor-Ingenieur
Dr.-Ing.**

genehmigte Dissertation
von

Jasper Behrendorf, M. Sc.

2023

Referent: Prof. Dr.-Ing. Michael Beer
Korreferenten: Prof. Enrico Zio
Prof. Dr.-Ing. Martin Achmus
Prof. Dr.-Ing. Ingo Neumann

Tag der Promotion: 7. November 2023

Eidesstattliche Versicherung

Hiermit versichere ich, die Regeln der geltenden Promotionsordnung zu kennen und eingehalten zu haben und mit einer Prüfung nach den Bestimmungen der Promotionsordnung einverstanden zu sein, die Dissertation selbst verfasst zu haben, keine Textabschnitte von Dritten oder eigener Prüfungsarbeiten ohne Kennzeichnung übernommen und alle von mir benutzten Hilfsmittel und Quellen in meiner Arbeit angegeben zu haben, Dritten weder unmittelbar noch mittelbar geldwerte Leistungen für Vermittlungstätigkeiten oder für die inhaltliche Ausarbeitung der Dissertation erbracht zu haben, die Dissertation noch nicht als Prüfungsarbeit für eine staatliche oder andere wissenschaftliche Prüfung oder die gleiche oder eine in wesentlichen Teilen ähnliche Arbeit bei einer anderen Hochschule als Dissertation eingereicht zu haben.

Hannover, 12. September 2023

.....
(Unterschrift)

Zusammenfassung

Mit der zunehmenden Größe und Komplexität moderner Infrastrukturnetze stellt sich die Herausforderung, effiziente und zuverlässige Methoden zur Analyse ihrer Zuverlässigkeit zu entwickeln. Besonderes Augenmerk muss dabei auf potenzielle Abhängigkeiten zwischen den Netzen gelegt werden, da deren Vernachlässigung zu katastrophalen Ausfällen führen kann. Darüber hinaus ist es von größter Wichtigkeit, alle Unsicherheiten angemessen zu behandeln. Die Überlebenssignatur ist eine jüngere Entwicklung zur effektiven Analyse komplexer Netze, die entscheidende Vorteile gegenüber traditionelleren Methoden bietet. Ihr wichtigstes Merkmal ist die vollständige Trennung der Systemstruktur von probabilistischen Informationen. Dadurch können verschiedene Phänomene wie Abhängigkeiten von Komponenten, Ausfälle aufgrund gemeinsamer Ursache und ungenaue Wahrscheinlichkeiten berücksichtigt werden, ohne dass die Netzstruktur neu ausgewertet werden muss.

In dieser kumulativen Dissertation werden mehrere wichtige Neuerungen für die Überlebenssignatur vorgestellt, die sich auf die strukturelle Auswertung des Systems sowie auf die Modellierung von Komponentenausfällen konzentrieren.

Es wird eine neue Methode vorgestellt, bei der Abhängigkeiten zwischen Komponenten und Netzwerken mit Hilfe von Vine-Copulas modelliert werden. Darüber hinaus werden aleatorische und epistemische Unsicherheiten durch die Anwendung von Wahrscheinlichkeitsboxen und unscharfen Copulas berücksichtigt. Durch die Nutzung der großen Anzahl verfügbarer Copula-Familien ist es möglich, unterschiedliche Effekte von Abhängigkeiten zu berücksichtigen. Das graphenbasierte Design der Vine-Copulas ergänzt sich gut mit den typischen Beschreibungen von Netzwerktopologien. Die vorgeschlagene Methode wird an einem anspruchsvollen Szenario basierend auf dem IEEE-Zuverlässigkeitstestsystem getestet, um ihre Effektivität zu demonstrieren und die Fähigkeit hervorzuheben, komplizierte Szenarien mit einer Vielzahl von abhängigen Fehlermechanismen zu modellieren.

Der numerische Aufwand, der zur analytischen Berechnung der Überlebenssignatur erforderlich ist, ist für große und komplexe Systeme nicht zu bewältigen. Daher werden in dieser Arbeit zwei Methoden zur Approximation der Überlebenssignatur vorgestellt. Im ersten Ansatz werden Systemkonfigurationen von geringem Interesse mit Hilfe der Perkolationstheorie ausgeschlossen, während die verbleibenden Teile der Signatur durch Monte-Carlo-Simulation geschätzt werden. Die Methode ist in der Lage, die Überlebenssignatur mit sehr kleinen Fehlern genau zu approximieren und gleichzeitig den Rechenaufwand drastisch zu reduzieren. Mehrere einfache Testsysteme sowie zwei reale Situationen werden verwendet, um die Genauigkeit und Effizienz zu verdeutlichen.

Mit zunehmender Netzgröße und -komplexität stößt jedoch auch dieses Verfahren an seine Grenzen. Daher wird eine zweite Methode vorgestellt, bei der der numerische Bedarf weiter reduziert wird. In dieser wird nicht die gesamte Überlebenssignatur approximiert, sondern nur einige strategisch ausgewählte Werte mit Hilfe der Monte-Carlo-Simulation berechnet und zum Aufbau eines Ersatzmodells auf der Grundlage normalisierter radialer Basisfunktionen verwendet. Die Unsicherheit, die sich aus der Approximation der Datenpunkte ergibt, wird anschließend durch ein Intervallvorhersagemodell propagiert, das Ober- und Untergrenzen für die verbleibenden Überlebenssignaturwerte und dadurch auch für die Zuverlässigkeit des Netzwerks schätzt. Da wenige Datenpunkte ausreichen, um das Intervallvorhersagemodell zu erstellen, können noch größere Systeme analysiert werden.

Mit steigender Komplexität nicht nur des Systems, sondern auch der einzelnen Komponenten selbst ergibt sich die Notwendigkeit, die Komponenten als Subsysteme zu modellieren, wodurch sich komplexe Systeme aus Systemen ergeben. Es wird eine Arbeit vorgestellt, in der eine zuvor entwickelte Methode zur Resilienzentscheidungsfindung für mehrdimensionale Szenarien erweitert wird, in denen die Subsysteme als Überlebenssignaturen dargestellt werden. Die Überlebenssignatur der Teilsysteme kann aufgrund der inhärenten Trennung der Strukturinformationen vor der Resilienzanalyse berechnet werden. Dies ermöglicht eine effiziente Analyse, bei der die Ausfallraten der Subsysteme für verschiedene resilienzsteigernde Ausstattungen direkt aus der Überlebensfunktion berechnet werden, ohne die Systemstruktur neu auszuwerten.

Zusätzlich zu den Fortschritten auf dem Gebiet der Überlebenssignatur wird in dieser Arbeit auch eine neue Bibliothek für die Quantifizierung von Unsicherheiten mit dem Namen `UncertaintyQuantification.jl` vorgestellt, die als Paket in der Programmiersprache Julia entwickelt wurde. Julia ist eine moderne dynamische High-Level-Programmiersprache, die sich ideal für Aufgaben wie Datenanalyse und wissenschaftliches Rechnen eignet. `UncertaintyQuantification.jl` wurde von Grund auf so entwickelt, dass es generalisiert, flexibel und gleichzeitig einfach zu benutzen ist. Das Framework wird ständig weiterentwickelt und hat zum Ziel, eine Toolbox zu werden, die state-of-the-art Algorithmen aus allen Bereichen der Unsicherheitsquantifizierung umfasst und als wertvolles Werkzeug sowohl für die Forschung als auch für die Industrie dienen soll. Zum aktuellen Funktionsumfang von `UncertaintyQuantification.jl` gehören simulationsbasierte Zuverlässigkeitsanalysen unter Verwendung einer Vielzahl von Verfahren zur Erzeugung von Stichproben, lokale und globale Sensitivitätsanalysen sowie Metamodellierungstechniken.

Schlüsselwörter: Netzwerkzuverlässigkeit, Überlebenssignatur, Abhängigkeiten, Copulas, Monte-Carlo-Simulation, radiale Basisfunktionen, Intervallvorhersagemodelle

Abstract

With the growing scale and complexity of modern infrastructure networks comes the challenge of developing efficient and dependable methods for analysing their reliability. Special attention must be given to potential network interdependencies as disregarding these can lead to catastrophic failures. Furthermore, it is of paramount importance to properly treat all uncertainties. The survival signature is a recent development built to effectively analyse complex networks that far exceeds standard techniques in several important areas. Its most distinguishing feature is the complete separation of system structure from probabilistic information. Because of this, it is possible to take into account a variety of component failure phenomena such as dependencies, common causes of failure, and imprecise probabilities without reevaluating the network structure.

This cumulative dissertation presents several key improvements to the survival signature ecosystem focused on the structural evaluation of the system as well as the modelling of component failures.

A new method is presented in which (inter)-dependencies between components and networks are modelled using vine copulas. Furthermore, aleatory and epistemic uncertainties are included by applying probability boxes and imprecise copulas. By leveraging the large number of available copula families it is possible to account for varying dependent effects. The graph-based design of vine copulas synergizes well with the typical descriptions of network topologies. The proposed method is tested on a challenging scenario using the IEEE reliability test system, demonstrating its usefulness and emphasizing the ability to represent complicated scenarios with a range of dependent failure modes.

The numerical effort required to analytically compute the survival signature is prohibitive for large complex systems. This work presents two methods for the approximation of the survival signature. In the first approach system configurations of low interest are excluded using percolation theory, while the remaining parts of the signature are estimated by Monte Carlo simulation. The method is able to accurately approximate the survival signature with very small errors while drastically reducing computational demand. Several simple test systems, as well as two real-world situations, are used to show the accuracy and performance.

However, with increasing network size and complexity this technique also reaches its limits. A second method is presented where the numerical demand is further reduced. Here, instead of approximating the whole survival signature only a few strategically selected values are computed using Monte Carlo simulation and used to build a surrogate model based on normalized radial basis functions. The uncertainty resulting from the approximation of the data points is then

propagated through an interval predictor model which estimates bounds for the remaining survival signature values. This imprecise model provides bounds on the survival signature and therefore the network reliability. Because a few data points are sufficient to build the interval predictor model it allows for even larger systems to be analysed.

With the rising complexity of not just the system but also the individual components themselves comes the need for the components to be modelled as subsystems in a system-of-systems approach. A study is presented, where a previously developed framework for resilience decision-making is adapted to multidimensional scenarios in which the subsystems are represented as survival signatures. The survival signature of the subsystems can be computed ahead of the resilience analysis due to the inherent separation of structural information. This enables efficient analysis in which the failure rates of subsystems for various resilience-enhancing endowments are calculated directly from the survival function without reevaluating the system structure.

In addition to the advancements in the field of survival signature, this work also presents a new framework for uncertainty quantification developed as a package in the Julia programming language called `UncertaintyQuantification.jl`. Julia is a modern high-level dynamic programming language that is ideal for applications such as data analysis and scientific computing. `UncertaintyQuantification.jl` was built from the ground up to be generalised and versatile while remaining simple to use. The framework is in constant development and its goal is to become a toolbox encompassing state-of-the-art algorithms from all fields of uncertainty quantification and to serve as a valuable tool for both research and industry. `UncertaintyQuantification.jl` currently includes simulation-based reliability analysis utilising a wide range of sampling schemes, local and global sensitivity analysis, and surrogate modelling methodologies.

Keywords: Network reliability, survival signature, dependencies, copulas, Monte Carlo simulation, radial basis function networks, interval predictor models

“If it weren’t for the last
minute, nothing would get
done.”

– Rita Mae Brown

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Dependencies	2
1.3	Uncertainties	3
1.3.1	Characterization of uncertainties	4
1.3.2	Modelling of uncertainties	5
1.4	Signatures	8
1.4.1	System signature	8
1.4.2	Survival signature	9
1.5	Copulas	12
1.5.1	Copula families	14
1.5.2	Pair Copula Construction	16
1.6	Resilience	21
1.6.1	Resilience metrics	21
1.6.2	Adapted systemic risk measure	22
1.6.3	Grid search algorithm	23
1.6.4	Resilience decision-making	24
1.7	Software for Uncertainty Quantification	24
1.8	Aims and objectives	26
1.9	Original contributions	28
1.10	Structure of the thesis	29
2	Reliability Analysis of Networks Interconnected With Copulas	31
2.1	Introduction	32
2.2	Copulas	33
2.2.1	The Gaussian Copula	34
2.2.2	Archimedean Copulas	35
2.2.3	Dependence	35
2.3	Copula Construction Methods	37
2.3.1	Hierarchical Archimedean Copulas	37
2.3.2	Pair Copula Construction	38

2.3.3	Vine Copulas	38
2.4	Modelling Dependencies	39
2.4.1	Common Cause of Failure	40
2.4.2	Interdependencies	41
2.4.3	Construction of the dependence structure	42
2.5	Reliability Analysis	42
2.5.1	Survival Signature	42
2.5.2	Survival Function	44
2.5.3	Simulation	44
2.5.4	Imprecise Reliability Analysis	45
2.6	Numerical Example	47
2.7	Conclusion	52
3	Numerically Efficient Computation of the Survival Signature for the Reliability Analysis of Large Networks	53
3.1	Introduction	54
3.2	Survival Signature	55
3.3	Percolation	58
3.3.1	Percolation processes	58
3.3.2	Percolation threshold and survival signature	59
3.4	Approximation of the survival signature	60
3.4.1	Percolation	62
3.4.2	Monte Carlo Approximation	63
3.4.3	Reliability analysis	64
3.5	Numerical examples	65
3.5.1	Example 1: Electricity transmission network	65
3.5.2	Example 2: Berlin metro system	67
3.6	Conclusion	69
4	Imprecise Survival Signature Approximation Using Interval Predictor Models	70
4.1	Introduction	71
4.2	Survival Signature	72
4.3	Monotone Radial Basis Function Networks	73
4.4	Approximation of the Survival Signature	75
4.4.1	Design of the NRBF network surrogate model	75
4.4.2	Extension to an interval predictor model	76
4.5	Numerical Example	77
4.6	Conclusions	80

5	Interval Predictor Model for the Survival Signature Using Monotone Radial Basis Functions	82
5.1	Introduction	83
5.2	Survival Signature	85
5.3	Monotone Radial Basis Function Networks	87
5.4	Interval Predictor Models	89
5.5	Imprecise Approximation of the Survival Signature	91
5.5.1	Design of the NRBF network surrogate model	91
5.5.2	Extension to an interval predictor model	92
5.5.3	Method validation	93
5.6	Numerical Example	96
5.7	Conclusion	98
6	Multidimensional Resilience Decision-Making for Complex and Substructured Systems	100
6.1	Introduction	101
6.2	Resilience decision-making	103
6.2.1	Resilience quantification	104
6.2.2	Adapted systemic risk measure	106
6.2.3	Grid search algorithm and the curse of dimensionality	107
6.3	Concept of survival signature	108
6.3.1	Structure function	108
6.3.2	Survival signature	108
6.3.3	Probability structure	109
6.3.4	Survival function	109
6.4	Proposed methodology	109
6.4.1	Definition of substructured systems	110
6.4.2	Extension of the adapted systemic risk measure	111
6.4.3	Augmentation of the resilience analysis	112
6.5	Multistage high-speed axial compressor	114
6.5.1	Model	114
6.5.2	Costs of endowment properties	116
6.5.3	Scenario	116
6.6	Complex system	119
6.6.1	Model	119
6.6.2	Costs of endowment properties	122
6.6.3	Scenario	122
6.7	<i>U-Bahn</i> and <i>S-Bahn</i> system of Berlin	125
6.7.1	Model	125

6.7.2	Costs of endowment properties	130
6.7.3	Scenario	131
6.8	Conclusion and outlook	135
7	UncertaintyQuantification.jl: A new Framework for Uncertainty Quantification in Julia	137
7.1	Introduction	138
7.2	Getting Started	139
7.2.1	Inputs	140
7.2.2	Models	142
7.3	Reliability Analysis	146
7.3.1	First order reliability method	146
7.3.2	Monte Carlo Simulation	147
7.4	Sensitivity Analysis	147
7.4.1	Local Sensitivity Analysis	148
7.4.2	Global Sensitivity Analysis	148
7.5	Metamodelling	149
7.5.1	Polyharmonic spline	149
7.6	Numerical Examples	150
7.6.1	Estimation of small failure probabilities	150
7.6.2	Global Sensitivity Analysis of the Ishigami function	152
7.6.3	Metamodelling and Global Sensitivity Analysis	153
7.7	Conclusion	154
8	Conclusions and outlook	156
8.1	Conclusions	156
8.2	Outlook	157
9	List of publications	159
	Bibliography	162

1 Introduction

1.1 Motivation

Efficient reliability analysis of complex infrastructure networks such as power grids, gas and water distribution systems or communication networks is of paramount importance for their safety and operation. The 2008 “Directive on European Critical Infrastructures” defines critical infrastructures as

“critical infrastructure” means an asset, system or part thereof [...] which is essential for the maintenance of vital societal functions, health, safety, security, economic or social well-being of people, and the disruption or destruction of which would have a significant impact [...] as a result of the failure to maintain those functions ([1]).

Society’s increasing reliance on the availability of these systems [2] highlights the need for adequate reliability analysis.

The study of network reliability has received extensive attention from researchers over the past decades [3–5], since Moore and Shannon [6] introduced the first probabilistic network model in 1956. Traditional methods for network reliability problems can be roughly categorized into different fields [7]. Direct enumeration methods aim to obtain the reliability of the network from its underlying structure through complete state enumeration or enumeration of minimal path/cut sets [8–11]. Instead of computing the reliability of a large network directly, decomposition methods subdivide the network into smaller more manageable subsystems and obtain the reliability of the full network from the reliabilities of the subsystems [12–17]. The application of Monte Carlo simulation to circumvent the difficulty of solving NP-hard problems has a long history [18–22]. Recently, the combination of Monte Carlo simulation with machine learning techniques has seen increasing research focus [23–25]. Others have applied fault tree analysis to the reliability analysis of power networks [26].

Recently, the survival signature [27] has emerged as a competent tool for the reliability analysis of complex networks. Based on its inherent separation of structural from probabilistic information it provides the means for numerically efficient reliability analysis. Since the structure of the system is evaluated separately, a plethora of effects can be included in the probabilistic modelling of the system failures. These include phenomena such as dependent failures, common cause failures, or imprecise probabilities. Traditional techniques often struggle to include these effects or become intractable for larger networks.

The survival signature is a generalization of the system signature [28] and has seen wide application since its introduction. Coolen et al. [29] used the survival signature to perform nonparametric predictive inference for system reliability. A generalization to unrepairable multi-state components was presented in [30]. A continuous-state survival function and corresponding surrogate model were developed by Winnewisser et al. [31]. George-Williams et al. [32] extended the survival signature to complex systems with non-repairable dependent failures. Other applications to dependent failures can be found in [33, 34]. Recently, Zheng and Zhang [35] have investigated Archimedean copulas to model dependent components. Other areas of interest where the survival signature has seen successful application are imprecise reliability analysis [33] and component importance [33, 36]. Efficient simulation methods for the survival signature were introduced by [37]. Ample research has also focussed on efficient computation of the survival signature [38–41].

1.2 Dependencies

Events such as the 2003 blackout in Italy have highlighted the need for adequate consideration of interdependencies between critical infrastructures [42]. In this event, cascading failures because of network interdependencies lead to a large-scale blackout. The failure of power stations directly caused the failure of nodes on the Internet which subsequently caused more power stations to break down. In this case, there are bidirectional dependencies between the power network and the Internet as power stations depend on the Internet for control and the nodes of the Internet depend on the power stations for electricity. This cascading breakdown is illustrated in Figure 1.1 where one failure in the power system in Figure 1.1a removes nodes from the Internet which in turn causes more nodes in the power system to fail so that after only two steps a large portion of the power net has failed in Figure 1.1c.

In general, dependencies are defined as connections between two networks, such as the electricity required to power nodes of the Internet as seen in the previous example. This is usually a unidirectional relationship where infrastructure a depends on infrastructure b but b does not depend on a [43]. On a lower level, dependencies may also be described between components inside an infrastructure.

For multiple infrastructures connected in a system-of-systems, dependencies are no longer unidirectional. Instead, there are bidirectional interdependencies where infrastructure a depends on infrastructure b through some connections and b is dependent on a through different connections. Infrastructures are *interdependent* when both are *dependent* on the other. [43]

Rinaldi et al. [43] classify interdependencies of critical infrastructures into four types. *Physical* dependencies describe the dependence of one system on the output of another. If a system depends on the information flowing through another system, these interdependencies are classified as *cyber*. *Geographic* interdependencies occur when multiple systems share the same location and are subject to the same environmental conditions. Finally, *logical* interdependencies

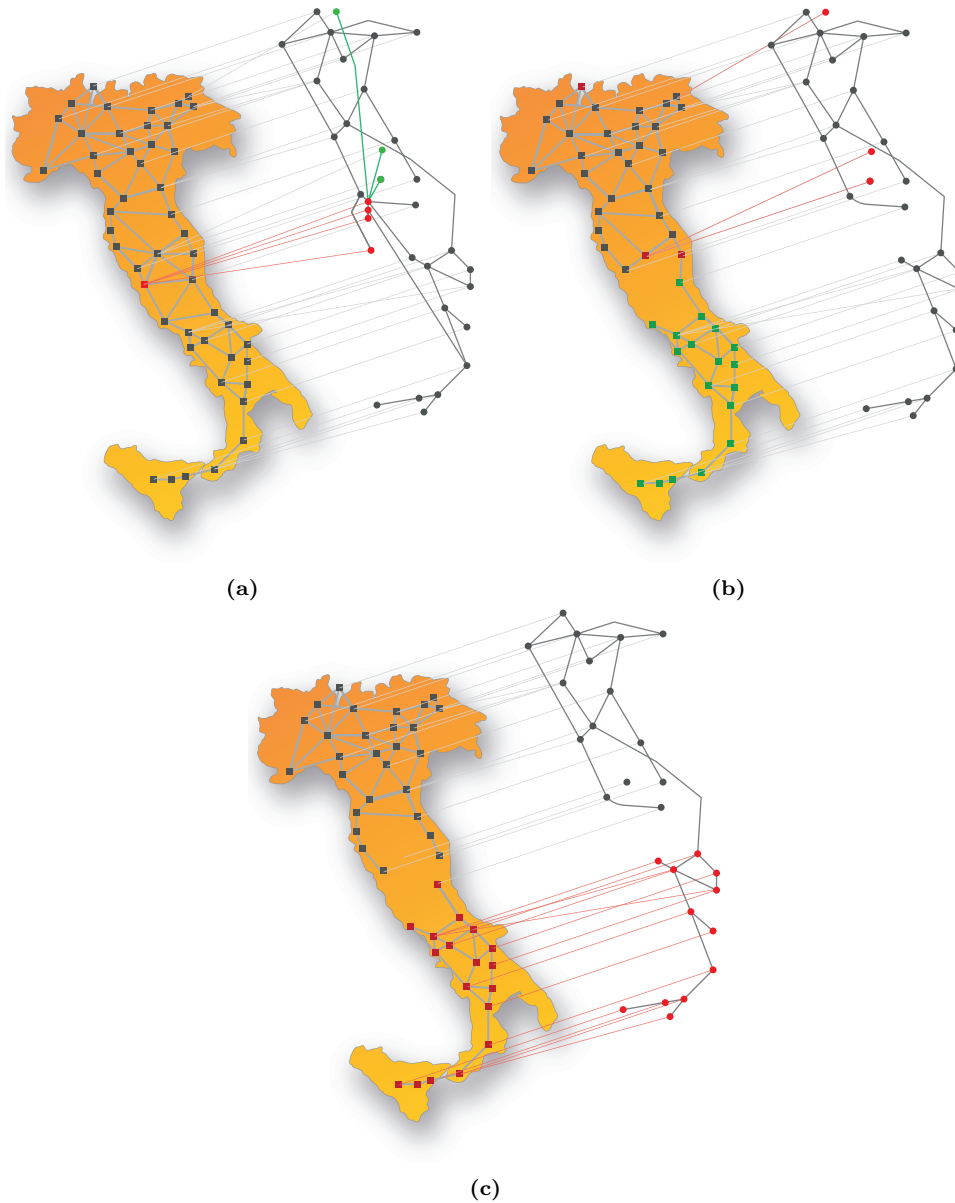


Figure 1.1: Cascading failures leading to the 2003 blackout in Italy. The figure presents the power grid as an overlay on the map of Italy and the Internet shifted above. In each subfigure, red nodes indicate newly failed nodes and green nodes are nodes that fail in the next step. Adapted from [42].

are any interdependency not covered by the previous three types.

The study of interdependent critical infrastructure networks is the target of increasing research focus, especially in the context of reliability [44–47] and resilience [42, 48, 49].

1.3 Uncertainties

This section defines different classes of uncertainties and provides brief introductions into modelling approaches.

1.3.1 Characterization of uncertainties

In line with Nikolaidis [50], the definition of *uncertainty* follows logically from the absence of *certainty* describing a state of absolute knowledge where everything there is to know about a process is known. This however is a theoretical and unachievable state in which deterministic models would be sufficient for the analysis of engineering systems. In reality, there is always a gap between certainty and the current state of knowledge resulting in *uncertainty*. This concept is further illustrated in Figure 1.2.

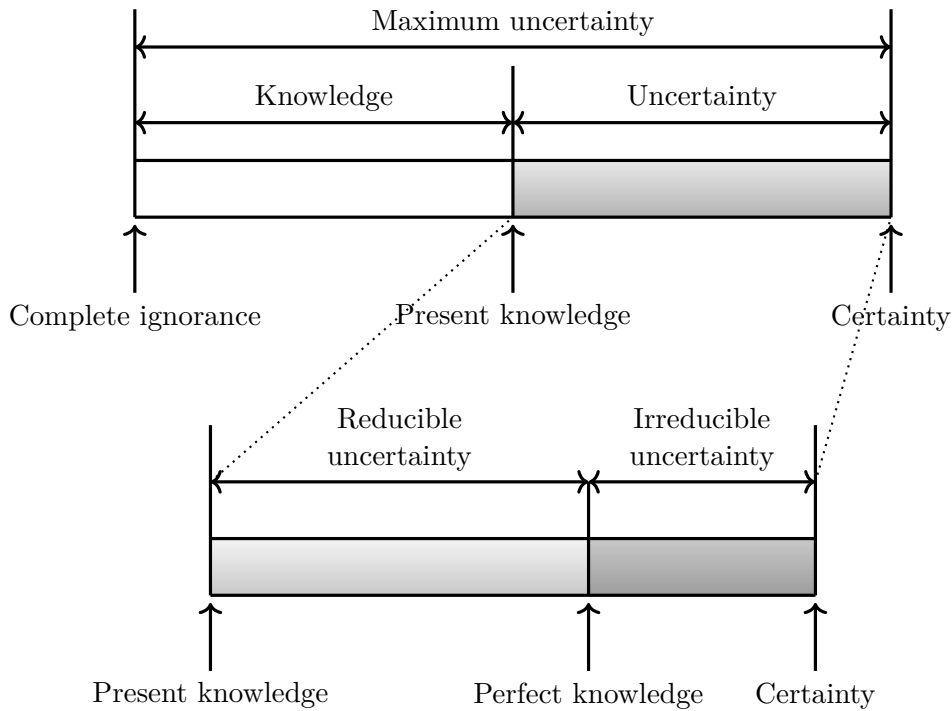


Figure 1.2: Characterization of reducible and irreducible uncertainties. Adapted from [51].

Although a topic of ongoing philosophical debate [51], it is largely accepted that uncertainty can be further classified into two types, namely *aleatory* and *epistemic* uncertainty [52]. The first type, *aleatory* uncertainties, are also called *irreducible* uncertainties or *variability* and describe the inherent randomness of a process. This could, for example, be variability in material properties, degradation of components, or varying external forces such as wind loads or earthquakes. Some researchers debate the existence of aleatory uncertainty under the assumption that if a process was fully understood it would no longer be random. Epistemic uncertainty is the uncertainty resulting from a lack of knowledge or vagueness and is also called *reducible* uncertainty as it can be reduced through the collection of additional data and information. If both types of uncertainties occur together this is known as *hybrid* uncertainty, see for example [53].

According to Bi et al. [54] uncertainties can be divided into four categories:

- Category I: Constant parameters without any associated uncertainty
- Category II: Parameters only subject to epistemic uncertainty represented as intervals
- Category III: Parameters with only aleatory uncertainties represented as precise probability distributions fully described by probability distributions.
- Category IV: Parameters subject to both aleatory and epistemic uncertainty represented as imprecise probabilities described by for example probability-boxes

1.3.2 Modelling of uncertainties

Next follows a brief introduction to the modelling of precise (category III) and imprecise probabilities (category IV).

A precise probability model of an experiment or random process is mathematically described by a probability space (Ω, \mathcal{F}, P) [55]. Here, Ω is the sample space containing all possible outcomes ω of the model or experiment expressed as

$$\Omega = \{\omega_1, \omega_2, \dots, \omega_n\}. \tag{1.1}$$

The σ -algebra (or σ -field) $\mathcal{F} \subseteq 2^\Omega$ is the event space and defined as a set of subsets of Ω containing the events under consideration. Events are the outcomes of an experiment that are of interest. To define the subset representing an event, all outcomes in Ω are considered. If the outcome leads to the event happening, it is added to the subset. Events containing only a single outcome are called *elementary* events. The empty set $\emptyset \in \mathcal{F}$ is an impossible event and the event containing all outcomes, i.e. Ω is a certain event. In addition, for any event $A \in \mathcal{F}$ follows $\bar{A} \in \mathcal{F}$ where \bar{A} is the complement of A , i.e. the event that A does not happen. Finally, the set-function P is a probability measure assigning a probability to each event in \mathcal{F} such that $P : \mathcal{F} \rightarrow [0, 1]$.

The definition of the probability measure is grounded in Kolmogorov's axioms [56]. For an elementary event $A \in \Omega$ the first axiom is

$$P(A) \geq 0, \tag{1.2}$$

which states that the probability assigned to an elementary event is non-negative. The second axiom

$$P(\Omega) = 1 \tag{1.3}$$

defines that the probability for the certain event is 1 while the third axiom states that for two mutually exclusive events A and B in Ω the probability that either one or both of the events will happen is the sum of the probabilities of the events. As such it allows defining probabilities

1 Introduction

for complex events such as

$$P(A \cup B) = P(A) + P(B), \quad (1.4)$$

if $A \cap B = \emptyset$. These three axioms in combination with set theory are sufficient to develop probability theory. For a thorough introduction to probability theory, refer to [55].

As a discrete example consider the throw of a six-sided dice. Here $\Omega = \{1, 2, 3, 4, 5, 6\}$ and \mathcal{F} contains all possible subsets of Ω . This includes elementary events such as the dice landing on a specific number $\{1\}$ or $\{6\}$ as well as complex events, for example the dice landing on an even $\{2, 4, 6\}$ or odd $\{1, 3, 5\}$ number. The probability assigned to each event is the cardinality of the event, i.e. the size of the set, divided by 6 (the number of possible outcomes ω). For example $P(\{1\}) = \frac{1}{6}$ and $P(\{2, 4, 6\}) = \frac{3}{6} = \frac{1}{2}$. The latter follows directly from a generalization of the third axiom since all the elementary events are mutually exclusive.

A random variable X is uniquely defined by its cumulative density function (CDF) $F : \mathbb{R} \rightarrow [0, 1]$. By definition, it returns the probability that the random variable will take a value less than or equal to x

$$F_X(x) = P_X(X \leq x). \quad (1.5)$$

A CDF is a right continuous and monotonically non-decreasing function which satisfies

$$\lim_{x \rightarrow -\infty} F_X(x) = 0 \quad (1.6)$$

and

$$\lim_{x \rightarrow \infty} F_X(x) = 1. \quad (1.7)$$

Consider a small interval of width Δx on the real line \mathbb{R} and the probability that the value of a random variable X falls in this interval

$$P(x < X \leq (x + \Delta x)). \quad (1.8)$$

This probability varies with the magnitude of Δx with a larger interval leading to a higher probability. Normalizing the probability as

$$\frac{1}{\Delta x} P(x < X \leq (x + \Delta x)) \quad (1.9)$$

gives the *density* of the probability over the interval. The limit of this density when the interval approaches 0 gives the so-called probability density function (PDF) of X . The PDF of a random variable can be obtained as the derivative of the CDF

$$f_X(x) = \frac{dF_X(x)}{dx} \quad (1.10)$$

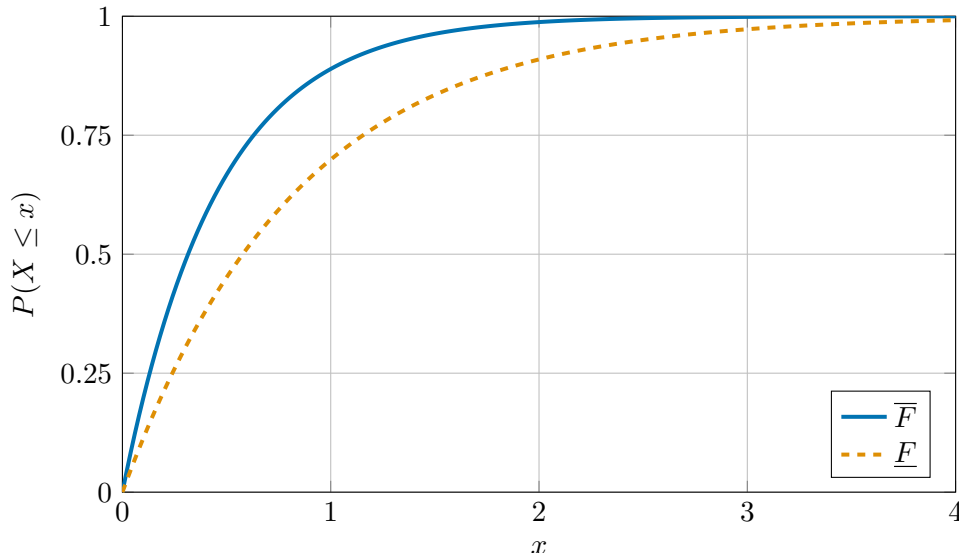


Figure 1.3: Example of an exponential p-box with $\lambda \in [1.2, 2.2]$.

if it exists. Conversely, the CDF can be defined as the integral of the PDF as

$$F_X(x) = \int_{-\infty}^x f_X(\lambda) d\lambda. \quad (1.11)$$

Using the PDF and CDF random variables subject to aleatory uncertainty can be described using well established probability theory.

Ferson et al. [57] introduced the notion of a *probability box* (p-box) for the modelling of imprecise probabilities. Consider two CDFs \underline{F} and \overline{F} with $\underline{F}(x) \leq \overline{F}(x)$ for all $x \in \mathbb{R}$. Then, $[\underline{F}(x), \overline{F}(x)]$ is the set of CDFs F such that $\underline{F}(x) \leq F(x) \leq \overline{F}(x)$. This set is called the p-box for an imprecisely known random variable X , where $\underline{F}(x)$ is the lower bound for the probability that X is smaller than or equal to x , and $\overline{F}(x)$ is the upper bound of this probability. For an example of a p-box described by two CDFs of the exponential distribution with parameter $\lambda \in [1.2, 2.2]$ refer to Figure 1.3. Here, only two CDFs are required to fully define the p-box. However, for distribution families described by multiple parameters four or more CDFs are needed to define the p-box. If only the outer CDFs are known and no further information about the CDFs between is available this is known as a distribution-free p-box where every possible CDF between the bounds is a valid CDF for the imprecise random variable X [57].

The epistemic uncertainty is propagated into the result by feeding the bounds of one or more p-boxes into a reliability analysis. As a result, the analysis returns an upper and lower bound on the reliability.

1.4 Signatures

Signatures are condensed representations of a system's topology. Their inherent separation of structural information from the probabilistic component failure times allows for efficient evaluation of the system reliability. Both the system signature [28] and its generalization the survival signature [27] are defined for *coherent* systems.

To begin, consider a system with m components. The state vector $\mathbf{x} = (x_1, x_2, \dots, x_m) \in \{0, 1\}^m$ represents the binary states of all components x_i in the system with $x_i = 1$ if a component is working and $x_i = 0$ if it is in a failed state. The components are arbitrarily labelled inside the state vector. However, it is important that this labelling is kept consistent throughout the analysis. The overall state of a system given a state vector is evaluated through the so-called structure function. This function $\varphi(\mathbf{x}) : \{0, 1\}^m \rightarrow \{0, 1\}$ maps each state vector \mathbf{x} for which the system is working to 1 and the state vectors of failed systems to 0.

A system is *coherent* under two conditions: The structure function is monotone and all n components are relevant to the system's functionality. In this case, the monotonicity of the structure function ensures that a system can not degrade (or fail) by restoring a failed component to working condition. Mathematically, this is defined as $\varphi(\mathbf{x}) \leq \varphi(\mathbf{y})$ for any two state vectors \mathbf{x} and \mathbf{y} if $\mathbf{x} \leq \mathbf{y}$. The comparison of state vectors is applied component-wise as $x_i \leq y_i$ for all $i \in \{1, 2, \dots, m\}$. This condition also implies $\varphi(\mathbf{0}) = 0$ and $\varphi(\mathbf{1}) = 1$.

A component is considered relevant if it contributes to the structure function. In the opposite case, component i is irrelevant if

$$\varphi(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_m) = \varphi(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_m) \quad (1.12)$$

for all possible combinations of $(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_m)$. Any component that is not relevant to the system can be removed to simplify the system while retaining the same functionality.

1.4.1 System signature

The system signature for coherent systems with binary state components was introduced by Samaniego [28]. Assuming i.i.d component failure times the system signature \mathbf{s} is a vector with m elements where the i th entry is the probability that the failure of i components results in the failure of the system. If $T_S > 0$ is the failure time of the system and T_i is the i th component failure then the signature is

$$\mathbf{s}_i = P(T_S = T_i). \quad (1.13)$$

Computation of the system signature is a combinatorial problem. The probability \mathbf{s}_i is obtained by generating all $m!$ permutations of component failures and calculating the fraction of permutations where the i th failure results in the failure of the system. From the assumptions about coherent systems follow that $\sum_i^m \mathbf{s}_i = 1$. This task is computationally demanding

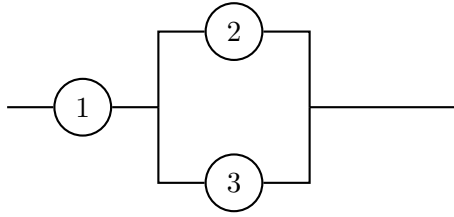


Figure 1.4: Simple system with three components.

for large systems. Even for a relatively small system with ten components, $10! = 39916800$ permutations of failure times must be evaluated.

As an example, consider a simple system of three components as shown in Figure 1.4. To obtain the signature of this system $3! = 6$ permutations of the component failure times have to be evaluated. Two permutations $X_1 < X_2 < X_3$ and $X_1 < X_3 < X_2$ lead to a system failure after one component fails, resulting in $\mathbf{s}_1 = \frac{1}{3}$. The remaining four permutations $X_2 < X_1 < X_3$, $X_2 < X_3 < X_1$, $X_3 < X_1 < X_2$, and $X_3 < X_2 < X_1$ all lead to system failure after the second component failure, which in turn means that $\mathbf{s}_2 = \frac{2}{3}$. A maximum of two component failures are required for this system to fail which implies $\mathbf{s}_3 = 0$. The full signature is therefore $\mathbf{s} = (\frac{1}{3}, \frac{2}{3}, 0)$.

The reliability of the system given by the survival function is then defined as

$$P(T_S > t) = \sum_i^m \mathbf{s}_i P(T_i > t). \quad (1.14)$$

If the failure times of the components follow a known distribution with CDF $F(t)$, the survival function can be analytically computed as

$$P(T_i > t) = \sum_{i=1}^m \mathbf{s}_i \sum_{j=0}^{i-1} \binom{m}{j} [F(t)]^j [1 - F(t)]^{m-j}. \quad (1.15)$$

1.4.2 Survival signature

Real world systems are often built from multiple different component types. Therefore, Coolen and Coolen-Maturi [27] generalized the system signature to systems with multiple types of components. This new signature is called the survival signature.

Consider a system of K component types. The survival signature is defined as the probability that the system is working while precisely l_k out of m_k components of each type $k \in \{1, 2, \dots, K\}$ are working as

$$\Phi(l_1, \dots, l_K) = \left[\prod_{k=1}^K \binom{m_k}{l_k}^{-1} \right] \times \sum_{\mathbf{x} \in S_{l_1, \dots, l_K}} \varphi(\mathbf{x}), \quad (1.16)$$

where the set S_{l_1, \dots, l_K} contains all possible state vectors for l_1, \dots, l_K working components. This set includes the states where no components of a certain type are functioning with $l_k = 0$. This

1 Introduction

makes the survival signature a K -dimensional array with dimensionality $(m_1+1) \times \dots \times (m_K+1)$. The individual values in this array will be referred to as *entries* throughout this dissertation.

As an example, consider a simple system of six components equally divided into two component types as shown in Figure 1.5. The survival signature of this system has $(k_1+1) \cdot (k_2+1) = 16$ entries. A closer look will be taken at the computation of $\Phi(1, 2)$. Since there are $\binom{m_1}{l_1} = \binom{3}{1} = 3$ possible configurations for one component of type 1 to work and $\binom{m_2}{l_2} = \binom{3}{2} = 3$ configurations for two components of type 2 to work, set $S_{1,2}$ contains nine state vectors. Of these nine, only the state vector where components 1, 3, and 6 are working will lead to a functioning system with $\varphi(1, 0, 1, 0, 0, 1) = 1$. As a result, the survival signature entry is calculated as $\Phi(1, 2) = \frac{1}{9}$. The full survival signature of the system is presented in Tab. 1.1. $2^6 = 64$ structure function evaluations are necessary to fully calculate it.

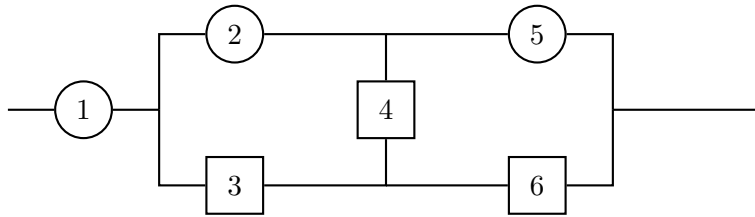


Figure 1.5: Simple system with six components divided into two types. The different node shapes refer to the component types.

Table 1.1: Survival signature of the network presented in Fig. 1.5.

l_1	l_2	$\Phi(l_1, l_2)$	l_1	l_2	$\Phi(l_1, l_2)$
0	0	0	2	0	0
0	1	0	2	1	0
0	2	0	2	2	4/9
0	3	0	2	3	6/9
1	0	0	3	0	1
1	1	0	3	1	1
1	2	1/9	3	2	1
1	3	3/9	3	3	1

The reliability of the system, which is the probability that it survives up to a point in time t , is given by the survival function as

$$P(T_s > t) = \sum_{l_1=0}^{m_1} \dots \sum_{l_K=0}^{m_K} \Phi(l_1, \dots, l_K) P\left(\bigcap_{k=1}^K \{C_t^k = l_k\}\right), \quad (1.17)$$

with C_t^k representing the number of components of type k functioning at time t . The separation of structural information (given by the survival signature) from the probabilistic information about the component failures can be clearly seen in the equation. As with the system signature,

there exists an analytical expression for the probabilistic part of the survival function in cases where the CDFs $F(t)_k$ of the failure time distributions of each component type are known as

$$P\left(\bigcap_{k=1}^K \{C_t^k = l_k\}\right) = \prod_{k=1}^K \left(\binom{m_k}{l_k} [F_k(t)]^{m_k-l_k} [1 - F_k(t)]^{l_k} \right). \quad (1.18)$$

Several simulation based algorithms have been introduced for cases where the CDFs are unknown [37].

The separation of probabilistic information enables the inclusion of various effects such as dependent component failures or imprecise failure time distributions. Feng et al. [58] have shown that when the failure times of the components are described as independent single-parameter p-box the upper and lower bounds of the system reliability $[\bar{P}, \underline{P}]$ in Eq. 1.18 can be directly obtained from the upper and lower bounds of the CDFs $[\bar{F}_k, \underline{F}_k]$

$$\bar{P}\left(\bigcap_{k=1}^K \{C_t^k = l_k\}\right) = \prod_{k=1}^K \left(\binom{m_k}{l_k} [\bar{F}_k(t)]^{m_k-l_k} [1 - \bar{F}_k(t)]^{l_k} \right) \quad (1.19a)$$

$$\underline{P}\left(\bigcap_{k=1}^K \{C_t^k = l_k\}\right) = \prod_{k=1}^K \left(\binom{m_k}{l_k} [\underline{F}_k(t)]^{m_k-l_k} [1 - \underline{F}_k(t)]^{l_k} \right). \quad (1.19b)$$

Returning to the example system given in Figure 1.5, assume the failure time distributions of both component types are defined as exponential p-boxes with the failure rates $\lambda_1 \in [0.8, 1.2]$ for component type 1 and $\lambda_2 \in [0.7, 0.9]$ for component type 2. The lower and upper bounds of the resulting reliability are presented in Figure 1.6.

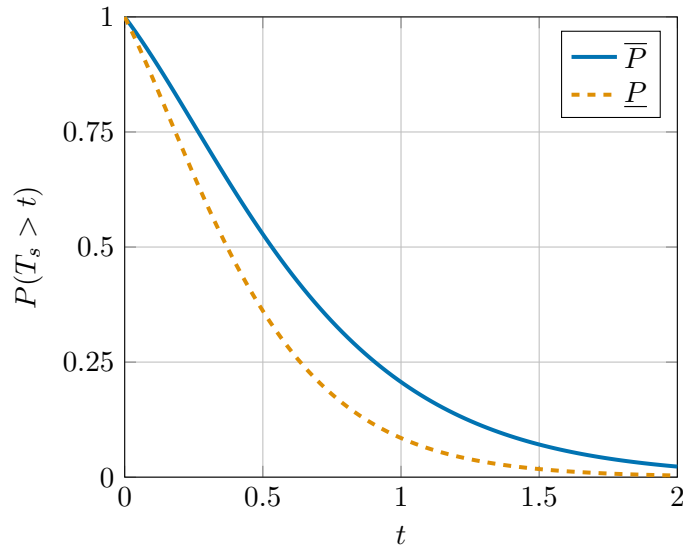


Figure 1.6: Upper and lower bounds for the reliability of the system shown in Figure 1.5.

1.5 Copulas

Copulas, from the Latin word for *bond* or *tie*, are multivariate distribution functions with the univariate marginal distributions all being $U(0, 1)$. Because these uniform marginals can be transformed to any marginal distribution through the inverse probability integral transform (using the inverse CDF) the copula essentially enables the coupling of different marginals into new multivariate distributions. Any multivariate distribution in dimension $d \geq 2$ can therefore be described by univariate marginals and a copula. This is known as Sklar's theorem [59].

Theorem 1.5.1 (Sklar's theorem). *Let H be a d -dimensional distribution function with marginals F_1, \dots, F_d . There exists a d -dimensional copula C such that*

$$H(\mathbf{x}) = C(F_1(x_1), \dots, F_d(x_d)), \forall \mathbf{x} \in \mathbb{R}^d \quad (1.20)$$

If the marginals F_1, \dots, F_d are continuous the copula is unique. Otherwise, it is only unique on $\text{Range}(F_1) \times \dots \times \text{Range}(F_d)$. If C is a d -dimensional copula and F_1, \dots, F_d are distribution functions, then the H is a d -dimensional distribution function with marginals F_1, \dots, F_d .

This separation of the copula from the marginal distribution functions provides great modelling flexibility and synergizes well with the separation of structural from probabilistic information in the survival signature presented in Eq. 1.17. For a thorough discussion of copulas, refer to [60, 61].

Copulas are invariant under strictly increasing transformation. As such, dependence among random variables can be studied by examining the underlying copula. This requires a dependence measure that is also invariant under strictly increasing transformations. Classically, *correlation* measured as the linear correlation coefficient ρ is used to measure dependence. Where the linear correlation coefficient ρ for two random variables is defined as

$$\rho(X, Y) = \frac{\text{Cov}(X, Y)}{\sqrt{\text{Var}(X)}\sqrt{\text{Var}(Y)}}, \quad (1.21)$$

with Var and Cov representing the variance and covariance.

However, ρ is not scale-invariant and may change when transforming from the copula to the marginal distributions. To illustrate this fact, consider the following example adapted from [62]. Figure 1.7 shows two scatter plots. On the left, 100 dependent samples have been drawn with $x_1 \sim U(0, 1)$ and $y_1 \sim U(0, x_1)$. On the right, these 100 samples are transformed from $[0, 1]$ to marginal distributions with $x_2 \sim \text{Exp}(1)$ and $y_2 \sim N(0, 1)$. Visually, these plots look quite different and computing the linear correlation coefficients $\rho(x_1, y_1) \approx 0.66776$ and $\rho(x_2, y_2) \approx 0.50486$ tells the same story. However, the underlying dependence structure has not changed at all. This makes ρ a poor measure of global dependence.

Two better suited scale-invariant dependence measures are Kendall's τ [63] and Spearman's ρ [64] both based on rank correlation instead of linear correlation. Kendall's τ will be used

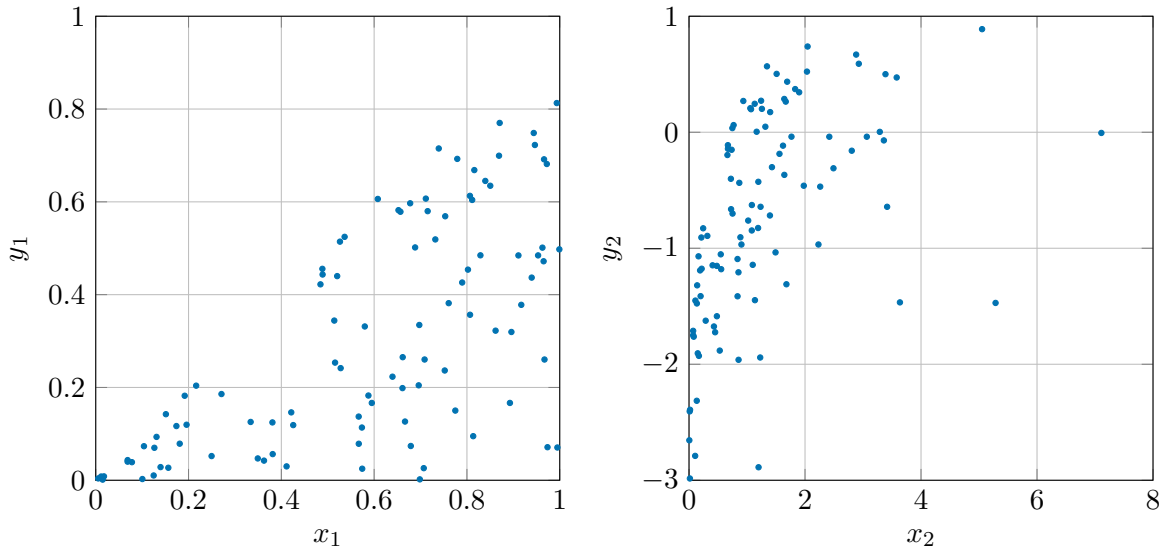


Figure 1.7: Comparison of dependent samples before and after transformation to marginal distributions.

throughout this dissertation.

Kendall's τ is defined based on *concordance*. Consider $(x_1, y_1), \dots, (x_n, y_n)$ to be a set of n observations of two random variables X and Y . Pairs of (x_i, y_i) and (x_j, y_j) are said to be *concordant* when $(x_i - x_j)(y_i - y_j) > 0$ and *discordant* if the reverse holds true. This means, that “large” values of one random variable are associated with “large” values of the other and the same for “small” values.

Let c be the number of concordant pairs and d be the number of discordant pairs, then Kendall's τ for the n observations is defined as

$$\tau = \frac{c - d}{c + d} = \frac{c - d}{\binom{n}{2}}. \quad (1.22)$$

Going back to the example presented in Figure 1.7 Kendall's τ for both plots is $\tau(x_1, y_1) = \tau(x_2, y_2) \approx 0.49818$. This proves, that it is better suited to measure dependence in a copula setting than the linear correlation coefficient.

The population version of Kendall's τ for X and Y is defined as the probability that a pair is concordant minus the probability that it is discordant for a random pair of observations

$$\tau = P[(X - X')(Y - Y') > 0] - P[(X - X')(Y - Y') < 0], \quad (1.23)$$

where (X', Y') is an independent copy of (X, Y) . To further underline the suitability of Kendall's τ one can show that it can be expressed solely based on the copula function (see, [61]) as

$$\tau = 4 \int_0^1 \int_0^1 C dC - 1. \quad (1.24)$$

1.5.1 Copula families

There are multiple families of copulas all exhibiting different dependence structures. Two of the most popular classes of copulas are presented here. Elliptical copulas, mainly the Gaussian copula, and Archimedean copulas.

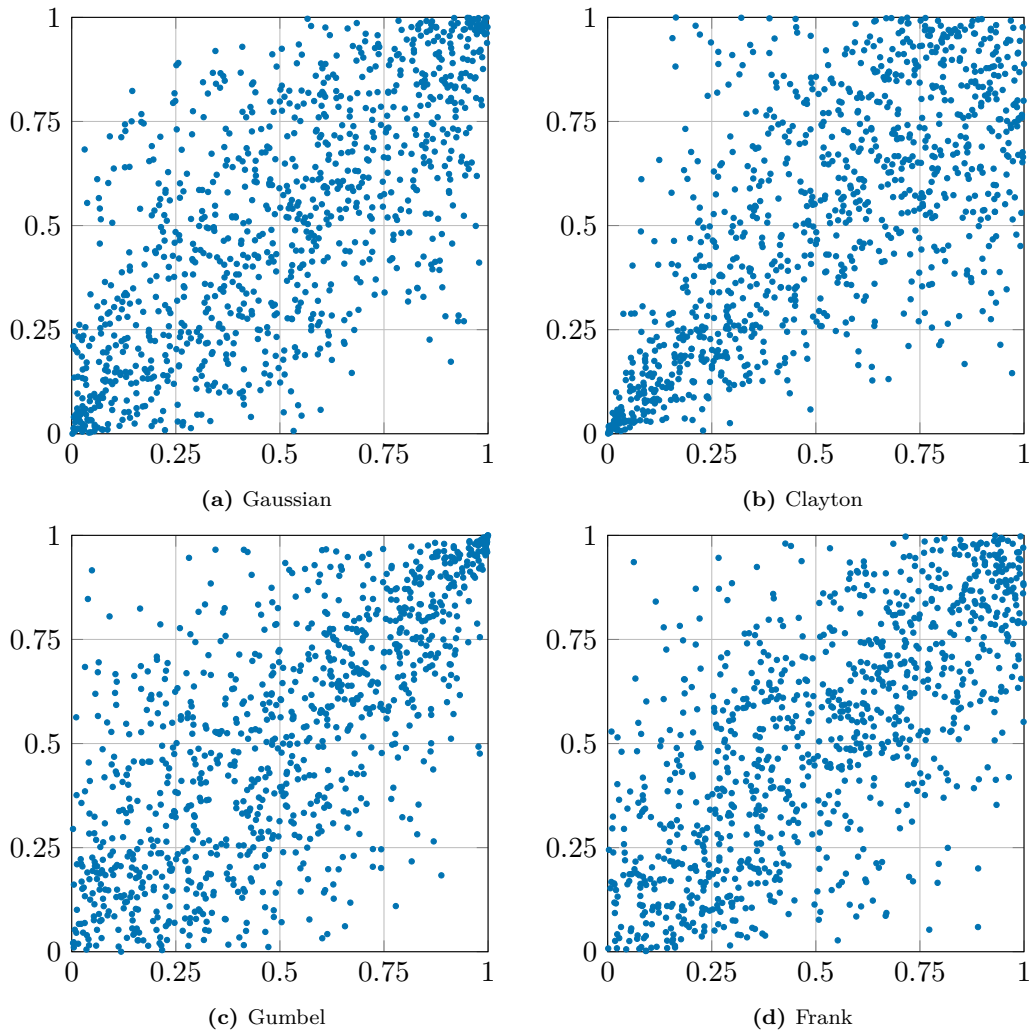


Figure 1.8: Samples drawn from different bivariate copulas families. The parameters have been chosen so that Kendall's $\tau = 0.5$.

Elliptical copulas

The family of elliptical copulas describes copulas underlying multivariate elliptical distributions. The most prominent elliptical distribution is the multivariate normal, providing the Gaussian copula. The d -dimensional Gaussian copula is defined as

$$C_R(u_1, \dots, u_d) = \Phi_R(\Phi^{-1}(u_1), \dots, \Phi^{-1}(u_d)), \quad (1.25)$$

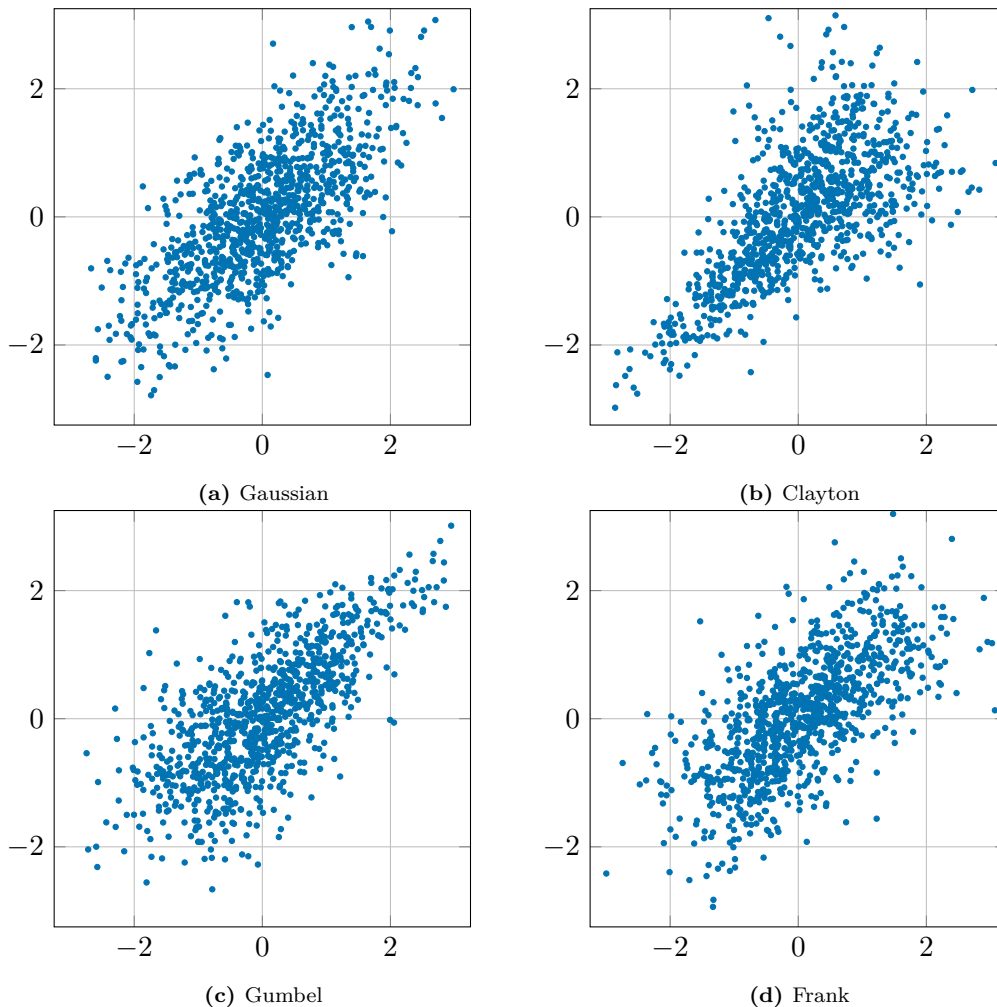


Figure 1.9: Samples drawn from the copulas presented in Figure 1.8 transformed to standard normal marginals.

where $R \in [-1, 1]^{d \times d}$ is a positive definite correlation matrix. Here, Φ_R is the joint CDF of the multivariate normal distribution with correlation matrix R and Φ^{-1} is the quantile function of the standard normal distribution [61]. A Gaussian copula with normal marginal distributions is the multivariate normal distribution.

Archimedean copulas

Archimedean copulas are another important family of copulas. Archimedean copulas are constructed from strictly monotone functions called *generators*. A generator $\varphi : [0, \infty] \rightarrow [0, 1]$ must be strictly decreasing with $\varphi(0) = 1$ and $\varphi(\infty) = 0$ [65]. Then, a d -dimensional Archimedean copula C_φ is defined as

$$C_\varphi(u_1, \dots, u_d) := \varphi(\varphi^{-1}(u_1) + \dots + \varphi^{-1}(u_d)), \quad (1.26)$$

where φ^{-1} is the inverse of the generator. Note that in some literature the order of φ and φ^{-1} is inverted. Some of the most popular parametric Archimedean copula families with their generators, inverses, and parameter domains are presented in Tab. 1.2. This family of copulas where the parameters control the strength of dependence through the generator provides a large spectrum of dependence structures.

Table 1.2: Generators, inverse generators, and parameter domains for some of the most popular Archimedean copulas.

Name	Generator $\varphi_\theta(t)$	Generator Inverse $\varphi_\theta^{-1}(t)$	Parameter θ
AMH ¹	$\log\left(\frac{1-\theta(1-t)}{t}\right)$	$\frac{1-\theta}{\exp(t)-\theta}$	$\theta \in [-1, 1)$
Clayton	$\frac{1}{\theta}(t^{-\theta} - 1)$	$(1 + \theta t)^{-1/\theta}$	$\theta \in [-1, \infty) \setminus \{0\}$
Frank	$-\log\left(\frac{\exp(-\theta t)-1}{\exp(-\theta)-1}\right)$	$-\frac{1}{\theta} \log(1 + \exp(-t)(\exp(-\theta) - 1))$	$\theta \in \mathbb{R} \setminus \{0\}$
Gumbel	$(-\log(t))^\theta$	$\exp(-t^{1/\theta})$	$\theta \in [1, \infty)$
Independence	$-\log(t)$	$\exp(-t)$	
Joe	$-\log(1 - (1-t)^\theta)$	$1 - (1 - \exp(-t))^{1/\theta}$	$\theta \in [1, \infty)$

¹ Ali-Mikhail-Haq

A special copula is the independence copula

$$C(u_1, \dots, u_d) = \prod_{i=1}^d u_i, \quad (1.27)$$

which implies independence between all variables. It can be defined either as a Gaussian copula with identity matrix as the correlation matrix R in Eq. 1.25 or alternatively as an Archimedean copula with generator $\varphi(t) = -\log(t)$ and inverse generator $\varphi(t)^{-1} = \exp(-t)$.

Figure 1.8 shows scatter plots for samples drawn from different bivariate copulas. The parameters of each copula have been chosen so that $\tau = 0.5$. Note the visible difference in the dependence structure for the same strength of dependence. This shows the flexibility of copulas when it comes to modelling various types of dependencies by applying different families. The samples obtained from the copulas transformed to standard normal marginal distributions are presented in Figure 1.9. The dependence in the underlying copula is still evident after the transformation to the marginals.

1.5.2 Pair Copula Construction

Constructing multivariate distribution functions is a difficult task. Various approaches have been introduced to build multivariate copulas from combinations of bivariate copulas because the application of simple families such as Archimedean copulas lacks flexibility to model complex dependency structures [61]. This is known as pair copula construction and has the

1 Introduction

additional benefit that the available theory on bivariate copulas is significantly more extensive in comparison to multivariate copulas [65].

Let $f_{1:d}(x_1, \dots, x_d)$ be the joint density function of d -dimensional vector of random variables $\mathbf{X} = (X_1, \dots, X_d)$. Here the subscript $f_{1:d}$ is a shorthand for $f_{1,2,\dots,d}$. This density can be expressed by a combination of conditional densities functions as

$$f_{1:d}(x_1, \dots, x_d) = f_1(x_1) \cdot f_{2|1}(x_2|x_1) \cdot f_{3|2,1}(x_3|x_2, x_1) \cdot \dots \cdot f_{d|1:(d-1)}(x_d|x_1, \dots, x_{d-1}). \quad (1.28)$$

By applying Sklar's theorem the conditional densities can be deconstructed into bivariate copula densities and densities of univariate marginals. Differentiating Eq. 1.20 with respect to a distribution with joint density $f(x_1, \dots, x_d)$, marginals f_j and marginal CDFs F_j , $j \in \{1, \dots, d\}$ leads to

$$f_{1:d}(x_1, \dots, x_d) = c_{1:d}(F_1(x_1), \dots, F_d(x_d)) \cdot f_1(x_1) \cdot \dots \cdot f_d(x_d), \quad (1.29)$$

where $c_{1:d}(\cdot)$ is the d -variate copula density. For the bivariate case this simplifies to

$$f_{1,2}(x_1, x_2) = f_1(x_1) \cdot f_{2|1}(x_2|x_1), \quad (1.30)$$

which will provide the basic building block for the multivariate density functions. It also yields

$$f_{2|1}(x_2|x_1) = \frac{f(x_1, x_2)}{f(x_1)} = c_{12}(F_1(x_1), F_2(x_2)) \cdot f_2(x_2) \quad (1.31)$$

as the conditional density of a random variable X_2 given X_1 which can be generalized to

$$f_{j|i}(x_j|x_i) = \frac{f(x_i, x_j)}{f(x_i)} = c_{ij}(F_i(x_i), F_j(x_j)) \cdot f_j(x_j). \quad (1.32)$$

As an example of how to decompose higher dimensional densities, consider a three-dimensional density function expressed through Eq. 1.28 as

$$f_{1,2,3}(x_1, x_2, x_3) = f_1(x_1) \cdot f_{2|1}(x_2|x_1) \cdot f_{3|1,2}(x_3|x_1, x_2). \quad (1.33)$$

Through Eq. 1.31 the second term of the three-dimensional density function can be expressed as a pair-copula and a marginal distribution. The third term can be further split by selecting either of the conditioning variables x_1 or x_2 where choosing x_2 leads to

$$\begin{aligned} f_{1,2,3}(x_1, x_2, x_3) &= f_1(x_1) \cdot \\ & c_{12}(F_1(x_1), F_2(x_2)) \cdot f_2(x_2) \cdot \\ & c_{13|2}(F_{1|2}(x_1|x_2), F_{3|2}(x_3|x_2)) \cdot f_{3|2}(x_3|x_2). \end{aligned} \quad (1.34)$$

The final term can again be deconstructed into another pair-copula and marginal distribution using Eq. 1.31 leading to the final and complete decomposition of the three-dimensional density

as

$$\begin{aligned}
 f_{1,2,3}(x_1, x_2, x_3) &= f_1(x_1) \cdot \\
 &\quad c_{12}(F_1(x_1), F_2(x_2)) \cdot f_2(x_2) \\
 &\quad c_{13|2}(F_{1|2}(x_1|x_2), F_{3|2}(x_3|x_2)) \cdot c_{23}(F_2(x_2), F_3(x_3)) \cdot f_3(x_3).
 \end{aligned} \tag{1.35}$$

The conditional densities that are arguments of the copulas can be obtained from the copula CDFs. This recursive methodology can be applied to decompose any multivariate density into combinations of bivariate copulas and univariate marginal. However, not all multivariate copulas can be modelled with this pair copula construction method. To allow for inference and model selection, the assumption is usually made that the pair-copulas only depend on the conditioning variables through the conditional distribution functions as their arguments. Hobæk Haff et al. [66] present which pair-copula decompositions satisfy this condition.

Vine copulas

During the decomposition of a multivariate density $f_{1:d}(x_1, \dots, x_d)$ into bivariate copulas and marginal densities many choices on conditioning variables have to be made, resulting in numerous possible pair-copula constructions. Bedford and Cooke [67] pioneered the vine copula approach as a graph-theory based tool to organize these decompositions.

Since vine copulas are defined as a set of *trees* the basic notion of a tree is quickly introduced. Let the tuple $G = (N, E)$ be an undirected graph composed of nodes N and edges E which are 2-element subsets of N as

$$E \subseteq [N]^2. \tag{1.36}$$

A *path* is a sequence of nodes n_1, \dots, n_k where for each node n_i there is an edge connecting it to the next and all nodes in the path are unique, i.e. not visited repeatedly. A special type of path where $n_1 = n_k$ is called a *cycle*. A *tree* is then defined as an acyclic undirected graph where all pairs of two nodes are connected by exactly one distinct path [68].

A *regular vine* (R-vine) of d elements is defined as a sequence of $d - 1$ trees $\mathcal{V} = (T_1, \dots, T_{d-1})$ with the following conditions: (1) T_1 is a tree of nodes $N_1 = 1, \dots, d$ and edges E_1 . (2) Each subsequent tree T_j for $j \geq 2$ is a tree with nodes $N_j = E_{j-1}$ and edges E_j . Meaning the nodes of tree T_j are the edges of T_{j-1} . And (3), each tree T_j for $j \in \{1, \dots, d - 1\}$ fulfils the *proximity condition* stating if $\{a, b\} \in E_j$ then $|a \cap b| = 1$. This condition ensures that any nodes connected in tree T_j for $j \geq 2$ share a common ancestor in tree T_{j-1} . Figure 1.10 shows a five-dimensional regular vine. Note how an edge in T_2 between 1, 2 and 3, 4 is forbidden by the proximity condition.

Starting from d nodes and $d - 1$ pair-copulas in T_1 followed by $d - 1$ nodes and $d - 1$ edges in T_2 until only two nodes and one edge exist in the final tree T_{d-1} one can see that the number of pair-copulas in a d -dimensional vine is

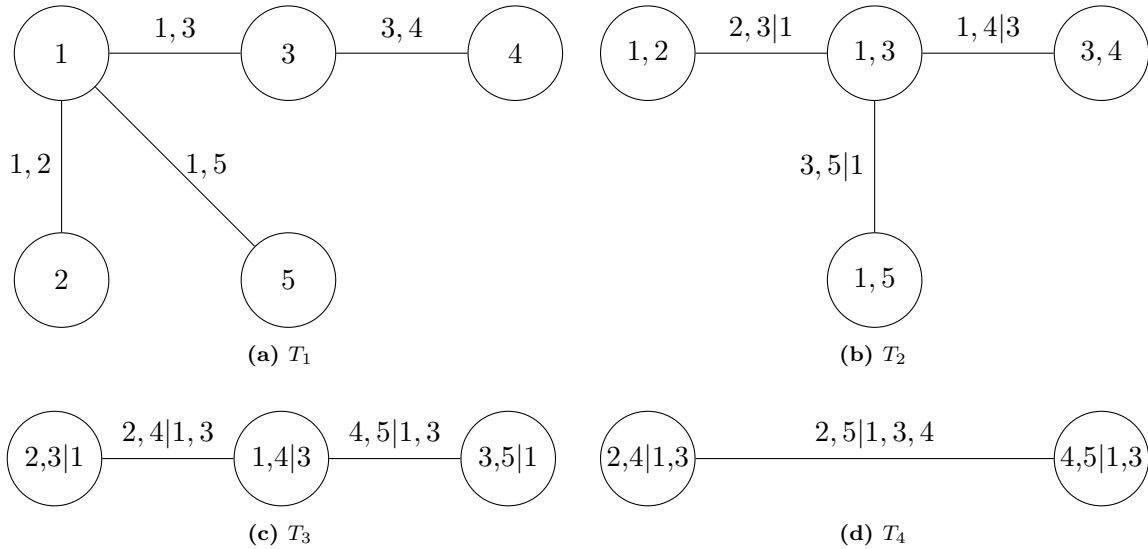


Figure 1.10: Graphical representation of the four trees constructing a five-dimensional regular vine.

$$(d-1) + (d-2) + \cdots + 2 + 1 = \frac{(d-1)d}{2} = \binom{d}{2}. \quad (1.37)$$

As such, a 5-dimensional vine already requires the evaluation of 10 bivariate copulas of which most involve conditional distribution functions. Assuming conditional independence for some pair-copulas allows to significantly reduce the complexity for higher dimensional vines. Going back to the example in Eq. 1.35, assuming that x_2 and x_3 are independent conditional on x_1 simplifies the final decomposition to

$$\begin{aligned} f_{1,2,3}(x_1, x_2, x_3) &= f_1(x_1) \cdot \\ & c_{12}(F_1(x_1), F_2(x_2)) \cdot f_2(x_2) \\ & c_{23}(F_2(x_2), F_3(x_3)) \cdot f_3(x_3), \end{aligned} \quad (1.38)$$

because the conditional independence implies $c_{3|12}(F_{1|2}(x_1|x_2), F_{3|2}(x_3|x_2)) = 1$.

Regular vines are very general. In fact, Morales-Nápoles [69] have shown that there are $d!/2 \cdot 2^{\binom{d-2}{2}}$ possible R-vines in d dimensions. A smaller subset of R-vines with only $d!/2$ possible structures known as C-vines (canonical vines) and D-vines provide specific ways to perform the density decomposition. In a C-vine there is a node $n \in N_i$ in each tree T_i for which $|\{e \in E_i | n \in e\}| = d-1$, meaning there is one central node that is connected to all other nodes. In a D-vine for each node $n \in N_i$ the number of edges is $|\{e \in E_i | n \in e\}| \leq 2$. The resulting structures for C- and D-vines in five dimensions are presented in Figure 1.11.

There is a straightforward sampling algorithm for sampling d dependent uniform variables in $[0, 1]^d$ from a vine copula. Start by sampling d independent uniform random numbers

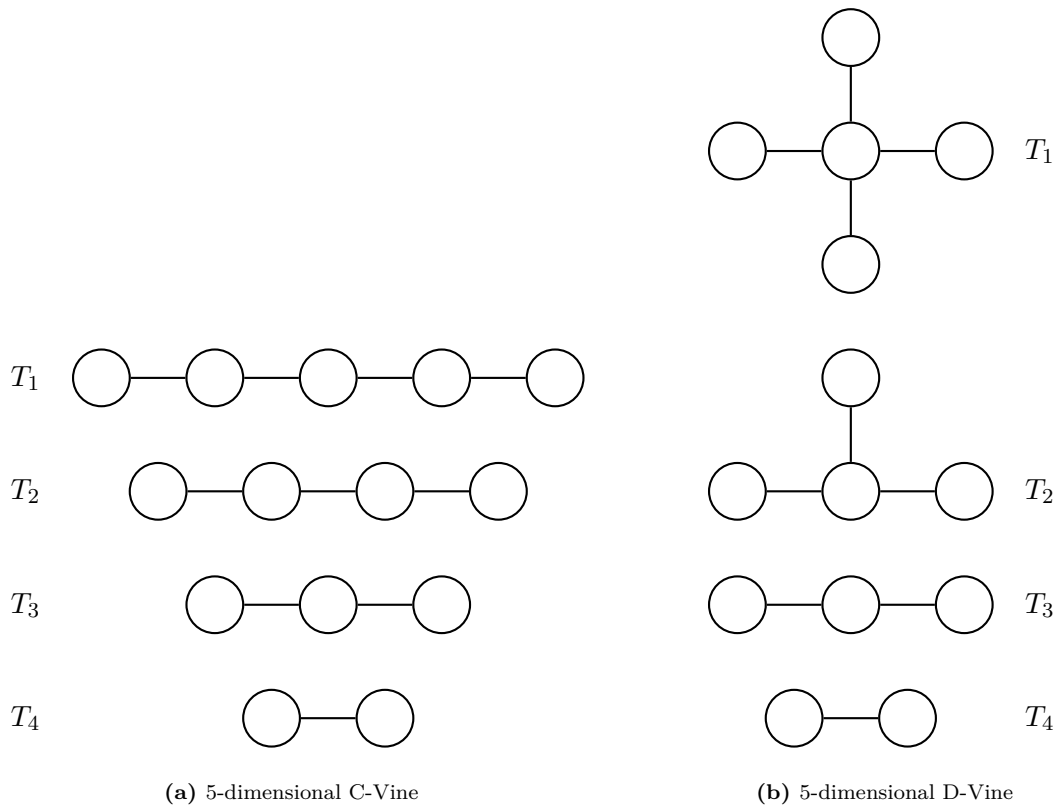


Figure 1.11: Graphical representations of C- and D Vines.

$u_i \sim U(0, 1)$ for $i \in \{1, \dots, d\}$ and compute

$$\begin{aligned}
 x_1 &= u_1, \\
 x_2 &= F_{2|1}^{-1}(u_2|x_1), \\
 x_3 &= F_{3|1,2}^{-1}(u_3|x_1, x_2), \\
 &\vdots \\
 x_n &= F_{n|1,\dots,d-1}^{-1}(u_d|x_1, \dots, x_{d-1}).
 \end{aligned}
 \tag{1.39}$$

Implementing this algorithm involves evaluation of the conditional distributions of the form $F(x|\mathbf{x})$ where \mathbf{x} is a vector of random variables. Joe [70] has shown that

$$F(x|\mathbf{x}) = \frac{\partial C_{x,x_j|\mathbf{x}_{-j}}(F(x|\mathbf{x}_{-j}), F(x_j|\mathbf{x}_{-j}))}{\partial F(x_j|\mathbf{x}_{-j})},
 \tag{1.40}$$

where \mathbf{x}_{-j} refers to the vector \mathbf{x} without element j . In the bivariate case and for uniform random variables u_1 and u_2 , i.e. $f(u_1) = f(u_2) = 1$, $F(u_1) = u_1$, and $F(u_2) = u_2$, this simplifies to

$$h(u_1, u_2; \Theta) = F(u_1|u_2) = \frac{\partial C_{u_1,u_2}(u_1, u_2, \Theta)}{\partial u_2}.
 \tag{1.41}$$

Here, the second parameter u_2 always refers to the conditioning variable and Θ is the set of parameters for the copula C_{u_1, u_2} [71]. The inverse of this function $h^{-1}(u_1, u_2; \Theta)$ with respect to u_1 is also required.

For a realization U_1, \dots, U_d of the vine copula, the samples of the copula distribution are obtained through the quantile functions of the marginals as

$$X_1 = F_1^{-1}(U_1), \dots, X_d = F_d^{-1}(U_d). \quad (1.42)$$

For a complete discussion on sampling copulas and the definitions of the h and h^{-1} functions of several bivariate copulas, see [65].

1.6 Resilience

This section provides a quick introduction to the concept of resilience and briefly presents the resilience decision-making methodology of Salomon et al. [72] which is extended and merged with the survival signature in Chapter 6.

Resilience, from the Latin “resilire” meaning “to bounce back” is a concept in which not only the robustness and reliability of a system is analysed but its ability to recover is also taken into account. A failure in a system results in an abrupt loss in system performance followed by a gradual recovery of the system through implemented measures [73]. The loss of performance after a disruptive event at time t_0 and the time it takes to recover, reaching normal levels at t_1 can be visualized in the *resilience triangle* as introduced by Bruneau et al. [73], see Figure 1.12.

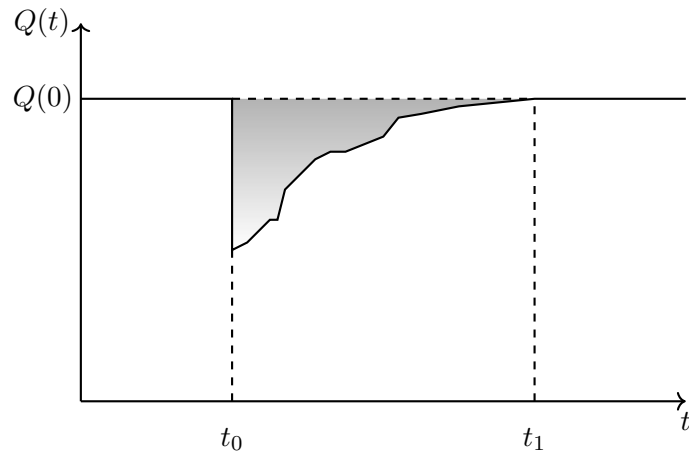


Figure 1.12: Resilience triangle. Adapted from [73].

1.6.1 Resilience metrics

The application of resilience relies on the availability of comprehensive quantitative resilience metrics to measure it. For a discussion of several available metrics refer to [74, 75].

In this dissertation the resilience metric by Ouyang et al. [76] is applied. It is a measure based on the ratio of the time-dependent system performance $Q(t)$ and target system performance $\mathcal{T}Q(t)$ integrated over the same time domain $[0, T]$. The metric is defined as the expected value $R = E[Y]$, where

$$Y = \frac{\int_0^T Q(t)dt}{\int_0^T \mathcal{T}Q(t)dt}. \quad (1.43)$$

A value of $R = 1$ indicates a system operating at target performance over the whole time period while for $R = 0$ the system is not working at all. The system performance $Q(t)$ is a stochastic process. Although the target performance $\mathcal{T}Q(t)$ can also be treated as a stochastic process, for convenience, it is assumed to be a non-random constant number in this work.

1.6.2 Adapted systemic risk measure

In Salomon et al. [72] the resilience metric is combined with an adapted systemic risk measure for complex systems. The original risk measure proposed by Feinstein et al. [77] is based on an input-output model and an acceptance criterion that represents safety standards and regulations. If (Ω, \mathcal{F}, P) is a probability space, $l \in \mathcal{N}$ is the number of entities in the system, and $k \in \mathbb{R}^l$ is a vector of controls. The random variable $Y_k(\omega)$ is the outcome of the system for each $\omega \in \Omega$ and control vector k . The controls k also called “endowments” are capital allocations to the entities in the system whose goal is to improve functionality. The input-output model $Y = (Y_k)_{k \in \mathbb{R}^l}$ is a non-decreasing random field in a vector space \mathcal{X} . For all $k_i \leq m_i$ and $i \in \{1, \dots, l\}$ monotonicity requires that $Y_k \leq Y_m$. This means, that allocating more resources will always improve the system. The set $\mathcal{A} \subseteq \mathcal{X}$ describes the acceptance criterion where the requirements of decision-makers are met.

Finally, the systemic risk measure is assembled from the input-output model and the acceptance criterion as the set of endowments resulting in model outputs satisfying the acceptance criterion as

$$R(Y; k) = \left\{ m \in \mathbb{R}^l \mid Y_{k+m} \in \mathcal{A} \right\}. \quad (1.44)$$

This systemic risk measure is now applied and adapted to systems with multiple component types and several endowment properties. Consider a system of l components divided into b different types each with n performance-enhancing properties. Any component a_i for $i \in \{1, \dots, l\}$ with properties $(\eta_{i1}, \eta_{i2}, \dots, \eta_{in})$ and component type $z_i \in \{1, 2, \dots, b\}$ can be described by the row vector

$$(a_i; z_i) = (\eta_{i1}, \eta_{i2}, \dots, \eta_{in}; z_i) \in \mathbb{R}^{1 \times n} \times \mathbb{N}. \quad (1.45)$$

Based on this vector notation, the system can be described by a matrix $A \in \mathbb{R}^{l \times n}$ and column

1 Introduction

vector $z \in \mathbb{N}^l$ containing the component types as

$$(A; z) = \begin{pmatrix} \eta_{11} & \eta_{12} & \cdots & \eta_{1n}; & z_1 \\ \eta_{21} & \eta_{22} & \cdots & \eta_{2n}; & z_1 \\ \vdots & \vdots & & \vdots & \vdots \\ \eta_{l1} & \eta_{l2} & \cdots & \eta_{ln}; & z_1 \end{pmatrix}. \quad (1.46)$$

Different realizations of A and z are fed into the input-output model as $Y_{(A;z)}$. Here it is assumed that the component types z are fixed and only the enhancing properties in A are varied. The corresponding risk measure is defined using the acceptance set

$$\mathcal{A} = \{X \in \mathcal{X} | E[X] \leq \alpha\}, \quad (1.47)$$

with $\alpha \in [0, 1]$ as

$$R(Y; K) = R(Y; (K; z)) = A \in \mathbb{R}^{l \times n} | Y_{(K+A;z)} \in \mathcal{A}. \quad (1.48)$$

As a result the resilience of the system described by $(K + A; z)$ is greater or equal to α . For simplicity, $K = 0$ is assumed and $R(Y; 0)$ is written as $R(Y)$. It may often be required to impose restrictions on A as well such as using the same property configurations for all components of the same type.

1.6.3 Grid search algorithm

A grid search algorithm [77] is used to calculate the set-valued systemic risk measure. To do so, the endowments that are under investigation are divided into a uniform grid. The search starts at the origin of this box which is assumed to be outside $R(Y)$, i.e. is not an acceptable configuration. From there, the acceptance criterion is evaluated for increasing grid points on the line identified by the direction $(1, 1, \dots, 1)$ until the first endowment configuration satisfying the acceptance criterion is found. The system will typically be analysed through stochastic simulation methods such as Monte Carlo simulation. Every time an acceptable point is found during the grid search all points with better endowment configurations can also be accepted. At the same time, all endowments below the first accepted points can be rejected. This results directly from the monotonicity condition.

Starting from the first accepted point a sub-box is defined using the points diagonally below (rejected) and above (accepted) of the accepted point. The other corners of this sub-box are then checked for acceptance. Based on the results, new sub-boxes will be defined from one accepted and one rejected point and subsequently evaluated. The algorithm terminates once all points have either been accepted or rejected defining a discrete grid-approximation of $R(Y)$. For illustrative purposes, Figure 1.13 shows what the results for an arbitrary system with two endowment properties $\eta_1 \in \{1, 2, \dots, 8\}$ and $\eta_2 \in \{1, 2, \dots, 8\}$ might look like.

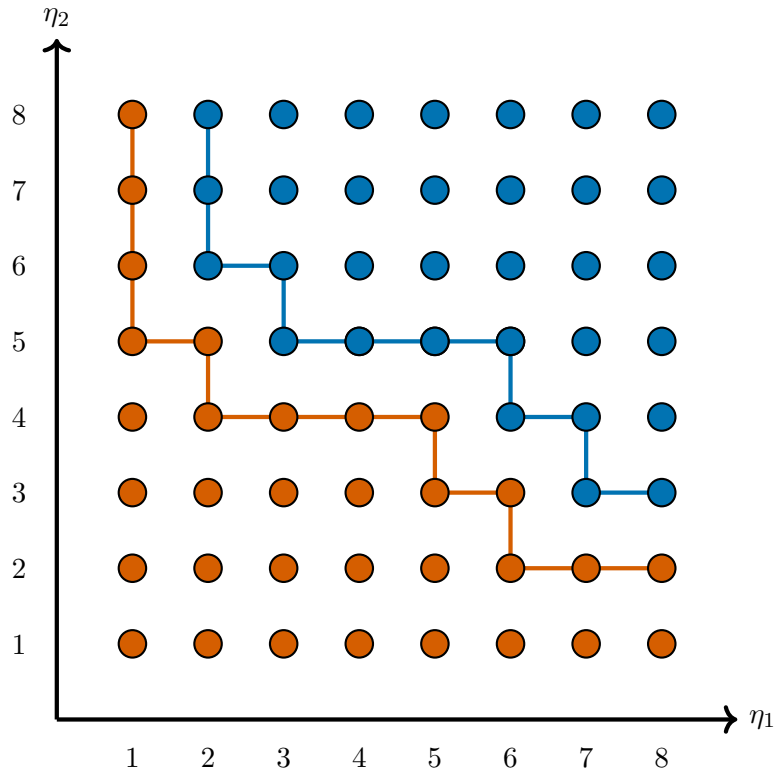


Figure 1.13: Results of the grid search algorithm for an arbitrary system with two endowment properties.

1.6.4 Resilience decision-making

A crucial part of the decision-making process is taking into account monetary restrictions. An increase in endowment properties can mean an exponential increase in the associated cost [78]. This can for example mean using higher quality materials or hiring extra maintenance personal. An increase in the reliability of components in a complex network can be associated with exponentially increasing costs.

For decision-making, a quantity of interest are the cheapest yet acceptable combinations of endowments. To evaluate the cost one must provide a cost function which takes the endowment configuration as arguments and returns the sum of all associated costs. The cheapest acceptable configurations will always be located right on the border of accepted endowment configurations seen in Figure 1.13.

1.7 Software for Uncertainty Quantification

With the increasing complexity of modern engineering structures and systems comes the need for efficient software to ensure their reliability and safety. For this, adequate propagation of uncertainties is of paramount importance. New advancements in simulation methods and analyses for uncertainty propagation are constantly made. Academia as well as industry can

benefit greatly from freely available and generalized frameworks that provide methods to analyse arbitrary systems.

Over the years, many such toolboxes have been developed for a large array of programming languages. Some examples follow in now particular order. *Dakota* [79] is a state-of-the-art software for optimization and uncertainty quantification developed at the Sandia National Laboratories. *OpenTURNS* [80] is an industrial software for uncertainty quantification in simulation developed jointly by EDF R&D, Airbus Group, Phimeca Engineering and IMACS. The Cossan Working Group develops *OpenCossan* [81], a general purpose software to quantify, mitigate and manage uncertainty for risk, reliability and resilient analyses. *UQlab* a general purpose uncertainty quantification framework is developed at ETH Zürich [82], and *UQpy* [83] a general purpose toolbox for modelling uncertainty in physical and mathematical systems is developed by the Shields Uncertainty Research Group at John Hopkins University.

A new framework called *UncertaintyQuantification.jl* is presented in this dissertation. An overview of the different frameworks with programming languages used, licences and latest release is presented in Tab. 1.3.

Table 1.3: List of some available software libraries for uncertainty quantification.

Name	Language	Licence	Latest Release ²
Dakota	C++	LGPL	6.1.8 (18.06.2023)
OpenCossan	MATLAB	GPL	April 2020 (22.04.2020)
OpenTurns	C++/ Python	LGPL	1.21 (20.06.2023)
UQlab	MATLAB	BSD-3	v2.0 (01.02.2022)
UQpy	Python	MIT	v4.1.2 (04.08.2023)
UncertaintyQuantification.jl	Julia	MIT	0.7.2 (05.09.2023)

² As of 06.09.2023

The new framework *UncertaintyQuantification.jl* [84] is developed as a modern “spiritual successor” to OpenCossan [81] based on the experience gained during the development of the former toolbox. The base of this toolbox was built by the group of Prof. Schuëller at the Institute for Engineering Mechanics of the University of Innsbruck originally only designed to perform stochastic structural analysis and written in Fortran as COSSAN (COmputation Stochastic Structural ANalysis) [85]. The software was completely rewritten in MATLAB and generalized to include a much larger set of algorithms as OpenCOSSAN at the Institute for Risk and Uncertainty at University of Liverpool [81] first released in 2011. The Institute for Risk and Reliability of the Leibniz University Hannover, Germany joined the development in 2015 followed by the Shanghai Institute of Disaster Prevention and Relief, Tongji University, China in 2017. In 2019 the host organization switched from the University of Liverpool to Strathclyde University.

With a long-standing history such as this, it is only expected that a software has accrued a large amount of technical debt. For this reason and issues with MATLAB being an expensive commercial programming language instead the decision was made to build an entirely new framework as an upgrade and replacement using the modern Julia [86] programming language.

Julia is a new language, having been published in version 1.0 in 2018. It is a modern dynamic programming language that is incredibly fast while remaining simple to use due to careful language design. Although the programmer has access to a sophisticated type system, the user is never forced to declare types, and type annotations are not required for performance. Julia is designed for high-performance computing, and the numerical demand of algorithms can often be distributed on existing computing clusters with only a few lines of code.

At the heart of Julia lies its *multiple dispatch* system. In multiple dispatch implemented functions are selected based on the argument types. Types are annotated in the function definitions using the `::` syntax. As an example, consider a function designed to square its input. For a scalar number this can be written as

```
square(x::Number) = x^2
```

This function can now be called as, for example, `square(4)` and it will correctly return $4^2 = 16$. However, if instead of a scalar number one would pass a vector to the function Julia would throw an error as no `square` function is defined for vector arguments. To define a function that squares the elements of a vector element-wise write

```
square(x::Vector) = x.^2
```

Now, executing `square([1,2,3,4])` would return `[1,4,9,16]` as expected. Julia is able to select the appropriate function for the argument through the multiple dispatch system. In the case of *UncertaintyQuantification.jl* multiple dispatch makes it possible to, for example, define sampling functions for various standard and quasi Monte Carlo simulation techniques. By passing the appropriate inputs to high-level algorithms these can be passed down to where the sampling is actually performed and Julia will at runtime select the appropriate sampling algorithm. As a result, complex algorithms can be defined for different types of input arguments with minimal code duplication.

1.8 Aims and objectives

The main objective of this dissertation is to improve upon the survival signature methodology in order to enable modelling of complex dependent behaviours and allow for the analysis of larger systems than previously possible. Based on the separation of system structure from probabilistic information, a key feature of the survival signature shown in Eq. 1.17, the proposed methods focus on different parts of the survival function.

On the probabilistic side, the focus lies on new methods for the modelling of complex dependent failure behaviours. Events such as the 2003 blackout in Italy [42] have highlighted the need for inclusion of interdependencies between networks in the analysis which can lead to catastrophic failures if disregarded. Additionally, competing failure modes such as common causes of failure or cascading failures must be included. While applications of the survival signature often assume full independence of component failures some studies have, for example, developed techniques based on Monte Carlo simulation to model common cause failures and cascading failures [32]. The goal in this thesis is to provide a generalized modelling framework for dependent failure based on copulas. Copulas have long been a proven method for complex dependency modelling in many fields such as finance [87, 88], insurance [89, 90] or enterprise risk management [91] and are becoming increasingly popular for engineering applications [92]. Multivariate copulas possess great modelling flexibility through separation of the dependency structure from the marginal distributions. Various bivariate copula families exist, providing different dependency structures for the modelling of entirely different causes of dependent failures. Through pair copula construction these bivariate copulas can be assembled to a single dependency structure for the whole system. Which in turn allows for efficient reliability analysis.

In regard to the structural part of the survival function, i.e. the survival signature, the methods presented in this thesis are focused on approximation techniques for large systems. Computation of the survival signature through Eq. 1.16 involves evaluation of every possible system state for each survival signature entry. With increasing size and complexity of the networks under analysis this analytical computation becomes impossible. Various studies trying to work around this issue have been published. Reed [38] and Reed et al. [39] proposed algorithms based on binary decision diagrams for exact computation of the survival signature. Another approach uses reliability block diagrams and the extended universal generating function [40]. However, these methods are both limited by memory requirements and unsuitable for large systems. This dissertation presents two alternatives to the exact computation of the survival signature to significantly reduce the required numerical effort. The first method applies percolation theory to exclude negligible parts of the survival signature and then uses Monte Carlo simulation to approximate the remaining entries based on a subset of all possible state vectors. The second technique builds a surrogate model of the survival signature instead. An interval predictor model is constructed from a few strategically selected data points which is then used to predict intervals for the remaining entries.

Branching out from the network reliability analysis to the study of resilience, an extension of a resilience decision-making technique [72] is presented where the application of survival signature allows for the efficient resilience analysis of complex substructured systems. The separation of the system structure from component failures presents an opportunity to simplify the analysis of modern systems where the components are also modelled as systems. The survival signature of these subsystems is computed ahead of the resilience analysis during which the component failure rates can be derived from the survival function.

In addition to the developments concerning the survival signature, this dissertation also presents a new framework for uncertainty quantification. The goal of this framework is to provide a complete and generalized toolbox where newly developed methods in all fields of uncertainty quantification can be efficiently implemented and shared with the wider community. The new framework is developed as a package in the Julia programming language and is freely available to everyone.

1.9 Original contributions

This dissertation consists of four journal papers and two conference contributions. At the core of the original contributions are several enhancements to the survival signature ecosystem.

The first journal publication presents a new method for the reliability analysis of complex dependent networks [93]. Various competing failure modes are modelled through different parametric copula families. Pair copula construction is used to build high-dimensional copulas from combinations of bivariate copulas allowing for high modelling flexibility. Epistemic uncertainty is included in the component failure times through p-boxes and imprecise copulas. A new Monte Carlo simulation method is also developed to obtain the upper and lower bounds of the network reliability. The proposed method is applied to a numerical example based on the IEEE reliability test system.

An approximation method for the survival signature is presented in the second journal paper [94]. In a first step, percolation theory is used to identify areas of the survival signature where the probability of a functioning system is close to zero and therefore negligible. Next, the remaining part is approximated by Monte Carlo simulation. Evaluating only a subset of all possible state vectors significantly reduces the numerical demand and allows for the analysis of larger systems. The method is applied to several toy examples in order to quantify the errors resulting from the percolation and Monte Carlo simulation. The new method is then applied to a real world example based on the S- and U-Bahn metro system of Berlin. The approximation of the survival signature for this large network, where the analytical solution is impossible to obtain, highlights the significance of the presented technique.

A method expanding on the Monte Carlo approximation technique is presented next. The new method is established in the first conference paper [95] and then further refined in the third journal article [96]. Instead of approximating each survival signature entry not excluded by application of percolation theory, only a few selected data points are used to build a surrogate model. This metamodel is a NRBF network constructed with additional constraints which ensure monotonicity of the resulting survival signature. The surrogate model is then extended to an IPM to account for the uncertainty of the Monte Carlo approximation of the data points. This imprecise model yields upper and lower bounds on the survival signature and therefore the network reliability. The method is validated against a toy example and then again applied to the Berlin metro example. An extra component type is used in the journal article to increase

the dimensionality of the survival signature. The presented method is able to accurately predict bounds on the survival signature and the reliability at a fraction of the computational demand of the full Monte Carlo simulation method and the analytical solution.

These four papers present the core contribution of this dissertation. In the next journal paper [97], the resilience decision-making method presented by Salomon et al. [72] is extended and merged with the survival signature to allow for efficient reliability analysis of large substructured systems. The survival signatures of the subsystems can be computed ahead of the resilience analysis. During the analysis, the component failures in the subsystem can then be efficiently propagated through the survival signature providing the failure rates of the top-level components through the derivative of the hazard functions. The method is applied to three numerical examples of increasing complexity. The largest example is again based on the Berlin metro system where the individual stations are now modelled as subsystems. Through the presented approach, the system is effectively reduced from 2776 components to the 306 top-level components (subsystems) for the resilience analysis. However, the influence of all individual components is still included in the analysis through the survival signatures. As a co-author, the author's contributions to this article focused mostly on conception, providing expertise on the survival signature, assisting with the extension of the grid search to multidimensional spaces, and implementing the proposed approach in software.

The second conference paper [98] and final contribution of this dissertation presents a new framework for uncertainty quantification called *UncertaintyQuantification.jl* [84]. The paper outlines the basic functionality of the framework and shows its application to a few numerical examples. In its current state the implemented algorithms include methods for reliability analysis using various simulation schemes, sensitivity analysis and metamodelling.

Two additional Julia packages have been developed as part of this dissertation. The survival signature approximation methods are implemented in the *SurvivalSignature.jl* [99] package while the resilience decision-making is provided in the *ResilienceDecisionMaking.jl* [100] package. All three modules are officially registered in the Julia *General* registry and publicly available for everyone on GitHub under the MIT licence.

1.10 Structure of the thesis

The remainder of the dissertation is structured as follows. Chapter 2 presents the method for the reliability analysis of complex interdependent networks where the dependencies are modelled through copulas. Because this approach is built on the survival signature, its inherent separation of network topology from component failures allows for efficient consideration of different dependent failure scenarios once without reevaluation of the system structure.

Instead of focusing on the probabilistic modelling of the component failures, Chapter 3 is focused on approximation of the survival signature for large systems. Monte Carlo simulation and percolation theory are applied to estimate the signature for systems where analytical

computation is impossible because of the prohibitive numerical demand.

Chapters 4 and 5 improve on the earlier method. Where previously all entries of the survival signature not excluded by percolation had to be approximated now only a few values are strategically selected and used to efficiently build a surrogate model of the survival signature, further reducing the numerical effort required.

In addition to the core developments of this thesis, Chapter 6 presents the extension of a resilience decision-making method. The existing method is merged with the survival signature to allow for efficient multidimensional resilience analysis of substructured systems.

A new framework for uncertainty quantification written in the Julia programming language is presented in Chapter 7.

Chapter 8 ends the thesis with some concluding remarks and outlook into possible areas for future research based on the presented methods.

2 | Reliability Analysis of Networks Interconnected With Copulas

Reliability Analysis of Networks Interconnected With Copulas

Jasper Behrendorf^a, Matteo Broggi^a, Michael Beer^{a,b,c}

^aInstitute for Risk and Reliability, Leibniz Universität Hannover, Hannover, Germany

^bInstitute for Risk and Uncertainty, University of Liverpool, Liverpool, United Kingdom

^cInternational Joint Research Center for Engineering Reliability and Stochastic Mechanics (ERSM), Tongji University, Shanghai, China

Published in *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering* on September 25, 2019

Abstract

With the increasing size and complexity of modern infrastructure networks rises the challenge of devising efficient and accurate methods for the reliability analysis of these systems. Special care must be taken in order to include any possible interdependencies between networks and to properly treat all uncertainties. This work presents a new approach for the reliability analysis of complex interconnected networks through Monte Carlo Simulation and survival signature. Application of the survival signature is key in overcoming limitations imposed by classical analysis techniques and facilitating the inclusion of competing failure modes. The (inter)dependencies are modelled using vine copulas while the uncertainties are handled by applying probability-boxes and imprecise copulas. The proposed method is tested on a complex scenario based on the IEEE reliability test system, proving its effectiveness and highlighting the ability to model complicated scenarios subject to a variety of dependent failure mechanisms.

Keywords: Survival signature, interdependencies, vine copulas, imprecise probabilities, Monte Carlo simulation.

2.1 Introduction

Reliability analysis of complex networks is an important task in the field of risk analysis. This importance is a result of the ever increasing size and complexity of modern critical infrastructure. At the same time, society is becoming increasingly reliant on the availability of these critical infrastructures such as water supply networks, electrical distribution networks or the internet. A breakdown of any of these systems can have a drastic impact on people's lives, as evident from the aftermath of recent natural disasters [101]. As a result, efficient and accurate methods for the reliability of these complex systems are required. However, history has shown that it is not sufficient to analyse these networks as individual units because the systems are often subject to complex interdependencies between one another. That is, failure in one network can potentially cascade into another network [42, 102]. For example, failures in a power grid due to natural disasters will drastically effect the communication network which in turn will inhibit the coordination of emergency personnel [103]. Therefore, it is of paramount importance to include and accurately model these interdependencies when analysing the reliability of networks.

Behrendorf et al. [104] presented a novel approach to the numerical reliability analysis of interdependent networks based on Monte Carlo simulation and survival signature. The

survival signature has the capability to fully separate the structure of a network from its probabilistic characteristics, allowing for efficient simulation while modelling dependencies in a probabilistic way [27]. Due to these characteristics it has constantly increased in popularity since its development, with new simulations techniques based on the signature being constantly developed (see for example [37]). In the previous the modelling of interdependencies between networks was limited to simple deterministic unidirectional causal links where failure of one component would result in the immediate failure of all dependent components. However, this approach lacks flexibility and does not allow to accurately capture the complex interdependencies between real world networks. As a result, a new methodology to model these interdependencies is required. Copulas have been successfully used to model dependence in enterprise risk management, finance, insurance, and environmental studies [62, 87, 105, 106]. Modelling dependencies with copulas is especially powerful as multivariate copulas allow to separate modelling of the marginal distributions from modelling the dependence structure [61]. Though the popularity of copulas for engineering applications has increased in the recent years [92, 107], literature is still scarce.

This work extends the previously developed method to allow for complex dependencies between nodes and networks as well as competing failure modes using multivariate copulas. This work is focused on using appropriate copulas to represent realistic dependency structures between different networks. The goal is to find a single dependency structure containing the complete dependency information. For this reason, different types of multivariate copulas such as hierarchical Archimedean copulas and vine copulas are investigated. The copula models are usually inferred from data or expert knowledge, both of which are subject to two types of uncertainty, namely aleatory and epistemic uncertainty. Aleatory uncertainty represents the natural randomness in process while epistemic uncertainty results from vagueness or lack of information [108]. Dealing with these uncertainties by imprecise reliability analysis results in bounds on the obtained survival function.

The remainder of this paper is outlined as follows. First, the basic notations and required definitions of copulas including measures of dependence is presented, followed by a discussion of copula construction methods. Then, the approach to modelling dependencies is presented. Next, the numerical method used to compute the network reliability is introduced. After discussion methods to handle uncertainties in the analysis, the proposed method is applied to a complex numerical example. Finally, the paper closes with some concluding remarks and an outlook into future works.

2.2 Copulas

This chapter introduces the basic theory on copulas as well as how they can be used to model dependencies in high dimensions. An overview of different parametric copula families is given. Additionally, measurements of dependence are introduced. For a comprehensive discussion of

copulas, see for example [60] or [61].

Copulas (from the Latin for ‘bond’ or ‘tie’) are functions that couple multivariate distribution functions to their one-dimensional marginal distributions functions and as such allow to separate modelling of the dependence structure from modelling the univariate marginals [60]. The foundation of the theory of copulas lies in what is known as Sklar’s theorem [59]. It states, that any multivariate distribution H can always be separated into its marginal distributions F_i and a copula function C . The theorem is valid in all dimensions $d \geq 2$.

Theorem 2.2.1 (Sklar’s theorem). *Let H be an d -dimensional distribution function with margins F_1, \dots, F_d . There exists an d -dimensional copula C such that for all \mathbf{x} in \mathbb{R}^d*

$$H(\mathbf{x}) = C(F_1(x_1), \dots, F_d(x_d)). \quad (2.1)$$

If the marginals F_1, \dots, F_d are continuous, then C is unique; otherwise, C is unique on $\text{Range}(F_1) \times \dots \times \text{Range}(F_d)$. Conversely, if C is a d -copula and F_1, \dots, F_d are distribution functions, then the function H defined by Eq. 2.1 is an d -dimensional distribution function with margins F_1, \dots, F_d .

Probabilistically, if C is a joint cumulative distribution function of a d -dimensional random vector on the unit cube $[0, 1]^d$ with uniform marginals, then $C : [0, 1]^d \rightarrow [0, 1]$ is a copula. It is noteworthy, that copulas are invariant under strictly increasing transformations, as stated by Theorem 2.2.2 [60].

Theorem 2.2.2. *For $d \geq 2$ let X_1, \dots, X_d be random variables with continuous distribution functions F_1, \dots, F_d , joint distribution function H and copula C . Let f_1, \dots, f_d be strictly increasing functions from \mathbb{R} to \mathbb{R} . Then $f_1(X_1), \dots, f_d(X_d)$ are random variables with continuous distribution functions and copula C . Thus, C is invariant under strictly increasing transformation of X_1, \dots, X_d .*

As such, any property of the joint distribution function that is invariant under strictly increasing transformation is in fact a property of the copula. As a result, this means, one can study dependence between random variables by studying the copula [62]. There exist multiple copula families with different dependence structures of which some of the most popular are presented in the following.

2.2.1 The Gaussian Copula

The d -dimensional Gaussian copula with positive definite correlation Matrix $\mathbf{R} \in [-1, 1]^{d \times d}$ is defined by

$$C_R(u_1, \dots, u_d) = \Phi_d(\Phi^{-1}(u_1), \dots, \Phi^{-1}(u_d)), \quad (2.2)$$

where $\Phi_d(\cdot; \mathbf{R})$ is the d -variate cumulative distribution of a $\mathbb{N}_d(0, \mathbf{R})$ random vector and Φ^{-1} denotes the inverse of the univariate standard Gaussian cumulative distribution function [61].

Table 2.1: Most popular Archimedean copulas with generators, generator inverses, and parameter domains.

Name	Generator $\varphi_\theta(t)$	Generator Inverse $\varphi_\theta^{-1}(t)$	Parameter θ
Ali-Mikhail-Haq	$\log\left(\frac{1-\theta(1-t)}{t}\right)$	$\frac{1-\theta}{\exp(t)-\theta}$	$\theta \in [-1, 1)$
Clayton	$\frac{1}{\theta}(t^{-\theta} - 1)$	$(1 + \theta t)^{-1/\theta}$	$\theta \in [-1, \infty) \setminus \{0\}$
Frank	$-\log\left(\frac{\exp(-\theta t)-1}{\exp(-\theta)-1}\right)$	$-\frac{1}{\theta} \log(1 + \exp(-t)(\exp(-\theta) - 1))$	$\theta \in \mathbb{R} \setminus \{0\}$
Gumbel	$(-\log(t))^\theta$	$\exp(-t^{1/\theta})$	$\theta \in [1, \infty)$
Independence	$-\log(t)$	$\exp(-t)$	
Joe	$-\log(1 - (1-t)^\theta)$	$1 - (1 - \exp(-t))^{1/\theta}$	$\theta \in [1, \infty)$

2.2.2 Archimedean Copulas

Archimedean copulas are an important class of copulas. Their popularity stems from a variety of reasons: they are easily constructed, the class holds a great number of different families and the copulas possess many excellent properties [60]. Additionally, the bivariate Archimedean copulas can be used in multivariate construction methods based on pairs of bivariate copulas [61]. A d -dimensional copula C_φ is classified as *Archimedean* if it admits to the representation

$$C_\varphi(u_1, \dots, u_d) := \varphi(\varphi^{-1}(u_1) + \dots + \varphi^{-1}(u_d)), \quad (2.3)$$

where the function $\varphi : [0, \infty] \rightarrow [0, 1]$ is called the *generator* of C_φ , φ^{-1} denotes its inverse and $u_1, \dots, u_d \in [0, 1]$ [65]. Table 2.1 shows some of the most popular one-parameter (governing the strength of dependence) Archimedean copula families with their generators, inverses and parameter domains.

2.2.3 Dependence

The study of dependence among random variables requires some form of dependence measurement. Typically, ‘correlation’ is used to describe different forms of dependence. However, in its technical meaning as the *linear correlation coefficient* ρ it is not ‘scale-invariant’ and as such does not remain unchanged under strictly increasing transformation [62]. Therefore, the more modern term ‘association’ is used instead of correlation. Two well known scale-invariant measures of association are the population versions of Kendall’s tau and Spearman’s rho. In this work, Kendall’s tau is applied in all cases.

Kendall’s tau is a measure of association based on *concordance*. A pair of random variables is concordant if ‘large’ values of one are associated with ‘large’ values of the other and the same holds for ‘small’ values. Formally, two observations (x_i, y_i) and (x_j, y_j) from a vector (X, Y) are *concordant* if $x_i < x_j$ and $y_i < y_j$, or *discordant* if $x_i > x_j$ and $y_i > y_j$. Alternatively, concordance can be expressed as $(x_i - x_j)(y_i - y_j) > 0$ and discordance as $(x_i - x_j)(y_i - y_j) < 0$.

Let (X, Y) denote a vector of continuous random variables and $\{(x_i, y_i), \dots, (x_n, y_n)\}$ a

sample of n observations from said vector. With c as the number of concordant pairs and d the number of discordant pairs among all possible $\binom{n}{2}$ pairs of observations (x_i, y_i) and (x_j, y_j) , Kendall's tau for the sample is defined as

$$t = \frac{c - d}{c + d} = (c - d) / \binom{n}{2}. \quad (2.4)$$

The value t may also be interpreted as the probability of concordance minus the probability of discordance for a random pair of observations (x_i, y_i) and (x_j, y_j) chosen from the sample. In turn, this can be applied to define the population version of Kendall's tau for random variables X and Y

$$\tau(X, Y) = P \left[(X - \tilde{X})(Y - \tilde{Y}) > 0 \right] - P \left[(X - \tilde{X})(Y - \tilde{Y}) < 0 \right], \quad (2.5)$$

where (\tilde{X}, \tilde{Y}) is an independent pair with the same distributions as (X, Y) [62]. Figure 2.1 shows four example scatter plots of samples generated from different bivariate copulas with the respective parameters chosen such that Kendall's tau equals 0.5, highlighting the individual dependence structure.

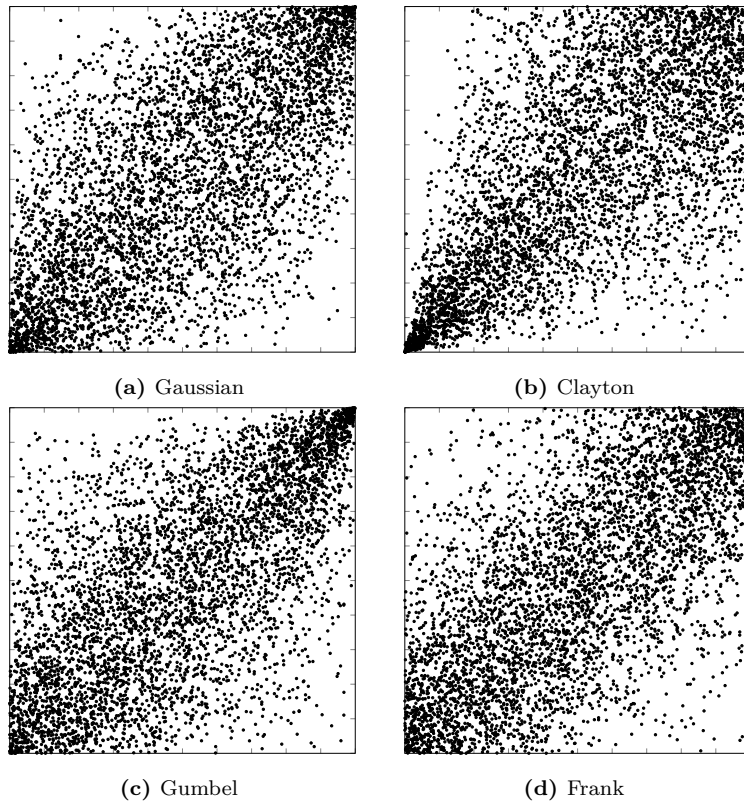


Figure 2.1: Samples drawn from different bivariate copulas where the parameters have been chosen so that Kendall's tau equals 0.5.

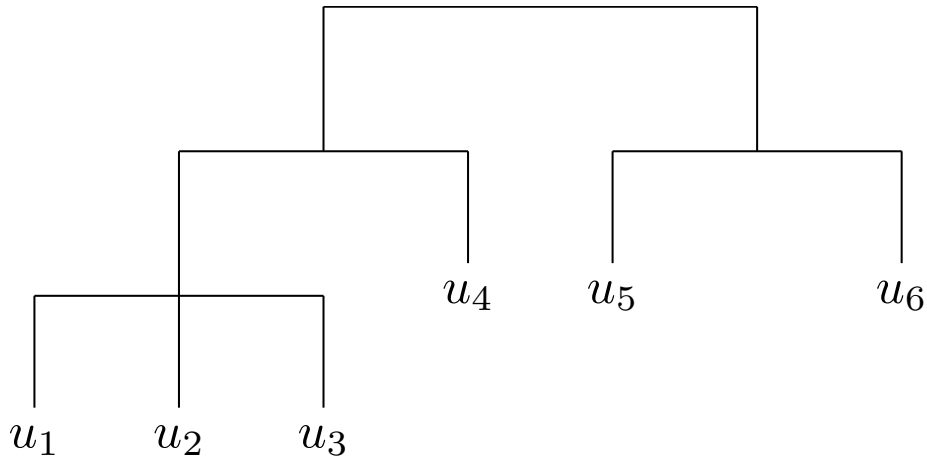


Figure 2.2: Structure of a 6-dimensional hierarchical Archimedean copula

2.3 Copula Construction Methods

Modelling dependencies inside and between networks requires a flexible dependence structure. Using one distinct copula family to sample failure times for all components in one or multiple networks is never precise enough. Therefore, the ability to combine different copula families in one structure is of utmost importance. This section presents two copula construction methods capable of this. These methods possess different modelling capabilities and strengths. For a discussion of additional methods and further details, see [61].

2.3.1 Hierarchical Archimedean Copulas

Hierarchical (alternatively: nested) Archimedean copulas are a class of copulas where groups of variables are connected by Archimedean copulas and these groups themselves are then coupled with another copula from one of the Archimedean families. This nesting structure may be repeated up to an arbitrary number of nesting levels. Figure 2.2 shows a visual representation of a hierarchical Archimedean copula with six variables in four groups as a dendrogram. Formally, hierarchical Archimedean copulas are defined by

$$C_{\varphi_0}(C_{\varphi_1}(u_{1,1}, \dots, u_{1,d_1}), \dots, C_{\varphi_J}(u_{J,1}, \dots, u_{J,d_J})) \quad (2.6)$$

where further nesting levels are defined recursively [65]. However, not all arbitrary combinations of $J + 1$ generators lead to Eq. 2.6 defining a valid copula.

The dependence in every group in this structure is governed by one parameter and variables that are close to each other (e.g, in the same group) share the same dependence [61]. This reduces the modelling flexibility substantially. An implementation of hierarchical Archimedean copulas can be found in the package `nacopula` for the statistical programming language **R** [109].

2.3.2 Pair Copula Construction

The goal of pair copula constructions (PCCs) is to build high-dimensional copulas from combinations of bivariate copulas and as such use the extensive theory on bivariate copulas to overcome limitations in the available literature on multivariate copulas [65].

Consider a vector of d random variables $\mathbf{X} = (X_1, \dots, X_d)$ with joint density function denoted by $f_{1:d}(x_1, \dots, x_d)$. The density can then be represented as a factorization of conditional densities:

$$f_{1:d}(x_1, \dots, x_d) = f_1(x_1) \cdot f_{2|1}(x_2|x_1) \cdot f_{3|2,1}(x_3|x_2, x_1) \cdot \dots \cdot f_{d|1:(d-1)}(x_d|x_1, \dots, x_{d-1}) \quad (2.7)$$

In the next step Sklar's theorem is applied to the conditional densities effectively splitting a multivariate density into bivariate copula densities and densities of univariate margins. Differentiating Eq. 2.1 with respect to a distribution with joint density $f(x_1, \dots, x_d)$, marginals f_j and marginal cdfs F_j , $j = 1, \dots, d$ leads to

$$f_{1:d}(x_1, \dots, x_d) = c_{1:d}(F_1(x_1), \dots, F_d(x_d)) \cdot f_1(x_1) \cdot \dots \cdot f_d(x_d), \quad (2.8)$$

where $c_{1:d}(\cdot)$ is the d -variate copula density. The bivariate case with pair-copula density $c_{1,2}(\cdot, \cdot)$ simplifies to

$$f_{1,2}(x_1, x_2) = c_{1,2}(F_1(x_1), F_2(x_2)) \cdot f_1(x_1) \cdot f_2(x_2), \quad (2.9)$$

which yields

$$f_{1|2}(x_1|x_2) = c_{1,2}(F_1(x_1), F_2(x_2)) \cdot f_1(x_1). \quad (2.10)$$

Equation 2.10 can be applied stepwise to Eq. 2.7 to fully decompose the multivariate density into bivariate copula densities and densities of univariate marginals. Note, that not all multivariate copulas can be modelled with this pair copula construction method.

2.3.3 Vine Copulas

Vines are a graphical representation of valid pair copula decompositions as sets of trees. Basic graph theory is used to define vines [65]. A regular vine (R-Vine) $\mathcal{V} = (T_1, \dots, T_{d-1})$ is defined as a tree sequence on d elements where:

- (1) T_1 is a tree with Nodes $N_1 = \{1, \dots, d\}$ and edges E_1 .
- (2) For $j \geq 2$, T_j is a tree with nodes $N_j = E_{j-1}$ and edges E_j .
- (3) For $j = 2, \dots, d-1$ and $\{a, b\}$ it must hold that $|a \cap b| = 1$.

The so called *proximity property* (3) states that, if an edge exists in T_j , $j \geq 2$ connecting a and b , in turn a and b must share a common node in T_{j-1} . Figure 2.3 shows a regular

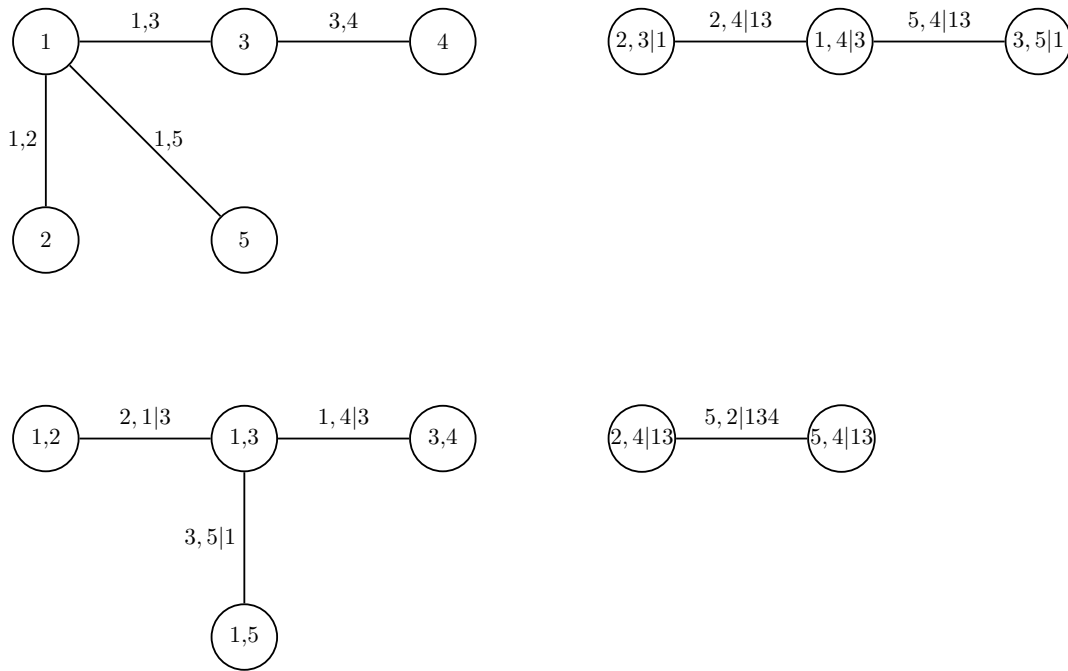


Figure 2.3: Graphical illustration of a four-dimensional copula as a regular vine.

vine representation of a 5-dimensional copula. There exist a multitude of d -dimensional R-vines. However, two sub-classes called C- and D-Vines are used almost exclusively. A regular vine \mathcal{V} is called a C-Vine if in each tree T_i there is one node that holds $n \in N_i$ such that $|\{e \in E_i | n \in e\}| \leq d - 1$. This condition states, that in each tree one node has the maximum degree (is connected to all other nodes). Alternatively, a D-Vine is characterised by each node $n \in N_i$ satisfying $|\{e \in E_i | n \in e\}| \leq 2$. Thus, any node may only have a maximum of two connections. Figure 2.4 shows the graphical structures of a five-dimensional C- and D-vine. Sampling of vine copulas is a non-trivial task. A regular vine on n variables possesses 2^{n-1} implied sampling orders [65]. Therefore, C- and D-Vines, where sampling is easier, are applied in all examples of this work with sampling from the vines being performed by the MATLAB toolbox `VineCopulaMatlab` [110].

2.4 Modelling Dependencies

Recalling from the introduction, the goal of this paper is to model complex dependencies between components of one system as well as interdependencies between components of different systems using copulas. For that reason, the previous section introduced some of the most popular copula families and different methods of constructing high dimensional copulas. This section deals with selecting appropriate copula families for different kinds of failures and investigating the usefulness of the copula construction methods in regards to the reliability analysis of complex networks.

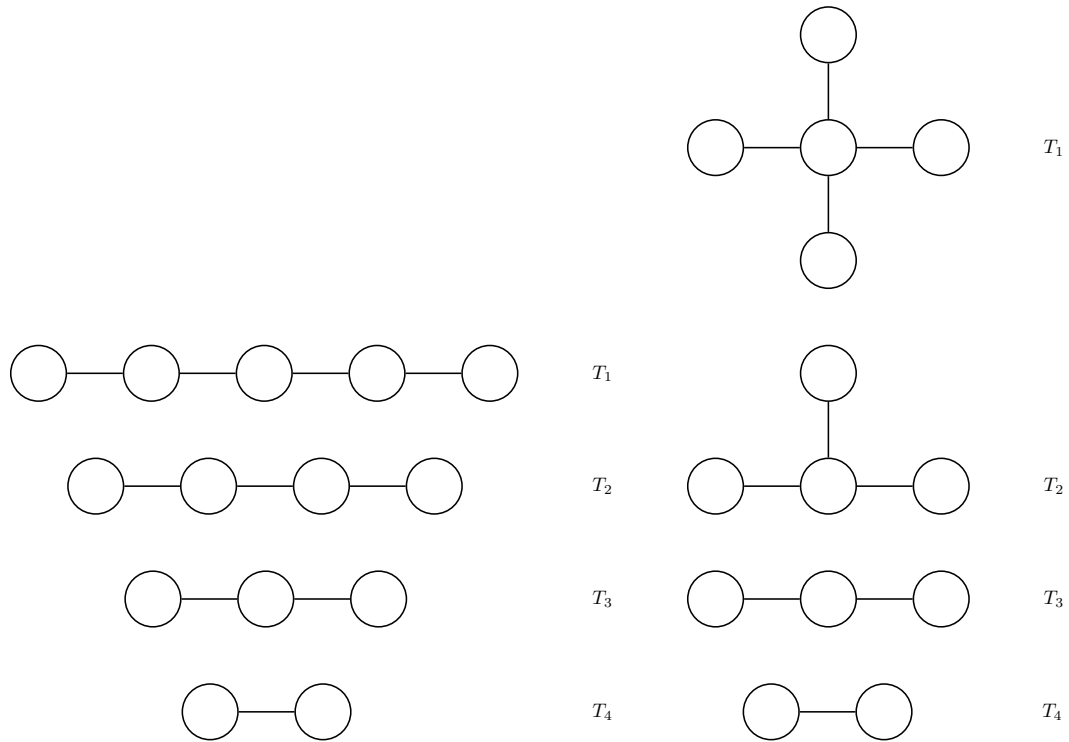


Figure 2.4: C-Vine (left) and D-Vine (right) in five dimensions.

Ideally, the dependency structures and therefore the copulas should be inferred from the measured component failure times of the dependent networks. However, as this data is rarely available for complex systems and the aim of this work is to prove the suitability of copulas in this framework, this is left for future work. Instead, a qualitative approach to the modelling of different kinds of failures is chosen. Examples of how to model two distinct classes of failures are given in the subsequent sections. These qualitative estimates could potentially serve as a basis for deducing the copula structure by Bayesian inference.

2.4.1 Common Cause of Failure

Common cause of failure is the event that two or more components fail simultaneously due to shared defects [111]. These weaknesses can include but are not limited to [112]:

- Manufacturing defects
- Errors by the maintenance or operator personal
- Shared environmental conditions

This work concentrates on the first weakness, manufacturing defects, especially those manifesting in early component life. The Clayton copula can be used to describe dependence between marginals where there is strong *lower tail* dependence. Lower (or upper) tail dependence is

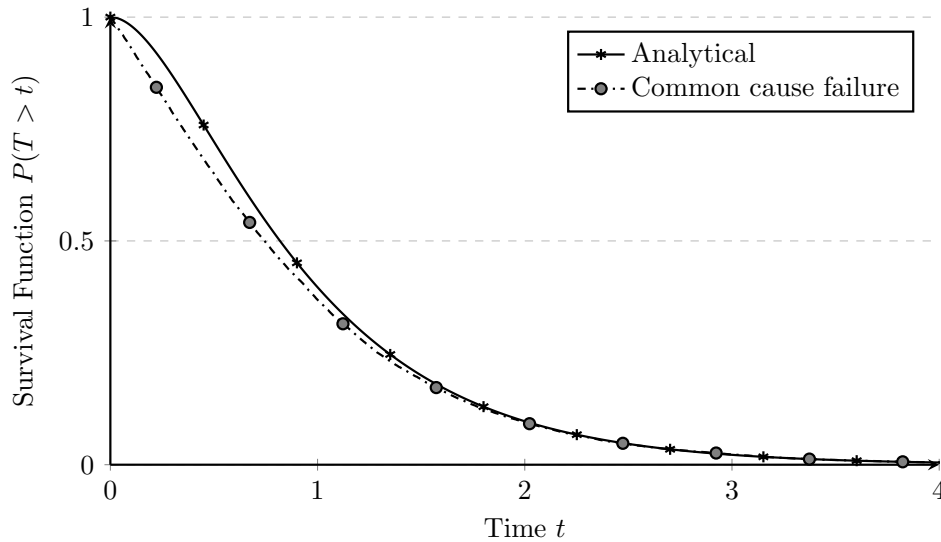


Figure 2.5: Reliability of a parallel system subject to common cause of failure.

concept expressing higher dependence in the lower-left (upper-right) quadrant of $[0, 1]^2$. This property of the Clayton copula is clearly evident from Fig. 2.1b where the samples in the lower-left quadrant are grouped closer together.

Consider a very simple system of two parallel components. The component failure times are assumed to be exponentially distributed with $\lambda = 1.5$ and are sampled from a bivariate Clayton copula with θ chosen such that Kendall's tau equals 0.3. Figure 2.5 shows a plot of the resulting reliability against the reliability in the independent case. The plot clearly shows how the lower tail dependence translates to the reliability of the system. Initially, the reliability is significantly reduced compared to the independent case. At later points in time, as the dependence weakens, this difference decreases. Contrarily, a copula exhibiting strong upper tail dependence might be used to model common cause failures at high component age.

2.4.2 Interdependencies

The treatment of interdependencies is not as simple as for common cause of failures. To understand the difficulties it is important to understand the two meanings dependence has in this case. When working with copulas, dependence is a measure of correlation or concordance and as is the nature of copulas, dependence is modelled independently of the marginals. As such, dependence in a statistical sense does not imply causality. However, this is exactly what interdependencies represent. If one component fails there is a chance that a dependent component will fail as well.

Consider two dependent components whose failure times are distributed with marginal distributions F_1 and F_2 and copula C , where $F_1 \neq F_2$. If failure times are sampled for both components from a fully dependent copula and apply the marginals using the inverse

transformation method, the failure times for the first component will still be distributed according to F_1 and the failures times for the second component will be distributed with F_2 . Even though perfect dependence is assumed, the components will not fail together. Since the copula approach separates the modelling of the dependence structure from modelling of the marginals, this causality can be included in the latter. In this case, a simple aggregation of the marginals is performed using the resulting strength of dependence (Kendall's tau) as a factor as shown in Eq. 2.11

$$U_1 = (1 - \tau) \cdot F_1^{-1}(u_1) + \tau \cdot F_2^{-1}(u_1) \quad (2.11)$$

2.4.3 Construction of the dependence structure

After selecting appropriate copula families to model the desired failure modes, the overall dependence structure for the network has to be selected. Three approaches exist based on the methods introduced in the previous section. The most straightforward approach is the application of multiple independent copulas to define dependence among groups of components. However, this does not allow for components to be connected to multiple other components by different copula families as one random variable can not be part of two independent copulas and as such is only suitable for simple scenarios. The two more advanced techniques for constructing high dimensional copulas presented are hierarchical Archimedean copulas and vine copulas. While HAC's offer more flexibility in terms of modelling the dependencies, they are still far more restrictive than vine copulas. This is largely due to their nested structure as compared to the graph based nature of vine copulas. Additionally, building a graph based dependence structure has obvious synergies with the reliability analysis of networks. For this reason, vine copulas are selected to build the overall dependence structure.

2.5 Reliability Analysis

This section recaps the numerical methodology used to compute the network reliability first introduced in [104]. It is based on the survival signature, an extension of the system signature, and Monte Carlo simulation.

2.5.1 Survival Signature

The survival signature is a novel tool for the quantification of system and network reliability based on the system signature [27, 28]. Both signatures allow for a separation of the system structure from its probabilistic characteristics such as component failure times. However, the system signature has a severe limitation in that it is only defined for systems made up of a single component type, which does not apply to complex networks. The survival signature addresses this drawback by generalizing the signature to systems with an arbitrary number of component types.

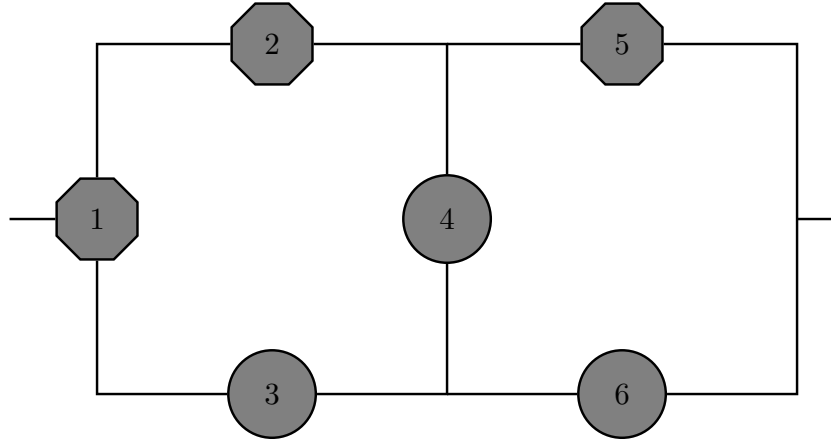


Figure 2.6: Network with six components equally divided into two component types.

Consider a system with m components. The state vector is defined as $\underline{x} = (x_1, \dots, x_m)$, where $x_i = 1$ indicates a component in working condition, while $x_i = 0$ indicates a component in a failed state. As such, the state vector represents the state of the individual components. The state of the full system is obtained by applying the structure function $\varphi(\underline{x})$ to the state vector. As before, $\varphi(\underline{x}) = 1$ indicates a working system and $\varphi(\underline{x}) = 0$ indicates that the system has failed. The structure function is defined based on the problem at hand. In this work, the structure function is assumed to return 1 if a path from any start node to any end node exists for the current network state. Calculating the survival signature for l out of m components working then becomes a combinatorial problem defined as

$$\Phi(l) = \binom{m}{l}^{-1} \sum_{\underline{x} \in S_l} \varphi(\underline{x}) \quad (2.12)$$

The survival signature is easily extended to systems with multiple component types. Consider a system with K component types, m_k components per type k ($k = 1, \dots, K$) and l_k out of m_k components per type in a working state, the survival signature becomes

$$\Phi(l_1, \dots, l_K) = \left[\prod_{k=1}^K \binom{m_k}{l_k}^{-1} \right] \times \sum_{\underline{x} \in S_{l_1, \dots, l_K}} \varphi(\underline{x}) \quad (2.13)$$

As an example, consider a system with two component types and three components per type as illustrated in Fig. 2.6. Here, node 1 is selected to be the start node and nodes 5 and 6 represent the end nodes. The full survival signature for the network is show in Table 2.2.

While algorithms to calculate the survival signature have already been available for a number of years [38, 113], efficient computation of the signature for systems with large numbers of components and types still poses a numerical challenge. A new approach attempting to reduce the high computational demand of the survival signature using graph theory and Monte Carlo approximation can be found in [114].

Table 2.2: Survival signature of the network shown in Fig. 2.6.

l_1	l_2	$\Phi(l_1, l_2)$	l_1	l_2	$\Phi(l_1, l_2)$
0	0	0	2	0	0
0	1	0	2	1	0
0	2	0	2	2	4/9
0	3	0	2	3	6/9
1	0	0	3	0	1
1	1	0	3	1	1
1	2	1/9	3	2	1
1	3	3/9	3	3	1

2.5.2 Survival Function

Based on the survival signature, the survival function is defined as

$$P(T_s > t) = \sum_{l_1=0}^{m_1} \dots \sum_{l_k=0}^{m_k} \Phi(l_1, \dots, l_K) P\left(\bigcap_{k=1}^K \{C_t^k = l_k\}\right) \quad (2.14)$$

This function gives the probability that a network is still working at time t , in other words the reliability of the system. The equation clearly shows the separation of structural information (survival signature on the left) and probabilistic information about component failures (right). This is beneficial as it allows to analyze the network once ahead of the reliability analysis instead of having to re-evaluate the structure every step of the way as with traditional techniques such as fault tree analysis. Additionally, this makes it possible to efficiently run multiple failure scenarios against a network.

2.5.3 Simulation

Component failure times are sampled from the vine copula, after selecting the number of desired samples N_{mc} and a sufficiently small time step, and transformed to their respective marginals. Next, for all combinations l_1, \dots, l_K from the survival signature and all time steps t the number of samples representing the exact same combination (amount of components still working at time t) are counted as $N_{l_1, \dots, l_K}(t)$. Then, the probabilistic part of the survival function is approximated by

$$P\left(\bigcap_{k=1}^K \{C_t^k = l_k\}\right) = \frac{N_{l_1, \dots, l_K}(t)}{N_{mc}} \quad (2.15)$$

In a final step, the partial reliabilities for all combinations are multiplied by their probability $\Phi(l_1, \dots, l_K)$, introducing the structural information into the reliability, and then summed yielding the full reliability of the network. This means that no computations must be performed for combinations where the probability in the survival signature is zero, further increasing the

efficiency of the simulation. This fact is especially useful in higher dimensions where large parts of the survival signature are negligible. A pseudo-algorithm illustrating how to obtain the survival function of the system based on the survival signature and the failure times sampled from the vine copula is given in Algorithm 1. The analytically and numerically computed survival functions for the network shown in Fig. 2.6 assuming independent exponential failure distributions for the components with $\lambda_1 = 0.8$ and $\lambda_2 = 1.6$ are presented in Fig. 2.7.

Pseudocode 1 Monte Carlo simulation for network reliability

Input:

Φ survival signature
 t_{fail} component failure times sampled from the vine copula
 v_{time} vector of time steps
 N_{mc} number of Monte Carlo samples

Output:

P Reliability of the network
function NETWORKRELIABILITY($\Phi, t_{fail}, v_{time}, N_{mc}$)
 for each l_1, \dots, l_K in S_{l_1, \dots, l_K} **do:** \triangleright Loop over all combinations of the survival signature
 if $\Phi(l_1, \dots, l_K) > 0$ **then:**
 for each t in v_{time} **do:**
 $N_{working} \leftarrow \text{sum}(t_{fail} > t)$ \triangleright Find components working at time t
 $N_{l_1, \dots, l_K} \leftarrow \text{sum}(N_{working} = l_1, \dots, l_K)$ \triangleright Count matching combinations
 $P_{partial}(l_1, \dots, l_K) \leftarrow N_{l_1, \dots, l_K} / N_{mc} \cdot \Phi(l_1, \dots, l_K)$
 end for
 end if
 end for
 $P \leftarrow \text{sum}(P_{partial})$ \triangleright Sum partial reliabilites yielding the full system reliability
end function

2.5.4 Imprecise Reliability Analysis

Two types of uncertainties must be considered during the reliability analysis, namely, aleatory and epistemic uncertainties. Aleatory uncertainty describes the natural randomness inherent in a process such as component degradation and external forces affecting the system (natural hazards, earthquakes, etc.), while epistemic uncertainty represents the uncertainty due to vagueness in information or a lack thereof. The latter is usually regarded as reducible through acquiring of additional data and information.

Aleatory uncertainty is naturally handled by the reliability analysis technique. Through assuming failure time distributions for the component failures and sampling these during Monte Carlo simulation, the randomness that the model is subject to is included in the analysis. However, the selection appropriate failure time distributions is typically based on either data or expert knowledge, neither of which yields perfect results. In turn, this introduces epistemic

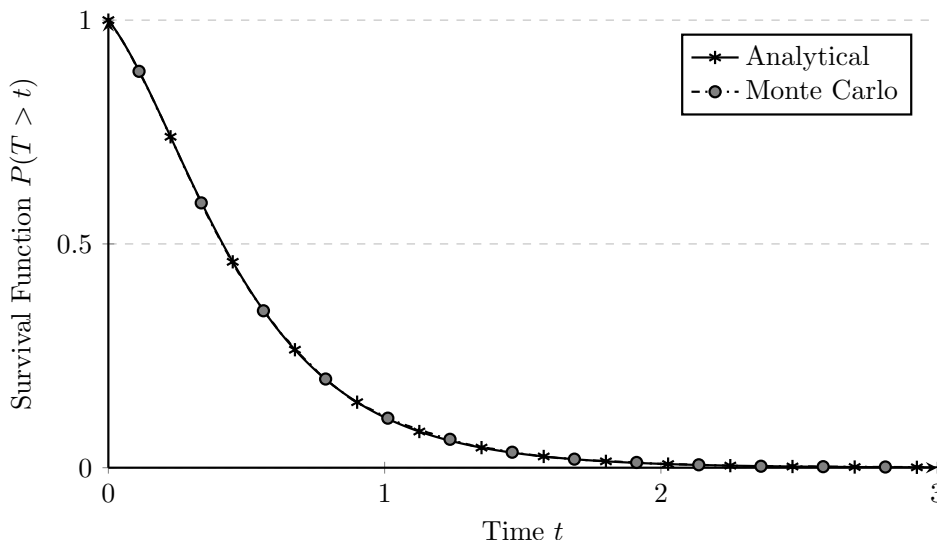


Figure 2.7: Survival function for the network in Fig. 2.6.

uncertainty into the model. These epistemic uncertainties can be tackled by using imprecise probability methods where instead of a single model, a set of plausible models is applied. As a result, the uncertainty propagates through the models and is ultimately reflected in the probability of failure, i.e. the reliability of the network.

Consider two non-decreasing functions \underline{F} and \overline{F} mapping the real line \mathbb{R} into $[0, 1]$, with $\underline{F}(x) \leq \overline{F}(x)$ for all $x \in \mathbb{R}$. Let $[\overline{F}(x), \underline{F}(x)]$ denote the set of non-decreasing functions mapping \mathbb{R} into $[0, 1]$ such that $\underline{F}(x) \leq F(x) \leq \overline{F}(x)$. When \overline{F} and \underline{F} circumscribe an imprecisely known probability distribution, $[\overline{F}, \underline{F}]$ is called the *probability box* or *p-box* of said probability distribution. As a result, if $[\overline{F}, \underline{F}]$ is a p-box for a random variable X whose distribution is known to be within the p-box, then $\overline{F}(x)$ and $\underline{F}(x)$ are the upper and lower bounds on $F(x)$, respectively [57]. An example of an exponential p-box with parameters $\lambda \in [1.2, 2.2]$ is shown in Fig. 2.8. In this case, only two CDFs need to be computed to fully define the p-box. However, for most families of distributions, four or more CDFs must be evaluated [57].

By feeding the bounds of the p-box into the reliability analysis, the epistemic uncertainty propagates into the result. Thus, instead of one survival function, the result is an upper and a lower bound [58]. Figure 2.9a shows an example of the upper and lower bounds obtained by performing a reliability analysis of a simple system of two parallel components of the same type, assuming the p-box shown in Fig. 2.8 for the failure time distributions.

Similarly to a p-box, instead of considering just one precise copula, a set of copulas can be considered to account for uncertainty in the dependencies. Pelesoni et al. [115] generalized p-boxes to the bivariate case. If \underline{C} and \overline{C} are two copulas such that $\underline{C} \leq \overline{C}$, then $[\underline{C}, \overline{C}]$ forms an *imprecise copula*. Let $[\underline{F}_X, \overline{F}_X]$ and $[\underline{F}_Y, \overline{F}_Y]$ be two univariate p-boxes for the random

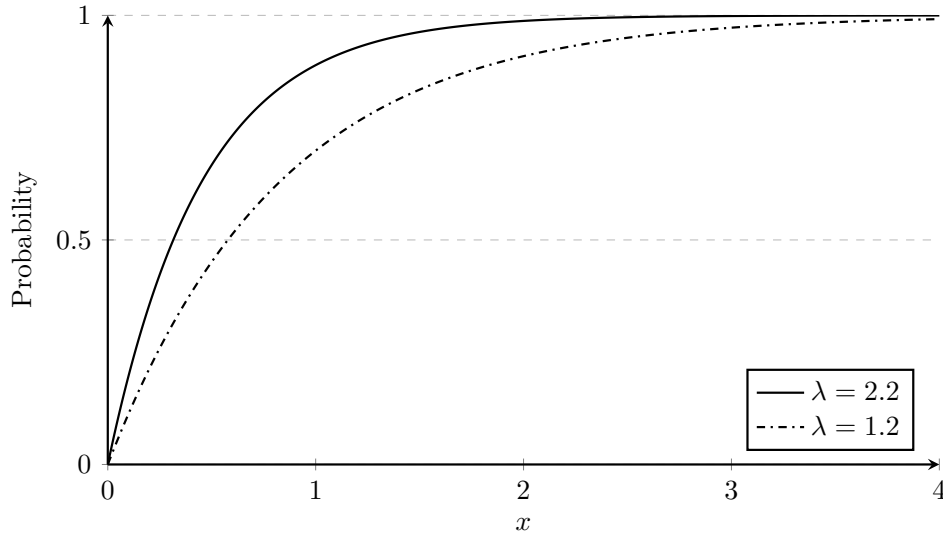


Figure 2.8: Example of an exponential p-box with $\lambda \in [1.2, 2.2]$.

variables X and Y respectively, then $[\underline{F}, \overline{F}]$ defined by

$$\underline{F}(x, y) = \underline{C}(\underline{F}_X(x), \underline{F}_Y(y)) \quad (2.16)$$

and

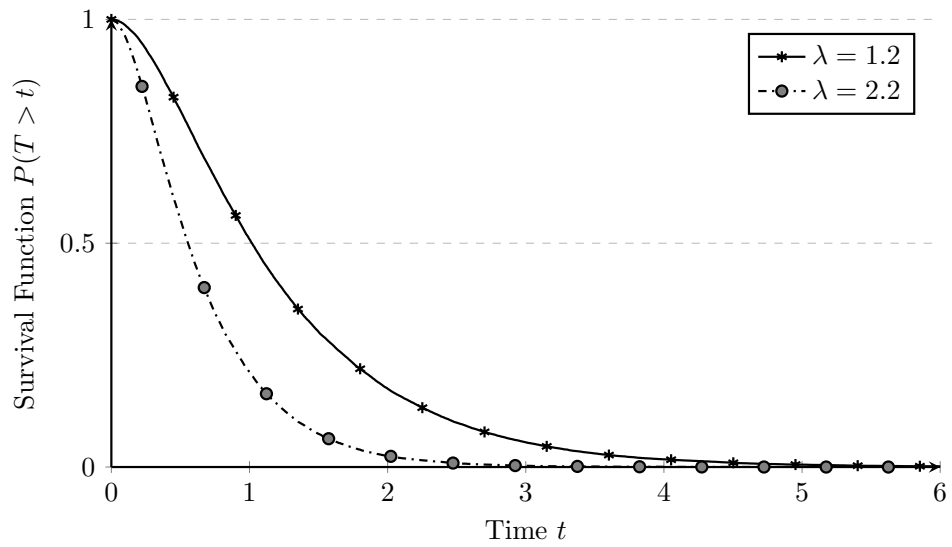
$$\overline{F}(x, y) = \overline{C}(\overline{F}_X(x), \overline{F}_Y(y)) \quad (2.17)$$

is a bivariate p-box [116]. As with the p-box, defining an imprecise copula from an interval on its parameters imposes bounds on the system reliability. Consider again a simple system of two parallel components, in this case interlinked by an imprecise Gaussian copula with $\rho \in [0.3, 0.6]$. The upper and lower bounds for the reliability are presented in Fig. 2.9b.

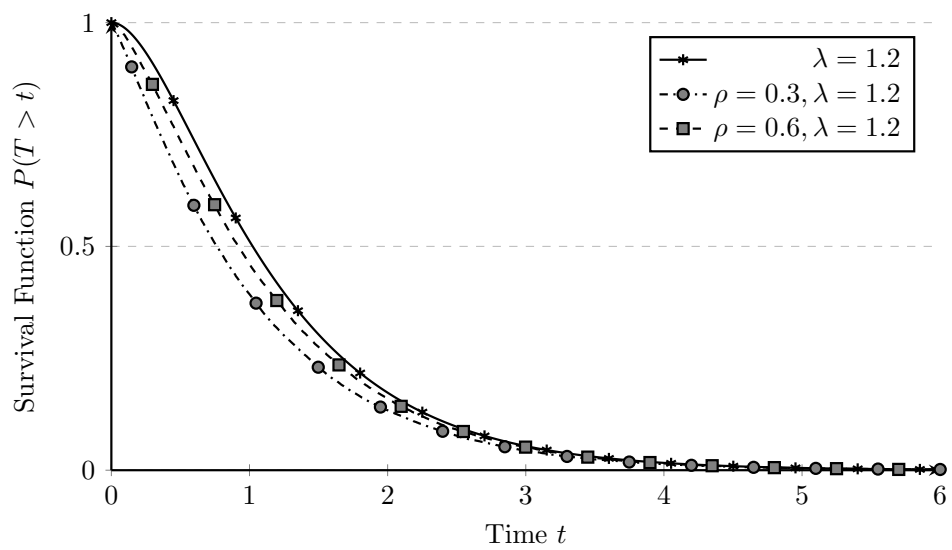
2.6 Numerical Example

The network structures for the following numerical example are taken from the IEEE Reliability Test System (RTS) [117]. The system is effectively split into two sub-systems (see Fig. 2.10 and Fig. 2.11) by removing the transformers that link the low power to the high power grid. Components in the networks are classified into five types. Component types 1 and 5 are the non-generating nodes in networks 1 and 2 respectively. The generating nodes are divided into three component types 2, 3 and 4. These represent different types of generators such as nuclear, oil or coal power plants. Note that this is no attempt at solving the IEEE RTS. The system is merely providing the network topology.

In a first step to obtain the reliability, the required survival signatures for both networks are calculated using the approach presented in [114]. Next, the vine copula that is used for sampling the individual component failure times is assembled from bivariate copulas. A common cause



(a) Probability-box



(b) Imprecise copula

Figure 2.9: Bounds on the reliability resulting from applying a p-box (a) or an imprecise Gaussian copula (b) to a simple system of two parallel components.

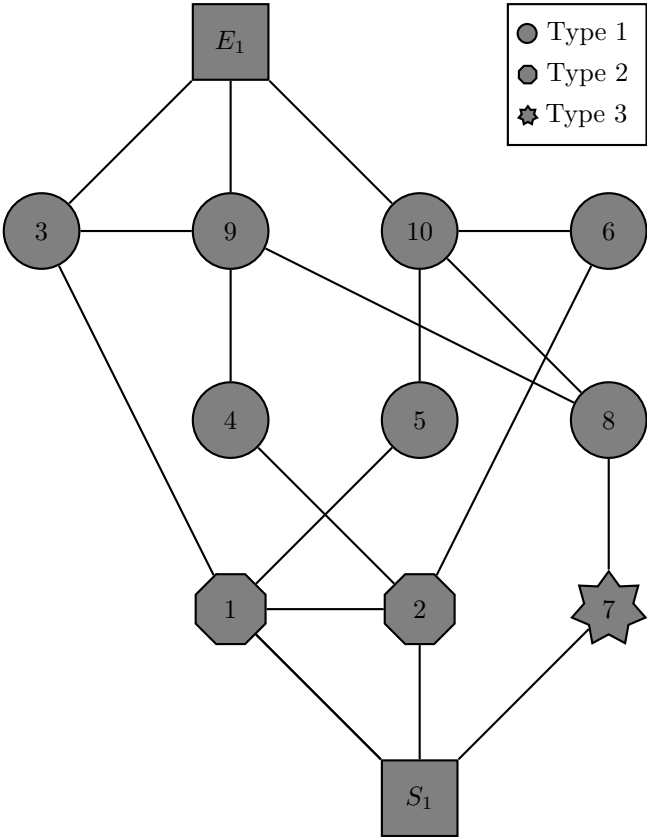


Figure 2.10: Structure of the first network taken from the IEEE RTS

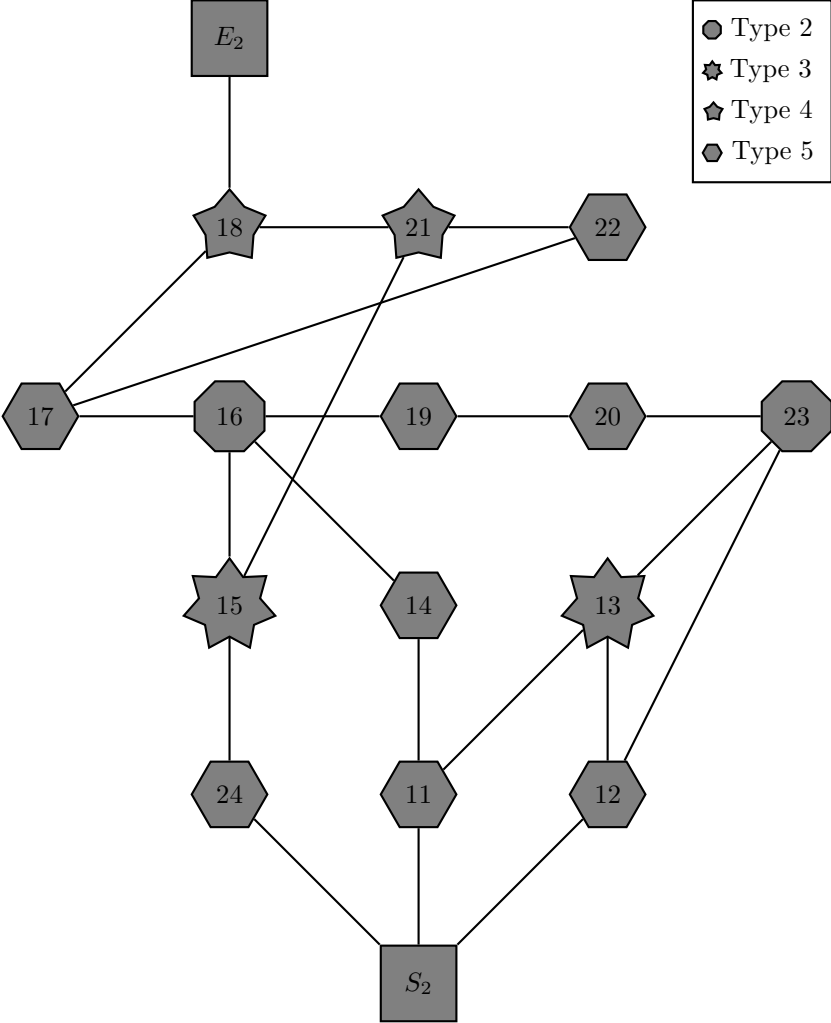


Figure 2.11: Structure of the second network taken from the IEEE RTS

Table 2.3: Failure rate ranges of the exponential marginal distributions and copula parameters used in the numerical example.

Parameter	Definition	Parameter range
λ_1	Failure rate of component type 1	$\lambda_1 \in [0.8, 1.2]$
λ_2	Failure rate of component type 2	$\lambda_2 \in [1.4, 1.5]$
λ_3	Failure rate of component type 3	$\lambda_3 \in [1.6, 1.9]$
λ_4	Failure rate of component type 4	$\lambda_4 \in [2.0, 2.3]$
λ_5	Failure rate of component type 5	$\lambda_5 \in [1.8, 2.2]$
τ_1	Clayton copula parameters on component type 2	$\tau_1 \in [0.1, 0.3]$
τ_2	Clayton copula parameters on component type 3	$\tau_2 \in [0.2, 0.4]$
τ_3	Clayton copula parameters on component type 4	$\tau_3 \in [0.1, 0.3]$
τ_4	Gaussian copula parameters between network 1 and 2	$\tau_4 \in [0.4, 0.8]$

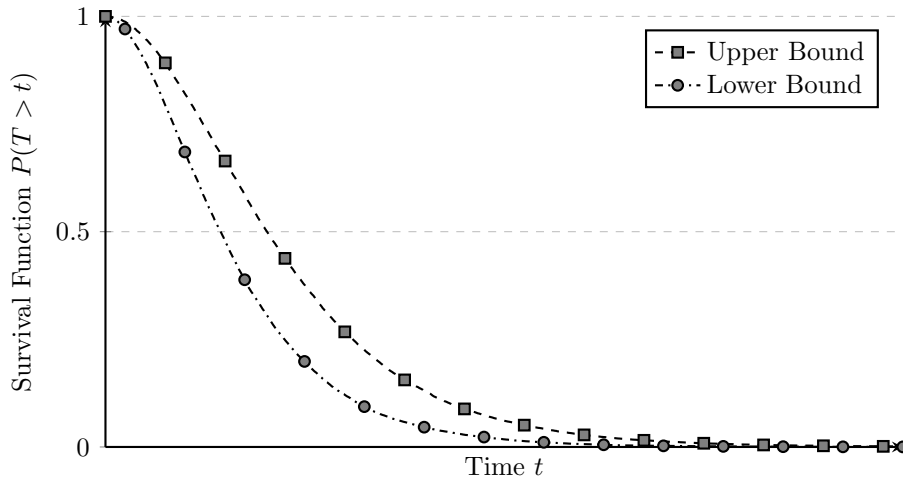


Figure 2.12: Bounds on the reliability of network 1.

of failure is set among the groups of nodes of types 2, 3, and 4 through imprecise bivariate Clayton copulas. Next, the transformers that were removed to split the network in two, are reintroduced as interdependencies between the nodes 3 and 11, 9 and 24 as well as 10 and 12 using imprecise bivariate Gaussian copulas. All one-dimensional marginal distributions are assumed to be exponentially distributed. The parameters for the marginals and the copulas are presented in Tab. 2.3.

Finally, the reliability analysis is performed using the previously introduced Monte Carlo simulation method. The upper and lower bounds of the reliability for network 1 is presented in Fig.2.12. For comparison, the plot also contains a deterministic reliability analysis (all mean values) of network 1.

2.7 Conclusion

This paper presented a novel approach to the modelling of complex dependencies in interdependent networks by leveraging multivariate copulas. Over the course of this work the necessary theory on copulas, dependence measures and pair copula construction techniques was discussed. Of the investigated structures vine copulas have shown to be ideally suited to model higher dimensional dependencies with sufficient flexibility. The capabilities of the proposed approach were highlighted using a scenario based on the network topology of the IEEE Reliability Test System. The application of vine copulas has proven to be able to represent a complicated model with multiple competing failure modes. It was shown that imprecision can easily be included in the reliability analysis. Nonetheless, the modelling flexibility of this method comes at a price. Finding a suitable vine copula structure is not a trivial task and greatly suffers from the curse of dimensionality.

In order to facilitate a transparent illustration of the approach, we have chosen a relatively small system which still carries the key features of a realistic system. The expansion to a large complex system would require a number of additional numerical challenges to be solved. This is beyond the scope of this paper and therefore left for future work. The challenges include deriving the vine copula model from data. Gruber and Czado [118] presented a promising method for Bayesian model selection and inference of vine copulas. The technique is the first to be able to jointly select the vine tree structure and the copula families. Additionally, the modelling aspects of other competing failure events such as external threats (e.g. earthquakes, tsunamis, terrorist attacks) must be closely investigated.

Acknowledgements

This project is partially supported by the European Union's Horizon 2020 Research and Innovation Programme RISE under grant agreement no. 730888 (RESET).

Funding Data

- The European Union's Horizon 2020 Research and Innovation Programme RISE under Grant Agreement No. 730888 (RESET) (Funder ID: 10.13039/100010661).

3 | Numerically Efficient Computation of the Survival Signature for the Reliability Analysis of Large Networks

Numerically Efficient Computation of the Survival Signature for the Reliability Analysis of Large Networks

Jasper Behrendorf^a, Tobias-Emanuel Regenhardt^a, Matteo Broggi^a, Michael Beer^{a,b,c}

^aInstitute for Risk and Reliability, Leibniz Universität Hannover, Hannover, Germany

^bInstitute for Risk and Uncertainty, University of Liverpool, Liverpool, United Kingdom

^cInternational Joint Research Center for Engineering Reliability and Stochastic Mechanics (ERSM), Tongji University, Shanghai, China

Published in *Reliability Engineering and System Safety* on July 24, 2021

Abstract

Societal growth thrives on the performance of critical infrastructure systems such as water supply systems, transportation networks or electrical distribution systems. This makes the reliability analysis of these systems a core focus for researchers today. The survival signature is a novel tool for analysing complex networks efficiently and outperforms traditional techniques in several key factors. Its most unique feature being a full separation of the system structure from probabilistic information. This in turn allows for the consideration of diverse component failure descriptions such as dependencies, common causes of failure and imprecise probabilities. However, the numerical effort to compute the survival signature increases with network size and prevents analysis of complex systems. This work presents a new method to approximate the survival signature, where system configurations of low interest are first excluded using percolation theory, while the remaining parts of the signature are approximated by Monte Carlo simulation. The approach is able to accurately approximate the survival signature with very small error at a massive reduction in computational demands. The accuracy and performance are highlighted using several simple test systems as well as two real world problems.

Keywords: Survival signature, Monte Carlo simulation, percolation, reliability analysis

3.1 Introduction

Critical infrastructure systems such as electrical, gas and water distribution systems, traffic networks and communication networks are cornerstones of modern societies. Our dependence on these systems comes with a demand for accurate reliability analyses to ensure their functionality. However, with increasing size and complexity, the analysis and assessment of reliability of these systems comes with an increase of computational effort. Extensive research on algorithms for the reliability analysis of systems and networks is readily available [21]. Past research has highlighted the importance of considering effects such as dependencies, common causes of failure and imprecision [42, 119, 120] during the analysis. This is where traditional approaches, e.g. fault tree analysis or reliability block diagrams reach their limits. A recent development in system analysis aiming to alleviate these shortcomings is the survival signature [27].

The survival signature was developed as a generalization of the systems signature [121] to allow for multiple component types. Perhaps the greatest advantage of the survival signature to traditional approaches is the full separation of system structure from probabilistic information.

This is a clear advantage over more common tools for system reliability, where the structure of the system needs to be modified or extra failure events need to be modelled to allow for consideration of the aforementioned effects, as for example pointed out by Li et al. [122]. Recent research showed that repairable systems [36], mission-stage-behaviour [123] and the combination of subsystems [29] can easily be implemented into the survival signature ecosystem. Reliability analysis using the survival signature has been widely studied in recent years, including a variety of simulation algorithms [37], dependent failures [32], imprecision [58] and more.

Just like the traditional techniques, the survival signature suffers greatly from the curse of dimensionality. This means, that with increasing network size and number of component types the numerical demand increases as well with non-polynomial growth. At the number of components and types typical of real infrastructures the computational effort required to evaluate the survival signature becomes prohibitive. Several works aimed at working around this limitation have been published in recent years. Reed [38] presented an efficient method of calculating the survival signature based on transforming a fault tree representation of a system to a binary decision diagram. While this method performs very well in cases where the fault tree or binary decision diagram is already known, it becomes increasingly impractical with growing network size/complexity. Another recently developed approach is based on the extended universal generating function (UGF) [40]. However, deriving the UGF is a non-trivial task itself and restricts the application to systems defined as reliability block diagrams.

This paper presents a new approach to the approximation of the survival signature based on percolation theory and Monte Carlo simulation. First, percolation theory is used to find areas of the survival signature that can be safely excluded [124]. Then, the remaining entries are approximated using Monte Carlo simulation [114]. The method is able to efficiently compute the signatures of arbitrary systems and structure functions.

The remainder of the paper is structured as follows. Section 3.2 presents the theory on the survival signature while Section 3.3 introduces percolation theory. The developed simulation algorithm is discussed in Section 3.4 including the application to toy examples and quantification of errors. Section 3.5 applies the proposed approach to more complex real world examples, followed by some concluding remarks in Section 3.6.

3.2 Survival Signature

The current state of a system consisting of m components can conveniently be described by a state vector $\underline{x} \in \{0, 1\}^m$. An entry $x_i = 1$ denotes a functional component i , while $x_i = 0$ indicates a non-functional or failed component. The labeling must be consistent although its initial choice is arbitrary. The global state of the system is defined by a structure function $\varphi : \{0, 1\}^m \rightarrow \{0, 1\}$. The structure function is defined for every possible state \underline{x} of the system and evaluates whether the system is operational ($\varphi(\underline{x}) = 1$) or not ($\varphi(\underline{x}) = 0$).

It is safe to assume in most cases that the system analysed is coherent. A system is labeled

as such if the structure function φ is not decreasing if the amount of working components $|\underline{x}|$ increases (and vice-versa), i.e., the repair of a component will not lead to a less functional system. A related, however not necessary assumption is that the system is always fully operational in the case of all components working ($\varphi(\underline{1}) = 1$) and not operational if all components are broken ($\varphi(\underline{0}) = 0$).

In case of systems consisting of multiple component types, let $K \geq 2$ be the number of component types, and m_k the amount of components of one specific type k . It follows that $\sum_{k=1}^K m_k = m$. As the labeling of the components in the state vector is arbitrary, it can be written in groups ordered by component type: $\underline{x} = (\underline{x}^1, \underline{x}^2, \dots, \underline{x}^K)$. Each of these sub-vectors now indicate the states of all components of that specific type, for example $\underline{x}^k = (x_1^k, x_2^k, \dots, x_{m_k}^k)$.

For any such kind of system, the survival signature $\Phi(l_1, l_2, \dots, l_K)$ is now defined as follows: Given that exactly l_k out of m_k components of every type are working, the probability that the system is operational is Φ . In other words, Φ denotes the percentage of working system configurations when l_k out of m_k components are working, without taking into account the reasons (e.g., failure modes) behind the failure of these components. Additionally, it should be pointed out that components of the same type are indistinguishable, i.e., it is possible to know how many components of a specific type are working, but not in which part of the system. Thus, the survival signature is a k -dimensional array of size $(m_1 + 1) \times (m_2 + 1) \times \dots \times (m_k + 1)$ (including case $l_k = 0$ that none of the components of a type are working) [27]. In the remainder of the paper, \underline{l} is used as a shorthand for l_1, l_2, \dots, l_K .

From the coherency assumptions mentioned above, it trivially follows that

$$\begin{aligned} \Phi(\underline{l} = \underline{0}) &= 0, \\ \Phi(\underline{l} = \underline{1}) &= 1, \\ \Phi(\underline{l}^a) &\leq \Phi(\underline{l}^b) \text{ if } l_k^a \leq l_k^b, \forall k \in (1, 2, \dots, K), \end{aligned} \quad (3.1)$$

where $\underline{0}$ and $\underline{1}$ refer to system configurations with all components failed or working respectively. The vectors \underline{l}^a and \underline{l}^b represent two arbitrary entries of the survival signature.

The direct computation of one specific entry $\Phi(\underline{l})$ of the survival signature is achieved by enumeration of all working states that satisfy the condition that l_k out of m_k components are functional for components $k = 1, \dots, K$. The total amount of possible combinations are $\binom{m_k}{l_k}$ for all k . The set of all these allowed combinations for all components is denoted by S_{l_1, \dots, l_K} . Thus the magnitude of this set is $\prod_{k=1}^K \binom{m_k}{l_k}$. The fraction of functional states over the amount of all possible states now yields the probability of the system being operational:

$$\Phi(\underline{l}) = \left[\prod_{k=1}^K \binom{m_k}{l_k} \right]^{-1} \times \sum_{\underline{x} \in S_{l_1, \dots, l_K}} \varphi(\underline{x}), \quad (3.2)$$

under the condition that the failure times of the individual components of one type are equally likely to occur. The complete survival signature for a simple system of two component types as shown in Fig. 3.1 is presented in Table 3.1.

The structural information of the system functionality is completely separated from the temporal behaviour of the individual components. The component failure times are included as the probability $P(C_t^k)$ that a specific amount C of component type k is functional at a given point in time t :

$$P(T_s > t) = \sum_{l_1=0}^{m_1} \dots \sum_{l_K=0}^{m_K} \Phi(l) P\left(\bigcap_{k=1}^K \{C_t^k = l_k\}\right), \quad (3.3)$$

providing the probability that the system failure time T_s is after the current point in time (the system's *survival function*). However, in the case that for any type k the failure times are independently and identically distributed (*iid*) with a known cumulative distribution function $F_k(t)$, the probabilistic part of the survival function can be simplified to

$$\begin{aligned} P\left(\bigcap_{k=1}^K \{C_t^k = l_k\}\right) &= \prod_{k=1}^K P(C_t^k = l_k) = \dots \\ \dots &= \prod_{k=1}^K \binom{m_k}{l_k} F_k(t)^{m_k - l_k} [1 - F_k(t)]^{l_k}. \end{aligned} \quad (3.4)$$

It is in this separation of structural and probabilistic information where the advantages of the survival signature compared to traditional approaches lie. Inclusion of complex effects such as imprecise probabilities or dependent component failures have no influence on the structural evaluation of the system. Note, that the method presented in this paper is only applicable to the structural (signature) part of the survival function. Simulation techniques for the probabilistic part of the equation and consideration of imprecision and dependencies are already available [37, 93].

In this work, application of the survival signature is restricted to systems consisting of binary components, in accordance with its original definition [27]. Generalization of the survival signature to multi-state systems is still actively being researched [30, 125].

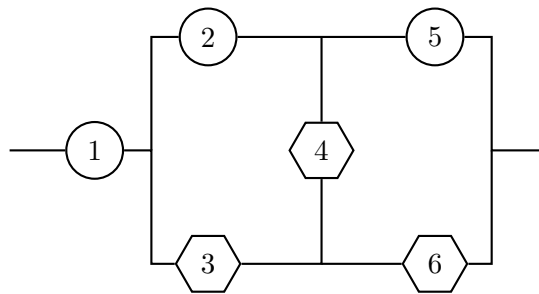


Figure 3.1: Example system with $K = 2$ component types and $m_1 + m_2 = 3 + 3 = 6$ components. The component types are represented by different shapes.

l_1	l_2	$\Phi(l_1, l_2)$	l_1	l_2	$\Phi(l_1, l_2)$
0	0	0	2	0	0
0	1	0	2	1	0
0	2	0	2	2	4/9
0	3	0	2	3	6/9
1	0	0	3	0	1
1	1	0	3	1	1
1	2	1/9	3	2	1
1	3	3/9	3	3	1

Table 3.1: Survival signature of the system in Fig. 3.1.

3.3 Percolation

If a system's structure function is given as a logical block diagram, it is referred to as a *reliability block diagram* (RBD). The individual components (or subsystems of several components) are represented by blocks connected through edges. Typically two completely reliable nodes are given on both ends of the diagram. The system is operational, if these two *source* (s) and *target* (t) nodes are connected.

Evaluation of the structure function is now equivalent to finding the *s-t-connectivity* of an undirected and unweighted graph as the edges of the RBD are exclusively logical connections without any distinction between edges. Additionally, the system can be decomposed into series- and parallel-subsystems to reduce network size.

In the context of survival signature computation the s-t-connectivity needs to be computed for all $\underline{x} \in S_{l_1, \dots, l_K}$ for all $l_k = 0, 1, \dots, m_k$ considering all K component types. This fact leads to high numerical costs even for medium-sized systems due to the increase in combinations of components to check.

The search of connections between two ends of complex networks or lattices (and the formation of clusters inside a network, which is a related concept) is one of the main aspects of *percolation theory*. Albeit usually introduced as a theory for multidimensional lattice networks [126], the results obtained in percolation theory can easily be applied to graphs and networks. Network robustness is one of the many applications of percolation to graph theory other than the classical use in material sciences. This provides an opportunity to improve the handling of vast state spaces during analysis of structure functions. Percolation theory has been successfully applied to network reliability problems in the past [127].

3.3.1 Percolation processes

A percolation process is defined as the random deletion of nodes from the network without any rewiring of the edges connected to the deleted nodes. The amount of deleted nodes can be determined as fraction of removed nodes, denoted by f .

In cases of small fractions f , i.e. where only a small number of nodes is removed from the network, this results in a high probability that a large, system-spanning structure connecting s and t (and all nodes if $f \rightarrow 0$) exists. This structure is called a *giant connected cluster* or *giant connected component* [128]. For increasing values of f , more nodes are removed eventually leading to the collapse of the network. The point where the giant connected cluster vanishes is denoted by the critical fraction f_c .

The connection of two individual nodes has little value when analysing networks/graphs in general, as opposed to reliability block diagrams. Thus, the probability $P_\infty(f)$ that a giant connected cluster exists [129] can be used to evaluate the overall state of the network instead.

One of the main outcomes of percolation theory for graphs is that the behavior of a network depends mainly on three different exponents and the critical value f_c [130]:

$$\begin{aligned} \langle s \rangle &\sim |f - f_c|^{-\gamma} \text{ (avg. cluster size),} \\ p_{\text{in}} &\sim (f - f_c)^\beta \text{ (prob. that random node is in cluster),} \\ \xi &\sim |f - f_c|^{-\nu} \text{ (mean distance in cluster).} \end{aligned} \quad (3.5)$$

This behaviour resembles phase transitions from statistical mechanics and the exponents are highly dependent on the global structure of the graph. The system tends to maintain its large connected component until $f_c \cdot \sum_{k=1}^K m_k$ are taken away. Then, the system collapses into smaller, isolated clusters. While the exponents are hard to obtain analytically, the critical fraction f_c (where the giant connected cluster vanishes) can easily be derived directly from the network structure.

3.3.2 Percolation threshold and survival signature

The *Molloy-Reed Criterion* is a simple condition for a giant cluster to exist: For a graph to contain a giant connected component, most nodes need to be connected to at least two other nodes [131]. For any graph, this can be expressed as

$$\kappa = \frac{\langle d^2 \rangle}{\langle d \rangle} > 2, \quad (3.6)$$

with d being the node degree (sum of all connections going into a node), and $\langle d \rangle$ and $\langle d^2 \rangle$ being the first and second moment of the degree distribution over the network. The value of κ can be directly computed from any graph representation, e.g. a double loop over the network's adjacency matrix. With the ratio κ the critical threshold for any system is obtained by

$$f_c = 1 - \frac{1}{\kappa - 1}, \quad (3.7)$$

without any further computation involving the system structure. From a survival signature point of view, this means that if more than a fraction of components f_c has failed, there is only

a negligible probability that the system is functional:

$$\sum_{k=1}^K l_k < (1 - f_c) \cdot \sum_{k=1}^K m_k \Rightarrow \Phi(\underline{l}) \approx 0. \quad (3.8)$$

This way the computation of all entries in the survival signature below that threshold can be omitted.

Equations (3.6) and (3.8) have also been independently developed by Cohen et al. [132].

3.4 Approximation of the survival signature

This section presents the developed method to estimate the survival signature. After application of the percolation theory the remainder of the survival signature is approximated using Monte Carlo simulation. The proposed technique is then applied to simple benchmarking examples in order to quantify the errors resulting from neglecting entries and approximation. A more complex numerical example is presented in the subsequent section.

The method is divided into two steps:

- (1) Identify the area to neglect based on the critical percolation threshold (Eq. 3.8) and set all entries to 0.
- (2) Use Monte Carlo simulation to approximate the remaining entries, effectively replacing the full combinatorial calculation (see Eq. 3.2) with a sampling approach.

The algorithm to approximate a single unknown survival signature entry $\Phi(\underline{l})$ is as follows. As long as neither a pre-selected maximum number of samples N or a target coefficient of variation C is reached, generate a random network state for the state vector \underline{l} denoted by s and increase the number of samples $n_{\underline{l}}$ by one. A simple way of choosing a random network state for a state vector is to randomly shuffle the components of type k and choose the first l_k for each $k = 1, \dots, K$.

Consider a system with 10 components divided into two types as $s_1 = [1, 3, 5, 7, 9]$ and $s_2 = [2, 4, 6, 8, 10]$. In order to generate a random system for an example state vector $\underline{l} = [3, 3]$ the component vectors are randomly permuted, e.g. $s_1 = [3, 7, 5, 9, 1]$ and $s_2 = [2, 10, 6, 4, 8]$. Selecting and merging subarrays of lengths l_1 and l_2 from s_1 and s_2 respectively, results in the random network state $s = [3, 7, 5, 2, 10, 6]$.

If the structure function evaluates the random network state as functioning, i.e. $\varphi(s) = 1$, increase the counter $w_{\underline{l}}$ by one. Next, update the approximation using

$$\Phi(\underline{l}) \approx \frac{w_{\underline{l}}}{n_{\underline{l}}} \quad (3.9)$$

and the current coefficient of variation by

$$c_l = \frac{\sqrt{(\Phi(l) - \Phi(l)^2)/n_l}}{\Phi(l)}. \quad (3.10)$$

A pseudo-code implementation of the algorithm is presented in Algorithm 2. Note, that entries where the number of possible network states is smaller than N are calculated analytically through Eq. 3.2.

Eq. 3.10 is based on the coefficient of variation definition for a standard Monte Carlo simulation with continuous random variables. However, since the survival signature approximation involves a very large number of possible combinations, and entries with only a small number of combinations are always calculated exactly, it can be applied here as well.

The complete algorithm has been implemented in the Julia package **SurvivalSignature.jl** and made publicly available on Github [99].

Pseudocode 2 Approximate survival signature entry

```

function APPROXIMATE( $l, \varphi, N, C$ )
     $c, n, w, \Phi \leftarrow 0$  ▷ Initialise variables
    while  $c > C$  and  $n \leq N$  do
         $n \leftarrow n + 1$ 
         $s \leftarrow$  random network state for  $l$ 
        if  $\varphi(s) = 1$  then
             $w \leftarrow w + 1$ 
        end if
         $\Phi \leftarrow w/n$ 
         $c \leftarrow \frac{\sqrt{(\Phi - \Phi^2)/n}}{\Phi}$ 
    end while
    return  $\Phi, c$  ▷ Signature entry and coefficient of variation
end function

```

To prove the suitability of the method and quantify the error resulting from the approximation, the proposed technique is applied to several example networks of varying sizes. The networks used are simple $n \times m$ grid networks of the form shown in Fig. 3.2. The nodes distributed among two types in such a way, that any component is only connected to components of the other type. Figure 3.3 shows the convergence of the approximation of $\Phi(14, 14)$ with increasing sample size for the 6×6 network. The associated mean squared errors are presented in Figure 3.4. Next, the algorithm is applied to the full networks in order to understand the resulting errors and their effect on a reliability analysis using the approximated survival signature.

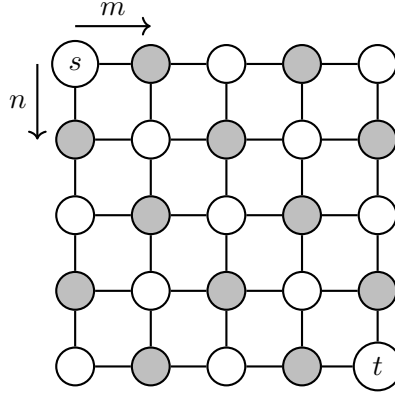


Figure 3.2: Simple grid network of size 5×5 with 25 nodes and 40 edges. The nodes labeled s and t are the source and target nodes.

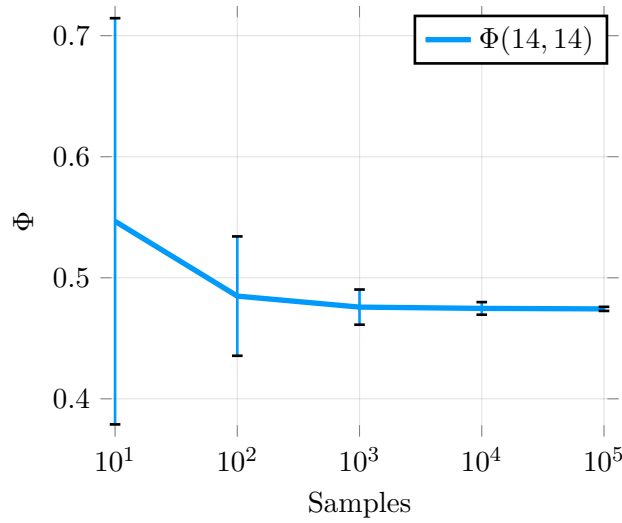


Figure 3.3: Convergence of the approximated signature entry $\Phi(14, 14)$ for the 6×6 network to the exact value of 0.4741594. The number of state vector evaluations needed for the exact solution is 9 363 600.

3.4.1 Percolation

In a first step, the exact survival signature for all networks is computed using the full combinatorial approach so that the approximation error can be calculated. Next, the critical threshold f_c of the networks is estimated using Eq. 3.7. For each network, a second survival signature denoted by Φ_{f_c} is created by copying all entries from the exact signature in line with Eq. 3.8. Then, the absolute error

$$E_{f_c} = \|\Phi - \Phi_{f_c}\|_F \quad (3.11)$$

and relative error

$$\tilde{E}_{f_c} = \frac{E_{f_c}}{\|\Phi\|_F} \quad (3.12)$$

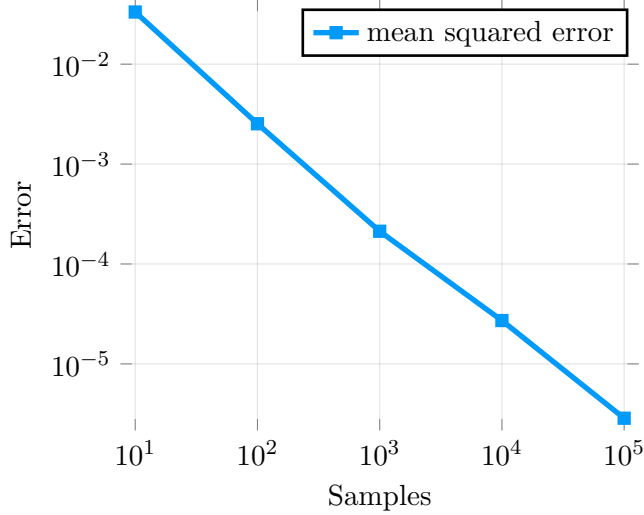


Figure 3.4: Evolution of the mean squared error when approximating the survival signature entry $\Phi(14, 14)$ for the 6×6 network with increasing sample size.

made by excluding the entries below the threshold is calculated. The results are presented in Tab. 3.2. It can be seen that the f_c slowly increases with network size which is also represented in the error. For example, the relative error made by excluding 120 entries from the survival signature of the 6×6 network is already less than 0.01 %.

$n \times m$	f_c	n	N	E_{f_c}	\tilde{E}_{f_c}
5×5	0.574468	66	182	0.000761789	0.000209180
5×6	0.584746	91	256	0.000524625	0.000129065
6×6	0.594595	120	361	0.000356815	0.000068137

Table 3.2: Critical thresholds and errors made by percolation of the grid networks. The number of neglected entries based on the critical threshold f_c is denoted by n . N represents the total number of entries in the survival signature.

Percolation is usually applied to large-scale networks. However, in the context of this work, the application is restricted so smaller networks where the computation of the analytical survival signature is still possible in order to be able to quantify the errors made by the exclusion of low interest system configurations. The effect of applying percolation to the survival signature of large-scale networks should be studied more closely in the future.

3.4.2 Monte Carlo Approximation

After applying the percolation based selection criterion, the entries of the survival signature not excluded by the critical percolation threshold are approximated using Monte Carlo simulation (see Section 3.4). The maximum number of samples used for each entry of the survival signature is increased with the network size to reach comparable levels of accuracy. As stated before, the

absolute error

$$E_{mc} = \|\Phi_{f_c} - \Phi_{mc}\|_F \quad (3.13)$$

and relative error

$$\tilde{E}_{mc} = \frac{E_{mc}}{\|\Phi_{f_c}\|_F} \quad (3.14)$$

introduced by the simulation-based approximation are calculated. Note that Φ_{mc} is compared to Φ_{f_c} instead of the exact survival signature Φ when quantifying the error. This is to ensure that estimated error results only from the Monte Carlo approximation. The results presented in Tab. 3.3 show that good accuracy can be reached with a reasonable amount of samples per entry. For example, using a maximum number of 10^4 samples for the 6×6 network leads to a relative error of less than 1%.

The number of structure function evaluations for this simulation is $n_\varphi = 2\,099\,933$ compared to the 2^{36} needed for the exact computation of the survival signature, a reduction of more than 99%. This great reduction in numerical effort also reflects in the computation time. The simulation takes less than 3 minutes using a single process where the exact computation took 18 hours highly parallelised on 64 processes.

$n \times m$	n	n_φ	E_{mc}	\tilde{E}_{mc}
5×5	2^{25}	399256	0.0207	0.0056
5×6	2^{30}	862400	0.0347	0.0085
6×6	2^{36}	2099933	0.0451	0.0086

Table 3.3: Absolute and relative errors made by approximation of the survival signature. The number of structure function evaluations required for the analytical solution is denoted by n while the number of evaluations required for the approximation is denoted by n_φ .

3.4.3 Reliability analysis

Finally, the exact and approximated survival signatures are applied to a network reliability analysis using the analytical solution of the survival function, see Eq. 3.3 and Eq. 3.4.

Let the failure times for components of type 1 have an Exponential distribution with $\lambda = 1$ and the failure times of components of type 2 have a Weibull distribution with $k = 2$ and $\lambda = 1$. The survival functions of the 6×6 network using the exact and approximated signatures are presented in Fig. 3.5. The plot clearly shows how the survival function P_{mc} using the approximated survival signature Φ_{mc} matches the one resulting from the exact signature. The relative error defined by $\tilde{E}_P = \|P - P_{mc}\|_F / \|P\|_F$ is approximately 0.13% in this case. The absolute and relative errors for the survival functions of all three test networks are shown in Tab. 3.4.

$n \times m$	E_P	\tilde{E}_P
5×5	0.0152	0.0011
5×6	0.0235	0.0015
6×6	0.0212	0.0013

Table 3.4: Absolute and relative errors resulting from a reliability analysis using the approximated survival signatures.

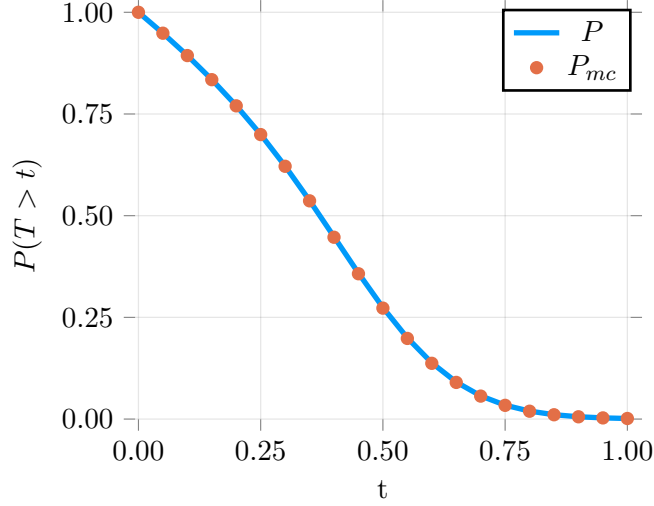


Figure 3.5: Survival functions of the 6×6 network. The approximated reliability of the network denoted by P_{mc} closely matches the analytical solution P .

3.5 Numerical examples

This section presents the application of the developed methodology to more involved, real world examples in comparison to the simple grid networks used in Section 3.4.

3.5.1 Example 1: Electricity transmission network

The first example used is a representative model of the electricity transmission network of Great Britain as presented in [133]. The network consists of 29 nodes that are split into two component types based on their bus type. Load buses are assigned component type 1, while voltage controlled buses are grouped in component type 2. The network’s slack bus (node 27) has not been separated into its own type to reduce the complexity of the problem and allow for relatively fast computation of the exact signature for comparison with the approximations. Note, that this is not an attempt at solving the underlying power flow problem but only a computation of the survival signature of the provided network topology. The network topology is displayed in Fig. 3.6.

A structure function is required in order to calculate the survival signature of the power network. The existence of s-t-connectivity as applicable for reliability block diagrams, shown in

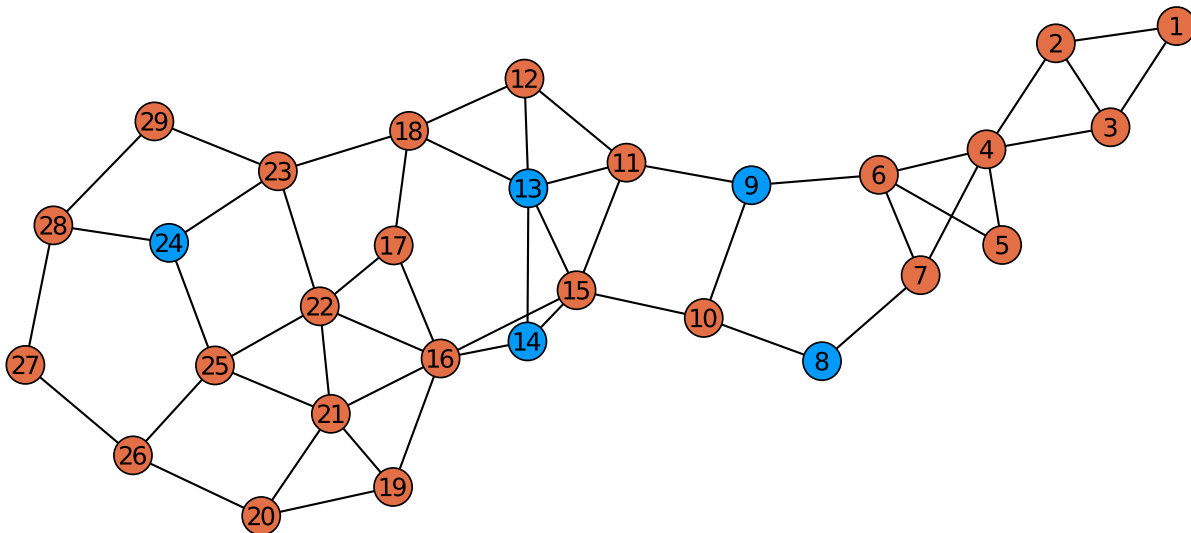


Figure 3.6: Topology of the electricity transmission network used in example 1. Components of type 1 are shown in blue. Components of type 2 are shown in orange. The locations of the nodes in the figure are not related to their actual geographical location.

Section 3.4, has little meaning for this network. Instead, the so called network efficiency [134] as defined in Eq. 3.15 is used to measure the state of the network for a given state vector.

$$E(G) = \frac{1}{n(n-1)} \sum_{i \neq j \in G} \frac{1}{d(i,j)} \quad (3.15)$$

where G is the network with n nodes and $d(i, j)$ denotes the length of the shortest path between two nodes i and j .

To define the (binary) structure function it is assumed that the network completely collapses once the loss of efficiency due to failing components exceeds 50% as

$$\varphi(\underline{x}) = \frac{E(G(\underline{x}))}{E(G)} < 0.5. \quad (3.16)$$

This value has been arbitrarily chosen for this example and implies no real world relevance. It should be noted, that this threshold is not related to the percolation threshold imposed in the first step of the algorithm.

Since this requires to compute the shortest paths between all components for every evaluated system configuration it is significantly more numerically demanding than the s-t-connectivity structure function used previously. An efficient algorithm to calculate the shortest paths is the Floyd-Warshall algorithm [135].

Survival signature approximations are performed with increasing sample sizes and compared to the exact solution to prove convergence. Figure 3.7 presents the mean squared errors and associated standard deviations resulting from 1000 repeated evaluations. A target coefficient of variation of 0.001 was used for all simulations. As evident from the plot, using a sample size of

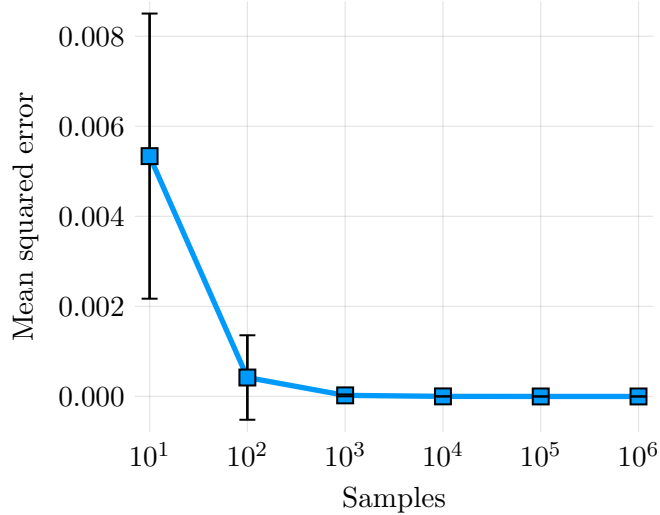


Figure 3.7: Evolution of the mean squared error resulting from approximation of the survival signature for the network shown in Fig. 3.6 with increasing number of samples used.

$1e4$ per survival signature entry already results in an adequately low error of $\approx 1.9e-6$. On a single processor core running at a clock speed of 3600 MHz the Monte Carlo approximation runs 509 s instead of 11 069 s for the full combinatorial evaluation.

3.5.2 Example 2: Berlin metro system

In this second example, the developed method is applied to the model of Berlin’s metro system taken from [72]. The model represents Berlin’s U-Bahn and S-Bahn systems which, due to the large number of interconnections between the systems, will be considered as a single system. The entire network consists of 306 nodes and 350 edges with nodes separated into two types based on their degree. Nodes with a maximum degree of two are grouped in type 1. Nodes with a degree larger than two are separated into type 2. This results in 245 nodes of type 1 and 61 of type 2. The topology of the network is presented in Fig. 3.8. The same structure function as in example 1 is applied to the network, see Eq. 3.16.

The full survival signature of this system has 15252 entries, the most numerically demanding entries being $\Phi(122, 30)$, $\Phi(122, 31)$, $\Phi(123, 30)$ and $\Phi(123, 31)$ each with a total number of approximately 6.69×10^{89} possible combinations to be evaluated. To put this into perspective, these are more combinations for a single entry than the estimated number of atoms in the observable universe ($\approx 10^{80}$). Therefore, calculating the analytical survival signature for this network is impossible using the traditional approaches on present day computers. However, it can be approximated using the presented simulation based technique.

The previous example has shown, that using $1e4$ samples per signature entry leads to sufficiently accurate results as evident from Fig. 3.7. The approximation of a single survival signature is essentially the same as computing a probability of failure. The minimum number

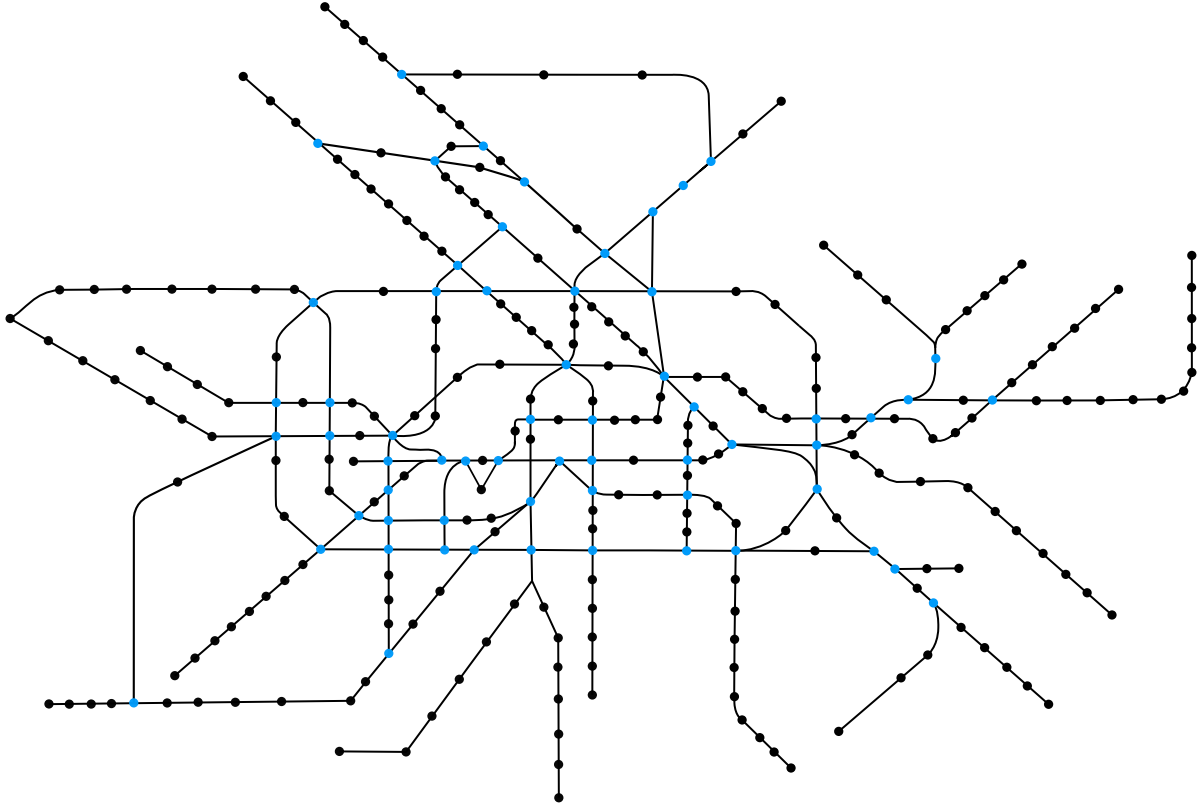


Figure 3.8: Topology of the Berlin metro system with 306 nodes. Nodes highlighted in blue represent stations with more than two connections. Adapted from [72].

of samples N required to estimate a probability of failure or in this case survival signature entry $\Phi(l)$ is defined by

$$N \leq \frac{1}{c^2 \cdot \Phi(l)}, \quad (3.17)$$

where c is the desired coefficient of variation [21]. This shows that the per entry accuracy of the survival signature approximation using a certain number of samples is the same regardless of network size. Based on this, $1e4$ samples are chosen to compute the survival signature of the metro system.

The approximation of the survival signature is still numerically demanding, running for 27 h 39 min 35 s using 64 threads on an AMD Ryzen Threadripper 3990X 64-Core Processor. Since no analytical solution is available for this system, the final error resulting from the approximation can not be quantified.

While the results show a significant reduction in numerical demand in comparison to the full combinatorial evaluation, it highlights the demand for more efficient sampling strategy. With increasing size and dimensions of the survival signature, brute force Monte Carlo simulation will still not be sufficient. Potentially, advanced Monte Carlo methods such as line sampling [136] or subset simulation [137] could drastically improve efficiency.

3.6 Conclusion

This paper presented a new approach to the approximation of the survival signature. The developed technique significantly outperforms traditional (full combinatorial) approaches in the number of required structure function evaluations and therefore in overall computation time. At the same time, the resulting approximation errors are sufficiently low enough for an accurate reliability analysis.

Both the application of percolation theory to reduce the number of computed signature entries and the Monte Carlo simulation require no additional information on the system other than the structure function. This allows the method to be applied to any problem where the numerical demand prohibits the full evaluation of the survival signature.

The viability of the new method is proved by comparing simple toy examples to their exact solutions and quantifying the errors. More complex and demanding numerical examples are used to show application to real world problems.

However, since the number of samples required for an accurate Monte Carlo approximation increases with growing network size and number of component types this approach will also reach its limits at some point. To analyse even larger and more complex systems the method must be extended to apply advanced simulation techniques such as line sampling or subset simulation in order to reduce the number of samples required to compute an entry of the survival signature.

The developed method is implemented in an open source Julia library and made available to fellow researchers.

CRedit authorship contribution statement

Jasper Behrendorf: Conceptualization, Methodology, Software, Writing – original draft, Visualization. **Tobias-Emanuel Regenhardt:** Conceptualization, Methodology, Writing – original draft. **Matteo Broggi:** Conceptualization, Methodology, Writing – review & editing. **Michael Beer:** Conceptualization, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The research work herein was supported by the German Research Foundation under Grant No. BE 2570/3–1 and BR 5446/1–1. This support is gratefully acknowledged.

4 | Imprecise Survival Signature Approximation Using Interval Predictor Models

Imprecise Survival Signature Approximation Using Interval Predictor Models

Jasper Behrendorf^a, Matteo Broggi^a, Michael Beer^{a,b,c}

^aInstitute for Risk and Reliability, Leibniz Universität Hannover, Hannover, Germany

^bInstitute for Risk and Uncertainty, University of Liverpool, Liverpool, United Kingdom

^cInternational Joint Research Center for Resilient Infrastructure & International Joint Research Center for Engineering Reliability and Stochastic Mechanics, Tongji University, Shanghai, China

Submitted to the *2023 IEEE Symposium Series on Computational Intelligence* on July 17, 2023

Abstract

This paper presents a novel technique for the approximation of the survival signature for very large systems. In recent years, the survival signature has seen promising applications for the reliability analysis of critical infrastructures. It outperforms traditional techniques by allowing for complex modelling of dependencies, common causes of failures and imprecision. However, as an inherently combinatorial method, the survival signature suffers greatly from the curse of dimensionality. Computation for very large systems, as needed for critical infrastructures, is mostly infeasible. New advancements have applied Monte Carlo simulation to approximate the signature instead of performing a full evaluation. This allows for significantly larger systems to be considered. Unfortunately, these approaches will also quickly reach their limits with growing network size and complexity. In this work, instead of approximating the full survival signature, we will strategically select key values of the signature to accurately approximate it using a surrogate radial basis function network. This surrogate model is then extended to an interval predictor model (IPM) to account for the uncertainty in the prediction of the remaining unknown values. In contrast to standard models, IPMs return an interval bounding the survival signature entry. The resulting imprecise survival signature is then fed into the reliability analysis, yielding upper and lower bounds on the reliability of the system. This new method provides a significant reduction in numerical effort enabling the analysis of larger systems where the required computational demand was previously prohibitive.

Keywords: Survival signature, interval predictor models, radial basis function networks

4.1 Introduction

The reliability analysis of critical infrastructure systems, such as electrical, gas and water distribution systems, traffic networks and communication networks is of paramount importance to the safety and development of modern societies. Our increasing dependence on the availability of these systems only escalates this fact. The analysis and assessment of reliability comes with increasingly higher computational demand due to the growing size and complexity of the infrastructure systems. In addition, phenomena such as dependencies inside or between these networks can have adverse effects and must not be disregarded [42]. Unfortunately, this is where traditional approaches such as fault tree analysis or reliability block diagrams quickly reach their limits. A modern development in system analysis built to circumvent these drawbacks is the survival signature [27].

The survival signature was introduced as a generalization of the system signature [121] to systems with multiple component types. It excels in particular where diverse effects such as dependencies, common causes of failure or imprecision need to be included in the reliability analysis. Through the separation of the system structure from probabilistic information about component failures it provides a flexible method where a variety of scenarios can be studied without unnecessary reevaluation of the system structure.

However, like the traditional methodologies, the survival signature suffers greatly from the curse of dimensionality. As a result, the numerical demand to compute the survival signature increases with non-polynomial growth with increasing network size and number of component types. As the number of components in critical infrastructure systems can quickly reach upwards of hundreds or thousands this computational demand to obtain the survival signature becomes prohibitive.

A number of promising solutions to this problem have been proposed over the recent years. These involve for example binary decision diagrams (BDD) [38], extended universal generating functions (UFG) [40] or Monte Carlo simulation [41, 94]. However, obtaining alternative system representations such as the BDD or UFG are non-trivial problems themselves, while standard Monte Carlo simulation will also quickly reach the limits of its feasibility for larger systems.

In this paper we propose a new method based on building an accurate surrogate model for the survival signature of large systems. In a first step, an adaptive strategy is employed to select which values of the survival signature to compute as data points for the surrogate model. These values are computed using the existing Monte Carlo method [94]. Then, a normalized radial basis function network is constructed from these data points to approximate the remaining values of the survival signature. Finally, the uncertainty resulting from the Monte Carlo simulation is used to extend the surrogate model to an interval predictor model (IPM). The uncertainty is propagated through the IPM which ultimately yields bounds on the survival signature.

The remainder of the paper is structured as follows. Section 4.2 presents the survival signature while Section 4.3 discusses the radial basis function based surrogate models. The newly proposed methodology is introduced in Section 4.4 followed by an application of the technique to a numerical example in Section 4.5. The paper closes with some concluding remarks in Section 4.6.

4.2 Survival Signature

Consider a system with K different component types and m_k components of each type $k \in \{1, 2, \dots, K\}$. The survival signature condenses the structural information of the system into a

probability that the system is working for l_k out of m_k components working per type as

$$\Phi(l_1, \dots, l_K) = \left[\prod_{k=1}^K \binom{m_k}{l_k}^{-1} \right] \times \sum_{\mathbf{x} \in S_{l_1, \dots, l_K}} \varphi(\mathbf{x}), \quad (4.1)$$

where S_{l_1, \dots, l_K} denotes the set of all state vectors of the system with $l_1 \dots, l_K$ working components and $\varphi(\mathbf{x})$ is the structure function which evaluates to 1 if the system is working for a given state vector and 0 if it is not.

A state vector $\mathbf{x} = (x_1, x_2, \dots, x_n)$ is the representation of a distinct system state where $x_i = 0$ indicates a component is failed and $x_i = 1$ indicates a working component for all $i \in \{1, 2, \dots, n\}$ components of the system and $n = \sum_{k=1}^K m_k$.

The survival signature is defined for coherent systems, for which it is monotonically non-decreasing. The monotone behavior can be expressed as

$$\begin{aligned} \Phi(l_1^a, \dots, l_K^a) \leq \Phi(l_1^b, \dots, l_K^b), \text{ if} \\ l_k^a \leq l_k^b, \forall k \in \{1, \dots, K\}, \end{aligned} \quad (4.2)$$

where the superscripts a and b refer to any two inputs of the survival signature. This monotonicity property is exploited later to design a monotone radial basis function network as a surrogate model.

The survival function, that is the reliability of the system at time t , is defined as

$$P(T_s > t) = \sum_{l_1=0}^{m_1} \dots \sum_{l_K=0}^{m_K} \Phi(l_1, \dots, l_K) P\left(\bigcap_{k=1}^K \{C_t^k = l_k\}\right), \quad (4.3)$$

where C_t^k denotes the number of components of type k functioning at time t . For known failure time distributions with cumulative distribution functions (CDF) $F_k(t)$ the probabilistic part of the survival function can be analytically computed as

$$P\left(\bigcap_{k=1}^K \{C_t^k = l_k\}\right) = \prod_{k=1}^K \binom{m_k}{l_k} [F_k(t)]^{m_k - l_k} [1 - F_k(t)]^{l_k}. \quad (4.4)$$

4.3 Monotone Radial Basis Function Networks

Radial basis function (RBF) networks [138] are a very simple type of artificial neural network with only one hidden layer, where the activation functions are radial basis functions, and a linear output layer. Radial basis functions in \mathbb{R}^d are defined with respect to a distance function between the input and a fixed point called a center. This distance is usually the Euclidean

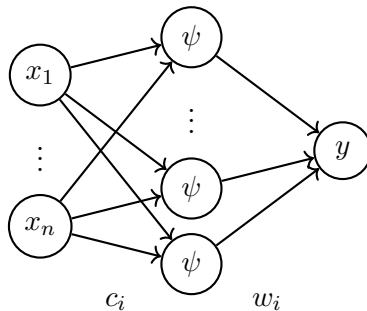


Figure 4.1: Graphical representation of a radial basis function neural network. From left to right: input layer, hidden layer, linear output layer.

distance defined as

$$\|x - c\| = \sqrt{\sum_{j=1}^d \frac{(x_j - c_j)^2}{2\sigma_j^2}}, \quad (4.5)$$

where $x \in \mathbb{R}^d$ is the input vector and the σ_j are normalization constants controlling the spread of the radial basis function in each dimension. The most common radial basis function is the Gaussian function

$$\psi(\|x - c\|) = e^{-\|x - c\|^2}. \quad (4.6)$$

Using the basis functions we can define the RBF network as

$$y(x) = \sum_{i=1}^N w_i \psi(\|x - c_i\|), \quad (4.7)$$

where N is the number of neurons, c_i is the center point associated with neuron i and w_i is the weight of that neuron in the linear output layer. Refer to Figure 4.1 for a simple graphical representation of an RBF network.

The output of the radial basis function network can be normalized as

$$y(x) = \frac{\sum_{i=1}^N w_i \psi(\|x - c_i\|)}{\sum_{i=1}^N \psi(\|x - c_i\|)}. \quad (4.8)$$

Normalized radial basis function (NRBF) networks have shown to improve extrapolation of the network when compared to regular RBF networks [139]. For the remainder of this paper, the basis functions used will be normalized as

$$\hat{\psi}_i(x) = \frac{\psi(\|x - c_i\|)}{\sum_{i=1}^N \psi(\|x - c_i\|)}. \quad (4.9)$$

In this work, we restrict the center locations of the NRBF network to a regular uniform grid over the entire input domain. This allows for easier selection of the σ_j shape parameters as well as greatly simplifying the monotonicity constraints. In the next step, the NRBF network will

be used as a surrogate model of the survival signature. Enforcing monotonicity of the surrogate model greatly improves its performance since the survival signature itself is a monotone function.

Monotonicity of an NRBF network can be enforced through simple linear constraints on the weights. For centers located on a grid, only sequences of weights lying on a line in the direction of the d dimensions need to be monotonic for the NRBF network to be monotonic [140].

Assuming M data points x_j with associated function values y_j for $j \in \{1, \dots, M\}$, the optimal weights w_i are determined as the solution to the following constrained least squares optimization problem

$$\begin{aligned} & \underset{w}{\text{minimize}} && \|\Psi w - y\|_2^2 \\ & \text{subject to} && w_p \leq w_q, \quad p, q \in \{1, \dots, N\}, \\ & && \text{if } c_p^\alpha < c_q^\alpha, c_p^\beta = c_q^\beta, \alpha \neq \beta, \end{aligned} \tag{4.10}$$

where the superscripts $\alpha, \beta \in \{1, \dots, d\}$ refer to the coordinates in dimensions α and β with $w = [w_1, \dots, w_N]^T$ and $y = [y_1, \dots, y_M]^T$. The matrix $\Psi \in \mathbb{R}^{M \times N}$ is defined as

$$\Psi_{ji} = \hat{\psi}_i(x_j). \tag{4.11}$$

This is a convex optimization problem which can be solved by standard solvers. Here we use the splitting cone solver (SCS) [141] in connection with JuMP [142].

4.4 Approximation of the Survival Signature

This section presents the proposed methodology to the approximation of the survival signature for very large systems. Equation (4.1) clearly shows that the computation of the survival signature is a combinatorial problem which greatly increases in numerical demand with increasing number of components and component types. Therefore, analytical computation of the survival signature is infeasible for very large systems. In [94] we presented an approach to approximating the survival signature using percolation to exclude negligible parts of the signature and approximating the remaining entries with Monte Carlo simulation. While this enabled the analysis of larger systems, this method still requires a significant number of system evaluations and will quickly reach prohibitive computational demand. In this paper, we propose an alternative solution where the survival signature is approximated through a radial basis function network based on a few strategically selected values. In addition, the uncertainty in the survival signature entries resulting from the Monte Carlo approximation is propagated through an interval predictor model and reflected as bounds on the survival signature.

4.4.1 Design of the NRBF network surrogate model

Similarly to the previously developed technique, this new approach starts by applying percolation theory to find the critical threshold f_c and exclude a significant portion of the survival signature

where the probability that the system functions is negligible, i.e. close to zero [94].

Next, let Ω be the set of remaining survival signature entries l_1, \dots, l_K for $l_k \in \{0, 1, \dots, m_k\}$ and $k \in \{1, \dots, K\}$ satisfying the condition

$$\sum_{k=1}^K l_k \geq (1 - f_c) \sum_{k=1}^K m_k. \quad (4.12)$$

We start by selecting M initial entries from Ω and calculate their survival signature values using Monte Carlo simulation. Note that values that require less than the number of Monte Carlo samples N_{MC} structure function evaluations are computed exactly. In order to have good coverage of the input space, we generate the M data points by creating a uniform grid between the lower and upper bounds of Ω and then select the nearest neighbors in Ω as the starting values. In addition, we create a uniform grid of centers c_i for $i = 1, \dots, N$ spread over the entire domain of Ω . The spread of the radial basis function σ_j is chosen as half of the distance between two successive centers in the j th dimension. These have shown to be a good starting point for NRBF networks [140]. Then, the weights for the initial monotone NRBF approximation of the survival signature are obtained by solving the optimization problem (4.10).

In the next step, new values of the survival signature to be approximated are selected and used to improve the surrogate model. This is performed by means of a Taylor expansion-based adaptive design (TEAD) [143]. The TEAD is a combination of exploration and exploitation based on nearest neighbor distance and Taylor expansion. This allows to both adequately cover the input space as needed as well as concentrate samples where the highest changes in the survival signature are located to provide an accurate surrogate model using a small sample size. At each point a new sample is selected from Ω and added to the M data points. Then the weights are updated by again solving the optimization problem (4.10). The search for new points is aborted once the change in weights falls below a defined threshold twice in a row.

4.4.2 Extension to an interval predictor model

Once an adequate surrogate model for the survival signature is obtained, it is extended to an interval predictor model. In difference to regular models, where a single output value is returned for any inputs, interval predictor models return an interval where the value is predicted to fall [144]. The goal here is to propagate the uncertainty of using Monte Carlo simulation to approximate individual entries of the survival signature through the surrogate model. By simply fitting a NRBF network to some estimated values of the survival signature the uncertainty of the Monte Carlo simulation represented by the coefficient of variation of the particular entries would be lost.

The IPM approach is simple. Instead of fitting a single surrogate model to the available data points we fit two models that act as upper and lower bounds. These models are built based on the points identified in the previous step. Both models use the exact same center locations c_i

and their difference in output is only influenced by two different sets of weights w_{\min} and w_{\max} . The bounds are defined as

$$\bar{y}(x, w_{\max}) = \Phi(x)^T w_{\max} \quad (4.13)$$

$$\underline{y}(x, w_{\min}) = \Phi(x)^T w_{\min}, \quad (4.14)$$

where $\Psi(x) = [\hat{\psi}_1(x), \dots, \hat{\psi}_N(x)]$.

The spread of the IPM, i.e. the separation between its limits, is defined as

$$\delta_y(x, w_{\max}, w_{\min}) = \Phi(x)^T (w_{\max} - w_{\min}). \quad (4.15)$$

The parameters w_{\max} and w_{\min} are given by the following optimization problem

$$\begin{aligned} & \underset{w_{\max}, w_{\min}}{\text{minimize}} && \frac{1}{N} \sum_j^N \delta_y(x_j, w_{\max}, w_{\min}) \\ & \text{subject to} && \underline{y}(x_j, w_{\min}) \leq \underline{y}_j \\ & && \bar{y}(x_j, w_{\max}) \geq \bar{y}_j \\ & && w_{\min} \leq w_{\max}, \end{aligned} \quad (4.16)$$

where $(x_j, \underline{y}_j, \bar{y}_j)$ for $j \in \{1, \dots, M\}$ are the data points. The bounds \underline{y}_j and \bar{y}_j are obtained as

$$\underline{y}_j = y_j - \gamma \cdot y_j \cdot cv_j \quad (4.17)$$

$$\bar{y}_j = y_j + \gamma \cdot y_j \cdot cv_j, \quad (4.18)$$

where cv_j is the coefficient of variation of the Monte Carlo approximation for y_j . The parameter $\gamma \geq 1$ can be adapted to obtain more conservative bounds. In addition to the constraints in (4.16) we also invoke the monotonicity constraints of (4.10) on w_{\max} and w_{\min} in order to ensure monotonicity of the IPM bounds.

Once the IPM is obtained by solving the optimization problem (4.16), the bounds on the survival signature can be fed into (4.3) to estimate the bounds of the imprecise survival function. Since bounds of the IPM are strictly monotone functions, it is sufficient to evaluate the survival function for the upper and lower bound. If the CDFs of the component failure times are also imprecise, the bounds on the reliability can be obtained by applying the vertex method [145].

4.5 Numerical Example

In this section we apply the proposed technique to a numerical example. The example used is that of the Berlin S- and U-Bahn system as presented in [94], see Fig. 4.2. The entire network consists of 306 nodes and 350 edges. The nodes are separated into two types based on their

degree, with type 1 consisting of nodes with degree of two or less, while type 2 contains all nodes with a degree larger than two. As a result, there are 245 nodes of type 1 and 61 of type 2. The full survival signature of this network has 15 252 entries. Let the network efficiency be defined as

$$E(G) = \frac{1}{n(n-1)} \sum_{u \neq v \in G} \frac{1}{d_{uv}}, \quad (4.19)$$

where G is the network consisting of n nodes and d_{uv} measures the length of the shortest path between nodes u and v [134]. Then, a structure function for the network analysis can be defined by the loss of efficiency for a given network state. Here we assume that the network fails to function once the loss of efficiency exceeds 50% as given by

$$\varphi(\mathbf{x}) = \frac{E(G(\mathbf{x}))}{E(G)} < 0.5. \quad (4.20)$$

Computing the full analytical survival signature given this structure function is impossible and even the Monte Carlo simulation requires more than 27 hours using 10 000 samples per entry distributed among 64 cores [94].

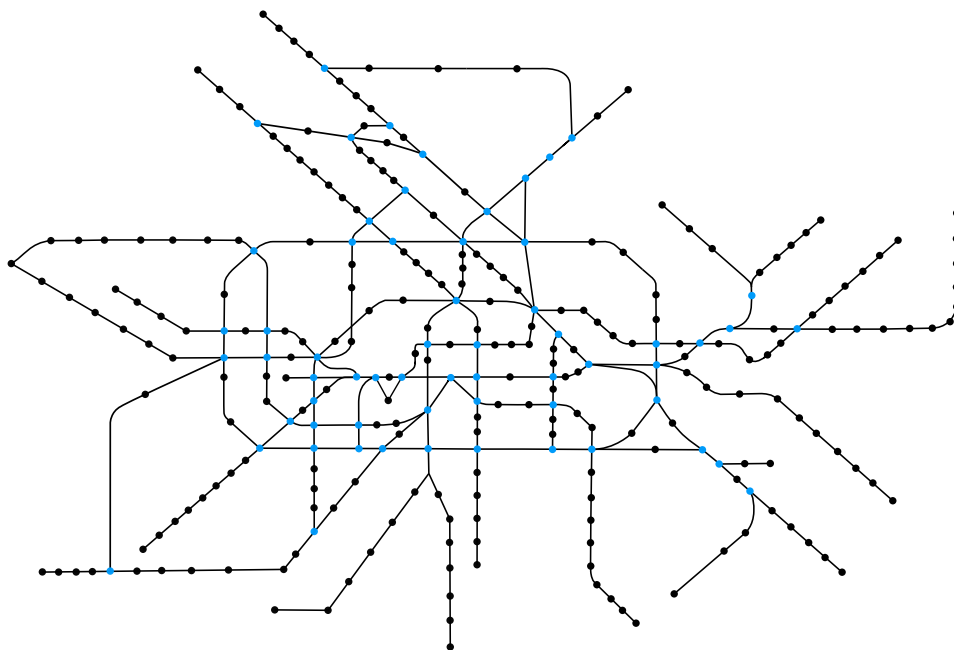


Figure 4.2: Topology of the Berlin metro system with 306 nodes. Nodes highlighted in blue represent stations with more than two connections. Adapted from [72].

The approximation of the survival signature starts by applying percolation to find the set of candidate points Ω . The critical threshold $f_c \approx 0.39550$ reduces Ω from 15 252 points down to 5673. 400 center points c_i are generated uniformly over the domain of Ω , of which 109 are excluded using (4.12), leaving 291 centers. Then, 25 initial points are selected from Ω . New points are chosen using the TEAD until the change in weights is less than $1e-3$ for two

consecutive points. The procedure aborts after 110 new points have been selected. The survival signature entries for all points are approximated using 10 000 Monte Carlo samples. Figure 4.3 shows the initial points as well as the new points chosen by the TEAD. The plot clearly presents how the new points selected by the adaptive technique are concentrated in the area of the survival signature with high relevance whereas only a few points are chosen where the survival signature is negligible. This ability to identify the important region of the signature allows to approximate it accurately using only a low number of evaluated entries.

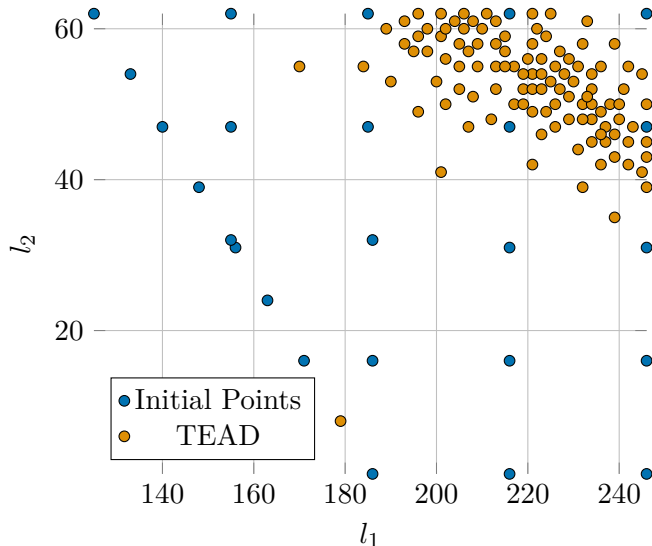


Figure 4.3: Initial data points and adaptive points selected by the TEAD.

In the final step, the coefficient of variation resulting from the approximation is used to define the upper and lower bounds of the evaluated survival signature entries and the weights for the IPM are obtained from the optimization (4.16).

We assume the failure times for component types 1 are exponentially distributed with $\lambda = 0.25$ while the failures of type 2 components are exponentially distributed with $\lambda = 0.5$. These values are arbitrarily chosen for illustrative purposes and have no real world relevance. The bounds on the resulting survival function are compared to the reliability of the full Monte Carlo approximation in Figure 4.4. For a closer look at the reliability bounds between $t = 0.25$ and $t = 0.5$ refer to Figure 4.5. As evident from the plots, the IPM is able to accurately predict upper and lower bounds on the reliability. This accuracy is especially remarkable due to the fact that it only requires the evaluation of 135 out of the 5673 survival signature values necessary for the full Monte Carlo approach. This is a significant reduction in numerical effort and enables the analysis of even larger networks than using the Monte Carlo method alone.

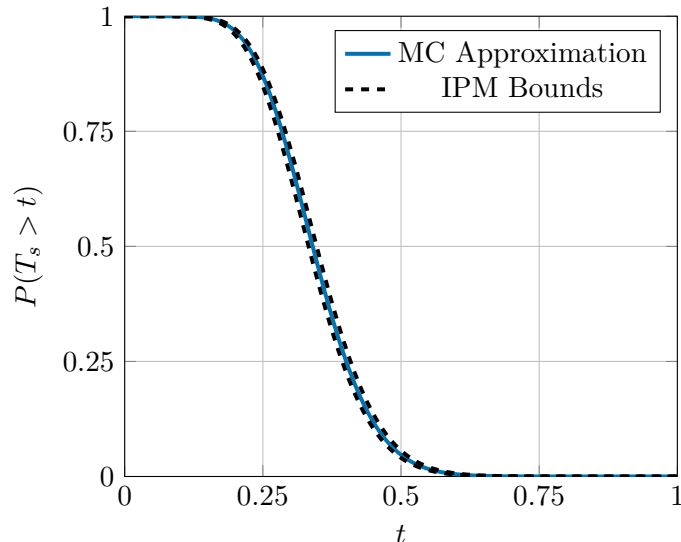


Figure 4.4: Bounds of the IPM surrogate model compared to the NRBF approximation.

4.6 Conclusions

This paper presents a novel approach to the approximation of the survival signature using normalized radial basis function networks and interval predictor models as surrogate models. The NRBF networks are able to accurately predict the entire survival signature given only a small fraction of the evaluated signature entries. The uncertainty resulting from using Monte Carlo simulation to approximate individual entries is efficiently propagated through the IPM to yield bounds on the survival signature. An adaptive procedure to select the data points for the surrogate model ensures high accuracy while keeping the numerical demand low. The effectivity of the developed method was proven by applying it to the analysis of a large system and comparing the results with a previously presented technique. The code developed as part of this work will be added to the open source Julia package *SurvivalSignature.jl* [99].

Future effort will be focused on improving the NRBF fit. While the number of centers and parameters controlling the spread of the basis functions chosen in this work serve as a good basis, optimizing these parameters to the specific problem could prove beneficial. Furthermore, the method should be applied to even larger systems with more than two component types to investigate the scalability.

By extending this method to include imprecise probabilities for the failure time distributions of the components such as probability boxes, a fully imprecise survival function can be devised.

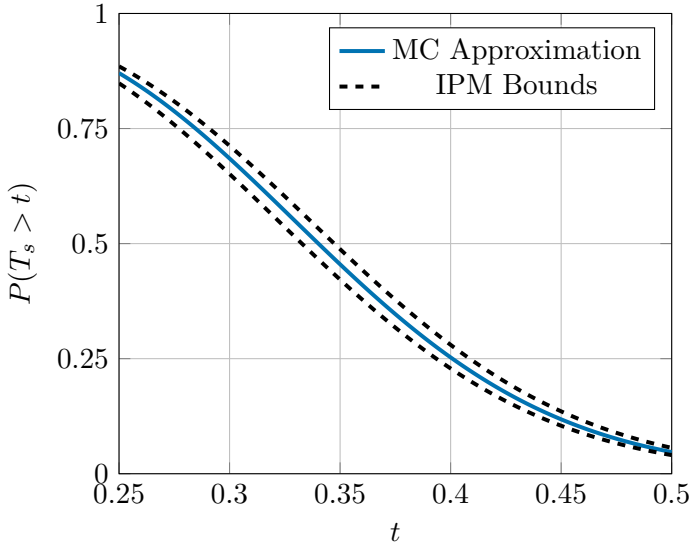


Figure 4.5: Bounds of the IPM surrogate model compared to the NRBF approximation between $t = 0.25$ and $t = 0.5$.

5 | Interval Predictor Model for the Survival Signature Using Monotone Radial Basis Functions

Interval Predictor Model for the Survival Signature Using Monotone Radial Basis Functions

Jasper Behrendorf^a, Matteo Broggi^a, Michael Beer^{a,b,c}

^aInstitute for Risk and Reliability, Leibniz Universität Hannover, Hannover, Germany

^bInstitute for Risk and Uncertainty, University of Liverpool, Liverpool, United Kingdom

^cInternational Joint Research Center for Resilient Infrastructure & International Joint Research Center for Engineering Reliability and Stochastic Mechanics, Tongji University, Shanghai, China

Submitted to *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems Part A: Civil Engineering* on August 25, 2023

Abstract

This research describes a novel method for approximating the survival signature for very large systems. Over the recent years, the survival signature has emerged as a capable tool for the reliability analysis of critical infrastructure systems. In comparison to traditional approaches it allows for complex modelling of dependencies, common causes of failures as well as imprecision. However, while it enables the consideration of these effects, as an inherently combinatorial method, the survival signature suffers greatly from the curse of dimensionality. Critical infrastructures typically involve upwards of hundreds of nodes. At this scale analytical computation of the survival signature is impossible using current computing capabilities. Instead of performing the full analytical computation of the survival signature some studies have focused on approximating it using Monte Carlo simulation. While this reduces the numerical demand and allows for larger systems to be analyzed, these approaches will also quickly reach their limits with growing network size and complexity. Here, instead of approximating the full survival signature, we build a surrogate model based on normalized radial basis functions where the data points required to fit the model are approximated by Monte Carlo simulation. The resulting uncertainty from the simulation is then used to build an interval predictor model (IPM) which estimates intervals where the remaining survival signature values are expected to fall. By applying this imprecise survival signature we can obtain bounds on the reliability. Because a low number of data points is sufficient to build the IPM this presents a significant reduction in numerical demand and allows for very large systems to be considered.

Keywords: Survival signature, Monte Carlo simulation, radial basis function network, interval predictor models, imprecise probabilities

5.1 Introduction

The reliability analysis of critical infrastructure systems, such as electricity, gas, and water distribution systems, transportation networks, and communication networks, is critical to modern society's safety and progress. Our growing reliance on the availability of these systems exacerbates this problem. The analysis of network reliability for these systems comes with increasingly higher computational demand due to their growing size and complexity. Additionally, complex phenomena such as dependencies inside or between these networks can lead to catastrophic failures and must be considered during the analysis [42]. Unfortunately, traditionally applied methodologies such as fault tree analysis or reliability block diagrams

are limited in their abilities to include these effects. The survival signature was developed to alleviate these shortcomings and enable a comprehensive reliability analysis [27].

The survival signature was introduced as a generalization of the system signature [121] to systems with multiple component types. A signature is a condensed representation of the system topology that allows efficient computation of the system reliability. Because of its separation of the system's structural information from the component failure times it allows for the consideration of a plethora of complex effects such as imprecision, common cause failures or dependencies. This provides a versatile method for studying a range of scenarios without having to reevaluate the system structure.

However, the survival signature, like traditional techniques, suffers heavily from the curse of dimensionality. The computational demand grows non-linearly with network size and number of component types and the computing burden to obtain the survival signature becomes prohibitive as the number of components in critical infrastructure systems rapidly approaches hundreds or thousands.

Research into solutions to this problem has seen considerable attention in recent years. Reed [38] and Reed et al. [39] proposed algorithms for the exact computation of the survival signature through binary decision diagrams (BDD). However, memory demand (RAM) limits these algorithms for larger systems or those with many component types. Another approach using reliability block diagrams and extended universal generating functions is similarly limited by memory requirements [40]. Estimation methods based on standard Monte Carlo simulation [94] and entropy-driven Monte Carlo simulation [41] have also been presented. While more efficient than the full evaluation of the survival signature these approaches also reach their limits with increasing network size. A recent development uses optimization and Monte Carlo simulation to efficiently estimate the two-terminal survival signature of networks with two component types [146].

In this paper we propose a new method where instead of using Monte Carlo simulation to estimate each entry of the survival signature, only a few strategically selected data points are computed using the existing Monte Carlo method [94] which are then used to construct a monotone normalized radial basis function network as a surrogate model of the full survival signature. In a second step, the surrogate model is extended to an interval predictor model (IPM) which effectively propagates the uncertainty associated with the Monte Carlo approximation of the data points through the model. In the end, we obtain upper and lower bounds of the survival signature and therefore the system reliability.

The remainder of the paper is structured as follows. We first present the survival signature followed by a discussion of (normalized) radial basis function based surrogate models and the theory of interval predictor models. The proposed methodology is then introduced and applied to a toy example followed by a larger numerical example to prove scalability of the method. The paper closes with concluding remarks and outlook into future research.

5.2 Survival Signature

The survival signature [27] was introduced as a generalization of the system signature [121] to multiple component types. It condenses the structural information of a system with K different component types into a probability that the system is functioning while l_k out of m_k components are in a functioning state for each type $k \in \{1, 2, \dots, K\}$. More precisely, it is defined as the percentage of all possible system states for l_k out of m_k working components that lead to a functioning system.

Let $\mathbf{x} \in \{0, 1\}^m$ be a state vector representing the state of all m components in the system with $m = \sum_{k=1}^K m_k$. An entry $x_i = 1$ denotes a functional component i while for $x_i = 0$ the component is non-functional or failed for all $i \in \{1, \dots, m\}$. Note that the labeling of components in the state vector is arbitrary but must be kept consistent. The global system state for any state vector can be evaluated through the structure function $\varphi : \{0, 1\}^m \rightarrow \{0, 1\}$, which evaluates to 1 if the system is working for a given state vector and 0 if it is not. The survival signature is then defined as

$$\Phi(l_1, \dots, l_K) = \left[\prod_{k=1}^K \binom{m_k}{l_k}^{-1} \right] \times \sum_{\mathbf{x} \in S_{l_1, \dots, l_K}} \varphi(\mathbf{x}), \quad (5.1)$$

where S_{l_1, \dots, l_K} denotes the set of all state vectors of the system with $l_1 \dots, l_K$ working components. The survival signature can therefore be represented as a K -dimensional array of size $(m_1 + 1) \times \dots \times (m_K + 1)$. This includes the cases with $l_k = 0$, where no components of type k are working. For the remainder of the paper we will refer to individual values in this K -dimensional array as survival signature *entries*.

The structure function depends on the system that is being analyzed. It can for example be defined as the s - t -connectivity of two source (s) and target (t) nodes in a system as used for the toy example to validate the proposed method. This is well known as the two-terminal reliability problem. For the real world numerical example in the penultimate section we define an alternative structure function based on the loss of network efficiency.

The survival signature is defined for coherent systems. In a coherent system, functionality does not decrease with growing number of working components. As such the survival signature is monotonically nondecreasing which can be expressed as

$$\begin{aligned} \Phi(l_1^a, \dots, l_K^a) &\leq \Phi(l_1^b, \dots, l_K^b), \text{ if} \\ l_k^a &\leq l_k^b, \forall k \in \{1, \dots, K\}, \end{aligned} \quad (5.2)$$

where the superscripts a and b refer to any two inputs of the survival signature. This information will help to obtain a better surrogate model by adding monotonicity constraints to the radial basis function networks in the next section.

Finally, the reliability of the system, that is the probability it survives up to a point in time

t , is defined as

$$P(T_s > t) = \sum_{l_1=0}^{m_1} \dots \sum_{l_K=0}^{m_K} \Phi(l_1, \dots, l_K) P\left(\bigcap_{k=1}^K \{C_t^k = l_k\}\right), \quad (5.3)$$

where C_t^k represents the number of components of type k functioning at time t . For known failure time distributions with cumulative distribution functions (CDF) $F_k(t)$ the probabilistic part of the survival function can be analytically computed as

$$P\left(\bigcap_{k=1}^K \{C_t^k = l_k\}\right) = \prod_{k=1}^K \binom{m_k}{l_k} [F_k(t)]^{m_k-l_k} [1 - F_k(t)]^{l_k}. \quad (5.4)$$

For cases where the CDFs are unknown, a number of simulation methods have been developed [37].

The structural information of the system, e.g. the survival signature, is completely separated from the probabilistic information on component failures. It is due to this separation that complex effects such as imprecision or dependent component failures can easily be included as they have no influence on the structural evaluation of the system. In case the failure times of the components are described as independent single parameter probability boxes (p-box) the upper and lower bounds of the system reliability $[\bar{P}, \underline{P}]$ can be obtained by feeding the upper and lower bounds of the CDFs $[\bar{F}_k, \underline{F}_k]$ into Eq. 5.4 [58]

$$\bar{P}\left(\bigcap_{k=1}^K \{C_t^k = l_k\}\right) = \prod_{k=1}^K \binom{m_k}{l_k} [\bar{F}_k(t)]^{m_k-l_k} [1 - \bar{F}_k(t)]^{l_k} \quad (5.5a)$$

$$\underline{P}\left(\bigcap_{k=1}^K \{C_t^k = l_k\}\right) = \prod_{k=1}^K \binom{m_k}{l_k} [\underline{F}_k(t)]^{m_k-l_k} [1 - \underline{F}_k(t)]^{l_k}. \quad (5.5b)$$

Evaluation of the complete survival signature is a numerically demanding task. In a previous work we presented an approach where Monte Carlo simulation is used to reduce the number of structure function evaluations in Eq. 5.1 [94]. While this enabled the analysis of larger systems, each individual entry of the survival signature still requires a significant number of structure function evaluations. As the number of entries in the survival signature grows with increasing system size and number of component types this technique also reaches its limits. In the previous study, the numerical example with 306 nodes and 2 component types already required 27 hours of computation time while using 64 cores for parallel computation of survival signature entries. Therefore, we propose to only evaluate a subset of survival signature entries and use them to fit a surrogate model. This surrogate model can then be used for the reliability analysis instead of the full survival signature.

5.3 Monotone Radial Basis Function Networks

This section presents the monotone radial basis function (RBF) networks which will be utilized as metamodels of the survival signature. RBF networks are a very simple type of artificial neural networks with only one hidden layer, and a linear output layer [138]. The name stems from the fact that radial basis functions are used as activation functions in the hidden layer. A distance function between the input and a fixed point known as a center is used to define these RBFs in \mathbb{R}^d . This distance is most commonly the Euclidean distance with a normalization constant in each dimension $j \in \{1, \dots, d\}$ defined as

$$\|x - c\| = \sqrt{\sum_{j=1}^d \frac{(x_j - c_j)^2}{2\sigma_j^2}}, \quad (5.6)$$

where $x \in \mathbb{R}^d$ is the input vector. The spread of the radial basis functions is controlled by the normalization constant σ_j . The most commonly used radial basis function is the Gaussian function

$$\psi(\|x - c\|) = e^{-\|x - c\|^2}. \quad (5.7)$$

In an alternative notation, instead of including the spread in the distance function through the σ_j values, it can be directly included in the radial basis function, usually represented as a shape parameter ϵ . For the Gaussian function this would be defined as follows

$$\psi(\|x - c\|) = e^{-\epsilon\|x - c\|^2}. \quad (5.8)$$

However, this reduction to a single shape parameter reduces modeling flexibility of the radial basis functions when compared to the σ_j as these allow to vary the spread of the basis function in each dimension. For the remainder of this paper, we assume all radial basis functions to be Gaussian of the form defined in Eq. 5.7. Figure 5.1 presents the one dimensional Gaussian radial basis function and how the choice of σ influences the spread of the function. For a thorough discussion of radial basis functions refer to Buhmann [147].

An RBF network is designed based on N neurons each with their own center points c_i and associated weight w_i for $i \in \{1, \dots, N\}$ and defined as

$$y(x) = \sum_{i=1}^N w_i \psi(\|x - c_i\|). \quad (5.9)$$

Estimation of the weights w_i can be done through linear least squares because y is linear in the weights. Figure 5.2 shows a graphical representation of an RBF network.

Heimes and van Heuveln [139] have shown that normalizing RBF networks improves their extrapolation capabilities in comparison to regular (unnormalized) networks. For the remainder

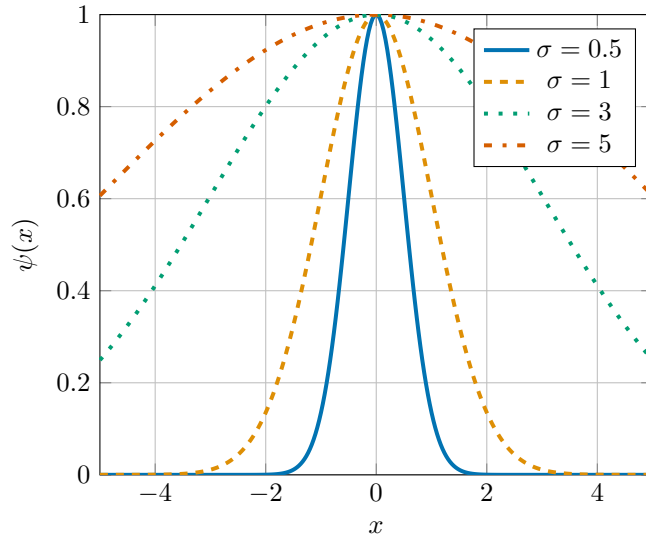


Figure 5.1: Gaussian radial basis functions centered at $c = 0.0$ and the influence of different spread values. Note how the functions become less narrow with increasing σ .

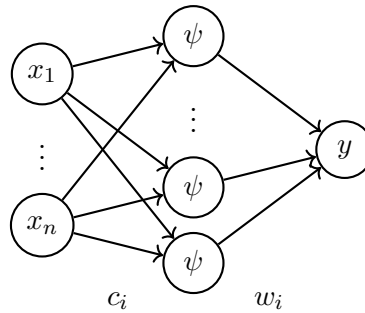


Figure 5.2: Graphical representation of a radial basis function neural network. From left to right: input layer, hidden layer, linear output layer.

of this paper, the applied basis functions will always be considered normalized as

$$\hat{\psi}_i(x) = \frac{\psi(\|x - c_i\|)}{\sum_{i=1}^N \psi(\|x - c_i\|)}. \quad (5.10)$$

The output of the normalized radial basis function (NRBF) network is then defined as

$$y(x) = \sum_{i=1}^N w_i \hat{\psi}_i(x). \quad (5.11)$$

In this study, we limit the NRBF network's center positions to a regular, uniform grid across the whole input domain. This makes it simpler to choose the σ_j shape parameters and considerably reduces the complexity of the monotonicity restrictions in the next step.

The NRBF network will later be utilised as a surrogate model for the survival signature. Since the survival signature is a monotone function, using a monotone surrogate model will in

turn lead to a significantly better fit. Simple linear constraints on the weights can be used to enforce the monotonicity of an NRBF network. As shown by Hušek [140], only weights of center points lying on a line pointing in the direction of any of the d dimensions must be monotonic for the entire NRBF network to be monotonic. Note, that this assumption only holds for the Gaussian radial basis functions.

The optimal weights under these monotonicity constraints are then found as the solution to the following constrained least squared optimization problem, assuming that there are M data points x_j with associated function values y_j for $j \in \{1, \dots, M\}$:

$$\begin{aligned} & \underset{w}{\text{minimize}} && \|\Psi w - y\|_2^2 \\ & \text{subject to} && w_p \leq w_q, \quad p, q \in \{1, \dots, N\}, \\ & && \text{if } c_p^\alpha < c_q^\alpha, c_p^\beta = c_q^\beta, \alpha \neq \beta. \end{aligned} \tag{5.12}$$

Here, the superscripts $\alpha, \beta \in \{1, \dots, d\}$ refer to the coordinates in dimensions α and β with $w = [w_1, \dots, w_N]^T$ and $y = [y_1, \dots, y_M]^T$. The matrix $\Psi \in \mathbb{R}^{M \times N}$ is defined as

$$\Psi_{ji} = \hat{\psi}_i(x_j). \tag{5.13}$$

As such, every weight is only bound by up to d constraints (one in each dimension; less at the edges of the domain). For example, a uniform grid of $20 \times 20 = 400$ center points requires 760 constraints for the entire NRBF network to be monotonic.

This is a convex optimization problem which can be solved with any standard solver. In this work we apply the splitting cone solver (SCS) [141] in connection with Convex.jl [148].

5.4 Interval Predictor Models

For larger systems, even individual entries of the survival signature can require prohibitive amounts of structure function evaluations to compute Eq. 5.1 in order to obtain the data required to fit the NRBF network. For this reason, we resort to the previously introduced method [94] and use Monte Carlo simulation to approximate these entries. However, approximating these entries has uncertainty attached to it which must not be disregarded. This section presents the necessary theory on interval predictor models (IPM) [149] which we will later use to propagate the uncertainty of the NRBF approximation into bounds on the survival signature.

Where a regular model such as a NRBF network returns a single value, an IPM returns an interval in which the output is expected to fall. A parametric IPM is obtained as

$$I(x, P) = \{y = p^T \xi(x), p \in P\}, \tag{5.14}$$

where x is the input vector, $\xi(x)$ is an arbitrary basis and p is a member of the hyper rectangle

P defined by

$$P = \{p : \underline{p} \leq p \leq \bar{p}\}. \quad (5.15)$$

Then, the IPM is given as

$$I(x, P) = [\underline{y}(x, \underline{p}, \bar{p}), \bar{y}(x, \underline{p}, \bar{p})], \quad (5.16)$$

with \underline{y} and \bar{y} being the lower and upper bounds of the IPM defined by

$$\underline{y}(x, \underline{p}, \bar{p}) = \bar{p}^T \left(\frac{\xi(x) - |\xi(x)|}{2} \right) + \underline{p}^T \left(\frac{\xi(x) + |\xi(x)|}{2} \right) \quad (5.17a)$$

$$\bar{y}(x, \underline{p}, \bar{p}) = \bar{p}^T \left(\frac{\xi(x) + |\xi(x)|}{2} \right) + \underline{p}^T \left(\frac{\xi(x) - |\xi(x)|}{2} \right). \quad (5.17b)$$

For the radial basis applied in this work the equations simplify to

$$\underline{y}(x, \underline{p}) = \underline{p}^T \xi(x) \quad (5.18a)$$

$$\bar{y}(x, \bar{p}) = \bar{p}^T \xi(x), \quad (5.18b)$$

since $\xi(x) = |\xi(x)|$.

The parameters \underline{p} and \bar{p} are determined by solving the following optimization problem

$$\begin{aligned} & \underset{\underline{p}, \bar{p}}{\text{minimize}} && \frac{1}{M} \sum_j^M \delta_y(x_j, \underline{p}, \bar{p}) \\ & \text{subject to} && \underline{y}(x_j, \underline{p}, \bar{p}) \leq y_j \\ & && \bar{y}(x_j, \underline{p}, \bar{p}) \geq y_j \\ & && \underline{p} \leq \bar{p}, \end{aligned} \quad (5.19)$$

where (x_j, y_j) for $j \in \{1, \dots, M\}$ are the data points and δ_y is the spread of the IPM, i.e. the separation between its limits defined as

$$\delta_y(x, \underline{p}, \bar{p}) = |\xi(x)|^T (\bar{p} - \underline{p}). \quad (5.20)$$

As an example consider 200 data points generated from the function

$$y(x) = (6x - 2) \cdot \sin(2x) + g \quad (5.21)$$

with $x \sim U(-3, 3)$ and the noise generating term $g \sim N(0, 1)$. We assume a linear combination of polynomials as the basis $\xi(x) = [1, x, x^2, x^3, x^4, x^5, x^6, x^7, x^8]$ and solve the optimization problem in Eq. 5.19. Figure 5.3 presents the data points and the resulting IPM envelopes. The bounds contain all the data points and approximate the underlying function well.

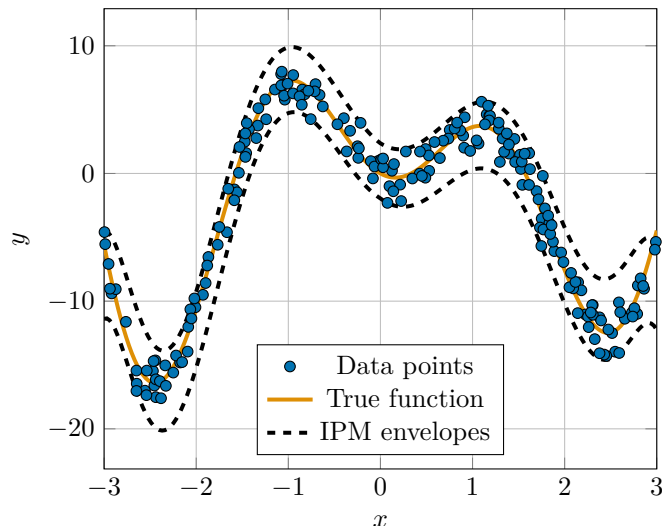


Figure 5.3: Interval predictor model envelopes around the noisy data points generated from the true function.

5.5 Imprecise Approximation of the Survival Signature

This section outlines the newly developed method to approximate the survival signature for very large systems. Eq. 5.1 demonstrates unequivocally that computing the survival signature is a combinatorial task, whose numerical demand significantly rises as the number of components and component types increases. As a result, for very large systems, analytical computing of the survival signature is impractical. We described a method for approximating the survival signature by utilizing percolation to remove insignificant portions of the signature and approximating the remaining entries with Monte Carlo simulation [94]. Although this made it possible to analyze larger systems, the method still necessitates a huge number of system assessments and will soon become computationally infeasible. In this article, we suggest a different approach in which the survival signature is approximated using a radial basis function network based on a few carefully chosen values. Additionally, an interval predictor model is used to propagate the uncertainty in the survival signature entries due to the Monte Carlo approximation, which is then reflected as bounds on the survival signature.

5.5.1 Design of the NRBF network surrogate model

Similarly to the previously proposed technique, this new approach begins by using percolation theory to determine the critical threshold f_c and exclude a major fraction of the survival signature where the chance that the system functions is minimal, i.e. near to zero [94].

Then, the set of remaining survival signature entries Ω is defined as all l_1, \dots, l_K for $l_k \in$

$\{0, 1, \dots, m_k\}$ and $k \in \{1, \dots, K\}$ where the following condition holds true

$$\sum_{k=1}^K l_k \geq (1 - f_c) \sum_{k=1}^K m_k. \quad (5.22)$$

To start, we calculate the survival signature for M initial data points spread uniformly over Ω . Values where the number of required structure function evaluations exceeds the defined maximum number of samples N_{MC} are approximated using Monte Carlo simulation. The rest are computed exactly using the analytical formulation. This ensures good coverage of the input space for the first approximation. In addition, we create a uniform grid of centers c_i for $i \in \{1, \dots, N\}$ distributed over the entire domain of Ω . The spread of the radial basis functions σ_j is chosen as the distance between two successive centers in the j th dimension, which has shown to yield good results. In the future optimal strategies for the spread selection should be devised. The solution to the optimization problem Eq. 5.12 yields the weights for the initial monotone NRBF model of the survival signature.

The surrogate model is then improved by selecting new values of the survival signature to be approximated. The Taylor expansion-based adaptive design (TEAD) [143], which is an exploration and exploitation technique based on nearest neighbor distances and Taylor expansion, is used to determine the new points. This allows to sufficiently cover the input space while also concentrating data in areas with the greatest changes in the survival signature in order to generate an accurate surrogate model with a small sample size. Each iteration of the TEAD chooses a new point from Ω and adds it to the M data points. Then the weights are updated by solving the optimization problem Eq. 5.12. When the change in weights falls below a predefined threshold three times in a row, the search for new points is terminated.

5.5.2 Extension to an interval predictor model

After obtaining an adequate surrogate model for the survival signature, it is expanded to an interval predictor model. In contrast to regular models, which produce a single output value for any input, IPMs return an interval within which the value will likely fall [149]. The reason for the extension to the IPM is that by simply fitting an NRBF network to the approximated values of the survival signature the uncertainty of the Monte Carlo simulation, reflected in the coefficient of variation, would be discarded. Instead, this uncertainty is propagated through the IPM yielding bounds on the survival signature.

The IPM theory defined earlier is slightly adapted to the NRBF basis and the data points evaluated in the previous step. The IPM utilizes the same center locations c_i for the upper and lower bounds whose difference in output is only influenced by two different sets of weights \underline{w}

and \bar{w} . The bounds are defined as

$$\bar{\Phi}(x, \bar{w}) = \Psi(x)^T \bar{w} \quad (5.23)$$

$$\underline{\Phi}(x, \underline{w}) = \Psi(x)^T \underline{w}, \quad (5.24)$$

where $\Psi(x) = [\hat{\psi}_1(x), \dots, \hat{\psi}_N(x)]$. The parameters \underline{w} and \bar{w} are given by the following optimization problem

$$\begin{aligned} & \underset{\underline{w}, \bar{w}}{\text{minimize}} && \frac{1}{N} \sum_j^N \delta_y(x_j, \underline{w}, \bar{w}) \\ & \text{subject to} && \underline{\Phi}(x_j, \underline{w}) \leq \underline{y}_j \\ & && \bar{\Phi}(x_j, \bar{w}) \geq \bar{y}_j \\ & && \underline{w} \leq \bar{w}, \end{aligned} \quad (5.25)$$

where $(x_j, [\underline{y}_j, \bar{y}_j])$ for $j \in \{1, \dots, M\}$ are the data points obtained in the previous step. The bounds \underline{y}_j and \bar{y}_j are computed from the survival signature entry approximations and the associated coefficient of variation cv_j by

$$\underline{y}_j = \max\{y_j - y_j \cdot cv_j, 0.0\}, \quad (5.26)$$

$$\bar{y}_j = \min\{y_j + y_j \cdot cv_j, 1.0\}. \quad (5.27)$$

In addition to the constraints in Eq. 5.25 we also invoke the monotonicity constraints of Eq. 5.12 on \underline{w} and \bar{w} in order to ensure monotonicity of the IPM envelopes.

After solving the optimization problem Eq. 5.25 to obtain the IPM, the bounds on the imprecise survival function can be computed from Eq. 5.3. Only two evaluations of the survival function are required, because both bounds of the IPM are monotone functions.

5.5.3 Method validation

We validate the proposed method by applying it to an artificial toy problem and comparing the survival signature bounds obtained from the IPM approximation to the full Monte Carlo approximation. Consider a simple grid network as presented in Figure 5.4. We select $n = 15$ and $m = 15$ and create a network with 225 nodes and 420 edges. The components are divided alternating into two types $k_1 = [1, 3, 5, \dots, 223]$ and $k_2 = [2, 4, 6, \dots, 225]$. As structure function $\varphi(\mathbf{x})$ we define the connectivity between the source node $s = 1$ and target node $t = 225$. The survival signature for this network contains 12882 entries of which the most numerically demanding entries have around $1.92e64$ different combinations to evaluate. Since analytical evaluation of this survival signature is practically impossible, the newly developed method will be compared to the older full Monte Carlo method [94] with up to 10000 samples per entry and a target coefficient of variation of $1e-3$. The TEAD for selecting additional points to compute

is aborted once the L2-norm of the difference between the weights before and after adding the new point is less than $1e-3$ three iterations in a row. Figure 5.5 shows the initial data points as well as the extra points selected by the TEAD algorithm. The graphic shows how the adaptive technique's new points are focused in areas with high survival signature relevance while choosing only a few points in areas where the survival signature is minimal. This ability to identify the essential section of the signature allows for accurate approximation with a small number of computed entries.

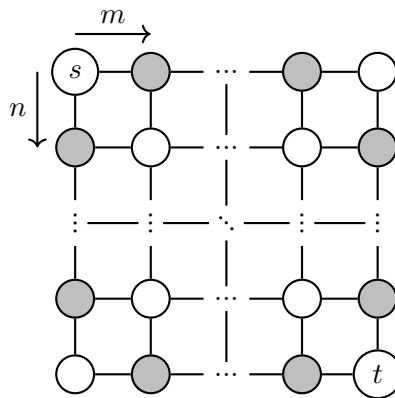


Figure 5.4: Simple grid network of size $n \times m$ with component types alternating between nodes. The nodes labelled s and t are the source and target nodes.

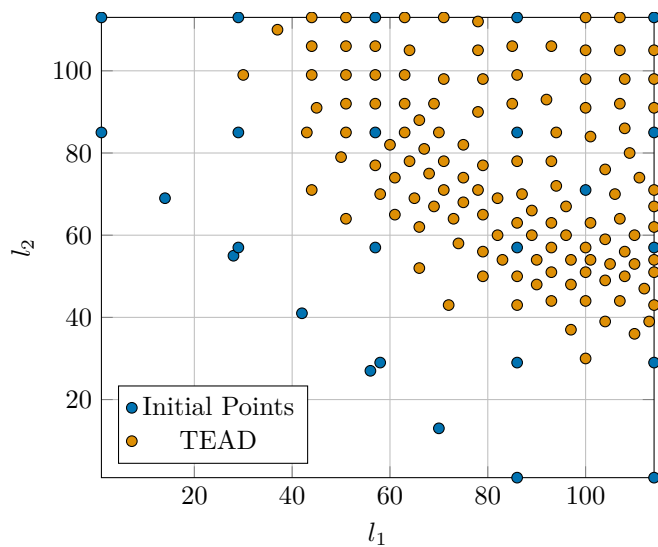


Figure 5.5: Initial data points and adaptive points selected by the TEAD.

We compute the spread of the IPM signature as defined in Eq. 5.20 in order to assess its quality. Note, that the spread is computed based on all entries in Ω not just the data points used to fit the IPM. The quality of the model improves with decreasing spread. The survival signature is approximated using different maximum samples sizes N_{MC} with the same uniform

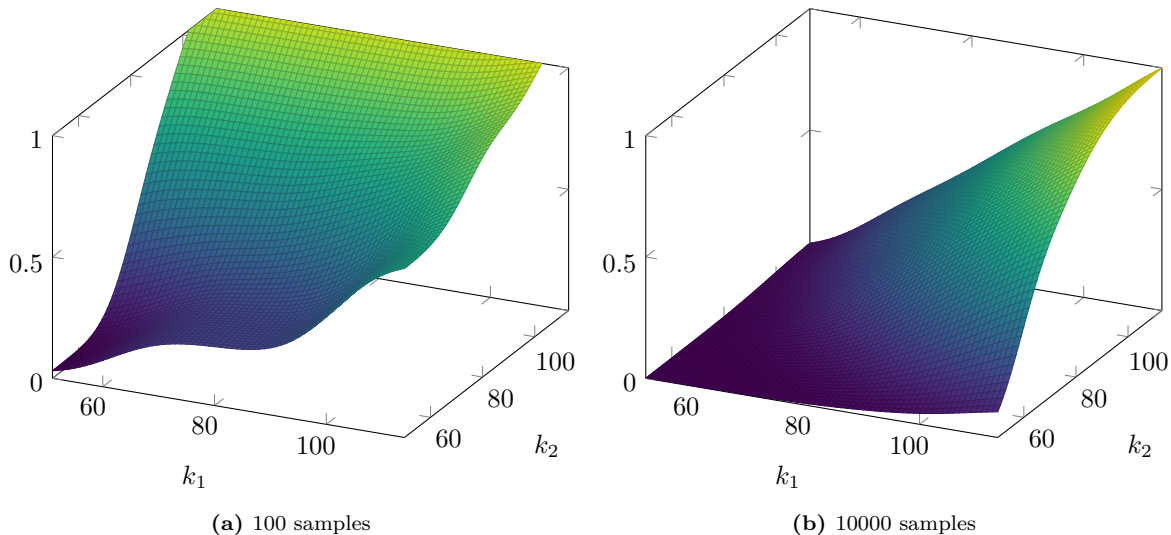


Figure 5.6: Upper bounds of the IPM survival signature for the 15×15 grid network using different samples sizes for individual entries.

Table 5.1: Spread and required data points for different sample sizes of the IPM survival signature for the 15×15 grid network. The number of structure function evaluations is denoted as n_φ .

N_{MC}	M	n_φ	$\delta_y(x)$
100	796	79303	0.29674
1000	296	293003	0.02999
10000	153	1500003	0.00715

grid of 163 center points, 15 in each dimension of which 62 are excluded using the percolation condition in Eq. 5.22. The results in Tab. 5.1 show that the spread reduces with increasing sample size. It can also be seen, that the number of data points M required to accurately fit the model decreases as well. So while the cost of each individual entry rises with larger sample sizes, fewer entries must be computed overall. The table also presents the number of required structure function evaluations n_φ . Comparing this to the analytical solution where $\approx 5.39e67$ evaluations are necessary shows how efficient this new method is. In comparison to the full Monte Carlo approximation with $N_{MC} = 10000$ where 33 172 770 state vectors must be evaluated, it achieves a further reduction in function evaluations of almost 95%. Figure 5.6 presents the relevant part of the upper bound approximation using $N_{MC} = 100$ and $N_{MC} = 10000$ samples. It is clear to see how the approximation using only 100 samples per entry is very rough and estimates large portions of the survival signature to be 1.0 while using more samples per entry leads to a much more precise signature.

Finally, we assume failure time distributions for the two component types to assess the network reliability. Let the failures of component type 1 be distributed exponentially with failure rate $\lambda = 1.0$. The failure times for component type 2 follow a Weibull distribution with shape $\alpha = 2.0$ and scale $\theta = 1.0$. Because of the survival signature's monotonicity, we can

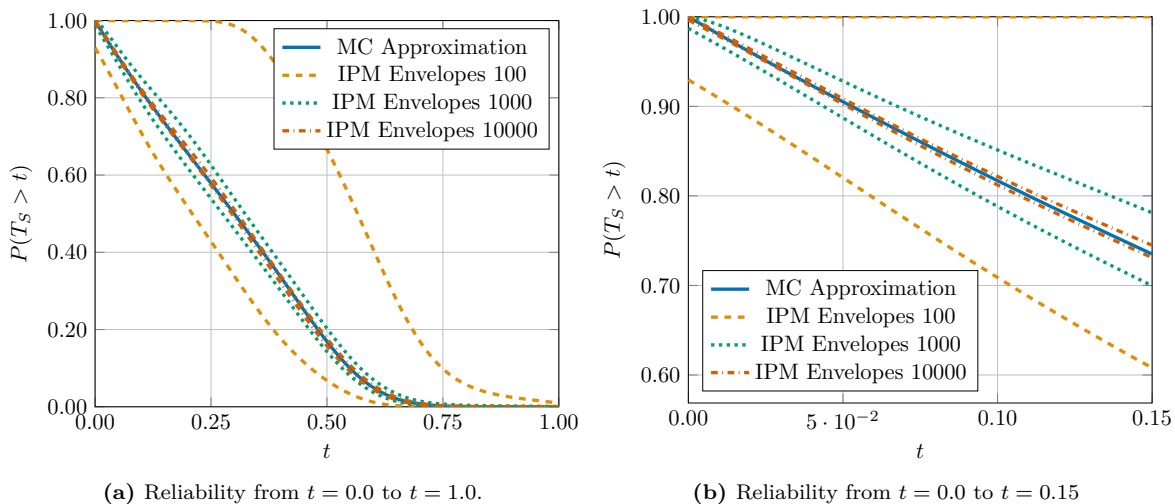


Figure 5.7: Bounds on the reliability of the 15×15 grid network for the different IPM survival signatures and the Monte Carlo approximation. A closer view from $t = 0.0$ to $t = 0.15$ is shown on the right.

simply evaluate Eq. 5.4 for the upper and lower bounds of the signature to obtain the bounds on the reliability. Figure 5.7 presents the reliability using the three different IPM approximations and compares them to the full Monte Carlo approximation. The plots show clearly how the bounds on the reliability shrink around the Monte Carlo solution when increasing the number of samples. Although only a rough estimate, even the approximation just using 100 samples is bounded around the Monte Carlo solution.

5.6 Numerical Example

In this section we apply the proposed technique to a numerical example. The Berlin S- and U-Bahn system is used as an example as presented in [94]. The topology of the network is presented in Fig. 5.8. The total network is made up of 306 nodes and 350 edges. The nodes are classified into three categories based on their degree, with type 1 consisting of nodes with degree of two or less, type 2 of nodes with a degree greater than two but lower than five and component type 3 of nodes having a degree of five or higher. As a result, there are 245 nodes of type 1, 53 of type 2 and 8 of type 3. This network's entire survival signature contains 119 556 entries.

We define a structure function based on the loss of network efficiency. In this context, network efficiency for a network G with n nodes is defined as

$$E(G) = \frac{1}{n(n-1)} \sum_{u \neq v \in G} \frac{1}{d_{uv}}, \quad (5.28)$$

where d_{uv} denotes the length of the shortest path between nodes u and v [134]. The shortest paths between all nodes can be efficiently calculated using the *Floyd-Warshall* algorithm [135].

The network is assumed to fail once the loss of efficiency exceeds 50 % as given by

$$\varphi(\mathbf{x}) = \frac{E(G(\mathbf{x}))}{E(G)} < 0.5. \quad (5.29)$$

Although the *Floyd-Warshall* algorithm is very efficient, this structure function has a much higher numerical demand than the one based on *s-t*-connectivity in the previous section. Because of this and the large size of the survival signature computing it analytically is impossible and even the Monte Carlo simulation requires vast amounts of computation time.

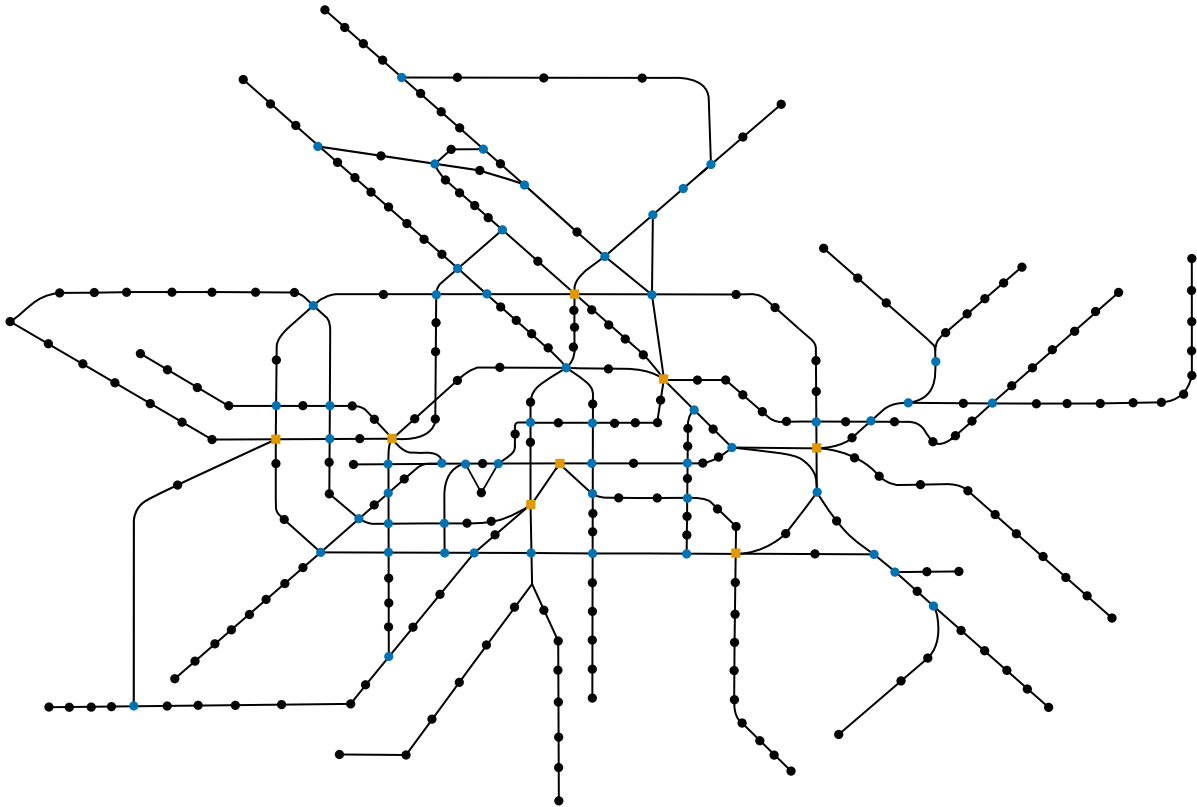


Figure 5.8: Topology of the Berlin metro system with 306 nodes. Nodes marked as ● represent stations with two or three connections. Stations with five or more connections are marked with ■. Adapted from Salomon et al. [72].

The survival signature approximation begins with percolation to obtain the set of candidate points Ω . The critical threshold $f_c \approx 0.39550$ reduces Ω from 119 556 to 45 441. 2250 center points c_i are generated uniformly over the domain of Ω , of which 603 are excluded using Eq. 5.22, leaving 1647 centers. Then, 125 starting points are uniformly selected from Ω . The TEAD is used to choose new points until the weights change is less than $1e-3$ for three consecutive points. After selecting 286 new points the algorithm aborts. The survival signature entries are approximated using 10 000 Monte Carlo samples and a target coefficient of variation of $1e-3$ for all points.

We assume that component failure times are exponentially distributed with $\lambda_1 = 0.3$, $\lambda_2 = 0.4$

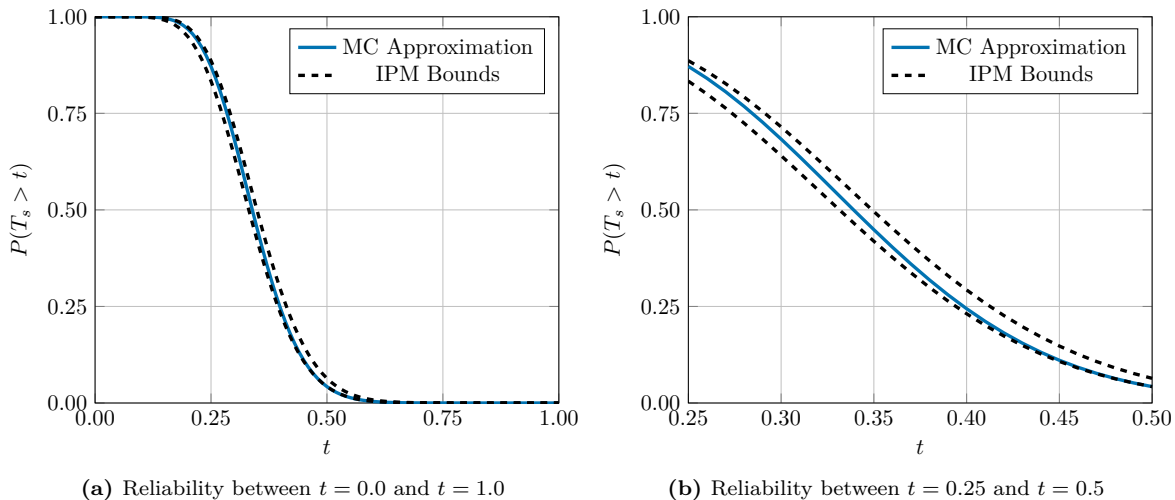


Figure 5.9: Bounds of the IPM surrogate model compared to the Monte Carlo approximation.

and $\lambda_3 = 0.4$. These values were arbitrarily chosen for demonstrative purposes and have no real world relevance. In Figure 5.9, the resulting bounds on the survival function are compared to the reliability of the entire Monte Carlo approximation. Only 406 out of the 119 556 survival signature values have been computed to build the IPM. This presents a significant reduction in numerical effort. In fact, only 3 960 256 structure function evaluations are required compared to the 385 919 420 of the Monte Carlo method, enabling the analysis of even larger networks than previously possible.

Next, rather than assuming precise probabilities for the component failure times, we represent them as p-boxes and create intervals for the distribution parameters as a result. The intervals for the exponential distribution parameters are chosen as $\lambda_1 \in [0.25, 0.35]$, $\lambda_2 \in [0.35, 0.45]$ and $\lambda_3 \in [0.45, 0.55]$. In this example, where only exponential distributions are used, we can obtain the lower and upper bounds of the reliability directly by injecting the upper and lower bounds of the parameters and IPM survival signature into Eq. 5.5. In cases where p-boxes of distributions with multiple parameters such as the Weibull distribution are involved, the reliability bounds can be calculated using a double-loop approach or optimization. Figure 5.10 depicts the bounds on the reliability using the IPM survival signature and the full Monte Carlo approximation. The graphic shows how the upper and lower bounds on the survival signature resulting from only the imprecise distributions are further widened when combined with the imprecise survival signature.

5.7 Conclusion

This research proposes a novel way to approximate the survival signature using normalized radial basis function networks and interval predictor models as surrogate models. Given only a small fraction of the evaluated signature entries, NRBF networks can reliably predict the whole

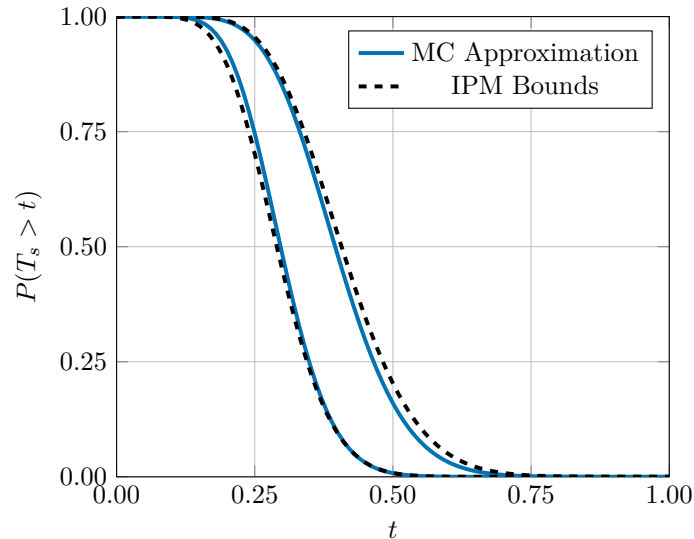


Figure 5.10: Bounds on the reliability of the Berlin metro system with imprecise failure time distributions for the IPM and the Monte Carlo approach.

survival signature. The uncertainty introduced by using Monte Carlo simulation to approximate individual entries is efficiently propagated through the IPM to produce survival signature bounds. An adaptive strategy for selecting data points for the surrogate model ensures excellent accuracy while keeping the numerical demand low. The effectivity of the developed method was demonstrated by applying it to the analysis of a simple designed toy example as well as a larger real world example based on the metro system of Berlin and comparing the results to those of the previously presented full Monte Carlo simulation technique. By combining the imprecise survival signature with imprecise probabilities a fully imprecise survival function is devised. The code developed for this work is added to the open source Julia package *SurvivalSignature.jl* and freely available to everyone [99].

In future work we will focus on improving the NRBF fit. While the number of centers and the spread selected in this study provide a good basis, optimizing these in the future would improve the design. Exploring alternative strategies to the TEAD for adaptively selecting the new points should also be considered. Furthermore, scalability for example with more than three dimensions must be investigated.

Acknowledgment

The authors would like to appreciate the support of the National Natural Science Foundation of China under grant 72271025.

6 | Multidimensional Resilience Decision-Making for Complex and Substructured Systems

Multidimensional Resilience Decision-Making for Complex and Substructured Systems

Julian Salomon^a, Jasper Behrendorf^a, Niklas Winnewisser^a, Matteo Broggi^a, Michael Beer^{a,b,c}

^aInstitute for Risk and Reliability, Leibniz Universität Hannover, Hannover, Germany

^bInstitute for Risk and Uncertainty, University of Liverpool, Liverpool, United Kingdom

^cInternational Joint Research Center for Resilient Infrastructure & International Joint Research Center for Engineering Reliability and Stochastic Mechanics, Tongji University, Shanghai, China

Published in *Resilient Cities and Structures* on November 02, 2022

Abstract

Complex systems, such as infrastructure networks, industrial plants and jet engines, are of paramount importance to modern societies. However, these systems are subject to a variety of different threats. Novel research focuses not only on monitoring and improving the robustness and reliability of systems, but also on their recoverability from adverse events. The concept of resilience encompasses precisely these aspects. However, efficient resilience analysis for the modern systems of our societies is becoming more and more challenging. Due to their increasing complexity, system components frequently exhibit significant complexity of their own, requiring them to be modeled as systems, i.e., subsystems. Therefore, efficient resilience analysis approaches are needed to address this emerging challenge.

This work presents an efficient resilience decision-making procedure for complex and substructured systems. A novel methodology is derived by bringing together two methods from the fields of reliability analysis and modern resilience assessment. A resilience decision-making framework and the concept of survival signature are extended and merged, providing an efficient approach for quantifying the resilience of complex, large and substructured systems subject to monetary restrictions. The new approach combines both of the advantageous characteristics of its two original components: A direct comparison between various resilience-enhancing options from a multidimensional search space, leading to an optimal trade-off with respect to the system resilience and a significant reduction of the computational effort due to the separation property of the survival signature, once a subsystem structure has been computed, any possible characterization of the probabilistic part can be validated with no need to recompute the structure.

The developed methods are applied to the functional model of a multistage high-speed axial compressor and two substructured systems of increasing complexity, providing accurate results and demonstrating efficiency and general applicability.

Keywords: Resilience, Decision-making, Survival signature, Reliability, Complex systems, Substructured systems.

6.1 Introduction

In today's highly developed societies, complex systems, such as infrastructure networks, industrial plants and jet engines are both ubiquitous and of paramount importance to the functioning of these modern societies. It is evident that these systems are exposed to a variety of harmful influences of natural, technical and anthropogenic origin. At the same time, as Punzo et al. highlight in [150], "It is an undeniable fact that modern day systems are more integrated, more interdependent, evolve at faster pace and, in a word, are more complex than the systems of the previous century [...]." Considering this high and increasing system complexity, it is

impractical to detect and prevent all potential negative impacts. Therefore, it is essential that new developments in engineering focus not only on monitoring and improving the robustness and reliability of systems, but also on their recoverability after adverse events [151]. The concept of resilience encompasses these aspects: analyzing and optimizing robustness, reliability and recovery of systems, from a technical and economic perspective [152–154]. Applying resilience to engineered systems leads to a paradigm shift. Secure systems cannot solely rely on strategies to prevent failures, but must include strategies for efficient recovery in the event of failure as well, see, e.g., [155, 156].

In engineering, the concept of resilience has steadily gained popularity in recent years [74, 150, 157]. The notion of “resilience” appears in various fields such as ecology, economics, and psychology, as well as in the context of mechanical systems, and is derived from the Latin word “resilire,” which means “to bounce back.” The concept of resilience was first introduced by Holling in the field of ecological systems [158]. Although several other definitions by various scientists followed, most of them have certain key aspects in common that were already captured by Holling’s early definition [159–163]. In [164], Ayyub provides a literature review and develops a comprehensive definition of resilience in the context of complex systems based on the content of the Presidential Policy Directive (PPD) on critical infrastructure security and resilience [165]. His definition provides a solid foundation for quantifying resilience.

Numerous options exist for improving the resilience of complex systems. However, resources are not unlimited and resilience cannot be increased at will. Therefore, it is essential not only to be able to distinguish and weigh between a variety of different resilience-enhancing measures, but to also consider their monetary aspects [166, 167]. In [72], Salomon et al. present a method for identifying the most cost-effective allocation of resilience-enhancing investments by merging the resilience metric of [76] and an adaptation of the systemic risk measure of [77]. Their approach allows for a direct comparison of the effects of heterogeneous controls on the resilience of a system over an arbitrary time period in a two-dimensional parameter space.

Additionally, current research in the context of resilience focuses on improved resilience quantification measures, as proposed in [168], and overarching frameworks for stakeholder decision-making, e.g., for transportation networks in the presence of seismic hazards [169]. For a comprehensive literature review on resilience assessment frameworks that balance resources and performance, see [170]. Other researchers recently studied the complexity of realistic infrastructure systems, failure consequences, recovery sequences, and varying external effects. In [171], for example, the authors revealed the vast complexity of modern critical infrastructures and their multi-factorial nature as cyber-human-physical systems and studied appropriate modeling and resilience analysis approaches. Further, the works [172] and [173] are concerned with the effects on decision-making when considering stakeholder preferences or enhancement and recovery strategies. External effects and challenges arising from climate change were studied in the context of resilience, e.g., in [174].

Various technical and infrastructural systems in today’s society are large and complex in

nature. In particular, when system components have such complexity that they themselves need to be modeled as systems, so-called systems of systems [175, 176], resulting in a significantly high number of components. This is in accordance to Batty, who highlights “A very simple definition of a complex system is ‘a system that is composed of complex systems’” [177]. As each of the subsystems affects the top-level system under consideration, this causes a significant increase in computational effort for system analysis and constitutes a major challenge [178, 179]. Therefore, it is particularly important to have tools capable of efficiently assessing all three resilience phases. Typically the reliability phase involves the most system evaluations, in particular when various different system configurations need to be assessed that have an impact on the probability structure of the subsystems and thus on the overall system. Therefore, a particularly efficient analysis approach is required for this phase.

An efficient approach to modeling the reliability of systems with multiple component types is provided by the concept of survival signature, introduced and discussed in [27, 180] by Coolen and Coolen-Maturi. Its major benefit over conventional approaches is the separation of the system structure from the probabilistic properties of the system components. Once the system structure has been analyzed, any possible probabilistic characterization can be tested without having to reevaluate any system states. Consequently, this approach reduces the computational cost of repeated model evaluations typically required in design and maintenance processes [37]. Current research is focused on multi-state components [37], common cause failures [33], multiple failure modes and dependent failures [32], approximation techniques for large systems [94] and reliability analysis in consideration of imprecision [181].

In this paper, theoretical fundamentals are summarized and the resilience decision-making method introduced in [72] is extended to multidimensional parameter spaces. Next, a novel and encompassing methodology is developed, consisting of its two major ingredients, the extended resilience decision-making method and the survival signature. This allows for an efficient and multidimensional resilience analysis of complex, large and substructured systems. The extension and novel methodology are then applied to a functional model of a multistage high-speed axial compressor, an arbitrary complex system as well as the U-Bahn and S-Bahn system of Berlin, to prove general applicability.

6.2 Resilience decision-making

Assessing the resilience of complex systems subject to technical or monetary constraints requires a sophisticated methodology to efficiently derive optimal decisions. In [72], Salomon et al. propose a versatile approach with three key elements, including a metric for resilience quantification, an adapted systemic risk measure, and a grid search algorithm that increases computational efficiency.

6.2.1 Resilience quantification

A suitable quantitative measure of resilience is a fundamental prerequisite for assessing resilience in engineering. In [74, 75, 182], the authors provide a comprehensive overview of resilience metrics in a systemic context. While Bergström et al. emphasize the general concept of resilience in the current literature as a critical link between increasing complexity of systems and their risk [74], Sun et al. focus on resilience of infrastructures and highlight the close link between resilience and functionality respectively performance measures [182]. Hosseini et al. proposed a general scheme for categorizing resilience quantification approaches [75]. In summary, performance-based resilience metrics are most widely used. These determine the resilience of a system by comparing its performance before and after a destructive event. Further subcategories relate to time in-/dependence and characterization as deterministic or probabilistic processes.

According to [183] and [75], performance-based and time-dependent metrics are capable of considering the following system and transition states before and after a disruptive event:

- The original stable state, i.e., the duration until a disruptive event occurs, relying on the reliability of the system.
- The system vulnerability, represented by a loss of performance after the occurrence of a disruptive event and the robustness counteracting the vulnerability and mitigating this performance loss. Both are governed by degradation characteristics of the system components.
- The system recoverability, characterized by the disrupted state of the system and its recovery to a new stable state.

An illustration of these phases and transitions is shown in Fig. 6.1. The performance level of the new stable state might differ from the performance level of the original state.

The area of performance loss between original and new stable state in Fig. 6.1 refers to the well-known principle of “*resilience triangle*” introduced by Bruneau et al. [73], as illustrated in Fig. 6.2. In their work, Bruneau et al. proposed a time-dependent, performance-based, and deterministic metric for resilience loss of a community due to seismic disasters as follows. Let t_0 be the time a disruptive event occurs and t_1 be the time of completed recovery. Further, $Q(t)$ denotes the quality of the community infrastructure at time t , specifying the type of system performance. Then, the metric is defined as:

$$R_{Br} = \int_{t_0}^{t_1} [100 - Q(t)] dt. \quad (6.1)$$

Note that the system performance is compared with a time-independent ideal performance of 100 in the considered interval of performance loss. The approach forms a strong basis for

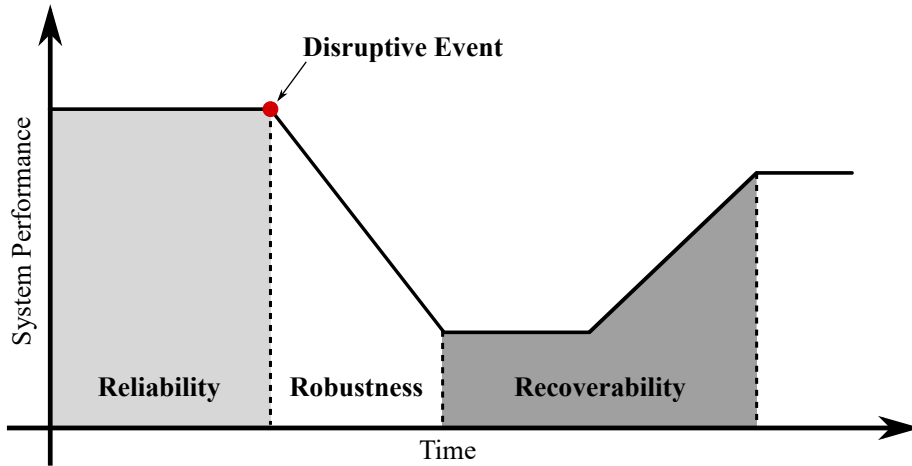


Figure 6.1: In the evolution of a system before and after the impact of a disruptive event, different phases can be distinguished: (i) the original stable state, (ii) disruptive impact, vulnerability, robustness, (iii) disrupted state and recovery; adapted from [183].

several, later proposed metrics in various contexts, see [184–186].

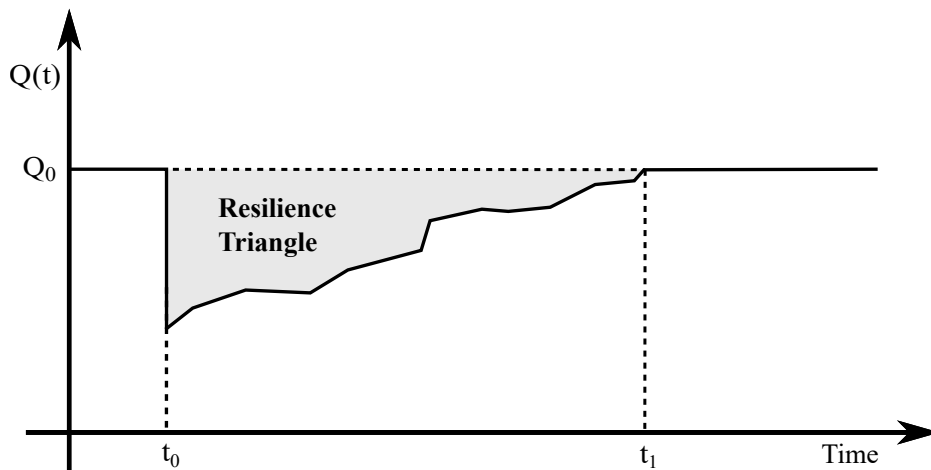


Figure 6.2: Resilience triangle; adapted from [73].

In [72], Salomon et al. utilize the probabilistic and time-dependent metric developed by Ouyang et al. [76]. The metric is defined as the expected ratio of the integral over the actual system performance $Q(t)$ from 0 to a given time T and the corresponding integral of a target system performance $\mathcal{T}Q(t)$ over the same time interval:

$$Res = E[Y], \tag{6.2}$$

where

$$Y = \frac{\int_0^T Q(t)dt}{\int_0^T \mathcal{T}Q(t)dt}. \tag{6.3}$$

Thereby, the system performance $Q(t)$ is a stochastic process. The target system performance $\mathcal{T}Q(t)$ can be generally considered as a stochastic process as well, however, for simplicity, $\mathcal{T}Q(t)$ may be assumed as a non-random and constant quantity $\mathcal{T}Q$. Assuming that the actual system performance does not exceed the target performance, the metric takes values between 0 and 1. For $Res = 1$, the system performance is equal to the target system performance, while $Res = 0$ indicates that the system is not functioning during the entire period under consideration.

6.2.2 Adapted systemic risk measure

In [77], Feinstein et al. proposed a general approach to measuring systemic risk, e.g., pursued in finance [187]. In [72], this risk measure was adapted and extended for the application to engineering systems as summarized in this section. The adapted systemic risk measure comprises a descriptive input-output model and an acceptance criterion that represents normative resilience standards of a regulatory authority.

Let a system be given with m components $i \in \{1, \dots, m\}$ of type $k_i \in \{1, 2, \dots, K\} \subseteq \mathbb{N}$ with e properties that influence the system performance $Q(t)$. These properties, hereafter referred to as “endowment properties”, affect system resilience and can be improved through capital allocations. Then, the component i is characterized by

$$(\mathbf{a}_i; k_i) = (\eta_{i1}, \eta_{i2}, \dots, \eta_{ie}; k_i) \in \mathbb{R}^{(1 \times e)} \times \mathbb{N}, \quad (6.4)$$

where $(\eta_{i1}, \eta_{i2}, \dots, \eta_{ie})$ are the numerical values of the e relevant endowment properties. Consequently, the entire system can be described by a tuple, consisting of the matrix $\mathbf{A} \in \mathbb{R}^{(m \times e)}$ and the column vector $\mathbf{z} \in \mathbb{N}^m$ that captures the component types:

$$(\mathbf{A}; \mathbf{z}) = \begin{pmatrix} \eta_{11} & \eta_{12} & \cdots & \eta_{1e} & z_1 \\ \eta_{21} & \eta_{22} & \cdots & \eta_{2e} & z_2 \\ \vdots & \vdots & & \vdots & \vdots \\ \eta_{m1} & \eta_{m2} & \cdots & \eta_{me} & z_m \end{pmatrix}. \quad (6.5)$$

The system under consideration is defined via a descriptive, non-decreasing input-output model $Y = Y_{(\mathbf{A}; \mathbf{z})}$ that is specified by this tuple and relates endowment properties to system performance. With respect to Eq. 6.2, the model output is specified as $Y = Y_{(\mathbf{A}; \mathbf{z})}$ dependent on the current endowment allocation $(\mathbf{A}; \mathbf{z})$.

Further, consider the following specific acceptance set

$$\mathcal{A} = \{X \in \mathcal{X} \mid E[X] \geq \alpha\} \quad (6.6)$$

for a normalized model output X and its expected value $E[X]$ with $\alpha \in [0, 1]$. Correspondingly,

the risk measure is defined as

$$R(Y) = \{ \mathbf{A} \in \mathbb{R}^{m \times e} \mid Y_{(\mathbf{A};z)} \in \mathcal{A} \}, \quad (6.7)$$

that is the set of all endowment property allocations \mathbf{A} such that the system reaches a resilience value greater or equal to α .

In practice, it might be necessary to impose structural restrictions on the matrix in Eq. 6.5. For example, consider the case that any component i of a specific type should be configured in the same way, i.e., the row vectors \mathbf{a}_i are claimed to be equal. In [77], Feinstein et al. capture such constraints by monotonously increasing functions $g_z : \mathbb{R}^p \rightarrow \mathbb{R}^{(m \times e)}$, $a' \mapsto (A; z)$ with $z \in \mathbb{R}^m$ denoting the component types. Such a function maps a lower-dimensional set of parameters $a' \in \mathbb{R}^p$ to the system description given in Eq. 6.5.

6.2.3 Grid search algorithm and the curse of dimensionality

According to [77] and [72], the measure of systemic risk might be determined via a combination of a grid search algorithm and stochastic simulations. The grid search algorithm operates in the space of all possible endowments, while stochastic simulations are employed to evaluate system resilience for the endowment allocations according to the grid search algorithm. The probabilistic resilience metric (Eq. 6.2 and Eq. 6.3) is estimated by means of Monte Carlo simulation. The grid search algorithm given in [77] consists of two phases and can be recapitulated as follows:

- (I) Search along the main diagonal of the space of endowment properties until the first acceptable combination is found based on the adapted systemic risk measure.
- (II) Identify the Pareto front between the set of acceptable endowments $R(Y)$ and its complement $R(Y)^c$ starting at the first accepted allocation.

The algorithm allows to compute the entirety of $R(Y)$ while significantly reducing the computational cost due to the assumed monotonicity property of the input-output model $Y_{(\mathbf{A};z)}$ given in Sec. 6.2.2. For a detailed description of a grid search algorithm for two dimensional problems, see [77], Ch. 4.

In [72] this algorithm was included in the resilience decision-making method and applied to case studies with two dimensional parameter spaces. In their work [77], Feinstein et al. point out that the grid search algorithm is applicable to higher dimensional problems “[...] *at the price of substantially larger computation times and required memory capacity*”. However, when analyzing real technical systems, it is often inevitable to consider a large number of influencing factors and thus a higher dimensionality of the parameter space. Therefore, in Sec. 6.5, an extension of the previously proposed resilience decision-making methodology to n -dimensional problems is applied to a four-dimensional functional model of an axial compressor and, in Sec. 6.7, as part of the novel methodology proposed in Sec. 6.4, it is applied to the *U-Bahn* and *S-Bahn* system of Berlin, addressing a five-dimensional problem.

6.3 Concept of survival signature

Introduced in [27], the concept of survival signature allows to compute the survival function of a system with multiple component types and attracted increasing attention for its advantageous features over the last decade. One of its merits is the high efficiency in repeated model evaluations due to the separation of the topological system reliability and the probability structure of system component failures. At the same time, the survival signature radically condenses information on topology. System components are of one type if their failure times are independent and identically distributed (*iid*) or exchangeable. This differentiation is important when it comes to modeling dependent component failure times [180]. A brief recap of the concept is provided in the following subsections. Detailed information about both the derivation of the concept and further applications can be found in [27, 58, 180].

6.3.1 Structure function

Let a system be given consisting of m components of a single type. Further, let $\mathbf{x} = (x_1, x_2, \dots, x_m) \in \{0, 1\}^m$ define the corresponding state vector of the m components, where $x_i = 1$ indicates a functioning state of the i -th component and $x_i = 0$ indicates a non-functioning state. Then, the structure function ϕ is a function of the state vector \mathbf{x} defining the operating status of the considered system: $\phi = \phi(\mathbf{x}) : \{0, 1\}^m \rightarrow \{0, 1\}$. Accordingly, $\phi(\mathbf{x}) = 1$ denotes a functioning system and $\phi(\mathbf{x}) = 0$ specifies a non-functioning system.

Suppose that a system consists of components of more than one type, i.e., $K \geq 2$. Then, the quantity of system components is denoted by $m = \sum_{k=1}^K m_k$, where m_k is the number of components of type $k \in \{1, 2, \dots, K\}$. Correspondingly, the state vector for each type is given by $\mathbf{x}^k = (x_1^k, x_2^k, \dots, x_{m_k}^k)$.

6.3.2 Survival signature

The survival signature summarizes the probability that a system is functioning as a function solely depending on the number of functioning components l_k per component type $k \in \{1, 2, \dots, K\}$. Assuming the failure times within a component type to be *iid* or exchangeable, the survival signature is defined as:

$$\Phi(l_1, l_2, \dots, l_K) = \left[\prod_{k=1}^K \binom{m_k}{l_k}^{-1} \right] \times \sum_{\mathbf{x} \in S_{l_1, l_2, \dots, l_K}} \phi(\mathbf{x}), \quad (6.8)$$

where $\binom{m_k}{l_k}$ corresponds to the total number of state vectors \mathbf{x}^k of type k and S_{l_1, l_2, \dots, l_K} denotes the set of all state vectors of the entire system for which $l_k = \sum_{i=1}^{m_k} x_i^k$. Consequently, the survival signature depends only on the topological reliability of the system, independent of the time-dependent failure behavior of its components that is described in Sec. 6.3.3. For more information on claimed exchangeability in practice, see [180, 181].

6.3.3 Probability structure

The probability structure of system components specifies the probability that a certain number of components of type k is functioning at time t . Accordingly, $C_k(t) \in \{0, 1, \dots, m_k\}$ represents the number of components of type k in a functioning state at time t . Further, assume the probability distribution for the failure times of type k to be known with $F_k(t)$, denoting the corresponding cumulative distribution function. Then,

$$\begin{aligned} P\left(\bigcap_{k=1}^K \{C_k(t) = l_k\}\right) &= \prod_{k=1}^K P(C_k(t) = l_k) \\ &= \prod_{k=1}^K \binom{m_k}{l_k} [F_k(t)]^{m_k - l_k} [1 - F_k(t)]^{l_k} \end{aligned} \quad (6.9)$$

describes the probability structure of the system, regardless of its topology.

6.3.4 Survival function

The survival function describes the probability of a system being in a functioning state at time t and results from Sec. 6.3.2 and 6.3.3 as:

$$P(T_s > t) = \sum_{l_1=0}^{m_1} \dots \sum_{l_K=0}^{m_K} \Phi(l_1, l_2, \dots, l_K) \times P\left(\bigcap_{k=1}^K \{C_k(t) = l_k\}\right), \quad (6.10)$$

where T_s denotes the random system failure time. Clearly, the concept of survival signature separates the time-independent topological reliability and the time-dependent probability structure. Thus, the survival signature, calculated once in a pre-processing step, can be reused for further evaluations of the survival function, which are necessary, for example, when analyzing a variety of different system configurations that affect the probability structure given a constant system topology. The survival signature can be stored in a matrix, thereby summarizing the topological reliability. The utilization of this matrix circumvents the repeated evaluation of the often computationally expensive structure function. Note that it is precisely these properties of the survival signature concept that provide an important advantage over conventional methods when system simulations need to be performed repeatedly [37]. In terms of computational demand, Monte Carlo simulation may be used to approximate the survival signature of large systems [94].

6.4 Proposed methodology

In this section, the proposed methodology for computationally efficient resilience analysis in the context of complex substructured systems is illustrated. The approach integrates the concept of survival signature described in Sec. 6.3 into the resilience decision-making framework

recapped in Sec. 6.2. First, the preparation of the complex system by means of a formalized substructuring approach is presented. Second, the novel methodology is proposed.

6.4.1 Definition of substructured systems

Assume a substructured system \mathcal{S} that is composed of a set of subsystems and a set of components. The subsystems can again be comprised of further subsystems and components. This substructuring approach can be conducted for $L \geq 1$ levels of subsystems, where only components exist at level $L + 1$. Components are directly associated with probability distributions describing their time-dependent probabilistic behavior. Note that the level 1 relates to the overall system level. Figure 6.3 illustrates the substructuring concept.

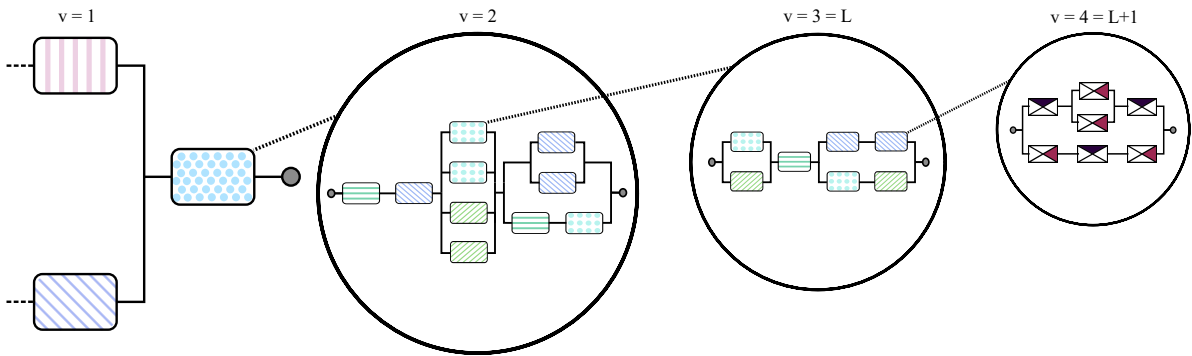


Figure 6.3: Illustration of the proposed substructuring concept.

Let there be n^v subsystems $\mathcal{S}_1^v, \mathcal{S}_2^v, \dots, \mathcal{S}_{n^v}^v$ and m^v components $\mathcal{C}_1^v, \mathcal{C}_2^v, \dots, \mathcal{C}_{m^v}^v$ at level $v = 1, 2, \dots, L$. During the analysis, the information on component behavior is propagated from level $L + 1$ to level 1 before determining the state s^0 of the overall system \mathcal{S} in dependence on various topological (sub)system structures. In the context of the resilience framework in Sec. 6.2, the state $s^0 \in S \subseteq \mathbb{R}^+$ with state space S of the overall system \mathcal{S} corresponds to the system performance $Q(t)$ that is basis for the resilience measure Res , see Eq. 6.2. Note that multiple resilience analyses might be conducted for various $Q(t)$. The quantity s^0 indicates system functionality from an ordered perspective and depends on the functionality of its directly subordinate subsystems and components. Given level $v = 1, \dots, L$, the dependency of the (sub)system state s_j^v on the state vector \mathbf{x}_j^v is modeled via the mapping $s_j^v = \phi_j^v(\mathbf{x}_j^v) \in \{0, 1\}$, where ϕ_j^v is a structure function, i.e., a topological rule for system functioning as presented in Sec. 6.3.1. The state vector is introduced as $\mathbf{x}_j^v = (s_1^w, s_2^w, \dots, s_{n_j^w}^w, c_1^w, c_2^w, \dots, c_{m_j^w}^w)$ for the j -th subsystem at level v with $j = 1, 2, \dots, n^v$ and $w = v + 1$. Thereby, $s_p^w, c_q^w \in \{0, 1\}$ denote the functionality of the p -th subsystem and q -th component, respectively. Further, n_j^w is the number of subsystems at level w contained in subsystem j at level v and $\sum_j^{n^v} n_j^w = n^w$. Analogously, m_j^w has the equivalent interpretation for components. At level $v = 1$, the notation reduces to $s^0 = \phi^0(\mathbf{x}^0)$. The state vectors at level L comprises only component states as

$\mathbf{x}_j^L = (c_1^w, c_2^w, \dots, c_{m_j^w}^w)$ with $j = 1, 2, \dots, n^L$, $c_i^w \in \{0, 1\}$ and $w = L + 1$.

The probability distributions governing the component states c_i^v are assumed to be known as CDF $F_k(t)$ for given component type k according to Sec. 6.3.3. Note that different subsystems might rely on the same component types. The assumption $s_j^v, c_i^v \in \{0, 1\}$ is due to the fact that the concept of survival signature is based on a binary-state consideration. However, multiple researchers work on extensions of the concept to a discrete or continuous multi-state consideration, see e.g. [30, 188–190].

6.4.2 Extension of the adapted systemic risk measure

In the resilience analysis of complex, substructured systems, it may be important that endowments can be formally assigned not only to system components but to other system structures, such as subsystems. To enable the incorporation of such endowment assignments in the novel methodology, the adapted systemic risk measure, cf. Sec. 6.2.2, is extended as follows.

Let a system, in addition to its m components, be given with a total of n subsystems $j \in \{1, \dots, n\}$ of $b_j \in \{1, 2, \dots, B\} \subseteq \mathbb{N}$ types over all system levels L with d endowment properties that influence the system performance $Q(t)$. Then, the subsystem j is characterized by

$$(\mathcal{S}_j; b_j) = (\xi_{j1}, \xi_{j2}, \dots, \xi_{jd}; b_j) \in \mathbb{R}^{(1 \times d)} \times \mathbb{N}, \quad (6.11)$$

where $(\xi_{j1}, \xi_{j2}, \dots, \xi_{jd}; b_j)$ are the numerical values of the d relevant endowment properties. The entire system is then, in addition to the description by the tuple consisting of the matrix $\mathbf{A} \in \mathbb{R}^{(m \times e)}$ and the column vector $\mathbf{z} \in \mathbb{N}^m$, capturing the components, described by the tuple composed of the matrix $\mathbf{D} \in \mathbb{R}^{(n \times d)}$ and the column vector $\mathbf{h} \in \mathbb{N}^n$, capturing the subsystems:

$$(\mathbf{D}; \mathbf{h}) = \begin{pmatrix} \xi_{11} & \xi_{12} & \cdots & \xi_{1d}; & h_1 \\ \xi_{21} & \xi_{22} & \cdots & \xi_{2d}; & h_2 \\ \vdots & \vdots & & \vdots & \vdots \\ \xi_{n1} & \xi_{n2} & \cdots & \xi_{nd}; & h_n \end{pmatrix}. \quad (6.12)$$

The system under consideration is defined via the descriptive, non-decreasing input-output model $Y = Y_{(\mathbf{A}; \mathbf{z}), (\mathbf{D}; \mathbf{h})}$ that is specified by both tuples and relates endowment properties to system performance. Again, with respect to Eq. 6.2, the model output is specified as $Y = Y_{(\mathbf{A}; \mathbf{z}), (\mathbf{D}; \mathbf{h})}$ dependent on the current endowment allocation for components $(\mathbf{A}; \mathbf{z})$ and subsystems $(\mathbf{D}; \mathbf{h})$. Then, with the specific acceptance set \mathcal{A} from Eq. 6.6, the extended adapted systemic risk measure is defined as

$$R(Y) = \left\{ \mathbf{A} \in \mathbb{R}^{m \times e}, \mathbf{D} \in \mathbb{R}^{(n \times d)} \mid Y_{(\mathbf{A}; \mathbf{z}), (\mathbf{D}; \mathbf{h})} \in \mathcal{A} \right\}, \quad (6.13)$$

that is the set of all endowment property allocations \mathbf{A} and \mathbf{D} such that the system reaches a resilience value greater or equal to α . Note that in this manner, equivalently, any performance-

influencing endowments, of any system structures, or even endowments independent of system structures, can be incorporated into the resilience decision-making analysis.

6.4.3 Augmentation of the resilience analysis

The system resilience Res is governed by the reliability, robustness and recoverability of a system as illustrated in Fig. 6.1. The magnitude of these quantities is influenced by the endowment allocations that are captured in the tuples $(A; z)$ and $(D; h)$. The assigned resilience-enhancing endowment properties $(\eta_{i1}, \eta_{i2}, \dots, \eta_{im})$ and $(\xi_{j1}, \xi_{j2}, \dots, \xi_{jd})$ can either relate to a specific quantity or a subset of the three quantities and correspond to different implementations in the overall system performance model, i.e., input-output model $Y_{(A;z),(D;h)}$.

The reliability is typically the most computationally challenging quantity when evaluating system resilience Res . Thus, this part of the computation is augmented by the concept of survival signature with its advantageous separation and compact storage properties as well as the fundamental substructuring approach proposed in the previous Sec. 6.4.1 in order to enable efficient resilience analyses of large and highly complex systems.

In a pre-processing step, the survival signatures $\Phi_j^v(l_1, l_2, \dots, l_K)$ of the $n = \sum_v^L n^v$ subsystems \mathcal{S}_j^v are computed based on the corresponding structure functions ϕ_j^v as described in Sec. 6.3.2. Subsequently, the survival signatures are utilized to efficiently retrieve the topological subsystem reliability (online) for varying endowment configurations.

In order to identify the set of all acceptable endowments $R(Y)$, repeated evaluations of $Y_{(A;z),(D;h)}$ are required according to the grid search algorithm – various endowment allocations in the search space spanned over discretized numerical values of $A \in \mathbb{R}^{m \times e}$ with $m = \sum_v^{L+1} m^v$ and $D \in \mathbb{R}^{(n \times d)}$ with $n = \sum_v^{L+1} n^v$ need to be evaluated analogous to Sec. 6.2.3. In each evaluation N stochastic simulations of $Y_{(A;z),(D;h)}$ have to be performed to obtain $E[Y]$, see Eq. 6.2, and corresponding status assignments according to the acceptance set \mathcal{A} in Eq. 6.6. Given the number of dimensions that need to be evaluated according to the grid search algorithm as M , the number of evaluations for $Q(t)$ is $M \cdot N \cdot u$ with u being the total number of time steps per simulation. Consequently, simulating system resilience is a complex, demanding and repeating challenge.

Computing the resilience directly relates to the computation of at least one structure function that can be any function that expresses the relation of interacting elements. The structure function can correspond to simple logical expressions, such as Reliability Block Diagrams (RBD) or fault trees, up to sophisticated simulation models, e.g., when assessing the network efficiency of a graph. In fact, such models often become extremely challenging in the context of real world systems. The evaluation of a global structure function including the entirety of all components at once might even be computationally unfeasible. In contrast, given a system in a substructured form \mathcal{S} as proposed in Sec. 6.4.1, the computation of the system functionality splits into the evaluation of multiple hierarchically ordered structure functions.

Such a consideration enables a wider range of application in terms of system size and complexity, especially when the computational capacity is limited.

The computational efficiency is further enhanced by application of the survival signature. Given a system, substructured according to Sec. 6.4.1 with $L \geq 2$, the computation of subsystem reliabilities can be propagated from level L to level 1 by evaluating the survival functions of subsystems \mathcal{S}_j^v based on the survival functions of \mathcal{S}_p^w instead of computing $s_j^v = \phi_j^v(\mathbf{x}_j^v)$ for each level. Coolen et al. proposed a methodology to merge survival signatures of specifically arranged subsystems in the context of substructured systems [29]. However, note that this approach differs from the one developed in the current paper. The survival function $P(T_{s_j^v} > t)$ of the j -th subsystem at level v is then computed according to Eq. 6.10 w.r.t. the survival signature $\Phi_j^v(l_1, l_2, \dots, l_K)$. At top-level 1, the failure rates of the subsystems \mathcal{S}_j^1 with $j = 1, 2, \dots, n^1$, utilized to sample subsystem functionality, can then be obtained via the cumulative hazard function and its derivative:

$$\lambda^{s_j^1}(t) = -\frac{d \ln P(T_{s_j^1} > t)}{dt}. \quad (6.14)$$

This enables to sample the subsystem state s_j^1 for time step (t_h, t_{h+1}) online with significantly reduced computational effort when evaluating the system resilience Res . The computation of Res then only involves $n^1 + m^1$ instead of $\sum_v^{L+1} m^v$ elements. In addition, significantly increased computational efficiency is achieved due to the separation of system topology and probability structure, the latter determined by the current endowment allocation. While the component probability structure varies, the topological reliability, independent of the endowment allocation, is captured in the survival signature in a compact manner and can be retrieved repeatedly with close to no costs. Note that subsystems of the same type share the same survival signature. This can be exploited for increased efficiency as well. In fact, the computational advantage of the proposed approach scales with size and complexity of the considered system \mathcal{S} . The developed and employed algorithm is outlined in Alg. 6.4.3 for illustrative purposes.

In order to prove efficiency and general applicability, the novel approach is applied to an arbitrary complex system in Sec. 6.6 and to the *U-Bahn* and *S-Bahn* system of Berlin in Sec. 6.7.

Algorithm 6.4.3

Step A Computation of the survival signatures for all subsystem \mathcal{S}_j^v with $v = 1, 2, \dots, L$ and $j = 1, 2, \dots, n^v$.

Step B Identification of the Pareto front by executing the grid search algorithm; each endowment allocation is evaluated by performing the following steps:

Step B1. Generation of the failure rate matrix with dimensions $n^1 \times T$ based on Eq. 6.14 for each subsystem and each timestep t_h with $h = 1, 2, \dots, u$ and generation of the failure rate matrix with dimensions $m^1 \times T$ for each component and

each timestep; if $L \geq 2$, the failure rate matrix for $v = 1$ for each subsystem is generated recursively from bottom to top by computing the survival functions.

Step B2. Perform N samples with time $t_h = 0$:

- a) Evaluate the system performance $Q(t_h)$.
- b) Sample possible failures of subsystems \mathcal{S}_j^1 for $j = 1, 2, \dots, n^1$ and components \mathcal{C}_i^1 for $i = 1, 2, \dots, m^1$ based on the failure rate matrices computed in Step B1.
- c) Check if any failed subsystem/component has recovered; if a subsystem/component recovers, set the time counter of its specific failure rate to 0.
- d) Set $t_h = t_{h+1} = t_h + \Delta t$ and repeat Steps a) – d) until $t_h = T$, i.e., the maximum time is reached.

Step B3. Obtain Res for the current endowment configuration via Eq. 6.2 and Eq. 6.3 over all time steps u and all samples N .

The complete algorithm has been implemented in the Julia package *ResilienceDecisionMaking.jl* and made publicly available on Github [100].

6.5 Multistage high-speed axial compressor

Axial compressors are complex, multi-component key elements of gas turbines. Therefore, it is critical in both design and maintenance to consider as many factors affecting system performance as possible to efficiently maximize compressor resilience. To address this challenge, the decision-making analysis proposed in [72] regarding system resilience is extended in order to deal with components, respectively factors, of different types.

6.5.1 Model

In [191], the authors present a functional model of a four-stage high-speed axial compressor from the Institute of Turbomachinery and Fluid Dynamics at Leibniz Universität Hannover, Germany, depicting its functionality as well as reliability characteristics. For detailed information about this particular axial compressor see [192–194].

The model captures the dependence of the overall performance of the compressor, i.e., the total-to-total pressure ratio and the total-to-total isentropic efficiency, on the surface roughness of the individual blades. These are arranged in rotor and stator rows. The model is based on the results of a sensitivity analysis of an aerodynamic model of the compressor and the so-called Relative Important Indices, cf. [58]. A network representation of the functional model is shown in Fig. 6.4. Each component represents either a stator (S1 - S4) or rotor (R1 - R4) row.

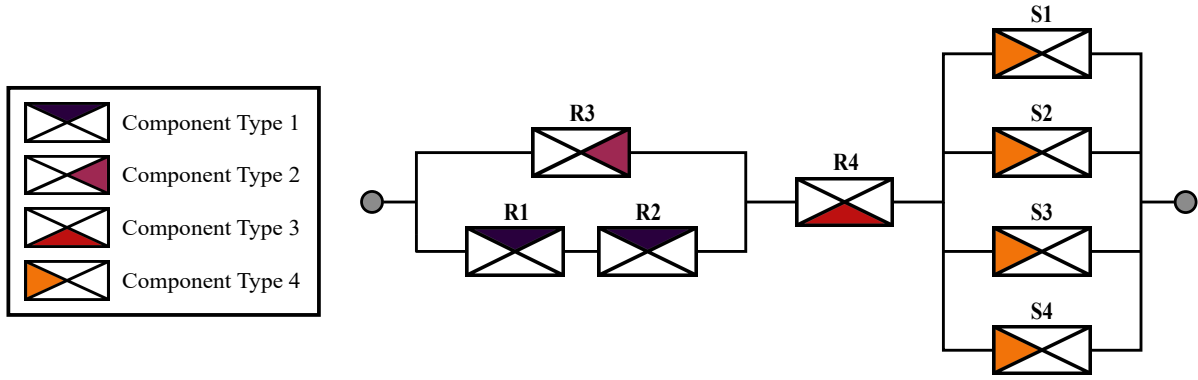


Figure 6.4: Functional model of the multistage high-speed axial compressor.

The rows are classified into $K = 4$ component types $k_i \in \{1, 2, 3, 4\} \forall i \in \{1, \dots, 8\}$. This classification, as well as the arrangement of the components, is based on the resulting effect of their blade roughness on the two performance parameters of the compressor. More precisely, an interruption between start and end implies that a roughness-induced performance variation of at least 25% is exceeded, corresponding to a non-functional compressor. This defines the system performance $Q(t)$ of the functional model for subsequent application of the resilience decision-making method. The system performance is determined at each time point t_h and is 1 if there is a path from start to end and 0 if this connection is interrupted. More detailed information about the functional model and its derivation can be obtained from [191].

For the resilience analysis, it is assumed that each row, i.e., each component of the functional model, is characterized by two endowment properties, a roughness resistance re and a recovery improvement rec , such that a component is fully described by $(a_i; k_i) = (re_i, rec_i; k_i)$. In this context, the roughness resistance can be interpreted as a qualitative coating that counteracts the roughening of the blade surfaces. Both the roughness resistance re_i and the recovery improvement rec_i of each row i are assumed to be functions of the component type k_i , i.e., $re_i = re_{i'}$, $rec_i = rec_{i'}$ if $k_i = k_{i'}$.

Each component of the functional model can fail randomly after system performance is calculated at time t_h . A failed component is considered as no longer being part of the model and does not contribute to the overall system performance at time t_{h+1} and at all subsequent times until it is completely recovered. The failure probability of a component i in the time interval (t_h, t_{h+1}) is assumed to be constant in time, cf. [191], and is specified by

$$P \{(a_i; k_i) \text{ fails during } (t_h, t_{h+1})\} = \Delta t \cdot \lambda_i \quad (6.15)$$

with

$$\lambda_i = 0.8 - 0.03 \cdot re_i, \quad (6.16)$$

where λ_i is the time-independent failure rate. Increasing the roughness resistance of a blade

row reduces the degradation of the surface and consequently the corresponding failure rate λ_i .

When a component i fails, its functionality is assumed to be immediately and completely recovered after a certain number of time steps, according to

$$r = r_{max} - rec_i \quad \text{with} \quad rec_i < r_{max} \quad (6.17)$$

where r_{max} is an upper bound on the number of time steps for recovery and rec_i is the recovery improvement that reduces the recovery duration. Note, that this recovery model corresponds to a one-step recovery profile and various alternative characteristic profiles of recovery are possible as well, cf., [164] and [153].

6.5.2 Costs of endowment properties

Optimal endowment properties are related to the quality of the components, and an increase in their production quality is associated with increasing costs. This should be taken into account in resilience decision-making. As discussed in [78], increasing the reliability of components in complex networks can be associated with an exponential increase in cost.

Increasing the endowment property of roughness resistance reduces the failure rate of blades in a row and thus improves reliability, see Eq. 6.15 and Eq. 6.16. Thus, its total cost is assumed to be

$$cost^{re} = \sum_{i=1}^8 price_{(re_i; k_i)}^{re} \cdot 1.2^{(re_i-1)}, \quad (6.18)$$

where re_i is the roughness resistance value of component i , k_i its type and $price_{(re_i; k_i)}^{re}$ an arbitrary common basic price. Accordingly an exponential relationship is assumed for the cost associated with recovery improvement:

$$cost^{rec} = \sum_{i=1}^8 price_{(rec; k_i)}^{rec} \cdot 1.2^{(rec_i-1)}. \quad (6.19)$$

The total cost $cost_{(A;z)}$ of an endowment is the sum of these costs:

$$cost_{(A;z)} = cost^{re} + cost^{rec}. \quad (6.20)$$

6.5.3 Scenario

In order to apply the decision-making method for resilience-enhancing endowments to the multistage high-speed axial compressor, the model parameter values and simulation parameter values shown in Tab. 6.1 are considered.

In a first step, the set of all acceptable endowments corresponding to a resilience value of at least $Res = 0.85$ over the considered time period is determined. Since any axial compressor blade improvement involves costs, the second step is to identify the most cost-efficient acceptable

Table 6.1: Parameter values for the resilience decision-making method for the functional model of the multistage high-speed axial compressor.

Parameter	Scenario
Acceptance threshold α	0.85
Number of time steps u	200
Length of a time step Δt	0.05
Maximum time T	10
Base failure rate λ	0.8
Roughness resistance re	$re_i \in \{1, \dots, 20\}$
Roughness resistance price $price_{(re_i; k_i)}^{re}$:	$800\text{€} \forall k_i \in \{1, 2, 3\}$ $500\text{€} \forall k_i = 4$
Maximum recovery time r_{max}	21
Recovery improvement rec	11
Recovery improvement price $price_{(rec; k_i)}^{rec}$:	600€
Sample size N	500

endowment, denoted as \hat{A} . The recovery improvement rec is assumed to be fixed for all components, regardless of the type, $rec_i = 11 \forall i \in \{1, \dots, m\}$ and the roughness resistance re is examined over $re_i \in \{1, \dots, 20\} \forall i \in \{1, \dots, m\}$. The roughness resistance values may be interpreted in ascending order as increasing quality levels of coatings.

Figure 6.5 illustrates the results of the grid search algorithm. It shows the roughness resistance combinations contained in $R(Y)$, i.e., all combinations that lead to a satisfying system resilience of at least $Res = 0.85$. It can be clearly seen that the roughness resistance of the blades of the fourth stage (component type 3) has the greatest influence on the system resilience. Combinations with coating qualities of $re_i \leq 15$ at the fourth stage are generally not sufficient to achieve an acceptable level of resilience, regardless of the endowment property values of the other component types. In addition, the roughness resistance of the four stators (component type 4) has the least influence on system resilience of all types. Here, a minimum coating quality of $re_i = 1$ as endowment is in various combinations already sufficient to achieve acceptable resilience values. The same applies to the rotors of component type 1 and type 2. However, the components of the other types require significantly higher coating qualities compared to the stators in order to compensate for the small roughness resistance values in these both types.

The design, maintenance and optimization of complex systems, such as an axial compressor, are invariably subject to monetary limitations. It is crucial for decision-making to be able to take these financial constraints into account. Therefore, Fig. 6.6 shows only those roughness resistance combinations included in $R(Y)$ that result in an acceptable system resilience of at least $Res = 0.85$ and are less expensive than a predefined cost limit for the total roughness resistance, that is arbitrarily assumed to be $cost^{re} = 40\,000\text{€}$ in this case study.

The results reveal that only configurations with low coating qualities for stators (component type 4) are below the cost limit. On the one hand, this is due to their aforementioned low

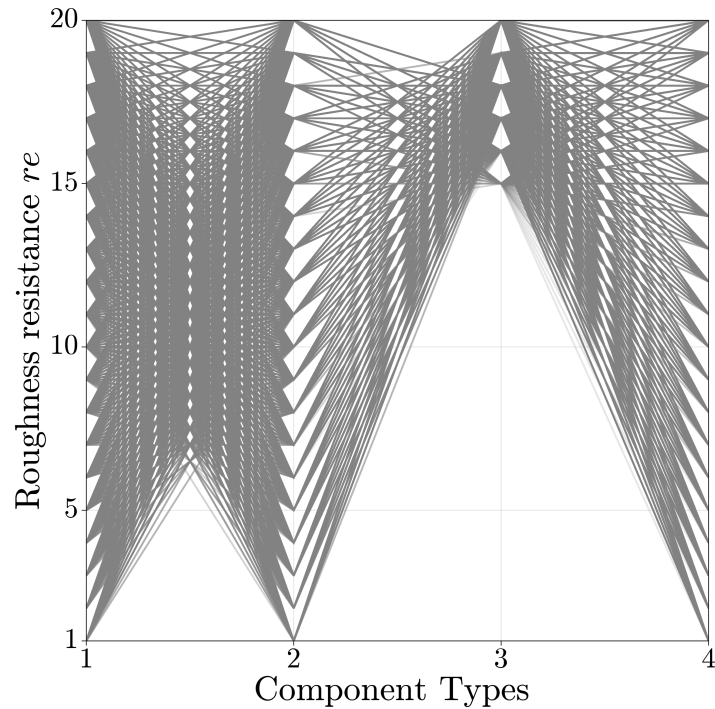


Figure 6.5: Numerical results of the 4D grid search algorithm for the functional model of the axial compressor with explored roughness resistance values.

influence on system resilience, and on the other hand to the high cost of the quality levels for the stators. Although the base price of 500€ is rather low, it is significantly higher in terms of cost for the entire component type than for the other types due to the higher total number of components of this type. In addition, only configurations that provide the highest quality levels of $re_i \geq 18$ for the type 3 rotor are acceptable and below the price limit. The roughness resistance of this rotor has such a large impact on system resilience that at lower quality levels, compensation by higher quality levels of the remaining stages would exceed the given budget. Although the roughness resistance of the rotor of component type 2 has a lower influence on the system resilience than that of component type 3, minimal quality levels of the coating can not be compensated by high qualities of the other components. Therefore, at least $re_i = 5$ for $k_i = 2$ is required to fulfill the acceptance criterion.

The grid search algorithm is able to reduce the numerical effort for the calculation of $R(Y)$ by about 98%. As a result, only 2% of the potential combinations of roughness resistance values need to be evaluated.

Taking into account the base prices in Tab. 6.1, the most cost-efficient endowment is characterized by roughness resistances of $re_i = 7$ for $k_i = 1$, $re_i = 13$ for $k_i = 2$, $re_i = 19$ for $k_i = 3$ and $re_i = 1$ for $k_i = 4$ for the respective components. In Fig. 6.6 the corresponding configuration is highlighted in blue. The final cost results from Eq. 6.20 as $cost_{(\hat{A};z)} = cost^{re} + cost^{rec} = 35\,209\text{€} + 29\,720\text{€} = 64\,929\text{€}$.

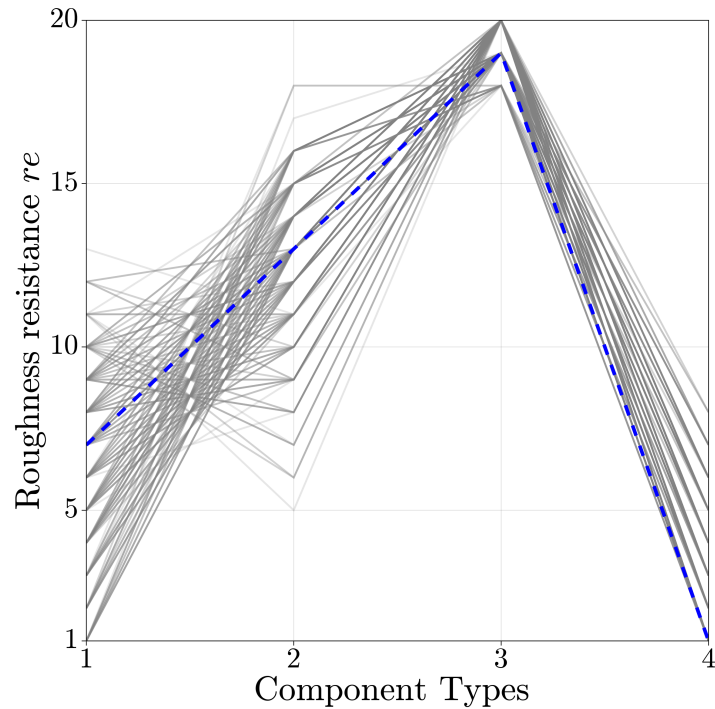


Figure 6.6: Numerical results of the 4D grid search algorithm for the functional model of the axial compressor with explored roughness resistance values and a cost threshold for roughness resistance of 40 000€.

6.6 Complex system

In [37] and [181] the authors apply their introduced simulation approaches for reliability analysis on an arbitrary complex system. In order to demonstrate the wide applicability and efficiency of the proposed methodology developed in this paper, this complex system is considered, adapted by means of substructuring, and an efficient resilience decision-making analysis is conducted.

6.6.1 Model

The arbitrary complex system consists of $n = 14$ subsystems, each assigned to one of $B = 6$ subsystem types. Figure 6.7 illustrates the complex system and the assignment of subsystems to their types. A connection between start node and target node indicates a functioning state and an interruption of this connection indicates a non-functioning state of the overall system. This defines the system performance $Q(t)$ of the functional model for subsequent application of the resilience decision-making method. The system performance is determined at each time point t_h and is 1 if there is a path from start to end and 0 if this connection is interrupted. Note that the complex system is thus formally an RBD. For illustration and simplicity, it is assumed that there is only one level of subsystems, i.e., $l = L = 1$, and thus $\mathbf{x}^s = (s_1, s_2, \dots, s_{14})$, $\mathcal{S}_j^1 = \mathcal{S}_j$, and $\lambda^{s_j^1}(t) = \lambda^{s_j}(t)$. Figure 6.8 illustrates the structure of the six subsystem types. These are formally RBDs as well. It is assumed that each subsystem of the same type is represented by

the same RBD. A subsystem \mathcal{S}_j is considered to be functional if a connection exists from start to end and non-functional if this connection is interrupted, i.e., $s_j \in \{0, 1\} \forall j \in \{1, \dots, 14\}$. Depending on the type, the subsystems consist of seven to ten components. Thus, the overall system is composed of $m = 106$ individual components.

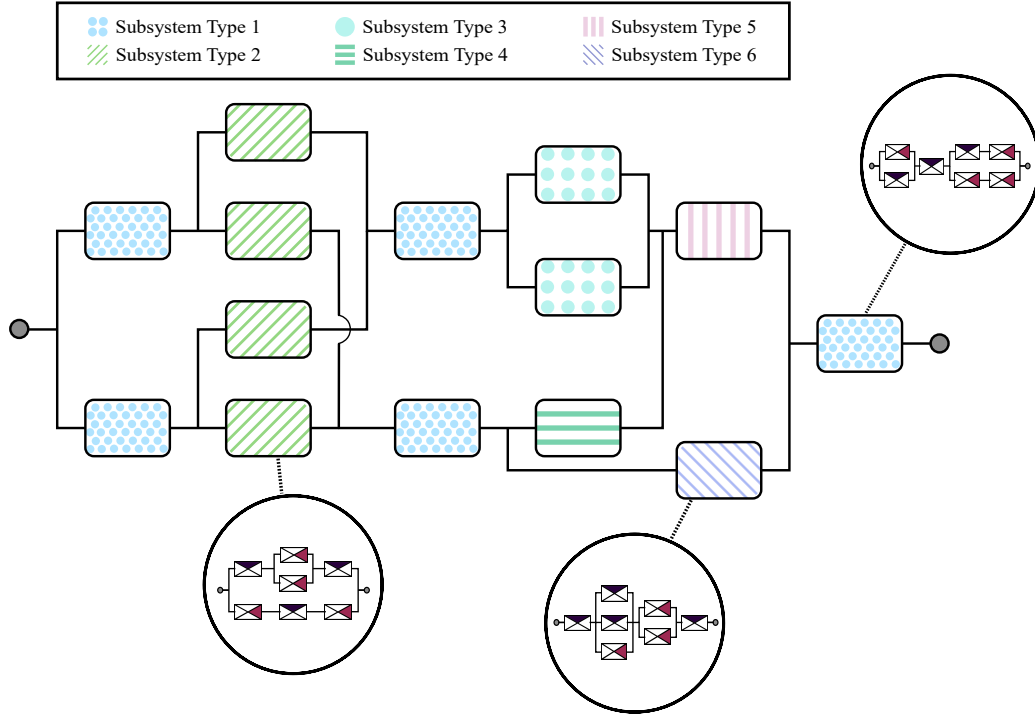


Figure 6.7: Representation of the arbitrary complex system with 14 components, adapted from [37].

The components are classified into $K = 2$ types $k_i \in \{1, 2\} \forall i \in \{1, \dots, 106\}$, i.e., 50 components of type 1 and 56 components of type 2. For the resilience analysis, each component of the model, is assumed to be characterized by an endowment property, that is the reliability improvement rel , such that a component is fully described by $(a_i; k_i) = (rel_i; k_i)$. Note that the reliability improvement rel_i of each component i is assumed to be function of the component type k_i , i.e., $rel_i = rel_{i'}$ if $k_i = k_{i'}$. Further, each component type, and thus each component, is characterized by a specific time-dependent failure behavior. In practice, the underlying distribution functions, describing this behavior, need to be derived from existing operational data. However, the consideration of real data is often highly challenging due to the inherent uncertainty caused by, e.g., lack of data, measurement inaccuracies, subjective expert knowledge, small sample sizes, etc. New developments in the context of the survival signature as introduced, e.g., in [181], allow for the efficient consideration and propagation of uncertainties through the entire model. They will be incorporated into the proposed methodology towards an imprecise resilience approach in future work of the authors. However, for the purpose of proof of concept and applicability, exponential distributions are considered for both component types in this

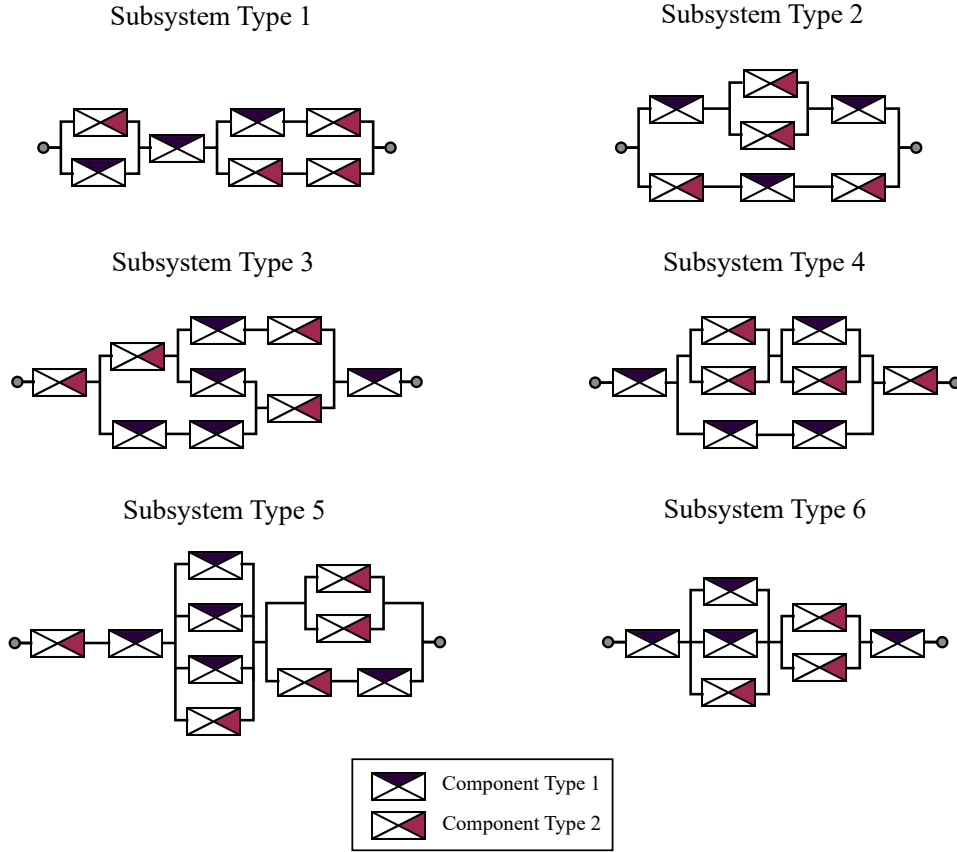


Figure 6.8: Representation of the $B = 6$ subsystem types of the complex system.

case study as

$$F_i(t; \lambda_i(\text{rel}_i)) = 1 - e^{-\lambda_i(\text{rel}_i)t} \text{ for } t \geq 0, \quad (6.21)$$

with

$$\lambda_i(\text{rel}_i) = \lambda_{i,\max} - \Delta\lambda_i \cdot \text{rel}_i, \quad (6.22)$$

being the failure rate of component i of type k depending on the corresponding reliability improvement rel_i . $\lambda_{i,\max}$ is the maximum failure rate and $\Delta\lambda_i$ denotes the failure rate reduction per reliability improvement rel_i that is assumed to be constant for each component type, leading to equidistant failure rate variations.

The simulation can be summarized as follows: after the system performance has been computed at time t_h , each subsystem \mathcal{S}_j of the complex system can fail at random based on the extracted and time-dependent failure rate $\lambda^{s_j}(t_h)$ from corresponding survival function, cf. Eq. 6.14. A failed subsystem is treated as no longer present in the model and does not contribute to the overall system performance $Q(t)$ at time t_{h+1} and all subsequent time points until it is fully recovered. The failure probability of a subsystem \mathcal{S}_j in the time interval (t_h, t_{h+1}) is

$$P \{ \mathcal{S}_j \text{ fails during } (t_h, t_{h+1}) \} = \Delta t \cdot \lambda^{s_j}(t_h). \quad (6.23)$$

If a subsystem \mathcal{S}_j failed, its functionality is assumed to be immediately and fully recovered after r time steps, again corresponding to a one-step recovery profile. It is assumed that a repaired subsystem and thus all components of the subsystem are in as-new original condition after repair. Note that this is an assumption for the sake of demonstration, and in reality deviating states might be obtained after repair, possibly depending on further endowment properties that affect the duration and quality of recovery. After recovery, the survival function of a subsystem is time-zeroed, such that the resulting failure rate per simulation step $\lambda^{s_j}(t_h)$ evolves over time equivalent to that of a subsystem in new condition.

6.6.2 Costs of endowment properties

The improvement of endowment properties is inevitably associated with costs. Increasing the endowment property “reliability improvement” reduces the failure rate of components and consequently of corresponding subsystems. Again, an exponential relationship between costs and improvements is assumed. Then the total costs can be defined as

$$cost_{(A;z)} = cost^{rel} = \sum_{i=1}^{106} price_{(rel_i;k_i)}^{rel} \cdot 1.2^{(rel_i-1)}, \quad (6.24)$$

where $(rel_i; k_i)$ is the reliability improvement value of component i , k_i its type and $price_{(rel_i;k_i)}^{rel}$ is an arbitrary common basic price.

6.6.3 Scenario

The considered model parameters and simulation parameters values for the application of the resilience decision-making method for complex and substructured systems to the arbitrary complex system illustrated in Fig. 6.7, are shown in Tab. 6.2. The recovery is assumed to be fixed with $r = 20$ time steps for all subsystems, regardless of the type. The reliability improvement rel_i is explored over $rel_i \in \{1, \dots, 10\} \forall i \in \{1, \dots, m\}$.

In a pre-processing step, the survival signatures of all 14 subsystems are determined. As an example, Tab. 6.3 depicts the survival signature values of subsystem type 5 of the complex system. For clarity, only the non-trivial survival signature values are shown, i.e., all values that are neither zero or one. Then the analysis starts as follows: In a first step, the set of all acceptable endowment configurations $R(Y)$, corresponding to a resilience value of at least $Res = 0.9$ over the considered time period, is determined according to Algorithm 6.4.3. Since any improvement of the system components is associated with costs, the second step is to identify the most cost-efficient acceptable endowment \hat{A} .

Figure 6.9 illustrates the results of the grid search algorithm. It shows the reliability improvement combinations contained in $R(Y)$, i.e. all combinations that lead to a satisfying system resilience. It can be seen, that the reliability improvement of components of type 1 is

Table 6.2: Parameter values for the resilience decision-making method on the arbitrary complex system.

Parameter	Scenario
Acceptance threshold α	0.90
Number of time steps u	200
Length of time step Δt	0.05
Maximum time T	10
Maximum failure rate $\lambda_{i,\max}$	$\lambda_{i,\max} = 0.15$ for $k_i = 1$ $\lambda_{i,\max} = 0.20$ for $k_i = 2$
Failure rate reduction $\Delta\lambda_i$	$\Delta\lambda_i = 0.014$ for $k_i = 1$ $\Delta\lambda_i = 0.019$ for $k_i = 2$
Reliability improvement rel_i	$rel_i \in \{1, \dots, 10\}$ for $k_i \in \{1, 2\}$
Reliability improvement price $price_{(rel_i;k_i)}^{rel}$	$price_{(rel_i;1)}^{rel} = 1\,000\text{€}$ $price_{(rel_i;2)}^{rel} = 2\,000\text{€}$
Recovery time steps r	20
Sample size N	500

more important, i.e., has a higher impact on the overall system resilience than the reliability improvement of components of type 2. For maximum reliability improvement values for type 1, i.e., $rel_i = 10$ for $k_i = 1$, even low reliability improvement values for type 2, i.e., $rel_i = 2$ for $k_i = 2$, are sufficient in order to fulfill the acceptance criterion and reach system resilience values of at least $Res = 0.90$. On the other hand, with maximum reliability improvement for components of type 2, i.e., $rel_i = 10$ for $k_i = 2$, a moderate reliability improvement for type 1 of at least $rel_i = 4$ for $k_i = 1$ is required to meet the acceptance criterion.

These results are plausible, since a detailed examination of the subsystem types and their topology, cf. Fig. 6.8, reveals that components of type 1 hold a total of six so-called bottleneck positions within the subsystems, i.e., positions where the failure of a single component interrupts the functioning of the entire subsystem, while components of type 2 occupy only three of these positions. This results in a higher influence of component type 1 on the functionality of the subsystems and thus ultimately in a higher influence on overall system resilience. Accordingly, the quality of reliability improvement of component type 1 is more relevant than that of component type 2. Looking at the probabilistic structure of the components, it is noticeable that the failure rate reduction for components of type 2 is greater than for components of type 1, i.e., the increase in reliability improvement for type 2 probabilistically generates a higher surplus value compared to improvements of type 1. However, this obviously cannot balance the influence gradient between both types and thus underlines the critical topological importance of type 1 components.

The design, maintenance and optimization of complex systems is typically restricted by economic limitations. It is crucial for decision-making to be able to take these monetary constraints into account. Assuming the arbitrary base prices in Tab. 6.2, the most cost-effective

Table 6.3: Non-trivial survival signature values of subsystems with $b_j = 5$ of the complex system, shown in Fig. 6.7 and Fig. 6.8.

l_1	l_2	$\Phi(l_1, l_2)$
2	4	1/25
3	3	3/50
2	5	3/25
4	3	3/20
3	4	9/50
2	6	1/5
5	3	11/50
6	3	3/10
3	5	3/10
4	4	33/100
3	6	2/5
5	4	23/50
4	5	12/25
6	4	3/5
4	6	3/5
5	5	16/25
6	5	4/5
5	6	4/5

acceptable endowment \hat{A} is specified by a reliability improvement configuration of $rel_i = 8$ for $k_i = 1$ and $rel_i = 4$ for $k_i = 2$ for the respective components. In Fig. 6.9, the corresponding configuration is highlighted. Note that due to the monotonicity of the input-output model and the monotonically increasing endowment costs, the most cost-efficient endowment can only be located on the dominant vertices of the Pareto front. Therefore, only these configurations need to be examined in terms of cost. The final cost results from Eq. 6.24 as $cost_{(\hat{A};z)} = 372\,695\text{€}$.

Due to the utilization of the grid search algorithm, the numerical effort required to compute $R(Y)$ is reduced. Only 23% of all possible configurations of reliability improvement values need to be evaluated. This reduction effect scales with the size and dimensionality of the endowment search space. By means of the novel resilience decision-making method, the considered complex system could be reduced from its entirety of 106 individual components to 14 components on the top-level with respect to the resilience analysis and the associated identification of all acceptable endowment configurations, which drastically reduces the computational effort. Nevertheless, all 106 components and their influence were considered by incorporating and propagating the subsystems' survival functions. Again, this effect scales with increasing complexity and size of the investigated systems.

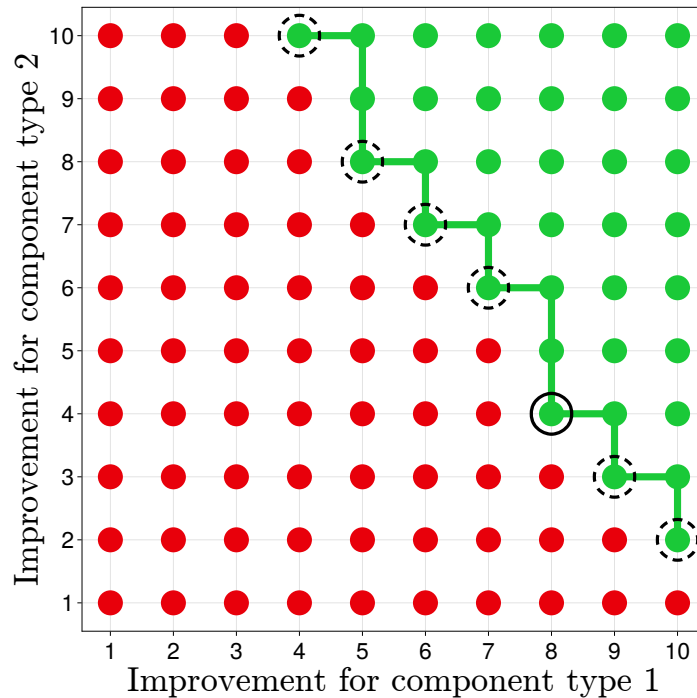


Figure 6.9: Numerical results of the 2D grid search algorithm for the complex system with explored reliability improvement values.

6.7 *U-Bahn* and *S-Bahn* system of Berlin

About two thirds of the total of 1.5 billion passengers per year are transported by Berlin’s subway *U-Bahn* and suburban trains *S-Bahn* [195, 196], making these two transport services the most used means of public transport in Berlin and thus of utmost importance for the German capital. Key infrastructures that are of such significant social and economic relevance to modern societies obviously and inevitably need to be as resilient as humanly possible. The applicability of the methodology developed in this work to large complex systems is demonstrated on a comprehensive model of the Berlin *U-Bahn* and *S-Bahn* system. The objective is to identify suitable resilience-enhancing properties for all stations in the system, taking into account monetary constraints. This allows the characterization of acceptable endowments for the system in terms of reliability, robustness, and recoverability. This approach can be applied not only to any phase during the life cycle of existing systems, but also to systems in the design phase, in order to optimize their resilience.

6.7.1 Model

Berlin’s *U-Bahn* and *S-Bahn* systems are highly interconnected systems that are linked by numerous stations. According to [197], they may therefore be considered as a unified system, hereafter referred to as “metro system”. In [72] the authors apply their introduced approach

for resilience decision-making to a model of the Berlin metro system. In order to demonstrate the wide applicability and efficiency of the proposed methodology developed in this work, this model is considered, extended and adapted by means of substructuring, and an efficient and multidimensional resilience decision-making analysis is conducted.

In [198], Zhang et al. proposed how mapping of metro networks into topological graphs can be conducted. Based on this, the Berlin metro system consists of 306 nodes for 306 metro stations and 350 edges for 350 connections between these stations. For simplicity, parallel connections are mapped to single edges in the model, and are assumed to be undirected. These assumptions reduce the complexity of the metro system. Figure 6.10 illustrates the graph representation.

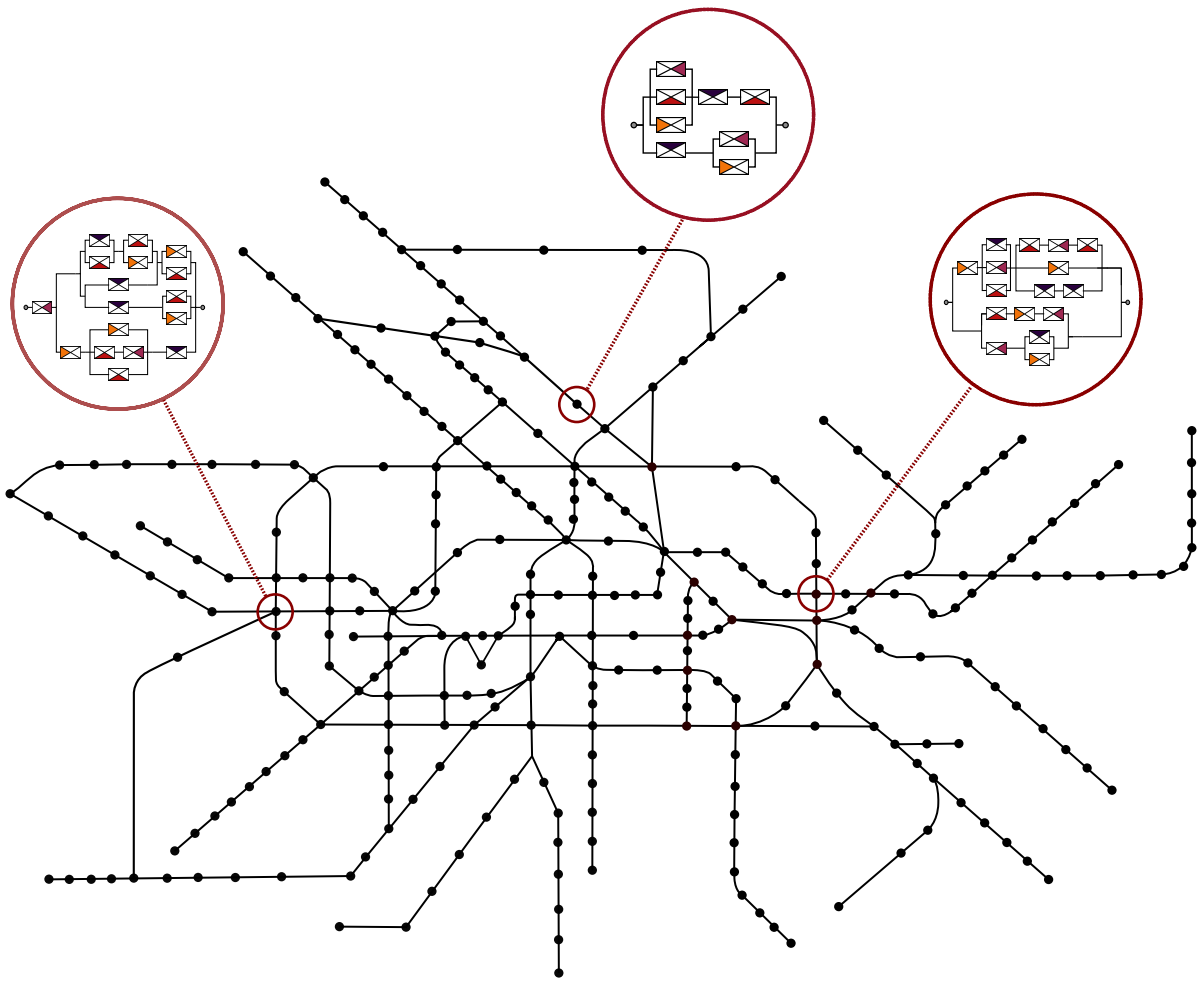


Figure 6.10: Topological network for the Berlin metro system.

The functionality of systems depends on the functionality of its components. However, the functionality of these components often depends again on the functionality of a variety of subcomponents, etc. A major challenge in modeling is therefore determining an appropriate level of detail.

The resilience decision-making methodology proposed in this paper allows for the incorporation of such subsystem structures by live propagation of corresponding reliability characteristics up to the top-level. Therefore, for the resilience analysis of the metro system, each metro station is modeled as a subsystem with own functionality and performance function. Again, for illustrative purposes and sake of convenience, assume that there is only one level of subsystems, i.e., $l = L = 1$, and thus $\mathbf{x}^s = (s_1, s_2, \dots, s_{306})$, $\mathcal{S}_j^1 = \mathcal{S}_j$ and $\lambda^{s_j^1}(t) = \lambda^{s_j}(t)$ with $n = 306$ subsystems respectively metro stations.

In terms of reliability modeling, subcomponents could correspond to structural elements, such as stairs, columns, ceilings, station rails as well as electric facilities, such as railway power supply, elevators, escalators, ventilation plants, information systems and illuminations. These subcomponents can be subdivided in terms of their functionality and relevance to the metro station, such as in rail operations related components and user accessibility related components. For illustrative purpose, the analysis is restricted to reliability modeling of metro stations. Therefore, functional models are defined for the metro station subsystems that are, as in the previous case study, formally RBDs. Again, a subsystem \mathcal{S}_j is considered to be functional if a connection from start to end exists and non-functional if this connection is interrupted, i.e., $s_j \in \{0, 1\} \forall j \in \{1, \dots, 306\}$. Figure 6.10 illustrates three of these subsystems for three different metro stations as an example. The metro stations are classified into $B = 6$ types, depending on the number of their connections to direct neighbors, i.e., stations with only one connection form subsystem type 1, stations with two direct neighbors form subsystem type 2, etc. For the analysis, each subsystem is assumed to be characterized by an endowment property, that is the recovery improvement rec , such that a metro station j with type b_j is described by $(\mathcal{S}_j; b_j) = (rec_j; b_j)$. Note, that the recovery improvement rec_j of each metro station is assumed to be a function of the station type b_j , i.e., $rec_j = rec_{j'}$ if $b_j = b_{j'}$. For simplicity, it is assumed that each metro station of a type is represented by the same RBD. Figure 6.11 displays the structure of all six subsystem types and Fig. 6.12 tabulates the number of individual metro stations per type.

Depending on the type and thus with increasing complexity related to the number of direct neighbors, also known as node degree, the subsystems consist of four up to twenty-one components. Taking into account the information from Fig. 6.12, the overall system therefore consists of a total of $m = 2776$ considered individual components.

The components are classified into $K = 4$ types $k_i \in \{1, 2, 3, 4\} \forall i \in \{1, \dots, 2776\}$. For the analysis, each component is assumed to be characterized by an endowment property, that is the reliability improvement rel , such that a component is fully described by $(a_i; k_i) = (rel_i; k_i)$. Note, that the reliability improvement rel_i of each component is assumed to be a function of the component type k_i , i.e., $rel_i = rel_{i'}$ if $k_i = k_{i'}$. Further, each component is characterized by a specific time-dependent failure behavior. For the purpose of proof of concept and applicability, for component type 1 and 3, i.e., $k_i = 1$ and $k_i = 3$, exponential distributions are considered according to Eq. 6.21 and Eq. 6.22. For component type 2 and 4, i.e., $k_i = 2$ and $k_i = 4$, two

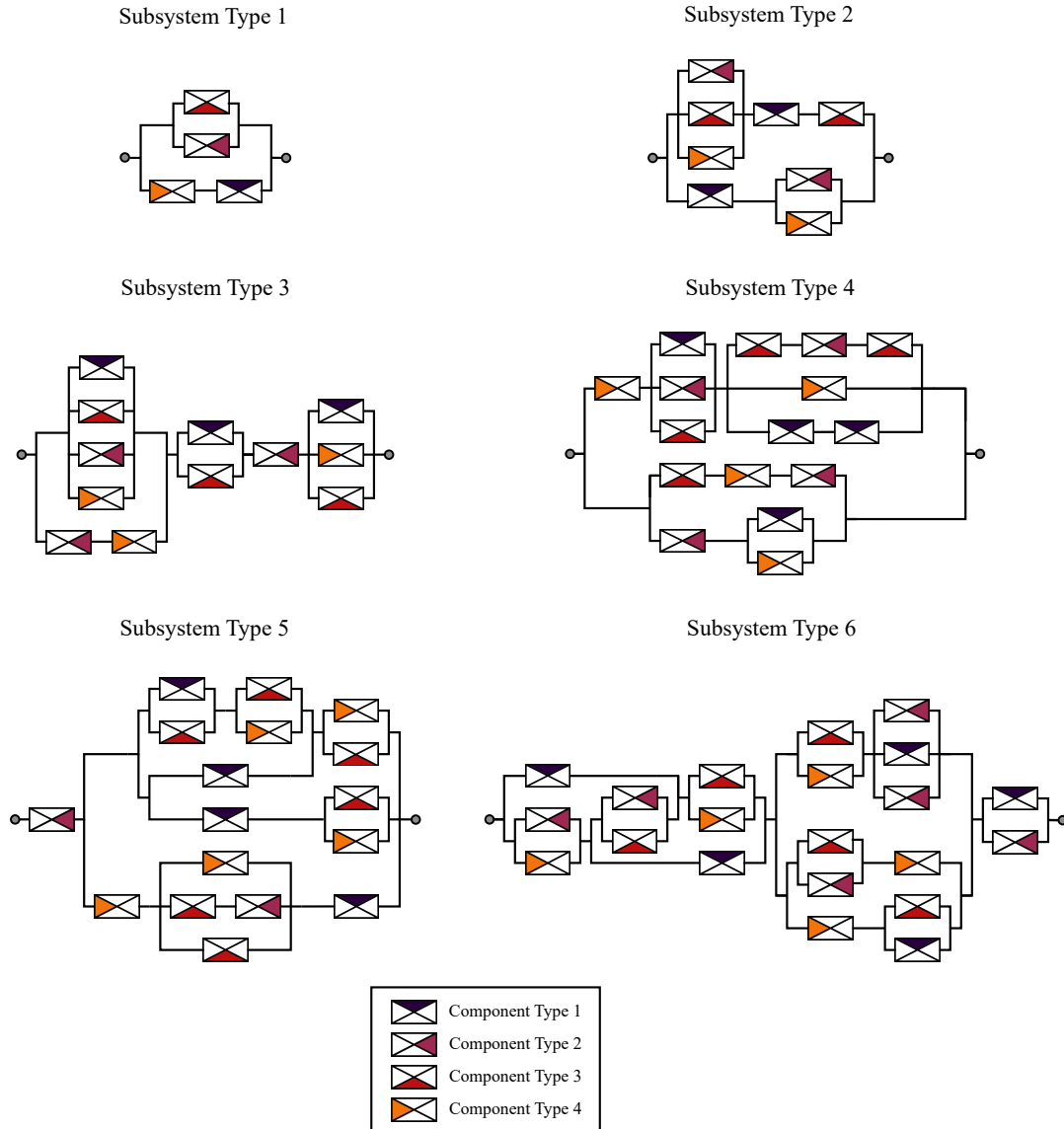


Figure 6.11: Representation of the $B = 6$ station types of the Berlin metro system.

parametric gamma distributions are considered. The cumulative distribution function of the gamma distribution can be derived based on its probability density function that is given in terms of the rate parameter $\lambda_i(rel_i)$ depending on the current reliability endowment value rel_i

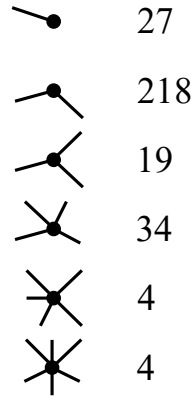


Figure 6.12: Number of individual metro stations per type.

of component i of type k_i as

$$f(t; \alpha_i, \lambda_i(\text{rel}_i)) = \frac{t^{\alpha_i-1} e^{-\lambda_i(\text{rel}_i)t} \lambda_i(\text{rel}_i)^{\alpha_i}}{\Gamma(\alpha_i)}, \quad (6.25)$$

for $t, \alpha_i, \lambda_i(\text{rel}_i) > 0$, where α_i is the shape parameter, $\lambda_i(\text{rel}_i)$ is the rate parameter, and $\Gamma(\alpha_i)$ is the well-known Gamma function. Consequently, the cumulative distribution function can be obtained by integration and with respect to the current endowment of component i it can be formulated as

$$F(t; \alpha_i, \lambda_i(\text{rel}_i)) = \int_0^t f(u; \alpha_i, \lambda_i(\text{rel}_i)) du. \quad (6.26)$$

$\lambda_i(\text{rel}_i)$ is again a function of the component specific reliability improvement and given by Eq. 6.22.

In order to perform a resilience analysis, the definition of an appropriate system performance measure for the metro system is imperative. As in [198] and [72], in this case study, the so-called network efficiency E_f is adopted as the relevant performance measure, i.e., $Q(t) = E_f(t)$. Zhang et al. justified in [198] their choice by stating that connectivity between individual metro stations is an essential criterion for evaluating metro operations. As described by Latora and Marchiori in [134], network efficiency is a quantitative indicator of network connectivity and is defined as:

$$E_f = \frac{1}{n(n-1)} \sum_{u \neq v} \frac{1}{d_{uv}} \quad (6.27)$$

with n the number of subsystems, i.e., metro stations in the network and d_{uv} the path length between metro station u and metro station v , i.e., the shortest distance between these stations. A comprehensive overview of algorithms to efficiently determining the path length d_{uv} between stations, such as the algorithms of Floyd, Dijkstra's, or Bellman-Ford, is provided in [199] and [200].

The simulation procedure corresponds to that from the previous case study and the failure

probability of a subsystem \mathcal{S}_j , i.e., metro station, in the time interval (t_h, t_{h+1}) is defined by Eq. 6.23. Unlike in the previous case study, a failed metro station is not entirely removed from the system, but remains in the set of metro stations; however, their node degree becomes 0, i.e., all existing connections to direct neighbors are removed. This assumption is essential, as the computation and interpretation of the system performance network efficiency depends on the number of nodes. The case study therefore relies on the fact that the number of nodes is constant.

If a subsystem \mathcal{S}_j failed, its functionality is assumed to be immediately and fully recovered after a certain number of time steps r :

$$r = r_{\max} - 2 \cdot rec_j \quad \text{with} \quad rec_j < r_{\max}, \quad (6.28)$$

where rec_j is the recovery improvement specific to the station \mathcal{S}_j and r_{\max} is an upper bound for number of time-steps for recovery. After recovery, all previous connections to other metro stations are assumed to be restored, unless these are in a state of failure. As each time-step has a specific length of $\Delta t = (T/u)$, the duration of the recovery process is $r \cdot (T/u)$. Again, this recovery model corresponds to a one-step recovery profile and as mentioned before, various alternative characteristic profiles of recovery are possible as well. A repaired station and thus all components of the station are assumed to be in a as-new original condition after repair. This is an assumption for the sake of demonstration, and deviating states are possible. After recovery, the survival function of a metro station is time-zeroed, such that the resulting failure rate per simulation step $\lambda^{s_j}(t_h)$ evolves over time equivalent to that of a station in new condition.

6.7.2 Costs of endowment properties

The improvement of both endowment properties, “reliability improvement” and “recovery improvement”, is inevitably associated with costs. Again, exponential relationships between total costs and improvements are assumed:

$$cost^{rel} = \sum_{i=1}^{2776} price_{(rel_i; k_i)}^{rel} \cdot 1.2^{(rel_i-1)}, \quad (6.29)$$

where rel_i is the reliability improvement value of component i , k_i its type and $price_{(rel_i; k_i)}^{rel}$ an arbitrary common basic price. Accordingly an exponential relationship is assumed for the total cost associated with recovery improvement:

$$cost^{rec} = \sum_{j=1}^{306} price_{(rec_j; b_j)}^{rec} \cdot 1.2^{(rec_j-1)}, \quad (6.30)$$

where rec_j is the recovery improvement value of station j , b_j its type and $price_{(rec_j; b_j)}^{rec}$ an arbitrary common basic price. The total cost $cost_{(A;z),(D;h)}$ of an endowment is the sum of

these costs:

$$cost_{(A;z),(D;h)} = cost^{rel} + cost^{rec}. \quad (6.31)$$

In practice, it is crucial to include the economic aspects of failure and recovery processes in detail in the resilience assessment. Mitigating resilience losses through system improvements imposes direct costs on stakeholders, such as improving component properties. Note, however, that for a comprehensive analysis, it is important to also consider indirect costs to the affected population and businesses, when the performance of a key system declines, as stated in [173]. Further, it is reasonable to incorporate the subjective preferences of stakeholders into the resilience assessment, as suggested in [172]. These considerations have the potential to significantly influence the outcome of a resilience decision-making process. Therefore, they should be integrated into the proposed methodology in future work by including additional cost conditions and discount rates for the corresponding deterioration and recovery sequences.

6.7.3 Scenario

In order to apply the resilience decision-making method to the Berlin metro system illustrated in Fig. 6.10, the model parameter and simulation parameter values, shown in Tab. 6.4, are considered. The recovery improvement rec_j is explored over $rec_j \in \{1, \dots, 10\} \forall j \in \{1, \dots, 306\}$, but considered to be equal for each station, regardless of the type b_j . The reliability improvement rel_i again is explored over $rel_i \in \{1, \dots, 10\} \forall i \in \{1, \dots, 2776\}$ for $k_i \in \{1, \dots, 4\}$.

In a pre-processing step, the survival signatures of all 306 metro stations are determined. As an example, Tab. 6.5 illustrates the non-trivial survival signature values, i.e., $\Phi(l_1, \dots, l_4) \neq 0$ and $\Phi(l_1, \dots, l_4) \neq 1$, of station type 2 of the metro system. Then, the set of all acceptable endowment configurations $R(Y)$, corresponding to a resilience value of at least $Res = 0.99$ over the considered time period, is determined according to Algorithm 6.4.3. Further, as any improvement of the system components and stations is associated with costs, the most cost-efficient acceptable endowment, denoted by the tuple (\hat{A}, \hat{D}) , is determined.

In Fig. 6.13 the results of the grid search algorithm are illustrated. It shows the accepted endowments contained in $R(Y)$, i.e. all combinations that lead to a satisfying resilience of the metro system. It is clearly visible that type 1 components as well as the recovery improvement of the metro stations have the greatest influence and thus the highest importance for the metro system. Only endowments with a reliability improvement of at least $rel_i = 8$ for type 1 components and endowments with a recovery improvement for all metro stations of at least $rec_j = 8$ lead to a system resilience meeting the acceptance criterion. In addition, type 2 components are of considerable relevance. Here, only endowments with a reliability improvement of at least $rel_i = 6$ are acceptable. The reliability improvements of type 3 and 4 components, on the other hand, are of less significance. For both types of components, there are numerous acceptable configurations that include minimum reliability improvement values for one of these types.

Table 6.4: Parameter values for the resilience decision-making method on the metro system of Berlin.

Parameter	Scenario
Acceptance threshold α	0.99
Length of time step Δt	0.05
Number of time steps u	200
Maximum time T	10
Shape parameter gamma distribution α_i	$\alpha_i = 1.2$ for $k_i = 2$ $\alpha_i = 2.6$ for $k_i = 4$
Maximum failure rate $\lambda_{i,\max}$	$\lambda_{i,\max} = 0.34$ for $k_i = 1$ $\lambda_{i,\max} = 0.43$ for $k_i = 2$ $\lambda_{i,\max} = 0.36$ for $k_i = 3$ $\lambda_{i,\max} = 0.66$ for $k_i = 4$
Failure rate reduction $\Delta\lambda_i$	$\Delta\lambda_i = 0.03$ for $k_i = 1$ $\Delta\lambda_i = 0.04$ for $k_i = 2$ $\Delta\lambda_i = 0.034$ for $k_i = 3$ $\Delta\lambda_i = 0.051$ for $k_i = 4$
Reliability improvement rel_i	$rel_i \in \{1, \dots, 10\}$ for $k_i \in \{1, \dots, 4\}$
Reliability improvement price $price_{(rel_i; k_i)}^{rel}$	$price_{(rel_i; 1)}^{rel} = 100\text{€}$ $price_{(rel_i; 2)}^{rel} = 200\text{€}$ $price_{(rel_i; 3)}^{rel} = 200\text{€}$ $price_{(rel_i; 4)}^{rel} = 400\text{€}$
Maximum recovery time r_{max}	22
Recovery improvement rec_j	$rec_j \in \{1, \dots, 10\}$
Recovery improvement price $price_{(rec_j; b_i)}^{rec}$	$price_{(rec_j; 1)}^{rec} = 100\text{€}$ $price_{(rec_j; 2)}^{rec} = 200\text{€}$ $price_{(rec_j; 3)}^{rec} = 300\text{€}$ $price_{(rec_j; 4)}^{rec} = 400\text{€}$ $price_{(rec_j; 5)}^{rec} = 500\text{€}$ $price_{(rec_j; 6)}^{rec} = 600\text{€}$
Sample size N	500

These results again prove to be plausible, as in the previous case study, upon closer examination of the topological structures of the metro system and its subsystems. Several U-Bahn and S-Bahn lines start and end in long chains of directly interconnected type 2 stations, see Fig. 6.10. The resilience analysis of the Berlin metro system published in [72] revealed that especially an interruption of these chains has a major negative impact on the network efficiency and thus on the resilience of the metro system. Accordingly, the importance of type 2 stations is particularly high not only due to their multiplicity in the system, but due to their topological contribution in terms of connectivity as well. Consequently, components of this station type have a significant impact on the resilience of the overall system. An examination of the type 2 subsystem model, see Fig. 6.11, shows that type 1 components take on a predominant position. Once both type 1 components in this subsystem fail, the entire metro station fails. No other components of a single type can cause this in station type 2.

The significant influence of type 2 components can easily be explained by examining the

Table 6.5: Non-trivial survival signature values of stations with $b_j = 2$ of the metro system, shown in Fig. 6.11.

l_1	l_2	l_3	l_4	$\Phi(l_1, \dots, l_4)$
2	2	1	1	1/4
2	1	1	2	1/4
2	2	2	1	3/8
2	2	1	2	3/8
2	1	2	2	3/8
3	2	1	1	1/2
2	3	1	1	1/2
2	1	3	1	1/2
3	1	1	2	1/2
2	3	1	2	1/2
2	1	1	3	1/2
2	2	1	3	1/2
2	3	1	3	1/2
2	2	2	2	9/16
3	2	2	1	3/4
2	3	2	1	3/4
2	2	3	1	3/4
3	2	1	2	3/4
3	1	2	2	3/4
2	3	2	2	3/4
2	1	3	2	3/4
2	1	2	3	3/4
2	2	2	3	3/4
2	3	2	3	3/4
3	2	2	2	7/8
2	2	3	2	7/8

type 3 and 5 station systems, see again Fig. 6.11. Of all stations, only here bottleneck positions exist, where the failure of a single component interrupts the functioning of the entire station. Both of these positions, in type 3 and type 5 stations, are occupied by type 2 components. Since both station types have three and five direct connections to other stations, they can be considered to be particularly interconnected and thus of high relevance to network efficiency and thus of high relevance to system resilience.

Type 3 and 4 components, on the other hand, do not occupy any particularly significant positions in the stations' systems. This explains their low influence. The enormous influence of the recovery improvement is intuitively explainable. As resilience is established via the integral of the actual system performance, each recovered metro station contributes directly and immediately to the network efficiency and thus to the resilience of the system. Therefore, improvement of this property results in an immediate and intuitive increase in resilience.

Assuming the arbitrary base prices in Tab. 6.4, the most cost-efficient acceptable endowment (\hat{A}, \hat{D}) results from a reliability improvement configuration of $rel_i = 10$ for $k_i = 1$, $rel_i = 9$

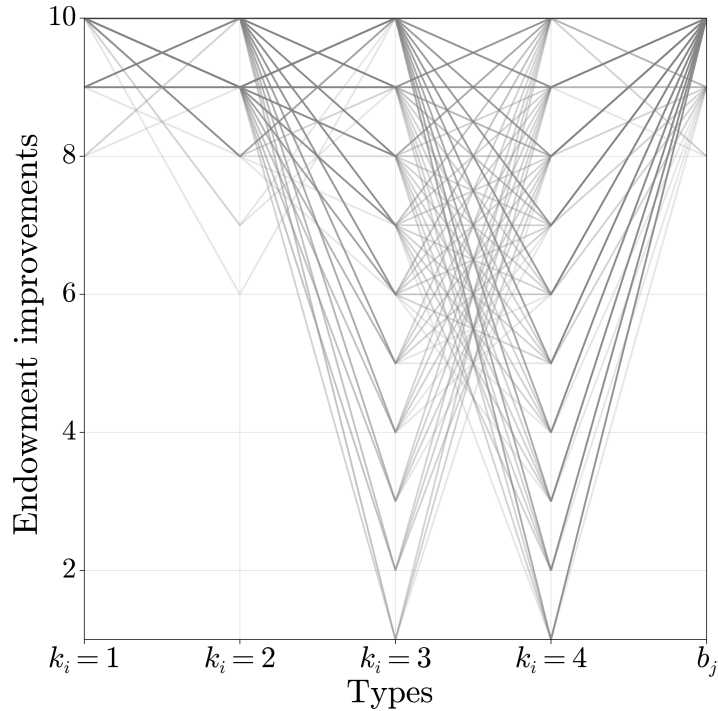


Figure 6.13: The set of all accepted endowments $R(Y)$ evaluated via the 5D grid search algorithm for the Berlin metro system with explored reliability improvement and recovery improvement values.

for $k_i = 2$, $rel_i = 7$ for $k_i = 3$, $rel_i = 2$ for $k_i = 4$ for components of type 1 to 4 and a maximum recovery improvement configuration of $rec_j = 10$ for $b_j \in \{1, 2, 3, 4\}$, i.e., all stations, regardless of their type. In Fig. 6.14, the corresponding configuration is highlighted. Due to the monotonicity of the input-output model and the assumed monotonically increasing endowment costs, only the endowment configurations on the dominant vertices of the Pareto front have to be examined for the identification of the most cost-efficient endowment. Therefore, only these endowment configurations are shown in Fig. 6.14. The resulting costs are given by Eq. (6.29, 6.30, 6.31) with $cost_{(\hat{A}, \hat{D})} = 1\,700\,829\text{€} + 361\,185\text{€} = 2\,062\,014\text{€}$.

Due to the utilization of the grid search algorithm, the computational effort could be significantly reduced in this case study as well – only 0.159% of all potential endowment configurations had to be examined in order to assign a distinct state to each configuration in the search space as accepted or not accepted. By means of the novel approach, the metro system could be reduced from its entirety of 2776 individual components to 306 components on the top-level with respect to the resilience analysis, drastically reducing the computational effort. Nevertheless, all 2776 components and their influence were considered. As in the case study of the axial compressor, not only the most cost-efficient endowment configuration can be identified but also investigations on configurations that are below certain budget limits can be conducted.

Note that, in this case study, as well as in the previous ones, various complexity variations

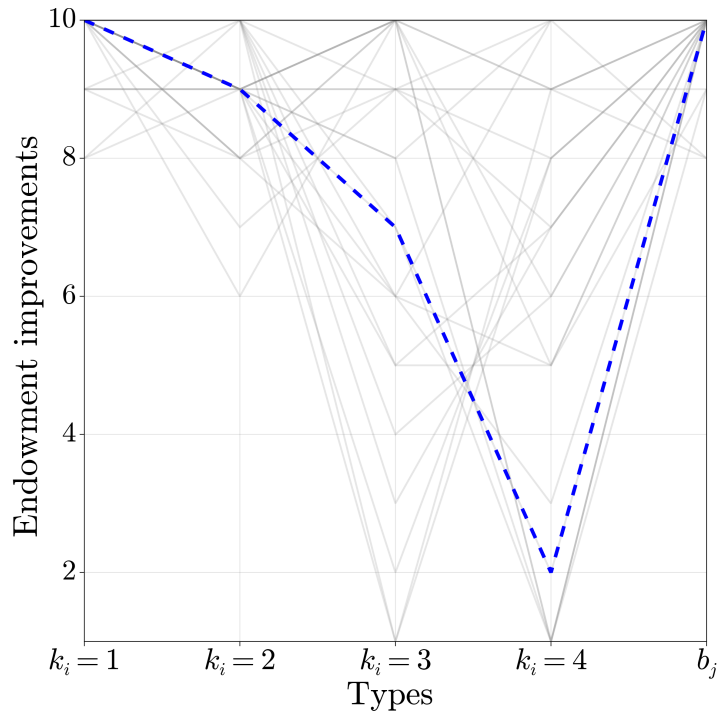


Figure 6.14: Dominant Pareto front endowments of the 5D grid search algorithm for the Berlin metro system with explored reliability improvement and recovery improvement values and the most cost-efficient endowment (\hat{A}, \hat{D}) is highlighted.

such as so-called cascading failures, see [42, 201, 202], are possible to implement due to the time-step-accurate simulation. In the case of infrastructure systems, e.g., the increasingly frequent natural disasters can thus be considered, that typically have an impact as local phenomena and affect stations that are geographically close to each other. It has already been shown in [72] that these can be taken into account in the resilience decision-making analysis of infrastructure systems.

6.8 Conclusion and outlook

This paper addresses the challenge of efficient multidimensional decision-making for complex and substructured systems between resilience-influencing parameters. By merging an extension of the resilience framework proposed in [72] with the survival signature, an efficient and novel methodology is derived. The approach allows for direct comparison of the impact of heterogeneous controls on system resilience, such as failure prevention and recovery improvement arrangements, both during the design phase as well as during any phase in the life cycle of already existing complex systems.

Due to the time-step accurate simulation of the system performance on system level during the resilience analysis, complexity extensions such as cascading failures and other dependency structures can be considered without difficulties. The new methodology has a high numerical

efficiency. The majority of the endowment properties examined affect the probability structure of the system components. The numerous changes in the probability structure caused by constantly changing endowment properties during the resilience analysis can be ideally covered with minimal effort due to the separation property of the survival signature.

The novel approach includes a substructuring approach for large, complex systems. This and the integration of the survival signature allow for the propagation of subsystem reliabilities through any number of system levels to the top-level and lead to a significant reduction of the computational load. This way, and with the extension of the adapted systemic risk measure, it is now possible to analyze systems with a large number of components in terms of their resilience.

Monetary restrictions can easily be included in the analysis. More precisely, not only the most cost-efficient, accepted endowment is identified, but subsets of the set of all accepted endowments below defined price levels can be formed. Budget limits can thus be specifically taken into account in the decision-making process.

The methodology is applied to three entirely distinct systems: A functional model of a multistage high-speed axial compressor, an arbitrary system consisting of numerous subsystems and components and a comprehensive substructured model of the metro system of Berlin, proofing wide and general applicability. All results obtained are plausible with the corresponding assumed model parameters. Note, that the approach can be utilized to systems of any kind.

In the development of our proposed methodology, some simplifying assumptions were made that do not accurately reflect reality. However, the authors strongly believe that the presented approach can be considered as a meaningful core development that, for a reality-based application on highly multifactorial systems, such as cyber-human-physical systems, should be combined with future as well as existing developments to ensure an efficient and comprehensive resilience decision-making analysis taking into account all technical and monetary aspects of modern societies.

Future work will address the incorporation of various existing extensions of the concept of survival signature, such as accounting for uncertainty and propagating it toward imprecise system resilience and considering multiple state or continuous component functionality. Further, future work regarding multidimensional parameter spaces must deal with the limitations in computing time and storage capacity in order to enable application to even higher-dimensional problems. Namely, techniques such as advanced sampling methods, e.g. Subset Simulation, see [203], must be investigated to further reduce numerical effort.

Acknowledgment

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) SFB 871/3 119193472 and SPP 2388 501624329.

7 | **UncertaintyQuantification.jl: A new Framework for Uncertainty Quantification in Julia**

UncertaintyQuantification.jl: A new Framework for Uncertainty Quantification in Julia

Jasper Behrendorf^a, Ander Gray^b, Matteo Broggi^a, Michael Beer^{a,c,d}

^aInstitute for Risk and Reliability, Leibniz Universität Hannover, Hannover, Germany

^bUnited Kingdom Energy Authority, Culham, United Kingdom ^cInstitute for Risk and Uncertainty, University of Liverpool, Liverpool, United Kingdom

^dInternational Joint Research Center for Engineering Reliability and Stochastic Mechanics, Tongji University, Shanghai, China

Published in *Proceedings of the 5th ECCOMAS Thematic Conference on Uncertainty Quantification in Computational Sciences and Engineering (UNCECOMP 2023)* in June 2023

Abstract

This work presents a new framework for uncertainty quantification developed as a package in the Julia programming language called `UncertaintyQuantification.jl`. Julia is a modern high-level dynamic programming language ideally suited for tasks like data analysis and scientific computing. `UncertaintyQuantification.jl` was developed from the ground up to be generalized and flexible while at the same time being easy to use. Leveraging the features of a modern language such as Julia allows to write efficient, fast and easy to read code. Especially noteworthy is Julia's core feature multiple dispatch which enables us to, for example, develop methods with a large number of varying simulation schemes such as standard Monte Carlo, Sobol sampling, Halton sampling, etc., yet minimal code duplication. Current features of `UncertaintyQuantification.jl` include simulation based reliability analysis using a large array of sampling schemes, local and global sensitivity analysis, meta modelling techniques such as response surface methodology or polynomial chaos expansion as well as the connection to external solvers by injecting values into plain text files as inputs. Through Julia's existing distributed computing capabilities all available methods can be easily run on existing clusters with just a few lines of extra code.

Keywords:

7.1 Introduction

The increasing complexity of modern engineering systems requires adequate simulation methods to ensure their safety and reliability. At the same time, a reliable analysis can only be performed when taking into consideration the present uncertainties. New analyses and simulation methods for the treatment of these uncertainties are constantly developed. A generic and freely available framework, which includes these methods and can be applied to a large array of engineering problems, can be a valuable tool for researchers, students, and industry alike.

Many such frameworks have been developed for a variety of programming languages. *Dakota* (C++) [79], *OpenTURNS* (C++/Python) [80], *OpenCossan* (MATLAB) [81], *UQLab* (MATLAB) [82], and *UQpy* (Python) [83] to name a few.

This work presents an alternative to these frameworks called *UncertaintyQuantification.jl* [84]. This new framework is developed as a module for the Julia programming language. The Julia language [86] has been developed as a new approach to scientific computing. Julia is designed to be fast yet easy to use at the same time. Its core principal *multiple dispatch* allows to write

simple reusable code, where the appropriate algorithm is automatically selected based on the input arguments to a function. This allows us to efficiently implement a variety of simulation techniques for the same underlying analysis with little extra effort. Julia was designed from the ground up with parallelization and high performance computing in mind. The large numerical demand of modern simulation methods can often be distributed to computing clusters with only a few extra lines. This makes Julia ideally suited for the development of a generic and complete package for uncertainty quantification.

UncertaintyQuantification is registered in the official *General* registry and can be installed from the Julia REPL by switching to the package manager with a closing square bracket `]`.

```
julia> ] add UncertaintyQuantification
```

The goal of *UncertaintyQuantification* is to build a complete and generalized framework for uncertainty quantification where newly developed methods are constantly implemented and shared with fellow researchers. *UncertaintyQuantification* is released under the MIT license and publicly available on Github. While development is still in the early stages, the basic features and key analyses have been completed. In its current version *UncertaintyQuantification* includes simulation based reliability analysis through standard and advanced Monte Carlo simulation, local and global sensitivity analysis, metamodels, and the ability to connect any of these methods to external solvers. This paper only presents the basic usage of the module. For a thorough explanation of all included simulations the reader is referred to the online documentation.

The remainder of the paper is structured as follows. In Section 7.2 we introduce the basics of *UncertaintyQuantification* including how to define input parameters, random variables and models. Section 7.3 and Section 7.4 present how to use the framework to perform reliability and sensitivity analyses, respectively. How to build metamodels as approximations is presented in Section 7.5, followed by a few numerical examples in Section 7.6. The paper closes with conclusions and a brief outlook into future development of the framework.

7.2 Getting Started

In this section we introduce the basic building blocks of *UncertaintyQuantification*. This includes the inputs such as `Parameter` or `RandomVariable` which will feed into any `Model` for a variety of different analyses. We will also present more advanced concepts including how to model dependencies between the inputs through copulas.

7.2.1 Inputs

Parameters

A **Parameter** is defined as a constant scalar value. In addition to value the constructor also requires a **Symbol** by which it can later be identified in the **Model**. A **Symbol** is a Julia object which is often used as a name or label. **Symbols** are defined using the `:` prefix. Parameters represent constant deterministic values. As an example we define a **Parameter** representing the gravity of Earth.

```
g = Parameter(9.81, :g)
```

Parameters are very handy when constants show up in the **Model** in multiple spaces. Instead of updating every instance in the **Model**, we can conveniently update the value by changing a single line.

Random Variables

A **RandomVariable** is essentially a wrapper around any **UnivariateDistribution** defined in the *Distributions.jl* package [204]. Similarly to the **Parameter**, the second argument to the constructor is a **Symbol** acting as a unique identifier. For example, a standard gaussian random variable is defined by passing `Normal()` and `:x` as arguments.

```
x = RandomVariable(Normal(), :x)
```

A list of all possible distributions can be generated by executing `subtypes(UnivariateDistribution)` in the Julia REPL (read-eval-print loop). Note that, *Distributions* is re-exported from *UncertaintyQuantification* and no separate `using` statement is necessary. In addition, the most important methods of the **UnivariateDistribution** including `pdf`, `cdf`, and `quantile`, are also defined for the **RandomVariable**.

Random samples can be drawn from a **RandomVariable** by calling the `sample` method passing the random variable and the desired number of samples.

```
samples = sample(x, 100) # sample(x, MonteCarlo(100))
```

The `sample` method returns a **DataFrame** with the samples in a single column. When sampling from a **Vector** of random variables these individual columns are automatically merged into one unified **DataFrame**. By default, this will use standard Monte Carlo simulation to obtain the samples. Alternatively, any of the quasi-Monte Carlo methods can be used instead.

```
samples = sample(x, SobolSampling(100))
samples = sample(x, LatinHypercubeSampling(100))
```


Many of the advanced simulations, e.g. line sampling [205] or subset simulation [137] require mappings to (and from) the standard normal space, and these are exposed through the `to_standard_normal_space!` and `to_physical_space!` methods respectively. These operate on a `DataFrame` and as such can be applied to samples directly. The transformation is done in-place, i.e. no new `DataFrame` is returned. As such, in the following example, the samples end up exactly as they were in the beginning.

```
to_standard_normal_space!(x, samples)
to_physical_space!(x, samples)
```

Dependencies

UncertaintyQuantification supports modelling of dependencies through copulas. By using copulas, the modelling of the dependence structure is separated from the modelling of the univariate marginal distributions. The basis for copulas is given by Sklar's theorem [59]. It states that any multivariate distribution H in dimensions $d \geq 2$ can be separated into its marginal distributions F_i and a copula function C .

$$H(x_1, \dots, x_d) = C(F_1(x_1), \dots, F_d(x_d)) \quad (7.1)$$

For a thorough discussion of copulas, see [61].

In line with Sklar's theorem we build the joint distribution of two dependent random variables by separately defining the marginal distributions.

```
x = RandomVariable(Normal(), :x)
y = RandomVariable(Uniform(), :y)
marginals = [x, y]
```

Next, we define the copula to model the dependence. *UncertaintyQuantification* supports Gaussian copulas for multivariate ($d \geq 2$) dependence as well as a list of Archimedean copulas for bivariate dependence. The Archimedean copulas are implemented as a wrapper around the *BivariateCopulas.jl* [206] package. Here, we define a Gaussian copula by passing the correlation matrix and then build the `JointDistribution` from the copula and the marginals.

```
cop = GaussianCopula([1 0.8; 0.8 1])
joint = JointDistribution(marginals, cop)
```

Figure 7.1 shows the samples drawn by sampling from the independent marginals and the joint distribution using the Gaussian copula.

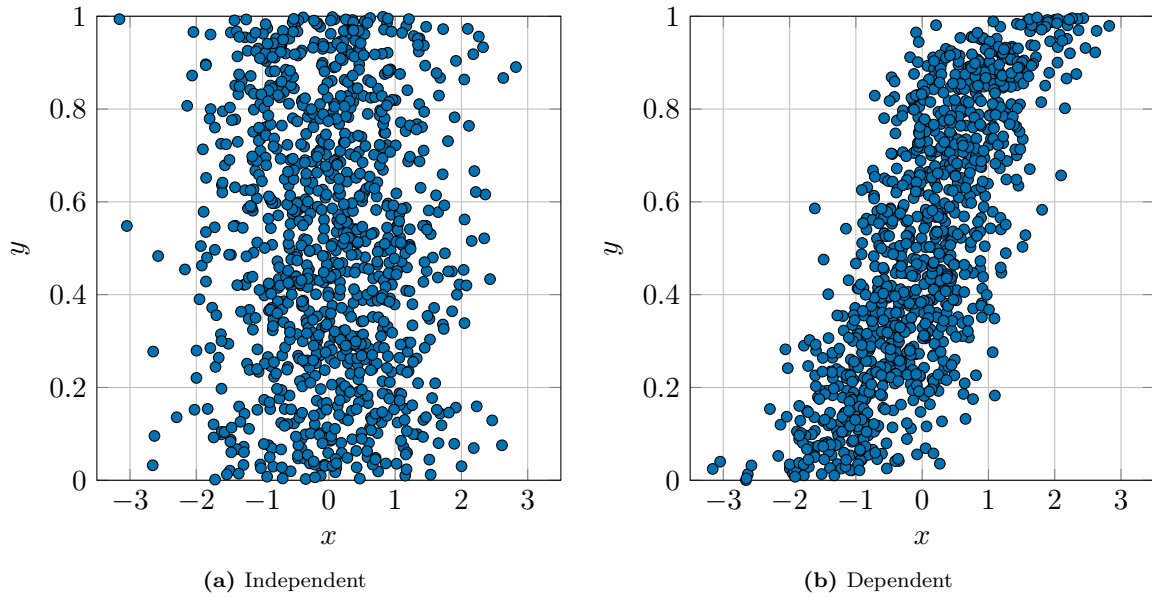


Figure 7.1: 1000 samples drawn from the independent marginal distributions (a) and the dependent joint distribution (b).

7.2.2 Models

In this section we present the models included in *UncertaintyQuantification*. A model, in its most basic form, is a relationship between a set of input variables $x \in \mathbb{R}^{n_x}$ and an output $y \in \mathbb{R}$. Currently, most models are assumed to return single-valued outputs. However, as seen later, the `ExternalModel` is capable of extracting an arbitrary number of outputs from a single run of an external solver.

Model

A `Model` is essentially a native Julia function operating on the previously defined inputs. Building a `Model` requires two things: a `Function`, which is internally passed a `DataFrame` containing the samples and must return a `Vector` containing the model response for each sample, and a `Symbol` which is the identifier used to add the model output into the `DataFrame`.

Suppose we wanted to define a `Model` which computes the distance from the origin of two variables x and y as z . We first define the function and then pass it to the `Model`.

```
function z(df::DataFrame)
    return @. sqrt(df.x^2 + df.y^2)
end
m = Model(z, :z)
```

An alternative for a simple model such as this, is to directly pass an anonymous function to the `Model`.

```
m = Model(df -> sqrt.(df.x.^2 .+ df.y.^2), :z)
```

After defining it, a `Model` can be evaluated on a set of samples by calling the `evaluate!` method. This will add the model outcome to the `DataFrame`. Alternatively, the response can be obtained as a vector, by calling the `Model` as a function.

```
samples = sample([x, y], MonteCarlo(1000))
evaluate!(m, samples) # add to the DataFrame
output = m(samples) # return a Vector
```

However, most of the time manual evaluation of the `Model` will not be necessary as it is done internally by whichever analysis is performed.

ParallelModel

With the basic `Model` it is up to the user to implement an efficient function which returns the model responses for all samples simultaneously. Commonly, this will involve vectorized operations as presented in the example. For more complex or longer running models, *UncertaintyQuantification* provides a simple `ParallelModel`. This model relies on the capabilities of the *Distributed* module, which is part of the standard library shipped with Julia. Without any present *workers*, the `ParallelModel` will evaluate its function in a loop for each sample. If one or more workers are present, it will automatically distribute the model evaluations. For this to work, *UncertaintyQuantification* must be loaded with the `@everywhere` macro in order to be loaded on all workers. In the following example, we first load *Distributed* and add four local workers. A simple model is then evaluated in parallel. Finally, the workers are removed.

```
using Distributed
addprocs(4) # add 4 local workers

@everywhere using UncertaintyQuantification

x = RandomVariable(Normal(), :x)
y = RandomVariable(Normal(), :y)

m = ParallelModel(df -> sqrt(df.x^2 .+ df.y^2), :z)

samples = sample([x, y], 1000)
evaluate!(m, samples)

rmprocs(workers()) # release the local workers
```

It is important to note, that the `ParallelModel` requires some overhead to distribute the

function calls to the workers. Therefore it performs significantly slower than the standard `Model` with vectorized operations for a simple function as in this example.

By using `ClusterManagers.jl` [207] to add the workers, the `ParallelModel` can easily be run on an existing compute cluster such as *Slurm*.

ExternalModel

The `ExternalModel` provides interaction with almost any third-party solver. The only requirement is, that the solver uses text-based input and output files in which the values sampled from the random variables can be injected for each individual run. The output quantities are then extracated from the files generated by the solver using one (or more) `Extractor`(s). This way, the simulation techniques included in this module, can be applied to advanced models in finite element software such as *OpenSees* or *Abaqus*.

The first step in building the `ExternalModel` is to define the folder where the source files can be found as well as the working directory. Here, we assume that the source file for a simple supported beam model is located in a subdirectory of our current working directory. Similarly, the working directory for the solver is defined. In addition, we define the exact files where values need to be injected, and any extra files required. No values will be injected into the files specified as extra. In this example, no extra files are needed, so the variable is defined as an empty `String` vector.

```
sourcedir = joinpath(pwd(), "demo/models")
sourcefiles = ["supported-beam.tcl"]
extrafiles = String[]
workdir = joinpath(pwd(), "supported-beam")
```

Next, we must define where to inject values from the random variables and parameters into the input files. For this, we make use of the `Mustache.jl` [208] and `Formatting.jl` [209] modules. The values in the source file must be replaced by triple curly bracket expressions of the form `{{{ :x }}}`, where `:x` is the identifier of the `RandomVariable` or `Parameter` to be injected. For example, to inject the Young's modulus and density of an elastic isotropic material in *OpenSees*, one could write the following.

```
nDMaterial ElasticIsotropic 1 {{{ :E }}} 0.25 {{{ :rho }}}
```

This identifies where to inject the values, but not in which format. For this reason, we define a `Dict{Symbol, String}` which maps the identifiers of the inputs to a Python-style format string. In order to inject our values in scientific notation with eight digits, we use the format string `".8e"`. For any not explicitly defined `Symbol` we can include `:*` as a fallback.

```
formats = Dict{:E => ".8e", :rho => ".8e", :* => ".12e"}
```

After formatting and injecting the values into the source file, it would look similar to this.

```
nDMaterial ElasticIsotropic 1 9.99813819e+02 0.25 3.03176259e+00
```

Now that the values are injected into the source files, the next step is to extract the desired output quantities. This is done using an **Extractor**. The **Extractor** is designed similarly to the **Model** in that it takes a **Function** and a **Symbol** as its parameters. However, where a **DataFrame** is passed to the **Model**, the working directory for the currently evaluated sample is passed to the function of the **Extractor**. The user defined function must then extract the required values from the file and return them. Here, we make use of the *DelimitedFiles* module to extract the maximum absolute displacement from the output file that *OpenSees* generated.

```
disp = Extractor(base -> begin
    file = joinpath(base, "displacement.out")
    data = readdlm(file, ' ')

    return maximum(abs.(data[:, 2]))
end, :disp)
```

An arbitrary number of **Extractor** functions can be defined in order to extract multiple output values from the solver.

The final step before building the model is to define the solver. The solver requires the path to the binary, and the input file. Optional command line arguments can be passed to the **Solver** through the **args** keyword. If the solver binary is not on the system path, the full path to the executable must be defined. Finally, the **ExternalModel** is assembled.

```
opensees = Solver(
    "OpenSees",
    "supported-beam.tcl";
    args = ""
)

ext = ExternalModel(
    sourcedir, sourcefiles, disp, opensees; formats=numberformats, workdir=workdir,
    ↪ extras=extrafiles
)
```

A full example of how to run a reliability analysis of a model defined in *OpenSees* can be found in the demo files of *UncertaintyQuantification*.

7.3 Reliability Analysis

Reliability analysis, as in the computation of the probability of failure of an engineering system, is facilitated through the simple method interface `probability_of_failure`. The method used to calculate this failure probability is selected internally based on the supplied arguments making use of Julia's multiple dispatch.

All methods to compute the probability of failure require the definition of a performance (limit-state) function $g(x)$ where $x \in \mathbb{R}^{n_x}$ is the set of input variables. By definition, failure of the system occurs for $g(x) < 0$. When implementing the limit-state, the Julia function operates on a `DataFrame` as seen with the `Model`. The `DataFrame` passed to the performance function includes the responses of all models.

7.3.1 First order reliability method

In the *first-order reliability method* (FORM) [210, 211], the random variables are transformed into the uncorrelated standard normal space and the limit-state function is approximated by a first-order Taylor series expansion at the design point. The design point being the point on the limit-state closest to the origin in the standard normal space.

Assume two random variables $x_1 \sim N(200, 20)$ and $x_2 \sim N(150, 10)$ with the failure domain being

$$F = \{(x_1, x_2) : x_2 > x_1\}. \quad (7.2)$$

The following script will compute the failure probability using FORM. The solution is obtained using the Hasofer-Lind-Rackwitz-Fiessler (HL-RF) algorithm.

```
form = FORM()

x1 = RandomVariable(Normal(200, 20), :x1)
x2 = RandomVariable(Normal(150, 10), :x2)

function g(df)
    return df.x1 .- df.x2
end

pf, beta, dp = probability_of_failure(g, [x1, x2], form)
```

The returned quantities are the probability of failure $p_f \approx 0.012673$, the reliability index $\beta \approx 2.23606$ and the design point $dp \approx (160, 160)$. In this case, the `probability_of_failure` function is called without a separate model, which can be included as the first input if desired. The `FORM` struct holds no information and its sole purpose is to dispatch to the correct method.

Currently *UncertaintyQuantification* only implements the HL-RF algorithm to compute the design point. To circumvent the algorithm's known shortcomings, we plan to add an alternative method, where the design point identification is solved as an optimization problem, in the future.

7.3.2 Monte Carlo Simulation

The probability of failure can also be approximated by a variety of Monte Carlo simulation techniques. The module includes standard Monte Carlo simulation as well as a number of quasi-Monte Carlo simulation methods such as Latin hypercube sampling or Sobol' sampling. In order to compute the probability of failure for the simple example presented in the previous subsection through standard Monte Carlo simulation, we need only replace the `FORM` struct with a `MonteCarlo` struct. The only parameter necessary to construct the `MonteCarlo` object is the desired number of samples to be used.

```
mc = MonteCarlo(10000)
pf, c, samples = probability_of_failure(g, [x1, x2], mc)
```

The probability of failure obtained from the simulation is $p_f = 0.0129$ with a coefficient of variation of 0.87475 which agrees with the results obtained by FORM in the previous section. In difference to FORM, any of the Monte Carlo methods will return the the coefficient of variation and the evaluated samples instead of the reliability index and the design point.

To run the simulation with quasi-Monte Carlo techniques, simply replace the `MonteCarlo` struct.

```
pf, c, samples = probability_of_failure(g, [x1, x2],
  ↪ LatinHypercubeSampling(10000))
pf, c, samples = probability_of_failure(g, [x1, x2], HaltonSampling(10000))
pf, c, samples = probability_of_failure(g, [x1, x2], SobolSampling(10000))
```

UncertaintyQuantification provides some advanced Monte Carlo methods, namely `LineSampling` and `SubSetSimulation`. For a more complex example using `SubSetSimulation` to estimate small failure probabilities, see Section 7.6.1.

7.4 Sensitivity Analysis

This section presents the methods for both local and global sensitivity analysis included in *UncertaintyQuantification*. Sensitivity analysis examines how the uncertainty of the model output can be attributed to the uncertainty of the inputs [212]. There are two major areas of sensitivity. Where local sensitivity analysis considers the influence of inputs in a certain point, global sensitivity analysis considers the complete input space.

7.4.1 Local Sensitivity Analysis

UncertaintyQuantification provides the computation of gradients for local, derivative based, sensitivity analysis through the `gradient` method. For a set of input variables and models the gradient can be calculated by passing a `DataFrame` of reference points to evaluate. The `gradient` method will return a new `DataFrame` of the same size, containing the gradients. Internally, finite difference methods are used to approximate the gradients.

```
grads = gradient(model, inputs, reference, output)
```

Gradients can alternatively be estimated after transformation to the standard normal space. However, this is mainly provided in order to find the important direction required for line sampling [136].

```
grads = gradient_in_standard_normal_space(model, inputs, reference, output)
```

7.4.2 Global Sensitivity Analysis

For global sensitivity analysis [212] *UncertaintyQuantification* provides Sobol' indices, a form of variance based sensitivity analysis. This type of sensitivity analysis quantifies how much of the variance of a model output can be attributed to each of the inputs. Two main sensitivity measures are defined. For a model $Y = f(X_1, X_2, \dots, X_n)$, the first order index is defined by

$$S_i = \frac{V_{X_i}(E_{\mathbf{X}_{\sim i}}(Y|X_i))}{V(Y)}, \quad (7.3)$$

where $\mathbf{X}_{\sim i}$ represents all input variables except i . The first order index captures the effect of varying X_i alone. The total effect index captures also the variance caused by all interactions of X_i with other variables. It is defined as

$$S_{T_i} = \frac{E_{\mathbf{X}_{\sim i}}(V_{X_i}(Y|\mathbf{X}_{\sim i}))}{V(Y)}. \quad (7.4)$$

To compute simulation based Sobol' indices we must first define the inputs and model as with all the previous examples. The indices can then be approximated by calling the `sobolindices` function. Quasi-Monte carlo methods like `SobolSampling` usually lead to better results than brute force Monte Carlo. The Sobol' indices for multiple models of the same inputs can be computed simultaneously by passing vectors of models and output symbols to the method.

```
# single model/output
indices = sobolindices(model, inputs, :y, SobolSampling(2000))
# multiple models/outputs
indices = sobolindices([m1, m2], inputs, [:y1, :y2], SobolSampling(2000))
```


The Sobol' indices are estimated according to Jansen (1999) [213] and Saltelli (2010) [214]. Bootstrapping is used to quantify the standard error of the estimators. For a complete example of how to compute Sobol' indices, see Section 7.6.2.

7.5 Metamodelling

When dealing with complex, long running models, a single run of the model can take hours to days. In these cases, it is infeasible to run any analysis requiring a large number of samples. Quantities such as Sobol' indices for sensitivity analysis simply can not be computed. In these cases we can apply so called *metamodels* or *surrogate models*. These models are constructed from a small selection of evaluations of the true model and aim to approximate it as closely as possible. The metamodel itself is easy to evaluate and enables the analyses which were inaccessible for the original model.

The metamodels currently included in *UncertaintyQuantification* are the response surface methodology, polyharmonic splines and polynomial chaos expansion. Several designs of experiments (central composite design, Box-behnken design, factorial design), are provided to fit response surface models.

7.5.1 Polyharmonic spline

As a simple example we are going to construct a polyharmonic spline metamodel for a simple one-dimensional test function. A polyharmonic spline [215] is a combination of a low degree polynomial term and polyharmonic radial basis functions of the form

$$s(x) = \sum_{i=1}^N w_i \varphi(|\mathbf{x} - \mathbf{c}_i|) + \mathbf{v}^T \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix}. \quad (7.5)$$

The radial basis functions are $\varphi(r) = r^p, p = 1, 3, 5, \dots$, and $\varphi(r) = r^p \ln(r), p = 2, 4, 6, \dots$, with degree p and $r = |\mathbf{x} - \mathbf{c}_i| = \sqrt{(\mathbf{x} - \mathbf{c}_i)^T (\mathbf{x} - \mathbf{c}_i)}$. The N center points to interpolate are denoted by \mathbf{c}_i .

The test function [216] is defined on $x \in [0, 1]$ as

$$f(x) = (6x - 2)^2 \sin(12x - 4). \quad (7.6)$$

To start, we set up the necessary input and the model. We sample 20 values of x using Sobol' sampling and evaluate the model for the sampled points. By using a quasi-Monte Carlo sampling scheme we make sure that the sampled points have adequate coverage of the input space.

Figure 7.2: Polyharmonic spline metamodel compared to the true function.

```
x = RandomVariable(Uniform(0, 1), :x)

m = Model(df -> (6 .* x .- 2) .^ 2 .* sin.(12 .* x .- 4), :y)

data = sample(x, SobolSampling(20))
evaluate!(m, data)
```

In order to construct the polyharmonic spline (and determine the required weights \mathbf{w} and \mathbf{v}), the sampled points are passed to the constructor along with the desired degree $p = 2$ and the `Symbol` of the output to fit the metamodel to.

```
ps = PolyharmonicSpline(data, 2, :y)
```

A plot comparing the metamodel to the original function and the sampled points can be seen in Figure 7.2. With just 20 sampled points, the spline is able to accurately approximate the test function.

7.6 Numerical Examples

This section presents a few more complex examples than the ones presented in the previous sections.

7.6.1 Estimation of small failure probabilities

Subset simulation [see 137] is an advanced Monte Carlo simulation technique for the estimation of small failure probabilities. Here we solve a simple problem [taken from 217, page 9] where the response y depends on two independent random variables x_1 and x_2 following a standard normal distribution. The simple linear model is defined by

$$y(x_1, x_2) = x_1 + x_2 \quad (7.7)$$

with the failure domain

$$F = \{(x_1, x_2) : x_1 + x_2 > 9\}. \quad (7.8)$$

The analytical probability of failure can be calculated as

$$p_f = 1 - \Phi\left(\frac{9}{\sqrt{2}}\right) \approx 1 \times 10^{-10}, \quad (7.9)$$

where Φ is the standard normal cumulative density function.

In order to solve this problem, we start by creating the two standard normal random variables and group them in a vector inputs.

```
x1 = RandomVariable(Normal(), :x1)
x2 = RandomVariable(Normal(), :x2)
inputs = [x1, x2]
```

Next we define the model as

```
y = Model(df -> df.x1 + df.x2, :y)
```

where the first input is our function (which must accept a **DataFrame**) and the second the **Symbol** for the output variable.

To estimate a failure probability we need a performance function. This function, which accepts a **DataFrame** similar to the **Model**, must return a vector which is negative if a failure occurs for a sample. Here, a failure occurs if y exceeds 9.

```
function g(df::DataFrame)
    return 9 .- df.y
end
```

Finally, we create the **SubSetSimulation** object and compute the probability of failure using a standard Gaussian proposal PDF. The value for the target probability of failure at each intermediate level is set to 0.1.

```
subset = SubSetSimulation(1000, 0.1, 10, Normal())
pf, cov, samples = probability_of_failure(y, g, inputs, subset)
```

Alternatively, instead of using the standard Subset simulation algorithm (which internally uses Markov Chain Monte Carlo), we can use **SubSetInfinity** to compute the probability of failure, see [218]. Here we use a standard deviation of 0.5 to create the proposal samples for the next level.

```
subset = SubSetInfinity(1000, 0.1, 10, 0.5)
pf, cov, samples = probability_of_failure(y, g, inputs, subset)
```

Figure 7.3 shows the samples at each level of the two Subset simulations. Both methods are able to estimate the small probability of failure with the MCMC based method yielding $p_f \approx 5.3912e - 10$ and Subset- ∞ $p_f \approx 3.13212e - 11$. The plots show, that Subset- ∞ required two extra levels to approximate the probability of failure. However, due to the numerical demand of the Markov chains, Subset- ∞ runs about twice as fast for this example.

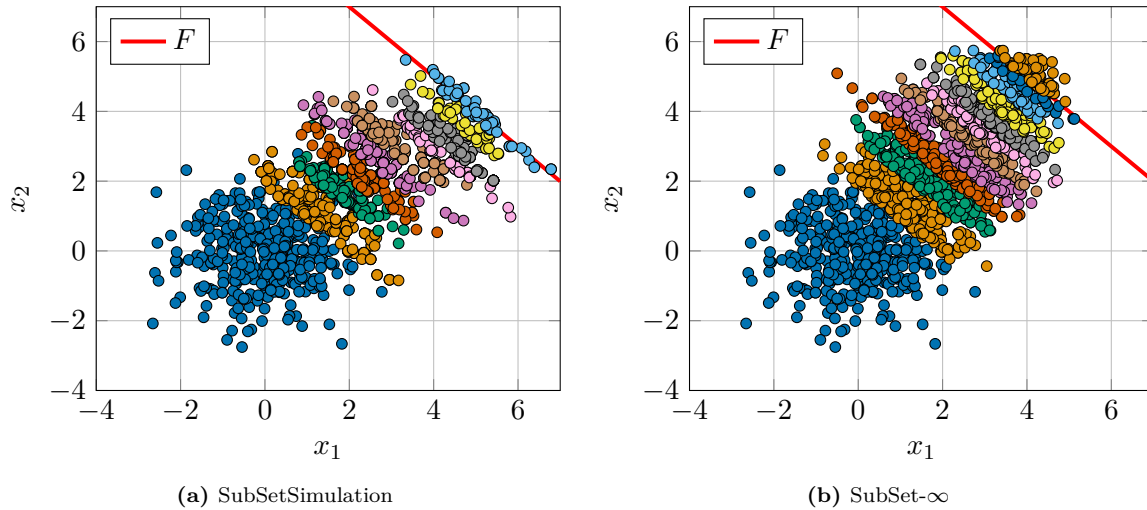


Figure 7.3: Samples at the different levels for the two Subset simulations. Both simulations have been started from the same seed samples at the first level.

7.6.2 Global Sensitivity Analysis of the Ishigami function

The Ishigami function [219] given by

$$f(x) = \sin(x_1) + a \sin(x_2)^2 + bx_3^4 \sin(x_1), \quad (7.10)$$

with $x_i \sim U(-\pi, \pi)$ for $i = 1, 2, 3$, is often used to benchmark global sensitivity analysis methods. It shows strong nonlinearity and has an interesting dependence on x_3 . The parameter a is typically set to 7 while b is either 0.1 [220] or 0.05 [221].

The following script will estimate the first order and total effect Sobol' indices of the Ishigami function using 10000 samples drawn from the Sobol' sequence. Using a quasi-Monte Carlo method instead of the brute force method improves convergence of the Sobol' indices.

```
x = RandomVariable.(Uniform(-pi,pi), [:x1, :x2, :x3])

a = Parameter(7, :a)
b = Parameter(0.1, :b)

function ishigami(df)
    return @. sin(df.x1) + df.a * sin(df.x2)^2 + df.b * df.x3^4 * sin(df.x1)
end

m = Model(ishigami, :y)

sobol = sobolindices(m, [x..., a, b], :y, SobolSampling(10000))
```

The Sobol' indices computed for the Ishigami function using 10000 samples are shown in

Table 7.1: Approximated first order and total effect Sobol' indices of the Ishigami function with estimated standard errors. The analytical solutions S_i and S_{T_i} are included for comparison.

x_i	\hat{S}_i	$\sigma_{\hat{S}_i}$	S_i	\hat{S}_{T_i}	$\sigma_{\hat{S}_{T_i}}$	S_{T_i}
x_1	0.291419	0.00964169	0.3138	0.536946	0.0129737	0.5574
x_2	0.436318	0.0102289	0.4424	0.439057	0.0130688	0.4424
x_3	-0.004927	0.0101326	0.0	0.243939	0.0130147	0.2436

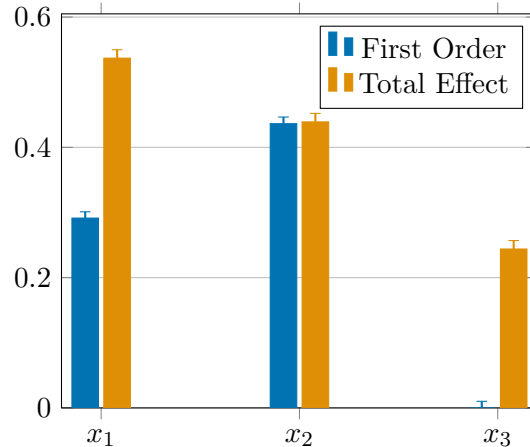
**Figure 7.4:** First order and total effect Sobol' indices of the Ishigami function.

Figure 7.4 and Table 7.1. S_3 being slightly negative can be attributed to numerical inaccuracies. Note also the interesting dependence on x_3 . While S_3 suggests no influence on the function at all, S_{T_3} proves that x_3 has significant influence through interactions.

Computing thousands of model evaluations will be impossible for a complex model. To work around this problem, we can build an accurate metamodel of our model and use it to compute the Sobol' indices as seen in the next section.

7.6.3 Metamodelling and Global Sensitivity Analysis

It is entirely possible to build a metamodel such as a response surface or polyharmonic spline based on a few model evaluations and use it to compute the Sobol' indices similarly to the previous section. However, a unique property of the polynomial chaos expansion (PCE) [222] is that it provides the Sobol' indices as a byproduct as they can be calculated from the coefficients of the PCE. Consider again the Ishigami function. In order to build a PCE metamodel we must first define the polynomial basis to be used.

We begin by selecting the degree $p = 8$ and the Legendre polynomials as a basis for the three input variables. Then we assemble the `PolynomialChaosBasis` from the combinations of degree up to p of our basis functions.

Table 7.2: First order and total effect Sobol' indices of the Ishigami function estimated from the polynomial chaos expansion.

	\hat{S}_i	\hat{S}_{T_i}
x_1	0.312994	0.555787
x_2	0.444213	0.444213
x_3	5.10399e-32	0.242794

```
p = 8
bases = [LegendreBasis(), LegendreBasis(), LegendreBasis()]
psi = PolynomialChaosBasis(bases, p)
```

Internally, *UncertaintyQuantification* will perform an isoprobabilistic transformation of the inputs to the support of the basis functions (in this case $[0, 1]$ for all inputs). Next, the PCE metamodel is built using Gauss quadrature and the Sobol' indices are computed directly from the polynomial chaos expansion.

```
pce, samples = polynomialchaos(inputs, m, psi, :y, GaussQuadrature())
sobol = sobolindices(pce)
```

The resulting indices are presented in Table 7.2. No errors are estimated when computing the Sobol' indices from a polynomial chaos expansion. With just 729 model evaluations for $p = 8$ one can see that the results are much closer to the analytical solution than using 10000 samples with the true model. Note, that the number of required quadrature nodes grows quickly with increasing degree and the number of inputs. This issue can be alleviated by applying sparse quadrature grids. However, the implementation of sparse quadrature and other methods of estimating the PCE coefficients in *UncertaintyQuantification* is still ongoing.

7.7 Conclusion

This paper presented a new framework for uncertainty quantification called *UncertaintyQuantification.jl*. The framework has been developed as a module in the Julia programming language making use of its modern features. This work serves as an introduction to the framework. We presented its basic usage and outlined how to perform a few key analyses including reliability analysis, sensitivity analysis and metamodeling.

Ongoing development of the framework is focused on improving existing methods such as extending the polynomial chaos expansion with sparse quadratures, as well as adding more algorithms. Features currently in development include more metamodels like Kriging and artificial neural networks, global sensitivity analysis with dependent inputs and the propagation of imprecise probabilities through probability boxes.

The source code of the framework is released under the MIT license and publicly available on Github. Contributions from other researchers are very welcome.

8 | Conclusions and outlook

This chapter summarizes the key developments presented in this thesis and provides an outlook into the direction of future research.

8.1 Conclusions

This dissertation discusses several important improvements to the survival signature ecosystem. New methodologies are presented, where copulas are used to model dependent failures while Monte Carlo simulation and surrogate models allow for efficient approximation of the survival signature. These methods, when combined, allow for the efficient numerical reliability analysis of complex coupled networks. The individual contributions are summarized next.

The first publication presented focuses on modelling dependencies in networks and interdependencies between different networks for the reliability analysis. Based on the separation of probabilistic and structural information inherent to the survival signature, this new method proposes to model the dependencies using copulas in the probabilistic part of the survival function. Failures cascade between networks solely based on the component failures sampled from the copulas. Pair-copula construction through vine copulas is suggested to model high-dimensional multivariate failure time distributions. In addition, the inclusion of imprecise probabilities in the copulas and the marginal distributions is presented. The concept has been applied successfully to an example based on the IEEE reliability test system.

Instead of the probabilistic modelling of component failures, the second contribution focuses on efficiently deriving the survival signature for large networks. The proposed method is divided into two steps. First, large parts of the survival signature can be excluded as negligible through the application of percolation theory. In the second step, the remaining entries are approximated using Monte Carlo simulation. The errors of both steps in comparison to the analytical solution are quantified using simple toy examples. It is shown, that the new method is able to accurately approximate the survival signature with a significant reduction in numerical effort. Finally, to demonstrate scalability to larger systems, the approach is applied to a network based on the Berlin S- and U-Bahn systems. With this new method, systems with hundreds of components can be analysed through the survival signature where previously the computational demand was prohibitive.

Expanding on this approximation method, the third and fourth publications develop a

surrogate model for the survival signature instead of fully approximating it. First, a radial basis function network is constructed based on a few initial data points estimated through the previous Monte Carlo technique. Next, this surrogate model is adaptively refined until it provides an accurate representation of the survival signature. In the final step, the uncertainty associated with the Monte Carlo simulation is propagated through an interval predictor model into the final survival signature. With only a few computed signature entries, this new method is able to accurately predict bounds on the survival signature and therefore the reliability. In comparison to the method based only on Monte Carlo simulation, this new technique provides another significant reduction in numerical demand. In the presented examples, a survival signature with 119 556 entries is accurately predicted with only 406 evaluated data points.

The fifth contribution extends a framework for resilience decision-making to multidimensional problems and substructured systems. The resilience of large systems can be efficiently quantified by computing the survival signature for the subsystems and deriving failure rates from the survival functions. The proposed approach can analyse systems with thousands of components. Simulations for different endowment configurations are often computationally demanding. This requirement is reduced greatly by exploiting the separation property of the survival signature and modifying only the probability structures of the component failure times.

The sixth and final paper introduces a new framework for uncertainty quantification. This new framework, developed in the modern Julia programming language, offers generalized algorithms for reliability analysis, sensitivity analysis, and numerous metamodelling approaches. It is open to the public and encourages contributions from the research community.

8.2 Outlook

The methods presented in this dissertation are able to efficiently estimate the reliability of large dependent networks. However, several potential areas for improvement have been identified during their development and are presented here.

In regard to modelling dependencies with copulas, especially vine copulas, it has been shown that constructing the copula model is a complex task. Alternative methods for modelling multivariate distributions could prove beneficial. For example, the recently presented class of Sliced-Normal [149] distributions allows for the characterization of complex dependencies with little modelling effort and could serve as a potential replacement of the copula model.

An area of interest for the approximation of the survival signature is the reduction of samples required to accurately estimate a single entry, which both presented methods will benefit from. Advanced Monte Carlo methods such as line sampling [136] or subset simulation [137, 217] suggest great potential and should be investigated.

In addition, for the surrogate model approach, different refinement schemes for the radial basis function network should be compared. Moreover, the current assumptions made for the number of centers and spread of the basis functions are rough and can certainly be optimized

in the future. Optimal centers and spread can potentially lead to the method requiring even less data points for an accurate representation of the true survival signature.

The current development focus for the *UncertaintyQuantification.jl* framework lies on feature parity with OpenCOSSAN and the implementation of completely new algorithms. Areas of special interest are the inclusion of more surrogate models such as Kriging and artificial neural networks, reliability based optimization, and the efficient propagation of imprecise probabilities.

9 | List of publications

Journal articles

- Behrendorf, J., Broggi, M. and Beer, M. ‘Interval Predictor Model for the Survival Signature Using Monotone Radial Basis Functions’. In: *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering* (2023). Submitted
- Salomon, J., Behrendorf, J., Winnewisser, N., Broggi, M. and Beer, M. ‘Multidimensional Resilience Decision-Making for Complex and Substructured Systems’. In: *Resilient Cities and Structures. Resilience of Civil Infrastructure to Multiple Hazards and Climate Change* 1.3 (2022), pages 61–78. DOI: 10.1016/j.rcns.2022.10.005
- Behrendorf, J., Regenhardt, T.-E., Broggi, M. and Beer, M. ‘Numerically Efficient Computation of the Survival Signature for the Reliability Analysis of Large Networks’. In: *Reliability Engineering & System Safety* 216 (2021), page 107935. DOI: 10.1016/j.ress.2021.107935
- Yang, L., Wang, P., Wang, Q., Bi, S., Peng, R., Behrendorf, J. and Beer, M. ‘Reliability Analysis of a Complex System with Hybrid Structures and Multi-Level Dependent Life Metrics’. In: *Reliability Engineering & System Safety* 209 (2021), page 107469. DOI: 10.1016/j.ress.2021.107469
- Behrendorf, J., Broggi, M. and Beer, M. ‘Reliability Analysis of Networks Interconnected With Copulas’. In: *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering* 5.4 (2019), 041006 (9 Pages). DOI: 10.1115/1.4044043

Book contributions

- Salomon, J., Behrendorf, J. and Beer, M. ‘Resilienz Baulicher Infrastruktur - Sicher Und Wirtschaftlicher Durch Dick Und Dünn’. In: *26. Dresdner Baustatik - Seminar - “Realität-Modellierung-Tragwerksplanung”*. Institut für Statik und Dynamik der Tragwerke, TU Dresden, 2022
- Behrendorf, J., Broggi, M. and Beer, M. ‘Efficient Reliability and Risk Analysis of Complex Interconnected Systems’. In: *Resilience Engineering for Urban Tunnels*. Amer-

ican Society of Civil Engineers, 2018, pages 43–54. ISBN: 978-0-7844-1513-9. DOI: 10.1061/9780784415139.ch04

Conference papers

- Behrendorf, J., Broggi, M. and Beer, M. ‘Imprecise Survival Signature Approximation Using Interval Predictor Models’. In: 2023 IEEE Symposium Series on Computational Intelligence. Submitted. Mexico City, Mexico, 2023
- Behrendorf, J., Gray, A., Broggi, M. and Beer, M. ‘UncertaintyQuantification.jl: A New Framework for Uncertainty Quantification in Julia’. In: *Proceedings of the 5th ECCOMAS Thematic Conference on Uncertainty Quantification in Computational Sciences and Engineering (UNCECOMP 2023)*. Edited by M. Papadrakakis, V. Papadopoulos and G. Stefanou. Athens, Greece, 2023
- Bittner, M., Behrendt, M., Behrendorf, J. and Beer, M. ‘Epistemic Uncertainty Quantification of Localised Seismic Power Spectral Densities’. In: Probabilistic Safety Assessment and Management PSAM 16. Honolulu, USA, 2022
- Salomon, J., Behrendorf, J., Winnewisser, N., Broggi, M. and Beer, M. ‘Resilience Decision-Making for Complex and Substructured Systems’. In: *Proceedings of the 8th International Symposium on Reliability Engineering and Risk Management*. Edited by M. Beer, E. Zio, K.-K. Phoon and B. M. Ayyub. Singapore: Research Publishing Services, 2022, pages 530–537. DOI: 10.3850/978-981-18-5184-1_MS-16-191-cd
- Salomon, J., Behrendorf, J., Broggi, M., Weber, S. and Beer, M. ‘Multidimensional Resilience Decision-Making On A Multistage High-Speed Axial Compressor’. In: *Proceedings of the 29th European Safety and Reliability Conference*. Edited by M. Beer and E. Zio. Singapore: Research Publishing Services, 2019, pages 1357–1364. DOI: 10.3850/978-981-11-2724-3_0992-cd
- Behrendorf, J., Brandt, S., Broggi, M. and Beer, M. ‘Efficient Approximation of the Survival Signature for Large Networks’. In: *Proceedings of the 6th International Symposium on Reliability Engineering and Risk Management*. Singapore: Research Publishing Services, 2018, pages 661–666. ISBN: 978-981-11-2726-7. DOI: 10.3850/978-981-11-2726-7_CRR14
- Behrendorf, J., Broggi, M. and Beer, M. ‘Imprecise Reliability Analysis of Complex Interconnected Networks’. In: *Safety and Reliability - Safe Societies in a Changing World*. 28th European Safety and Reliability Conference. Edited by S. Haugen, A. Barros, C. Gulijk, T. Kongsvik and J. E. Vinnem. London: CRC Press, 2018, pages 2589–2594. ISBN: 978-1-351-17466-4. DOI: 10.1201/9781351174664-325

- Behrendorf, J., Broggi, M., Brandt, S. and Beer, M. 'Numerically Efficient Reliability Analysis of Interdependent Networks'. In: *Safety and Reliability. Theory and Applications*. 27th European Safety and Reliability Conference. Edited by M. Cepin and R. Bris. London: CRC Press, 2017. DOI: 10.1201/9781315210469-298

Bibliography

- [1] Council of the European Union. ‘Council Directive 2008/114/EC of 8 December 2008 on the Identification and Designation of European Critical Infrastructures and the Assessment of the Need to Improve Their Protection’. In: *Official Journal of the European Union L 345* 51.2008 (2008), pages 74–82.
- [2] Boin, A. and McConnell, A. ‘Preparing for Critical Infrastructure Breakdowns: The Limits of Crisis Management and the Need for Resilience’. In: *Journal of Contingencies and Crisis Management* 15.1 (2007), pages 50–59. DOI: 10.1111/j.1468-5973.2007.00504.x.
- [3] Albert, R. and Barabási, A.-L. ‘Statistical Mechanics of Complex Networks’. In: *Reviews of Modern Physics* 74.1 (2002), pages 47–97. DOI: 10.1103/RevModPhys.74.47.
- [4] Pérez-Rosés, H. ‘Sixty Years of Network Reliability’. In: *Mathematics in Computer Science* 12.3 (2018), pages 275–293. DOI: 10.1007/s11786-018-0345-5.
- [5] Brown, J. I., Colbourn, C. J., Cox, D., Graves, C. and Mol, L. ‘Network Reliability: Heading out on the Highway’. In: *Networks* 77.1 (2021), pages 146–160. DOI: 10.1002/net.21977.
- [6] Moore, E. F. and Shannon, C. E. ‘Reliable Circuits Using Less Reliable Relays’. In: *Journal of the Franklin Institute* 262.3 (1956), pages 191–208. DOI: 10.1016/0016-0032(56)90559-2.
- [7] Gaur, V., Yadav, O. P., Soni, G. and Rathore, A. P. S. ‘A Literature Review on Network Reliability Analysis and Its Engineering Applications’. In: *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability* 235.2 (2021), pages 167–181. DOI: 10.1177/1748006X20962258.
- [8] Abraham, J. ‘An Improved Algorithm for Network Reliability’. In: *IEEE Transactions on Reliability* 1 (1979), pages 58–61. DOI: 10.1109/TR.1979.5220476.
- [9] Hao, Z., Yeh, W.-C., Wang, J., Wang, G.-G. and Sun, B. ‘A Quick Inclusion-Exclusion Technique’. In: *Information Sciences* 486 (2019), pages 20–30. DOI: 10.1016/j.ins.2019.02.004.

- [10] Gertsbakh, I. and Shpungin, Y. ‘Direct Network Reliability Calculation’. In: *Network Reliability: A Lecture Course*. Edited by I. Gertsbakh and Y. Shpungin. SpringerBriefs in Electrical and Computer Engineering. Singapore: Springer, 2020, pages 21–34. ISBN: 9789811514586. DOI: 10.1007/978-981-15-1458-6_3.
- [11] Alsharif, F. H. A., Mudhar, G. A. and Hassan, Z. A. H. ‘A Modified Technique to Compute the Minimal Path Sets for the Reliability of Complex Network’. In: *Journal of Physics: Conference Series* 1999.1 (2021), page 012083. DOI: 10.1088/1742-6596/1999/1/012083.
- [12] Singh, B. and Ghosh, S. K. ‘Network Reliability Evaluation by Decomposition’. In: *Microelectronics Reliability* 34.5 (1994), pages 925–927. DOI: 10.1016/0026-2714(94)90017-5.
- [13] Carlier, J. and Lucet, C. ‘A Decomposition Algorithm for Network Reliability Evaluation’. In: *Discrete Applied Mathematics*. First International Colloquium on Graphs and Optimization 65.1 (1996), pages 141–156. DOI: 10.1016/0166-218X(95)00032-M.
- [14] Hardy, G., Lucet, C. and Linnios, N. ‘K-Terminal Network Reliability Measures With Binary Decision Diagrams’. In: *IEEE Transactions on Reliability* 56.3 (2007), pages 506–515. DOI: 10.1109/TR.2007.898572.
- [15] Jane, C.-C. and Lai, Y.-W. ‘A Practical Algorithm for Computing Multi-State Two-Terminal Reliability’. In: *IEEE Transactions on Reliability* 57.2 (2008), pages 295–302. DOI: 10.1109/TR.2008.920792.
- [16] Liu, W. and Li, J. ‘An Improved Cut-Based Recursive Decomposition Algorithm for Reliability Analysis of Networks’. In: *Earthquake Engineering and Engineering Vibration* 11.1 (2012), pages 1–10. DOI: 10.1007/s11803-012-0093-5.
- [17] Liu, T., Bai, G., Tao, J., Zhang, Y.-A. and Fang, Y. ‘An Improved Bounding Algorithm for Approximating Multistate Network Reliability Based on State-Space Decomposition Method’. In: *Reliability Engineering & System Safety* 210 (2021), page 107500. DOI: 10.1016/j.res.s.2021.107500.
- [18] Fishman, G. S. ‘A Monte Carlo Sampling Plan for Estimating Network Reliability’. In: *Operations Research* 34.4 (1986), pages 581–594. DOI: 10.1287/opre.34.4.581.
- [19] Yeh, W.-C., Lin, Y.-C., Chung, Y. Y. and Chih, M. ‘A Particle Swarm Optimization Approach Based on Monte Carlo Simulation for Solving the Complex Network Reliability Problem’. In: *IEEE Transactions on Reliability* 59.1 (2010), pages 212–221. DOI: 10.1109/TR.2009.2035796.
- [20] Gertsbakh, I. B. and Shpungin, Y. *Models of Network Reliability: Analysis, Combinatorics, and Monte Carlo*. Boca Raton: CRC Press, 2012. 217 pages. ISBN: 978-0-429-16616-7. DOI: 10.1201/b12536.

- [21] Zio, E. *The Monte Carlo Simulation Method for System Reliability and Risk Analysis*. Springer Series in Reliability Engineering. London: Springer London, 2013. ISBN: 978-1-4471-4587-5. DOI: 10.1007/978-1-4471-4588-2.
- [22] Zuev, K. M., Wu, S. and Beck, J. L. ‘General Network Reliability Problem and Its Efficient Solution by Subset Simulation’. In: *Probabilistic Engineering Mechanics* 40 (2015), pages 25–35. DOI: 10.1016/j.probengmech.2015.02.002.
- [23] Stern, R. E., Song, J. and Work, D. B. ‘Accelerated Monte Carlo System Reliability Analysis through Machine-Learning-Based Surrogate Models of Network Connectivity’. In: *Reliability Engineering & System Safety* 164 (2017), pages 1–9. DOI: 10.1016/j.res.s.2017.01.021.
- [24] Pan, Q. and Dias, D. ‘An Efficient Reliability Method Combining Adaptive Support Vector Machine and Monte Carlo Simulation’. In: *Structural Safety* 67 (2017), pages 85–95. DOI: 10.1016/j.strusafe.2017.04.006.
- [25] Davila-Frias, A., Yodo, N., Le, T. and Yadav, O. P. ‘A Deep Neural Network and Bayesian Method Based Framework for All-Terminal Network Reliability Estimation Considering Degradation’. In: *Reliability Engineering & System Safety* 229 (2023), page 108881. DOI: 10.1016/j.res.s.2022.108881.
- [26] Volkanovski, A., Čepin, M. and Mavko, B. ‘Application of the Fault Tree Analysis for Assessment of Power System Reliability’. In: *Reliability Engineering & System Safety* 94.6 (2009), pages 1116–1127. DOI: 10.1016/j.res.s.2009.01.004.
- [27] Coolen, F. P. A. and Coolen-Maturi, T. ‘Generalizing the Signature to Systems with Multiple Types of Components’. In: *Complex Systems and Dependability*. Edited by W. Zamojski, J. Mazurkiewicz, J. Sugier, T. Walkowiak and J. Kacprzyk. Volume 170. Advances in Intelligent and Soft Computing. Berlin, Heidelberg: Springer, 2013, pages 115–130. ISBN: 978-3-642-30661-7. DOI: 10.1007/978-3-642-30662-4_8.
- [28] Samaniego, F. J. *System Signatures and Their Applications in Engineering Reliability*. Redacted by F. S. Hillier. Volume 110. International Series In Operations Research & Management Science. Boston, MA: Springer US, 2007. ISBN: 978-0-387-71796-8. DOI: 10.1007/978-0-387-71797-5.
- [29] Coolen, F. P., Coolen-Maturi, T. and Al-nefaiee, A. H. ‘Nonparametric Predictive Inference for System Reliability Using the Survival Signature’. In: *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability* 228.5 (2014), pages 437–448. DOI: 10.1177/1748006X14526390.
- [30] Eryilmaz, S. and Tuncel, A. ‘Generalizing the Survival Signature to Unrepairable Homogeneous Multi-State Systems’. In: *Naval Research Logistics* 63.8 (2016), pages 593–599. DOI: 10.1002/nav.21722.

- [31] Winnewisser, N. R., Salomon, J., Broggi, M. and Beer, M. ‘The Concept of Diagonal Approximated Signature: New Surrogate Modeling Approach for Continuous-State Systems in the Context of Resilience Optimization’. In: *Disaster Prevention and Resilience* 3.2 (2023), page 4. DOI: 10.20517/dpr.2023.03.
- [32] George-Williams, H., Feng, G., Coolen, F. P., Beer, M. and Patelli, E. ‘Extending the Survival Signature Paradigm to Complex Systems with Non-Repairable Dependent Failures’. In: *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability* 233.4 (2019), pages 505–519. DOI: 10.1177/1748006X18808085.
- [33] Feng, G., George-Williams, H., Patelli, E., Coolen, F. P. A. and Beer, M. ‘An Efficient Reliability Analysis on Complex Non-Repairable Systems with Common-Cause Failures’. In: *Safety and Reliability - Safe Societies in a Changing World*. 28th European Safety and Reliability Conference. Edited by S. Haugen, A. Barros, C. Gulijk, T. Kongsvik and J. E. Vinnem. London: CRC Press, 2018, pages 2531–2537. ISBN: 978-1-351-17466-4. DOI: 10.1201/9781351174664-318.
- [34] Eryilmaz, S. ‘Reliability of Systems With Multiple Types of Dependent Components’. In: *IEEE Transactions on Reliability* 65.2 (2016), pages 1022–1029. DOI: 10.1109/TR.2016.2521758.
- [35] Zheng, Y. and Zhang, Y. ‘Reliability Analysis for System with Dependent Components Based on Survival Signature and Copula Theory’. In: *Reliability Engineering & System Safety* 238 (2023), page 109402. DOI: 10.1016/j.ress.2023.109402.
- [36] Najem, A. and Coolen, F. P. A. ‘System Reliability and Component Importance When Components Can Be Swapped upon Failure’. In: *Applied Stochastic Models in Business and Industry* 35.3 (2019), pages 399–413. DOI: 10.1002/asmb.2420.
- [37] Patelli, E., Feng, G., Coolen, F. P. A. and Coolen-Maturi, T. ‘Simulation Methods for System Reliability Using the Survival Signature’. In: *Reliability Engineering & System Safety*. Special Section: Applications of Probabilistic Graphical Models in Dependability, Diagnosis and Prognosis 167 (2017), pages 327–337. DOI: 10.1016/j.ress.2017.06.018.
- [38] Reed, S. ‘An Efficient Algorithm for Exact Computation of System and Survival Signatures Using Binary Decision Diagrams’. In: *Reliability Engineering & System Safety* 165 (2017), pages 257–267. DOI: 10.1016/j.ress.2017.03.036.
- [39] Reed, S., Löfstrand, M. and Andrews, J. ‘An Efficient Algorithm for Computing Exact System and Survival Signatures of K-terminal Network Reliability’. In: *Reliability Engineering & System Safety* 185 (2019), pages 429–439. DOI: 10.1016/j.ress.2019.01.011.
- [40] Xu, B.-H., Yang, D.-Z., Qian, C., Feng, Q., Ren, Y., Wang, Z.-L. and Sun, B. ‘A New Method for Computing Survival Signature Based on Extended Universal Generating Function’. In: *2019 International Conference on Quality, Reliability, Risk, Maintenance,*

- and Safety Engineering (QR2MSE)*. 2019, pages 597–602. DOI: 10.1109/QR2MSE46217.2019.9021190.
- [41] Di Maio, F., Pettorossi, C. and Zio, E. ‘Entropy-Driven Monte Carlo Simulation Method for Approximating the Survival Signature of Complex Infrastructures’. In: *Reliability Engineering & System Safety* 231 (2023), page 108982. DOI: 10.1016/j.ress.2022.108982.
- [42] Buldyrev, S. V., Parshani, R., Paul, G., Stanley, H. E. and Havlin, S. ‘Catastrophic Cascade of Failures in Interdependent Networks’. In: *Nature* 464 (2010), pages 1025–1028. DOI: 10.1038/nature08932.
- [43] Rinaldi, S., Peerenboom, J. and Kelly, T. ‘Identifying, Understanding, and Analyzing Critical Infrastructure Interdependencies’. In: *IEEE Control Systems Magazine* 21.6 (2001), pages 11–25. DOI: 10.1109/37.969131.
- [44] Zio, E. and Sansavini, G. ‘Modeling Interdependent Network Systems for Identifying Cascade-Safe Operating Margins’. In: *IEEE Transactions on Reliability* 60.1 (2011), pages 94–101. DOI: 10.1109/TR.2010.2104211.
- [45] Di Giorgio, A. and Liberati, F. ‘A Bayesian Network-Based Approach to the Critical Infrastructure Interdependencies Analysis’. In: *IEEE Systems Journal* 6.3 (2012), pages 510–519. DOI: 10.1109/JSYST.2012.2190695.
- [46] Falahati, B., Fu, Y. and Wu, L. ‘Reliability Assessment of Smart Grid Considering Direct Cyber-Power Interdependencies’. In: *IEEE Transactions on Smart Grid* 3.3 (2012), pages 1515–1524. DOI: 10.1109/TSG.2012.2194520.
- [47] Falahati, B. and Fu, Y. ‘Reliability Assessment of Smart Grids Considering Indirect Cyber-Power Interdependencies’. In: *IEEE Transactions on Smart Grid* 5.4 (2014), pages 1677–1685. DOI: 10.1109/TSG.2014.2310742.
- [48] Fang, Y.-P. and Zio, E. ‘An Adaptive Robust Framework for the Optimization of the Resilience of Interdependent Infrastructures under Natural Hazards’. In: *European Journal of Operational Research* 276.3 (2019), pages 1119–1136. DOI: 10.1016/j.ejor.2019.01.052.
- [49] Bellè, A., Abdin, A. F., Fang, Y.-P., Zeng, Z. and Barros, A. ‘A Resilience-Based Framework for the Optimal Coupling of Interdependent Critical Infrastructures’. In: *Reliability Engineering & System Safety* 237 (2023), page 109364. DOI: 10.1016/j.ress.2023.109364.
- [50] Nikolaidis, E. ‘Types of Uncertainty in Design Decision Making’. In: *Engineering Design Reliability Handbook*. Edited by S. Singhal, D. M. Ghiocel and E. Nikolaidis. Boca Raton: CRC Press, 2004. ISBN: 978-0-429-20461-6. DOI: 10.1201/9780203483930.

- [51] Aughenbaugh, J. M. and Paredis, C. J. J. ‘The Value of Using Imprecise Probabilities in Engineering Design’. In: *Journal of Mechanical Design* 128.4 (2005), pages 969–979. DOI: 10.1115/1.2204976.
- [52] Kiureghian, A. D. and Ditlevsen, O. ‘Aleatory or Epistemic? Does It Matter?’ In: *Structural Safety*. Risk Acceptance and Risk Communication 31.2 (2009), pages 105–112. DOI: 10.1016/j.strusafe.2008.06.020.
- [53] Dang, C., Wei, P., Faes, M. G. R. and Beer, M. ‘Bayesian Probabilistic Propagation of Hybrid Uncertainties: Estimation of Response Expectation Function, Its Variable Importance and Bounds’. In: *Computers & Structures* 270 (2022), page 106860. DOI: 10.1016/j.compstruc.2022.106860.
- [54] Bi, S., Broggi, M., Wei, P. and Beer, M. ‘The Bhattacharyya Distance: Enriching the P-box in Stochastic Sensitivity Analysis’. In: *Mechanical Systems and Signal Processing* 129 (2019), pages 265–281. DOI: 10.1016/j.ymsp.2019.04.035.
- [55] Park, K. I. *Fundamentals of Probability and Stochastic Processes with Applications to Communications*. Cham: Springer International Publishing, 2018. ISBN: 978-3-319-68074-3 978-3-319-68075-0. DOI: 10.1007/978-3-319-68075-0.
- [56] Kolmogoroff, A. *Grundbegriffe der Wahrscheinlichkeitsrechnung*. Berlin, Heidelberg: Springer, 1933. ISBN: 978-3-642-49596-0 978-3-642-49888-6. DOI: 10.1007/978-3-642-49888-6.
- [57] Ferson, S., Kreinovich, V., Ginzburg, L., Sentz, F. and Meyers, D. *Constructing Probability Boxes and Dempster-Shafer Structures*. SAND2015-4166J. Albuquerque, NM: Sandia National Laboratory, 2015. DOI: 10.2172/809606.
- [58] Feng, G., Patelli, E., Beer, M. and Coolen, F. P. A. ‘Imprecise System Reliability and Component Importance Based on Survival Signature’. In: *Reliability Engineering & System Safety* 150 (2016), pages 116–125. DOI: 10.1016/j.ress.2016.01.019.
- [59] Sklar, M. ‘Fonctions de repartition a n dimensions et leurs marges’. In: *Publ. inst. statist. univ. Paris* 8 (1959), pages 229–231.
- [60] Nelson, R. B. *An Introduction to Copulas*. 2nd edition. Springer Series in Statistics. New York, NY: Springer, 2007. 272 pages. ISBN: 978-0-387-28678-5.
- [61] Joe, H. *Dependence Modeling with Copulas*. Volume 134. Monographs on Statistics & Applied Probability. New York: Chapman and Hall/CRC, 2014. ISBN: 978-1-4665-8322-1.
- [62] Schirmacher, D. Y. and Schirmacher, E. *Multivariate Dependence Modeling Using Pair-Copulas*. 2008-m-as08-1. Chicago, IL: Society of Actuaries, 2008, page 52.
- [63] Kendall, M. *Rank Correlation Methods*. 4th edition. Rank Correlation Methods. London: Griffin, 1970. ISBN: 0-85264-199-0.

- [64] Spearman, C. ‘The Proof and Measurement of Association between Two Things’. In: *The American Journal of Psychology* 15.1 (1904), pages 72–101. DOI: 10.2307/1412159.
- [65] Mai, J.-F. and Scherer, M. *Simulating Copulas: Stochastic Models, Sampling Algorithms and Applications*. Volume 4. Series in Quantitative Finance. London: Imperial College Press, 2012. 312 pages. ISBN: 978-1-84816-874-9. URL: <https://doi.org/10.1142/p842>.
- [66] Hobæk Haff, I., Aas, K. and Frigessi, A. ‘On the Simplified Pair-Copula Construction — Simply Useful or Too Simplistic?’ In: *Journal of Multivariate Analysis* 101.5 (2010), pages 1296–1310. DOI: 10.1016/j.jmva.2009.12.001.
- [67] Bedford, T. and Cooke, R. M. ‘Vines: A New Graphical Model for Dependent Random Variables’. In: *The Annals of Statistics* 30.4 (2002), pages 1031–1068. DOI: 10.1214/aos/1031689016.
- [68] Diestel, R. *Graph Theory*. 5th edition. Volume 173. Graduate Texts in Mathematics. Berlin, Heidelberg: Springer, 2017. ISBN: 978-3-662-53621-6 978-3-662-53622-3. DOI: 10.1007/978-3-662-53622-3.
- [69] Morales-Nápoles, O. ‘Counting Vines’. In: *Dependence Modeling*. Edited by D. Kurowicka and H. Joe. World Scientific, 2010, pages 189–218. ISBN: 978-981-4299-87-9. DOI: 10.1142/9789814299886_0009.
- [70] Joe, H. ‘Families of M-Variate Distributions with Given Margins and $m(m-1)/2$ Bivariate Dependence Parameters’. In: *Lecture Notes-Monograph Series* 28 (1996), pages 120–141. URL: <https://www.jstor.org/stable/4355888>.
- [71] Aas, K., Czado, C., Frigessi, A. and Bakken, H. ‘Pair-Copula Constructions of Multiple Dependence’. In: *Insurance: Mathematics and Economics* 44.2 (2009), pages 182–198. DOI: 10.1016/j.insmathco.2007.02.001.
- [72] Salomon, J., Broggi, M., Kruse, S., Weber, S. and Beer, M. ‘Resilience Decision-Making for Complex Systems’. In: *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering* (2020). DOI: 10.1115/1.4044907.
- [73] Bruneau, M., Chang, S. E., Eguchi, R. T. et al. ‘A Framework to Quantitatively Assess and Enhance the Seismic Resilience of Communities’. In: *Earthquake Spectra* 19.4 (2003), pages 733–752. DOI: 10.1193/1.1623497.
- [74] Bergström, J., Van Winsen, R. and Henriqson, E. ‘On the Rationale of Resilience in the Domain of Safety: A Literature Review’. In: *Reliability Engineering and System Safety* 141 (2015), pages 131–141. DOI: 10.1016/j.ress.2015.03.008.
- [75] Hosseini, S., Barker, K. and Ramirez-Marquez, J. E. ‘A Review of Definitions and Measures of System Resilience’. In: *Reliability Engineering & System Safety* 145 (2016), pages 47–61. DOI: 10.1016/j.ress.2015.08.006.

- [76] Ouyang, M., Dueñas-Osorio, L. and Min, X. ‘A Three-Stage Resilience Analysis Framework for Urban Infrastructure Systems’. In: *Structural Safety* (2012), pages 23–31. DOI: 10.1016/j.strusafe.2011.12.004.
- [77] Feinstein, Z., Rudloff, B. and Weber, S. ‘Measures of Systemic Risk’. In: *SIAM Journal on Financial Mathematics* 8.1 (2017), pages 672–708. DOI: 10.1137/16M1066087.
- [78] Mettas, A. ‘Reliability Allocation and Optimization for Complex Systems’. In: *Reliability and Maintainability Symposium, 2000. Proceedings. Annual* (2000), pages 216–221. DOI: 10.1109/RAMS.2000.816310.
- [79] Adams, B. M., Bohnhoff, W. J., Dalbey, K. R. et al. *Dakota, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 6.13 User’s Manual*. SAND2020-12495. Sandia National Lab. (SNL-NM), Albuquerque, NM (United States), 2020. DOI: 10.2172/1817318.
- [80] Baudin, M., Dutfoy, A., Iooss, B. and Popelin, A.-L. ‘Open TURNS: An Industrial Software for Uncertainty Quantification in Simulation’. In: *Handbook of Uncertainty Quantification*. Switzerland: Springer, 2017. URL: doi.org/10.1007/978-3-319-12385-1_64.
- [81] Patelli, E., Broggi, M., Angelis, M. de and Beer, M. ‘OpenCossan: An Efficient Open Tool for Dealing with Epistemic and Aleatory Uncertainties’. In: *Vulnerability, Uncertainty, and Risk: Quantification, Mitigation, and Management*. 2014, pages 2564–2573. DOI: 10.1061/9780784413609.258.
- [82] Marelli, S. and Sudret, B. ‘UQLab: A Framework for Uncertainty Quantification in Matlab’. In: *Vulnerability, Uncertainty, and Risk: Quantification, Mitigation, and Management*. 2014, pages 2554–2563. DOI: 10.1061/9780784413609.257.
- [83] Olivier, A., Giovanis, D. G., Aakash, B. S., Chauhan, M., Vandanapu, L. and Shields, M. D. ‘UQpy: A General Purpose Python Package and Development Environment for Uncertainty Quantification’. In: *Journal of Computational Science* 47 (2020), page 101204. DOI: 10.1016/j.jocs.2020.101204.
- [84] Behrendorf, J., Gray, A., Agarwal, G., Perin, A., Luttmann, M. and Knipper, L. *FriesischScott/UncertaintyQuantification.jl*. Version 0.6.1. Zenodo, 2023. DOI: 10.5281/zenodo.3993816.
- [85] Schuëller, G. I. and Pradlwarter, H. J. ‘Computational Stochastic Structural Analysis (COSSAN) – a Software Tool’. In: *Structural Safety*. Structural Reliability Software 28.1 (2006), pages 68–82. DOI: 10.1016/j.strusafe.2005.03.005.

Bibliography

- [86] Bezanson, J., Edelman, A., Karpinski, S. and Shah, V. B. ‘Julia: A Fresh Approach to Numerical Computing’. In: *SIAM Review* 59.1 (2017), pages 65–98. DOI: 10.1137/141000671.
- [87] Cherubini, U., Luciano, E. and Vecchiato, W. *Copula Methods in Finance*. John Wiley & Sons, 2004. 312 pages. ISBN: 978-0-470-86345-9.
- [88] Patton, A. J. ‘A Review of Copula Models for Economic Time Series’. In: *Journal of Multivariate Analysis*. Special Issue on Copula Modeling and Dependence 110 (2012), pages 4–18. DOI: 10.1016/j.jmva.2012.02.021.
- [89] Tang, A. and Valdez, E. A. *Economic Capital and the Aggregation of Risks Using Copulas*. 2009. DOI: 10.2139/ssrn.1347675. preprint.
- [90] Mejdoub, H. and Arab, M. B. ‘A Multivariate Analysis for Risk Capital Estimation in Insurance Industry: Vine Copulas’. In: *Asian Development Policy Review* 5.2 (2017), pages 100–119. URL: <https://ideas.repec.org//a/asi/adprev/v5y2017i2p100-119id278.html>.
- [91] Gatzert, N., Schmeiser, H. and Schuckmann, S. ‘Enterprise Risk Management in Financial Groups: Analysis of Risk Concentration and Default Risk’. In: *Financial Markets and Portfolio Management* 22.3 (2008), pages 241–258. DOI: 10.1007/s11408-008-0081-y.
- [92] Yan, J. ‘Multivariate Modeling with Copulas and Engineering Applications’. In: *Springer Handbook of Engineering Statistics*. Edited by H. Pham. Springer Handbooks. London: Springer, 2006, pages 973–990. ISBN: 978-1-84628-288-1. DOI: 10.1007/978-1-84628-288-1_51.
- [93] Behrendorf, J., Broggi, M. and Beer, M. ‘Reliability Analysis of Networks Interconnected With Copulas’. In: *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering* 5.4 (2019), 041006 (9 Pages). DOI: 10.1115/1.4044043.
- [94] Behrendorf, J., Regenhardt, T.-E., Broggi, M. and Beer, M. ‘Numerically Efficient Computation of the Survival Signature for the Reliability Analysis of Large Networks’. In: *Reliability Engineering & System Safety* 216 (2021), page 107935. DOI: 10.1016/j.ress.2021.107935.
- [95] Behrendorf, J., Broggi, M. and Beer, M. ‘Imprecise Survival Signature Approximation Using Interval Predictor Models’. In: 2023 IEEE Symposium Series on Computational Intelligence. Submitted. Mexico City, Mexico, 2023.
- [96] Behrendorf, J., Broggi, M. and Beer, M. ‘Interval Predictor Model for the Survival Signature Using Monotone Radial Basis Functions’. In: *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering* (2023). Submitted.

- [97] Salomon, J., Behrens Dorf, J., Winnewisser, N., Broggi, M. and Beer, M. ‘Resilience Decision-Making for Complex and Substructured Systems’. In: *Proceedings of the 8th International Symposium on Reliability Engineering and Risk Management*. Edited by M. Beer, E. Zio, K.-K. Phoon and B. M. Ayyub. Singapore: Research Publishing Services, 2022, pages 530–537. DOI: 10.3850/978-981-18-5184-1_MS-16-191-cd.
- [98] Behrens Dorf, J., Gray, A., Broggi, M. and Beer, M. ‘UncertaintyQuantification.jl: A New Framework for Uncertainty Quantification in Julia’. In: *Proceedings of the 5th ECCOMAS Thematic Conference on Uncertainty Quantification in Computational Sciences and Engineering (UNCECOMP 2023)*. Edited by M. Papadrakakis, V. Papadopoulos and G. Stefanou. Athens, Greece, 2023.
- [99] Behrens Dorf, J. *FriesischScott/SurvivalSignature.jl*. Zenodo, 2021. DOI: 10.5281/zenodo.4306114.
- [100] Behrens Dorf, J., Salomon, J. and Winnewisser, N. *ResilienceDecisionMaking.Jl*. Zenodo, 2022. DOI: 10.5281/zenodo.7034998.
- [101] UN-OCHA. *Philippines: Typhoon Haiyan Situation Report No. 19*. 2013. URL: <https://reliefweb.int/report/philippines/philippines-typhoon-haiyan-situation-report-no-19-28-november-2013>.
- [102] Leavitt, W. M. and Kiefer, J. J. ‘Infrastructure Interdependency and the Creation of a Normal Disaster: The Case of Hurricane Katrina and the City of New Orleans’. In: *Public Works Management & Policy* 10.4 (2006), pages 306–314. DOI: 10.1177/1087724X06289055.
- [103] Comfort, L. K. and Haase, T. W. ‘Communication, Coherence, and Collective Action: The Impact of Hurricane Katrina on Communications Infrastructure’. In: *Public Works Management & Policy* 10.4 (2006), pages 328–343. DOI: 10.1177/1087724X06289052.
- [104] Behrens Dorf, J., Broggi, M., Brandt, S. and Beer, M. ‘Numerically Efficient Reliability Analysis of Interdependent Networks’. In: *Safety and Reliability. Theory and Applications*. 27th European Safety and Reliability Conference. Edited by M. Cepin and R. Bris. London: CRC Press, 2017. DOI: 10.1201/9781315210469-298.
- [105] Goodwin, B. K. and Hungerford, A. ‘Copula-Based Models of Systemic Risk in U.S. Agriculture: Implications for Crop Insurance and Reinsurance Contracts’. In: *American Journal of Agricultural Economics* 97.3 (2014), pages 879–896. DOI: 10.1093/ajae/aau086.
- [106] Zhang, L. and Singh, V. P. ‘Bivariate Flood Frequency Analysis Using the Copula Method’. In: *Journal of Hydrologic Engineering* 11.2 (2006), pages 150–164. DOI: 10.1061/(ASCE)1084-0699(2006)11:2(150).

- [107] Ram, M. and Singh, S. B. ‘Analysis of Reliability Characteristics of a Complex Engineering System under Copula’. In: *Journal of Reliability and Statistical Studies* (2009), pages 91–102. URL: <https://journals.riverpublishers.com/index.php/JRSS/article/view/22097>.
- [108] Beer, M., Ferson, S. and Kreinovich, V. ‘Imprecise Probabilities in Engineering Analyses’. In: *Mechanical Systems and Signal Processing* 37.1-2 (1-2 2013), pages 4–29. DOI: 10.1016/j.ymsp.2013.01.024.
- [109] Hofert, M. and Maechler, M. ‘Nested Archimedean Copulas Meet R: The Nacopula Package’. In: *Journal of Statistical Software* 39 (2011), pages 1–20. DOI: 10.18637/jss.v039.i09.
- [110] Kurz, M. *VineCopulaMatlab Toolbox*. 2016. URL: <https://github.com/MalteKurz/VineCopulaMatlab>.
- [111] Watson, I. A. *Review of Common Cause Failures*. NCSR Report R27. UKAEA, 1981. URL: <https://www.osti.gov/etdeweb/biblio/5462206>.
- [112] Hanks, B. J. ‘An Appreciation of Common Cause Failures in Reliability’. In: *Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering* 212.1 (1998), pages 31–35. DOI: 10.1243/0954408981529277.
- [113] Aslett, L. J. M. *Reliability Theory: Tools for Structural Reliability Analysis*. Version 0.1.5. URL: <https://www.louisaslett.com/>.
- [114] Behrendorf, J., Brandt, S., Broggi, M. and Beer, M. ‘Efficient Approximation of the Survival Signature for Large Networks’. In: *Proceedings of the 6th International Symposium on Reliability Engineering and Risk Management*. Singapore: Research Publishing Services, 2018, pages 661–666. ISBN: 978-981-11-2726-7. DOI: 10.3850/978-981-11-2726-7_CRR14.
- [115] Pelessoni, R., Vicig, P., Montes, I. and Miranda, E. ‘Bivariate P-Boxes’. In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 24.02 (2016), pages 229–263. DOI: 10.1142/S0218488516500124.
- [116] Montes, I., Miranda, E., Pelessoni, R. and Vicig, P. ‘Sklar’s Theorem in an Imprecise Setting’. In: *Fuzzy Sets and Systems*. Special Issue on Uncertainty and Imprecision Modelling in Decision Making (EUROFUSE 2013) 278 (2015), pages 48–66. DOI: 10.1016/j.fss.2014.10.007.
- [117] Grigg, C., Wong, P., Albrecht, P. et al. ‘The IEEE Reliability Test System-1996. A Report Prepared by the Reliability Test System Task Force of the Application of Probability Methods Subcommittee’. In: *IEEE Transactions on Power Systems* 14.3 (1999), pages 1010–1020. DOI: 10.1109/59.780914.

- [118] Gruber, L. F. and Czado, C. ‘Bayesian Model Selection of Regular Vine Copulas’. In: *Bayesian Analysis* 13.4 (2018), pages 1111–1135. DOI: 10.1214/17-BA1089.
- [119] Nan, C. and Sansavini, G. ‘A Quantitative Method for Assessing Resilience of Interdependent Infrastructures’. In: *Reliability Engineering & System Safety* 157 (2017), pages 35–53. DOI: 10.1016/j.ress.2016.08.013.
- [120] Rocchetta, R. and Patelli, E. ‘Imprecise Probabilistic Framework for Power Grids Risk Assessment and Sensitivity Analysis’. In: *Risk, Reliability and Safety: Innovating Theory and Practice*. 26th European Safety and Reliability Conference. Edited by L. Walls, M. Revie and T. Bedford. London: CRC Press, 2016, pages 2789–2796. ISBN: 978-1-138-02997-2. DOI: 10.1201/9781315374987-424.
- [121] Boland, P. J. and Samaniego, F. J. ‘The Signature of a Coherent System and Its Applications in Reliability’. In: *Mathematical Reliability: An Expository Perspective*. Edited by R. Soyer, T. A. Mazzuchi and N. D. Singpurwalla. International Series in Operations Research & Management Science. Boston, MA: Springer, 2004, pages 3–30. ISBN: 978-1-4419-9021-1. DOI: 10.1007/978-1-4419-9021-1_1.
- [122] Li, Y., Coolen, F. P. A., Zhu, C. and Tan, J. ‘Reliability Assessment of the Hydraulic System of Wind Turbines Based on Load-Sharing Using Survival Signature’. In: *Renewable Energy* 153 (2020), pages 766–776. DOI: 10.1016/j.renene.2020.02.017.
- [123] Huang, X., Aslett, L. J. M. and Coolen, F. P. A. ‘Reliability Analysis of General Phased Mission Systems with a New Survival Signature’. In: *Reliability Engineering & System Safety* 189 (2019), pages 416–422. DOI: 10.1016/j.ress.2019.04.019.
- [124] Regenhardt, T.-E., Wei, L., Broggi, M. and Beer, M. ‘Applying Graph Theory and Lifeline Reliability to the System Survival Signature’. In: *Proceedings of the 6th International Symposium on Reliability Engineering and Risk Management*. Proceedings of the 6th International Symposium on Reliability Engineering and Risk Management. Singapore: Research Publishing Services, 2018, pages 681–686. ISBN: 978-981-11-2726-7. DOI: 10.3850/978-981-11-2726-7_CRR19.
- [125] Liang, H., Mi, J., Bai, L. and Cheng, Y. ‘Reliability Analysis of Discrete Multi-State Systems Based on Survival Signature’. In: *IOP Conference Series: Materials Science and Engineering* 1043.5 (2021), page 052029. DOI: 10.1088/1757-899X/1043/5/052029.
- [126] Broadbent, S. R. and Hammersley, J. M. ‘Percolation Processes: I. Crystals and Mazes’. In: *Mathematical Proceedings of the Cambridge Philosophical Society* 53.3 (1957), pages 629–641. DOI: 10.1017/S0305004100032680.
- [127] Li, D., Zhang, Q., Zio, E., Havlin, S. and Kang, R. ‘Network Reliability Analysis Based on Percolation Theory’. In: *Reliability Engineering & System Safety* 142 (2015), pages 556–562. DOI: 10.1016/j.ress.2015.05.021.

- [128] Bunde, A. and Havlin, S. ‘Percolation I’. In: *Fractals and Disordered Systems*. Edited by A. Bunde and S. Havlin. Berlin, Heidelberg: Springer, 1996, pages 59–114. ISBN: 978-3-642-84868-1. DOI: 10.1007/978-3-642-84868-1_2.
- [129] Callaway, D. S., Newman, M. E. J., Strogatz, S. H. and Watts, D. J. ‘Network Robustness and Fragility: Percolation on Random Graphs’. In: *Physical Review Letters* 85.25 (2000), pages 5468–5471. DOI: 10.1103/PhysRevLett.85.5468.
- [130] Havlin, S. and Bunde, A. ‘Percolation II’. In: *Fractals and Disordered Systems*. Edited by A. Bunde and S. Havlin. Berlin, Heidelberg: Springer, 1996, pages 115–176. ISBN: 978-3-642-84868-1. DOI: 10.1007/978-3-642-84868-1_3.
- [131] Molloy, M. and Reed, B. ‘A Critical Point for Random Graphs with a given Degree Sequence’. In: *Random Structures & Algorithms* 6.2-3 (1995), pages 161–180. DOI: 10.1002/rsa.3240060204.
- [132] Cohen, R., Erez, K., ben-Avraham, D. and Havlin, S. ‘Resilience of the Internet to Random Breakdowns’. In: *Physical Review Letters* 85.21 (2000), pages 4626–4628. DOI: 10.1103/PhysRevLett.85.4626.
- [133] Bukhsh, W. A. and McKinnon, K. *Network Data of Real Transmission Networks*. Power Systems Test Case Archive. 2013. URL: <https://www.maths.ed.ac.uk/optenergy/NetworkData/index.html>.
- [134] Latora, V. and Marchiori, M. ‘Efficient Behavior of Small-World Networks’. In: *Physical Review Letters* 87.19 (2001), page 198701. DOI: 10.1103/PhysRevLett.87.198701.
- [135] Floyd, R. W. ‘Algorithm 97: Shortest Path’. In: *Communications of the ACM* 5.6 (1962), page 345. DOI: 10.1145/367766.368168.
- [136] Pradlwarter, H. J., Schuëller, G. I., Koutsourelakis, P. S. and Charmpis, D. C. ‘Application of Line Sampling Simulation Method to Reliability Benchmark Problems’. In: *Structural Safety. A Benchmark Study on Reliability in High Dimensions* 29.3 (2007), pages 208–221. DOI: 10.1016/j.strusafe.2006.07.009.
- [137] Au, S.-K. and Beck, J. L. ‘Estimation of Small Failure Probabilities in High Dimensions by Subset Simulation’. In: *Probabilistic Engineering Mechanics* 16.4 (2001), pages 263–277. DOI: 10.1016/S0266-8920(01)00019-4.
- [138] Broomhead, D. and Lowe, D. ‘Multivariable Functional Interpolation and Adaptive Networks’. In: *Complex Systems* 2.3 (1988).
- [139] Heimes, F. and van Heuveln, B. ‘The Normalized Radial Basis Function Neural Network’. In: *SMC’98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.98CH36218)*. Volume 2. 1998, 1609–1614 vol.2. DOI: 10.1109/ICSMC.1998.728118.

- [140] Hušek, P. ‘On Monotonic Radial Basis Function Networks’. In: *IEEE Transactions on Cybernetics* (2022), pages 1–11. DOI: 10.1109/TCYB.2022.3185827.
- [141] O’Donoghue, B., Chu, E., Parikh, N. and Boyd, S. ‘Conic Optimization via Operator Splitting and Homogeneous Self-Dual Embedding’. In: *Journal of Optimization Theory and Applications* 169 (2016), pages 1042–1068. DOI: 10.1007/s10957-016-0892-3.
- [142] Lubin, M., Dowson, O., Dias Garcia, J., Huchette, J., Legat, B. and Vielma, J. P. ‘JuMP 1.0: Recent Improvements to a Modeling Language for Mathematical Optimization’. In: *Mathematical Programming Computation* (2023). DOI: 10.1007/s12532-023-00239-3.
- [143] Mo, S., Lu, D., Shi, X., Zhang, G., Ye, M., Wu, J. and Wu, J. ‘A Taylor Expansion-Based Adaptive Design Strategy for Global Surrogate Modeling With Applications in Groundwater Modeling’. In: *Water Resources Research* 53.12 (2017), pages 10802–10823. DOI: 10.1002/2017WR021622.
- [144] Crespo, L. G., Giesy, D. P. and Kenny, S. P. ‘Interval Predictor Models with a Formal Characterization of Uncertainty and Reliability’. In: *53rd IEEE Conference on Decision and Control*. 2014, pages 5991–5996. DOI: 10.1109/CDC.2014.7040327.
- [145] Dong, W. and Shah, H. C. ‘Vertex Method for Computing Functions of Fuzzy Variables’. In: *Fuzzy Sets and Systems. An International Journal in Information Science and Engineering* 24.1 (1987), pages 65–78. DOI: 10.1016/0165-0114(87)90114-X.
- [146] Da Silva, D. B. L. and Sullivan, K. M. *An Optimization-Based Monte Carlo Method for Estimating the Two-Terminal Survival Signature of Networks with Two Component Classes*. Submitted. 2023. URL: https://industrial-engineering.uark.edu/_resources/TwoTerminal-SurvivalSig-MC.pdf.
- [147] Buhmann, M. D. *Radial Basis Functions: Theory and Implementations*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge: Cambridge University Press, 2003. ISBN: 978-0-521-63338-3. DOI: 10.1017/CB09780511543241.
- [148] Udell, M., Mohan, K., Zeng, D., Hong, J., Diamond, S. and Boyd, S. ‘Convex Optimization in Julia’. In: *2014 First Workshop for High Performance Technical Computing in Dynamic Languages*. 2014, pages 18–28. DOI: 10.1109/HPTCDL.2014.5.
- [149] Crespo, L. G., Kenny, S. P. and Giesy, D. P. ‘Interval Predictor Models With a Linear Parameter Dependency’. In: *Journal of Verification, Validation and Uncertainty Quantification* 1.021007 (2016). DOI: 10.1115/1.4032070.
- [150] Punzo, G., Tewari, A., Butans, E., Vasile, M., Purvis, A., Mayfield, M. and Varga, L. ‘Engineering Resilient Complex Systems: The Necessary Shift toward Complexity Science’. In: *IEEE Systems Journal* 14.3 (2020), pages 3865–3874. DOI: 10.1109/JSYST.2019.2958829.

- [151] Tran, H. T., Balchanos, M., Domercant, J. C. and Mavris, D. N. ‘A Framework for the Quantitative Assessment of Performance-Based System Resilience’. In: *Reliability Engineering & System Safety* 158 (2017), pages 73–84. DOI: 10.1016/j.ress.2016.10.014.
- [152] Cimellaro, G. P., Reinhorn, A. M. and Bruneau, M. ‘Framework for Analytical Quantification of Disaster Resilience’. In: *Engineering Structures* 32.11 (2010), pages 3639–3649. DOI: 10.1016/j.engstruct.2010.08.008.
- [153] Ayyub, B. M. ‘Practical Resilience Metrics for Planning, Design, and Decision Making’. In: *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering* 1.3 (2015), page 04015008. DOI: 10.1061/ajrua6.0000826.
- [154] Fang, Y., Pedroni, N. and Zio, E. ‘Optimization of Cascade-Resilient Electrical Infrastructures and Its Validation by Power Flow Modeling’. In: *Risk Analysis* 35.4 (2015), pages 594–607. DOI: 10.1111/risa.12396.
- [155] Cimellaro, G. P. *Urban Resilience for Emergency Response and Recovery*. Geotechnical, Geological and Earthquake Engineering. Switzerland: Springer International Publishing, 2016. ISBN: 978-3-319-30655-1. DOI: 10.1007/978-3-319-30656-8.
- [156] Sharifi, A. ‘Urban Resilience Assessment: Mapping Knowledge Structure and Trends’. In: *Sustainability* 12.15 (2020), page 5918. DOI: 10.3390/su12155918.
- [157] Patriarca, R., Bergström, J., Di Gravio, G. and Costantino, F. ‘Resilience Engineering: Current Status of the Research and Future Challenges’. In: *Safety Science* 102 (2018), pages 79–100. DOI: 10.1016/j.ssci.2017.10.005.
- [158] Holling, C. S. ‘Resilience and Stability of Ecological Systems’. In: *Annual Review of Ecology and Systematics* 4 (1973), pages 1–23. DOI: 10.1146/annurev.es.04.110173.000245.
- [159] Fiksel, J. ‘Designing Resilient, Sustainable Systems’. In: *Environmental Science and Technology* 37.23 (2003), pages 5330–5339. DOI: 10.1021/es0344819.
- [160] Little, R. G. ‘Toward More Robust Infrastructure: Observations on Improving the Resilience and Reliability of Critical Systems’. In: *Proceedings of the 36th Annual Hawaii International Conference on System Sciences, HICSS 2003* (2003). DOI: 10.1109/HICSS.2003.1173880.
- [161] Hollnagel, E., Woods, D. E. and Levensen, N. *Resilience Engineering: Concepts and Precepts*. Aldershot, UK: Ashgate Publishing, 2006.
- [162] Bruneau, M. and Reinhorn, A. ‘Exploring the Concept of Seismic Resilience for Acute Care Facilities’. In: *Earthquake Spectra* 23.1 (2007), pages 41–62. DOI: 10.1193/1.2431396.

- [163] Youn, B. D., Hu, C. and Wang, P. ‘Resilience-Driven System Design of Complex Engineered Systems’. In: *Journal of Mechanical Design* 133.10 (2011), page 101011. DOI: 10.1115/1.4004981.
- [164] Ayyub, B. M. ‘Systems Resilience for Multihazard Environments: Definition, Metrics, and Valuation for Decision Making’. In: *Risk Analysis* 34.2 (2014), pages 340–355. DOI: 10.1111/risa.12093.
- [165] Presidential Policy Directive (PPD). *Critical Infrastructure Security and Resilience*. 2013. URL: <https://obamawhitehouse.archives.gov/the-press-office/2013/02/12/presidential-policy-directive-critical-infrastructure-security-and-resil>.
- [166] Gilbert, S. and Ayyub, B. M. ‘Models for the Economics of Resilience’. In: *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering* 2.4 (2016). DOI: 10.1061/AJRUA6.0000867.
- [167] Fang, Y. and Sansavini, G. ‘Optimizing Power System Investments and Resilience against Attacks’. In: *Reliability Engineering and System Safety* 159 (2017), pages 161–173. DOI: 10.1016/j.ress.2016.10.028.
- [168] Singh, R. R., Bruneau, M., Stavridis, A. and Sett, K. ‘Resilience Deficit Index for Quantification of Resilience’. In: *Resilient Cities and Structures* 1.2 (2022), pages 1–9. DOI: 10.1016/j.rcns.2022.06.001.
- [169] Alipour, A. and Shafei, B. ‘An Overarching Framework to Assess the Life-Time Resilience of Deteriorating Transportation Networks in Seismic-Prone Regions’. In: *Resilient Cities and Structures* 1.2 (2022), pages 87–96. DOI: 10.1016/j.rcns.2022.07.002.
- [170] Capacci, L., Biondini, F. and Frangopol, D. M. ‘Resilience of Aging Structures and Infrastructure Systems with Emphasis on Seismic Resilience of Bridges and Road Networks: Review’. In: *Resilient Cities and Structures* 1.2 (2022), pages 23–41. DOI: 10.1016/j.rcns.2022.05.001.
- [171] Duan, S. and Ayyub, B. M. ‘Assessment Methods of Network Resilience for Cyber-Human-Physical Systems’. In: *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering* 6.1 (2020), page 03119001. DOI: 10.1061/AJRUA6.000102.
- [172] Emanuel, R. and Ayyub, B. ‘Assessing Resilience Model Responsiveness in the Context of Stakeholder Preferences in Decision Support Systems’. In: *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering* 5.2 (2019), page 04019005. DOI: 10.1061/AJRUA6.0001002.

- [173] Saadat, Y., Ayyub, B. M., Zhang, Y., Zhang, D. and Hongwei Huang. ‘Resilience-Based Strategies for Topology Enhancement and Recovery of Metrorail Transit Networks’. In: *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering* 6.2 (2020), page 04020017. DOI: 10.1061/AJRUA6.0001057.
- [174] Zhang, Y., Ayyub, B. M. and Fung, J. F. ‘Projections of Corrosion and Deterioration of Infrastructure in United States Coasts under a Changing Climate’. In: *Resilient Cities and Structures* 1.1 (2022), pages 98–109. DOI: 10.1016/j.rcns.2022.04.004.
- [175] Nielsen, C. B., Larsen, P. G., Fitzgerald, J., Woodcock, J. and Peleska, J. ‘Systems of Systems Engineering: Basic Concepts, Model-Based Techniques, and Research Directions’. In: *ACM Computing Surveys* 48.2 (2015), pages 1–41. DOI: 10.1145/2794381.
- [176] Jamshidi, M. *Systems of Systems Engineering: Principles and Applications*. CRC press, 2017. ISBN: 978-1-315-21894-6.
- [177] Batty, M. ‘Complexity in City Systems: Understanding, Evolution, and Design’. In: *A Planner’s Encounter with Complexity*. Routledge, 2016, pages 99–122.
- [178] Kolowrocki, K. *Reliability of Large and Complex Systems*. Elsevier, 2014.
- [179] Negi, S. and Singh, S. B. ‘Reliability Analysis of Non-Repairable Complex System with Weighted Subsystems Connected in Series’. In: *Applied Mathematics and Computation* 262 (2015), pages 79–89. DOI: 10.1016/j.amc.2015.03.119.
- [180] Coolen, F. P. and Coolen-Maturi, T. ‘The Structure Function for System Reliability as Predictive (Imprecise) Probability’. In: *Reliability Engineering & System Safety* 154 (2016), pages 180–187. DOI: 10.1016/j.ress.2016.06.008.
- [181] Salomon, J., Winnewisser, N., Wei, P., Broggi, M. and Beer, M. ‘Efficient Reliability Analysis of Complex Systems in Consideration of Imprecision’. In: *Reliability Engineering & System Safety* 216 (2021), page 107972. DOI: 10.1016/j.ress.2021.107972.
- [182] Sun, W., Bocchini, P. and Davison, B. D. ‘Resilience Metrics and Measurement Methods for Transportation Infrastructure: The State of the Art’. In: *Sustainable and Resilient Infrastructure* 5.3 (2020), pages 168–199. DOI: 10.1080/23789689.2018.1448663.
- [183] Henry, D. and Ramirez-Marquez, J. E. ‘Generic Metrics and Quantitative Approaches for System Resilience as a Function of Time’. In: *Reliability Engineering and System Safety* 99 (2012), pages 114–122. DOI: 10.1016/j.ress.2011.09.002.
- [184] Zobel, C. W. ‘Representing Perceived Tradeoffs in Defining Disaster Resilience’. In: *Decision Support Systems* 50.2 (2011), pages 394–403. DOI: 10.1016/j.dss.2010.10.001.
- [185] Adams, T. M., Bekkem, K. R. and Toledo-Durán, E. J. ‘Freight Resilience Measures’. In: *Journal of Transportation Engineering* 138.11 (2012), pages 1403–1409. DOI: 10.1061/(ASCE)TE.1943-5436.0000415.

- [186] Sahebjamnia, N., Torabi, S. A. and Mansouri, S. A. ‘Integrated Business Continuity and Disaster Recovery Planning: Towards Organizational Resilience’. In: *European Journal of Operational Research* 242.1 (2015), pages 261–273. DOI: 10.1016/j.ejor.2014.09.055.
- [187] Weber, S. and Weske, K. ‘The Joint Impact of Bankruptcy Costs, Fire Sales and Cross-Holdings on Systemic Risk in Financial Networks’. In: *Probability, Uncertainty and Quantitative Risk* 2.1 (2017). DOI: 10.1186/s41546-017-0020-9.
- [188] Liu, Y., Shi, Y., Bai, X. and Liu, B. ‘Stress–Strength Reliability Analysis of Multi-State System Based on Generalized Survival Signature’. In: *Journal of Computational and Applied Mathematics* 342 (2018), pages 274–291. DOI: 10.1016/j.cam.2018.03.041.
- [189] Yi, H., Cui, L. and Balakrishnan, N. ‘Computation of Survival Signatures for Multi-State Consecutive-k Systems’. In: *Reliability Engineering & System Safety* 208 (2021), page 107429. DOI: 10.1016/j.ress.2021.107429.
- [190] Qin, J. and Coolen, F. P. ‘Survival Signature for Reliability Evaluation of a Multi-State System with Multi-State Components’. In: *Reliability Engineering & System Safety* 218 (2022), page 108129. DOI: 10.1016/j.ress.2021.108129.
- [191] Miro, S., Willeke, T., Broggi, M., Seume, J. R. and Beer, M. ‘Reliability Analysis of an Axial Compressor Based on One-Dimensional Flow Modeling and Survival Signature’. In: *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering* 5.3 (3 2019). DOI: 10.1115/1.4043150.
- [192] Braun, M. and Seume, J. ‘Forward Sweep in a Four-Stage High-Speed Axial Compressor’. In: *ASME Turbo Expo 2006: Power for Land, Sea, and Air* (2006), pages 141–152. DOI: 10.1115/GT2006-90218.
- [193] Hellmich, B. and Seume, J. ‘Causes of Acoustic Resonance in a High-Speed Axial Compressor’. In: *Journal of Turbomachinery* 130.3 (2008), 031003 (9 pages). DOI: 10.1115/1.2775487.
- [194] Siemann, J., Krenz, I. and Seume, J. ‘Experimental Investigation of Aspiration in a Multi-Stage High-Speed Axial-Compressor’. In: *ASME Turbo Expo 2016: Turbomachinery Technical Conference and Exposition* (2016). DOI: 10.1115/GT2016-56440.
- [195] Berliner Verkehrsbetriebe. *Lagebericht Und Jahresabschluss 2017*. BVG / BVG, 2018. URL: <http://unternehmen.bvg.de/index.php?section=downloads%7B%5C%7Ddownload=3128>.
- [196] S-Bahn Berlin. 2018. URL: <https://sbahn.berlin/das-unternehmen/unternehmensprofil/s-bahn-berlin-auf-einen-blick/>.
- [197] Zuev, K. M., Wu, S. and Beck, J. L. ‘General Network Reliability Problem and Its Efficient Solution by Subset Simulation’. In: *Probabilistic Engineering Mechanics* 40 (2015), pages 25–35. DOI: 10.1016/j.probengmech.2015.02.002.

Bibliography

- [198] Zhang, J., Zhao, M., Liu, H. and Xu, X. ‘Networked Characteristics of the Urban Rail Transit Networks’. In: *Physica A: Statistical Mechanics and its Applications* 392.6 (2013), pages 1538–1546. DOI: 10.1016/j.physa.2012.11.036.
- [199] Dreyfus, S. E. ‘An Appraisal of Some Shortest-Path Algorithms’. In: *Operations Research* 17.3 (1969), pages 395–412. DOI: 10.1287/opre.17.3.395.
- [200] Zhan, F. B. and Noon, C. E. ‘Shortest Path Algorithms: An Evaluation Using Real Road Networks’. In: *Transportation Science* 32 (1998), pages 65–73. DOI: 10.1287/trsc.32.1.65.
- [201] Crucitti, P., Latora, V. and Marchiori, M. ‘Model for Cascading Failures in Complex Networks’. In: *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics* 69.4 (2004), 045104 (4 pages). DOI: 10.1103/PhysRevE.69.045104.
- [202] Dueñas-Osorio, L. and Vemuru, S. M. ‘Cascading Failures in Complex Infrastructure Systems’. In: *Structural Safety* 31.2 (2009), pages 157–167. DOI: 10.1016/j.strusafe.2008.06.007.
- [203] Au, S.-K. and Wang, Y. *Engineering Risk Assessment with Subset Simulation*. John Wiley & Sons, 2014. DOI: 10.1002/9781118398050.
- [204] Besançon, M., Papamarkou, T., Anthoff, D., Arslan, A., Byrne, S., Lin, D. and Pearson, J. ‘Distributions.jl: Definition and Modeling of Probability Distributions in the JuliaStats Ecosystem’. In: *Journal of Statistical Software* 98 (2021), pages 1–30. DOI: 10.18637/jss.v098.i16.
- [205] Koutsourelakis, P. S., Pradlwarter, H. J. and Schuëller, G. I. ‘Reliability of Structures in High Dimensions, Part I: Algorithms and Applications’. In: *Probabilistic Engineering Mechanics* 19.4 (2004), pages 409–417. DOI: 10.1016/j.probengmech.2004.05.001.
- [206] Gray, A., Behrendorf, J., amrods and Schilling, C. *AnderGray/BivariateCopulas.jl*. Version 0.1.4. 2023. DOI: 10.5281/zenodo.7822922.
- [207] *JuliaParallel/ClusterManagers.jl*. Version 0.4.4. 2023. URL: <https://github.com/JuliaParallel/ClusterManagers.jl>.
- [208] *Jverzani/Mustache.jl*. Version 1.0.14. URL: <https://github.com/jverzani/Mustache.jl>.
- [209] *JuliaIO/Formatting.jl*. Version 0.4.2. 2020. URL: <https://github.com/JuliaIO/Formatting.jl>.
- [210] Hasofer, A. M. and Lind, N. C. ‘Exact and Invariant Second-Moment Code Format’. In: *Journal of the Engineering Mechanics Division* 100.1 (1974), pages 111–121. DOI: 10.1061/JMCEA3.0001848.

- [211] Rackwitz, R. and Flessler, B. ‘Structural Reliability under Combined Random Load Sequences’. In: *Computers & Structures* 9.5 (1978), pages 489–494. DOI: 10.1016/0045-7949(78)90046-9.
- [212] Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M. and Tarantola, S. *Global Sensitivity Analysis. The Primer*. Chichester, UK: John Wiley & Sons, Ltd, 2007. ISBN: 978-0-470-05997-5. DOI: 10.1002/9780470725184.
- [213] Jansen, M. J. W. ‘Analysis of Variance Designs for Model Output’. In: *Computer Physics Communications* 117.1 (1999), pages 35–43. DOI: 10.1016/S0010-4655(98)00154-4.
- [214] Saltelli, A., Annoni, P., Azzini, I., Campolongo, F., Ratto, M. and Tarantola, S. ‘Variance Based Sensitivity Analysis of Model Output. Design and Estimator for the Total Sensitivity Index’. In: *Computer Physics Communications* 181.2 (2010), pages 259–270. DOI: 10.1016/j.cpc.2009.09.018.
- [215] Beatson, R. K., Powell, M. J. D. and Tan, A. M. ‘Fast Evaluation of Polyharmonic Splines in Three Dimensions’. In: *IMA Journal of Numerical Analysis* 27.3 (2007), pages 427–450. DOI: 10.1093/imanum/drl027.
- [216] Forrester, A. I. J., Sóbester, A. and Keane, A. J. *Engineering Design via Surrogate Modelling: A Practical Guide*. Wiley, 2008. ISBN: 978-0-470-06068-1. URL: 10.1002/9780470770801.
- [217] Zuev, K. ‘Subset Simulation Method for Rare Event Estimation: An Introduction’. 2015. DOI: 10.48550/arXiv.1505.03506.
- [218] Patelli, E. and Au, S. K. ‘Efficient Monte Carlo Algorithm for Rare Failure Event Simulation’. In: *Proceedings of the 12th International Conference on Applications of Statistics and Probability in Civil Engineering (ICASP12)*. Edited by T. Haukaas. Vancouver, Canada, 2015. DOI: 10.14288/1.0076089.
- [219] Ishigami, T. and Homma, T. ‘An Importance Quantification Technique in Uncertainty Analysis for Computer Models’. In: *Proceedings. First International Symposium on Uncertainty Modeling and Analysis*. 1990, pages 398–403. DOI: 10.1109/ISUMA.1990.151285.
- [220] Marrel, A., Iooss, B., Laurent, B. and Roustant, O. ‘Calculations of Sobol Indices for the Gaussian Process Metamodel’. In: *Reliability Engineering & System Safety* 94.3 (2008), pages 742–751. DOI: 10.1016/j.ress.2008.07.008.
- [221] Sobol’, I. M. and Levitan, Yu. L. ‘On the Use of Variance Reducing Multipliers in Monte Carlo Computations of a Global Sensitivity Index’. In: *Computer Physics Communications* 117.1 (1999), pages 52–61. DOI: 10.1016/S0010-4655(98)00156-8.

Bibliography

- [222] Sudret, B. ‘Global Sensitivity Analysis Using Polynomial Chaos Expansions’. In: *Reliability Engineering & System Safety*. Bayesian Networks in Dependability 93.7 (2008), pages 964–979. DOI: 10.1016/j.ress.2007.04.002.

Curriculum Vitae

Personal details

Name	Jasper Behrendorf
Date of birth	23 November 1989
Place of birth	Hannover, Germany
Nationality	German

Education

since 04/2017	PhD student Institute for Risk and Reliability, Leibniz Universität Hannover
04/2014 – 09/2016	Master of Science, Computational Engineering Thesis: <i>Comparing isogeometric mesh generation techniques for curve-to-area parametrization</i> Leibniz Universität Hannover
10/2009 – 03/2014	Bachelor of Science, Computational Engineering Leibniz Universität Hannover Thesis: <i>Konzeptionierung und softwaretechnische Umsetzung einer Prüfstandssimulation für Schulungszwecke</i>

Work experience

since 09/2016	Research assistant Institute for Risk and Reliability, Leibniz Universität Hannover
07/2015 – 09/2016	Student assistant Institute for Risk and Reliability, Leibniz Universität Hannover
11/2014 – 05/2015	Intern Bentley Motors Limited, Crewe, United Kingdom
02/2011 – 09/2014	Student assistant Institut für Bauinformatik, Leibniz Universität Hannover