Springer Series in Reliability Engineering

Chao Hu · Byeng D. Youn
Pingfeng Wang

Engineering Design under Uncertainty and Health Prognostics

Springer

Springer Series in Reliability Engineering

Series editor

Hoang Pham, Piscataway, USA

More information about this series at http://www.springer.com/series/6917

Chao Hu · Byeng D. Youn
Pingfeng Wang

Engineering Design under Uncertainty and Health Prognostics

Springer

Chao Hu
Department of Mechanical Engineering
Iowa State University
Ames, IA
USA

and

Department of Electrical and Computer Engineering
Iowa State University
Ames, IA
USA

Byeng D. Youn
School of Mechanical and Aerospace Engineering
Seoul National University
Seoul
Korea (Republic of)

Pingfeng Wang
Department of Industrial and Enterprise Systems Engineering
University of Illinois at Urbana–Champaign
Urbana, IL
USA

ISSN 1614-7839 ISSN 2196-999X (electronic)
Springer Series in Reliability Engineering
ISBN 978-3-319-92572-1 ISBN 978-3-319-92574-5 (eBook)
https://doi.org/10.1007/978-3-319-92574-5

Library of Congress Control Number: 2018942617

Printed on acid-free paper

This Springer imprint is published by the registered company Springer International Publishing AG part of Springer Nature
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

Failures of engineered systems can result in enormous repair/replacement costs and can also cause life-threatening consequences, such as explosion and fire. Since the 1980s, major industries and government agencies worldwide have faced increasing challenges in ensuring the reliability and safety of engineered systems. Examples of failures in attaining high reliability include the Chernobyl disaster in Russia (1986), the collapse of the I-35W Mississippi River Bridge in the USA (2007), and the lithium-ion (Li-ion) battery fire/smoke issues on Boeing 787 Dreamliners in the USA and Japan (2013). The rapidly increasing failure costs associated with these failures, along with stringent reliability and safety requirements, have resulted in considerable research attention directed toward developing probabilistic analysis and design methods that can be used to analyze and improve the reliability and safety of engineered systems. This book presents a comprehensive description of these probabilistic methods. The text is packed with many practical engineering examples (e.g., electric power transmission systems, aircraft power generating systems, and mechanical transmission systems) and exercise problems. It is an up-to-date, fully illustrated reference suitable for both undergraduate and graduate engineering students, researchers, and professional engineers who are interested in exploring the fundamentals, implementation, and applications of probabilistic analysis and design methods. The probabilistic methods presented in this book include (i) conventional statistical methods and Bayesian statistics for data analysis; (ii) direct and smart Monte Carlo simulation methods, first- and second-order reliability methods, and stochastic response surface methods for reliability analysis in design; (iii) reliability-based design optimization; and (iv) signal processing, classification, and regression methods for reliability analysis in operation (prognostics and health management). The contents of the book provide a sufficient working knowledge to enable all interested parties to perform probabilistic analysis and design.

Chapters 2 and 3 present fundamental probability theory and commonly used statistical data analysis techniques. This foundation provides a sufficient background for the in-depth investigations in later chapters that address the above-mentioned challenges. The statistical methods presented in these two chapters serve as an

important tool for reliability engineers because they provide both descriptive and analytical ways to deal with the uncertainty in simulation or experimental data.

The rest of the chapters in this book address the two main challenges in the practice of life-cycle reliability engineering: (i) how to design a reliable engineered system during the design stage and (ii) how to achieve high operational reliability during operation of a system. Reliability-based design with time-independent reliability analysis tackles the first challenge. Addressing this challenge is the primary focus of this book. Chapter 4 presents the fundamentals of reliability analysis, including the concepts and formulations of time-independent and time-dependent reliability analyses. Chapter 5 introduces state-of-the-art techniques for time-independent reliability analysis, including the first- and second-order reliability methods (FORM/SORM), direct and smart Monte Carlo simulation (MCS), and emerging stochastic response surface methods. Chapter 6 discusses the advanced topic of reliability analysis: time-dependent reliability analysis in design. Chapter 7 explains how to design a reliable product by introducing reliability-based design optimization (RBDO).

Health monitoring, diagnostics, prognostics, and management strategies have been proposed to address the second challenge. These techniques have been cohesively integrated to the point where a new discipline has emerged: prognostics and health management (PHM). In recent years, PHM has been successfully applied to many engineered systems to assess their health conditions in real time under actual operating conditions and to adaptively enhance life-cycle reliabilities through condition-based maintenance, which allows for the anticipation and prevention of unexpected system failures. Chapter 8 discusses the current state-of-the-art techniques from this emerging discipline and Chap. 9 presents successful practices in several engineering fields.

Each chapter has an extensive collection of examples and exercises, including engineering and/or mathematical examples that illustrate the material in that chapter. Supplemental exercises are also provided to allow learners to practice application to improve understanding of the topics discussed in that chapter.

The authors would like to acknowledge that the work presented in this book was partially supported by the following organizations: the US National Science Foundation (NSF): the NSF I/UCRC Center for e-Design, the US Army TARDEC, the US Nuclear Regulatory Commission (NRC), the Maryland Industrial Partnerships Program (MIPS), the Korea Institute of Energy Technology Evaluation and Planning (KETEP), the National Research Foundation (NRF) of Korea, the Korea Institute of Machinery and Materials, the Korea Agency for Infrastructure Technology Advancement (KAIA), General Motors, LG Electronics, and Samsung Electronics.

Ames, USA — Chao Hu
Seoul, Korea (Republic of) — Byeng D. Youn
Urbana, USA — Pingfeng Wang

Contents

Chapter 1
Basic Concepts of Reliability

In the past few decades, reliability has been widely recognized as of great importance in engineering product design and development. Hence, considerable advances have been made in the field of *reliability-based design optimization* (*RBDO*), resulting in new techniques for analyzing and improving the reliability of an engineered system, while taking into account various sources of uncertainty (e.g., material properties, loads, and geometric tolerances). Additionally, advanced *maintenance* strategies have been developed to help ensure systems operate reliably throughout their lifetime. This chapter introduces basic concepts of reliability, provides an overview of the history of reliability engineering, presents the way reliability efforts can be integrated into product design and development, and provides a framework for the material that will be covered in the subsequent chapters.

1.1 Concept of Reliability

Failures of engineered systems cause significant economic and societal losses. Although today's U.S. industry spends more than $200 billion each year on reliability and maintenance [1], catastrophic unexpected failures of engineered systems still take place due to the ever-growing system complexity and increasing levels of stress (e.g., high pressure, high temperature, and high irradiation field) imposed on these systems. The rapidly increasing failure costs, along with stringent reliability and safety requirements, have resulted in considerable research efforts directed towards the fields of reliability, risk, and maintainability, which, in turn, has led to significant research advances in these fields. The practice of reliability engineering not only aims at designing an engineered system that performs its required function without failure at the very beginning of operation (i.e., $t = 0$), but also focuses on maintaining the functionality of the system and avoiding catastrophic failure under stated conditions for a specified period of time (i.e., $t > 0$). Due to the existence of

C. Hu et al., *Engineering Design under Uncertainty and Health Prognostics*,
Springer Series in Reliability Engineering,
https://doi.org/10.1007/978-3-319-92574-5_1

various uncertainty sources (e.g., material properties, loads, operating conditions), an event in which a specific system performs its required function under stated conditions for a specified period of time should be treated as a random phenomenon, the characterization of which requires the use of the methods of probability and statistics. The probability of this event is referred to as the system's *reliability*, which, from the perspective of statistics, can be described as the ratio of the number of systems that successfully perform their required functions to the number of all statistically identical systems (here, we assume that there are an infinite number of systems).

In reliability or risk assessment of engineered systems, probability and statistics are both important, relevant, and useful subjects. A clear distinction must be drawn between the two subjects in order to promote a better understanding.

- *Probability* is primarily a *theory*-oriented branch of mathematics. Probability involves computation of the likelihood of specific events. It enables us to analyze consequences under strict assumptions.
- *Statistics* is primarily an *application*-oriented branch of mathematics. Statistics focuses on the analysis of the frequency of past events (or quantification of uncertainty based on available observations from the real world). Statistics can be used to set up appropriate assumptions for probability analysis.

In summary, in this context, reliability is a probability, the quantification of which in the real-world requires the use of statistics. The elaboration above clarifies the main issue we attempt to address in this book, specifically: How can reliability be analyzed using probability and statistical methods?

The term "reliability," based on the temporal characteristic, can be categorized into the following two types:

(i) *Time-independent reliability*—referring to the probability that a system's performance (e.g., fatigue, corrosion, and fracture) meets its marginal value (or requirement) under uncertainty at the very beginning of operation ($t = 0$); and
(ii) *Time-dependent reliability*—referring to the probability that the system meets this requirement for a stated period of time over the system's expected lifespan ($t > 0$).

It should be noted that the concept of *time-independent reliability* is mainly used for designing an engineered system in a way that ensures a high built-in reliability; whereas, the concept of *time-dependent reliability* is often employed to design an engineered system and/or its affiliated PHM system to attain a high operational reliability. The purpose of this book is to present advanced methods and tools that can be used to quantify these two types of reliability, as well as to introduce techniques and ways that can be leveraged to eliminate or avoid failure, or reduce the probability of failure to an acceptable level. The ultimate goal of this field is to enable engineered systems to achieve and sustain near-zero breakdown performance.

1.2 History of Reliability Engineering

Reliability is an important attribute of any modern engineered system. The use of "reliability" can be traced back to the year 1816, when an English poet Samuel Taylor Coleridge, for the first time, coined the word "reliability" in praise of his friend [2] in the following lines:

> … he inflicts none of those small pains and discomforts which irregular men scatter about them, and which in the aggregate so often become formidable obstacles both to happiness and utility; while on the contrary he bestows all the pleasures, and inspires all that ease of mind on those around him or connected with him, which perfect consistency, and (if such a word might be framed) absolute reliability …

Even since this initial "apologetic" beginning, reliability has grown into an omnipresent attribute that plays an essential role in the safe and effective operation of almost any modern engineered system. The persuasive nature of reliability in the common public and academic community cannot be overstated. In fact, a quick search of the word "reliability" with Google results in over 100 million results found on the web [3].

From 1816 through 2015, the occurrence of several key events and developments has led to the establishment of reliability engineering as a scientific discipline. This scientific discipline started to be established in the mid-1950s and subsequently maintained rapid development through tremendous support from a vast community of academia, industry, and government constituents. Despite numerous social, cultural, and technological achievements enabled by reliability engineering, many challenges still await in the future. In what follows, we briefly review key events and developments in the chronological history of reliability engineering.

Our review starts before the first use of the word "reliability" and points out the essential theoretical foundation of reliability, i.e., the theory of probability and statistics, which has supported the establishment of reliability engineering as a scientific discipline. The theory was initially developed in 1600s to address a series of questions and interests in gaming and gambling by Blaise Pascal and Pierre de Fermat [3]. In the 1800s, Laplace further expanded the application domain of this theory into many practical problems. In addition to probability and statistics as the theoretical enabler for the emergence of reliability engineering, the concept of mass production for standardized parts (rifle manufacturing by the Springfield armory in 1795 and Ford Model T car production in 1913) also played an essential role as a practical enabler [3].

Besides these two enablers, the catalyst for the rise of reliability engineering has been recognized as the vacuum tube, or more specifically, the triode invented by Lee de Forest in 1906. By initializing the electronic revolution, the vacuum tube led to a series of applications such as the radio, television, and radar. The vacuum tube, which was deemed as the active element contributing to the victory of the Allies in World War II, was the primary source of equipment failure due to its much more (about four times more) frequent occurrence of failure than all other equipment.

After the war, these failure events motivated the US Department of Defense to organize investigations into these events, which eventually led to the emergence of reliability engineering as a scientific discipline in the 1950s. This new discipline was, for the first time, consolidated and synthesized in the Advisory Group on Reliability of Electronic Equipment (AGREE) report in 1957. The AGREE was jointly established in 1952 between the Department of Defense and the American Electronics Industry for the following missions [4]:

(1) To recommend measures that would result in equipment that is more reliable;
(2) To help implement reliability programs in government and civilian agencies;
(3) To disseminate a better education on reliability.

With the objective to achieve higher reliability, military-funded projects were launched and a great deal of effort was devoted to failure data collection and root cause analysis. Furthermore, the specification of quantitative reliability requirements emerged as the beginning of a contractual aspect of reliability. These reliability requirements necessitated the development of reliability prediction techniques to estimate and predict the reliability of a component before it was built and tested. The milestone of this development was the publication of a major report (TR-1100) titled "Reliability Stress Assessment for Electronic Equipment" by the Radio Corporation of America, a major manufacturer of vacuum tubes. Analytical models for estimating component failure rates were introduced in the report, which then facilitated the publication of the influential military standard MH-217 in 1961. This standard is still being used today for reliability prediction [5]. A timeline of the aforementioned key events that contributed to the emergence of reliability engineering is shown in Fig. 1.1.

In the 1960s, the decade of the first development phase of reliability engineering, the discipline proceeded along two tracks [3]:

(1) Increased specialization in the discipline, consisting of increased sophistication of statistical techniques (e.g., redundancy modeling, Bayesian statistics, Markov chains), the emergence of the discipline of Reliability Physics to identify and model physical causes of failure, and the development of a separate subject, Structural Reliability, to assess the structural integrity of buildings, bridges, and other construction;

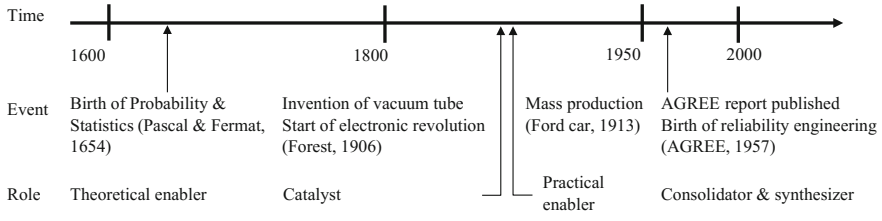


Fig. 1.1 Timeline of key events leading to the birth of reliability engineering

(2) Shifting of the focus from component reliability to system reliability to deal with increasingly complex engineered systems (e.g., ICBMs, the swing-wing F-111, the US space shuttle).

The 1970s witnessed work in three broad areas that characterized the development of reliability engineering:

(1) Increased interest in system reliability and safety of complex engineered systems (e.g., nuclear power plants [6]);
(2) A new focus on software reliability, due to the ever-increasing reliance on software in many safety-critical systems [7];
(3) Design of an incentive program, Reliability Improvement Warranty, to foster improvements in reliability.

During the last three and a half decades, from 1980 to 2015, significant technical advances and practical applications have been achieved by academia, government, industry and/or multilateral collaboration of these stakeholders, with an aim of addressing the challenges posed by the increasing complexity of modern engineered systems. An increasing number of publications on reliability engineering can be found in well-known journals, such as *IEEE Transactions on Reliability*, and *Reliability Engineering & Systems Safety*. These efforts and developments have enabled reliability engineering to become a well-established, multidisciplinary field that endeavors to address the following challenging questions [8]:

(1) Why does a system fail? This question is studied by analyzing failure causes and mechanisms and identifying failure consequences.
(2) How can reliable systems be designed? This is studied by conducting reliability analysis, testing, and design optimization.
(3) How can high operational reliability be achieved throughout a system's life-cycle? This question is studied by developing and implementing health monitoring, diagnostics, and prognostics systems.

This book attempts to address the second and third questions from the perspective of engineering design. Our focus will be placed on reliability-based design considering various sources of uncertainty, i.e., addressing the second question. The emerging discipline of prognostics and health management (PHM), which has received increasing attention from the reliability community, will be briefly discussed to address the third question.

1.3 Reliability Practices in a Product's Life-Cycle

Reliability engineering has been widely recognized as an important discipline that is practiced throughout the life-cycle of a product. The responsibilities of a reliability engineer in product design and development are specified by the famous

military standard MIL-STD-785, Reliability Program for Systems and Equipment Development and Production, as follows:

> Tasks shall focus on the prevention, detection, and correction of reliability design deficiencies, weak parts, workmanship defects. Reliability engineering shall be an integral part of the item design process, including design changes. The means by which reliability engineering contributes to the design, and the level of authority and constraints on the engineering discipline, shall be identified in the reliability program plan. An efficient reliability program shall stress early investment in reliability engineering tasks to avoid subsequent costs and schedule delays

The statements above suggest that reliability engineering be treated as an integral part of product development, rather than as a separate activity with little to no interaction with other development activities [9]. The flowchart of a product reliability program is presented in Fig. 1.2, where five reliability-related tasks are integrated into three major phases of the new product development process, specifically: (i) concept phase (Tasks I and II), (ii) design phase (Tasks III and IV), and (iii) production phase (Task V).

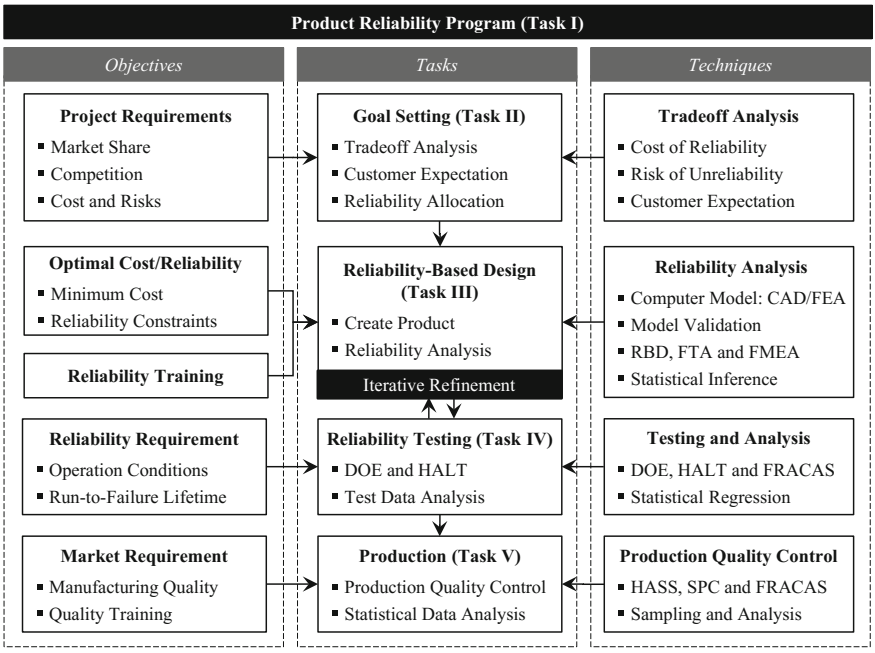


Fig. 1.2 Overview of product reliability plan with specific tasks. Acronyms are defined as follows: DOE, Design of Experiment; HALT, Highly Accelerated Life Testing; CAD, Computer-Aided Design; FEA, Finite Element Analysis; RBD, Reliability Block Diagram; FTA, Fault Tree Analysis; FMEA, Failure Modes and Effects Analysis; FRACAS, Failure Report and Corrective Action System; HASS, Highly Accelerated Stress Screening; and SPC, Statistical Process Control

Each of the five tasks is briefly discussed as follows:

Task I—*Creating a Product Reliability Plan*: To integrate the reliability effort into the new product development process, an organization should create a product reliability plan, which clearly specifies all activities of the reliability program, along with agreed-upon responsibilities for completing and reporting all tasks. The product reliability plan can be viewed as a comprehensive record of mandatory procedures that provide discipline and accountability for the reliability effort. A well-prepared reliability plan enhances the organization's confidence in its competence to accomplish the specified tasks, and provides a clear guide for the product development team to manage the reliability.

Task II—*Setting a Reliability Goal*: The concept phase identifies market-driven product features (e.g., cost targets, forecasted demand, and essential product functions), defines a set of design requirements on the basis of the product features, and develops a design concept to meet these requirements. During the concept phase, the reliability engineer in the product development team must establish a fully stated reliability goal. The reliability goal must contain three components, listed as follows [9]:

(1) A definition of failure based on the product's function, which covers all relevant failure modes.
(2) A description of the use conditions under which the product will be stored, transported, operated, and maintained.
(3) A statement of the reliability requirements, and/or a statement of the most critical failure modes.

Reliability goal setting should take into account customer expectations and market status, and achieve a good balance between the cost of reliability and the risk of unreliability. Then, the system-level reliability goal is allocated to subsystem-level and component-level reliability goals such that meaningful goals and responsibilities are defined for individual contributors. A clearly stated reliability goal allows team members to hold consistent views about the reliability, and to comprehend the importance of reliability while making design or production decisions [10]. A clearly defined reliability goal also ensures that appropriate time and resources are spent on reliability during the concept, design, and manufacturing phases of the product life-cycle. Furthermore, the reliability goal, stated in a way that is measureable, allows the leadership team to track progress toward the time when customers' expectations of reliability are met.

Task III—*Conducting Reliability-Based Design*: During the design phase, the reliability and/or design engineers apply various design methods and techniques to enhance the ultimate product reliability. The task of designing reliability into the product, called reliability-based design, often entails the development and validation of a reliability model for the product. This generally consists of three major steps: (i) developing a mathematical or computer simulation model of the physical system (i.e., the product); (ii) validating the developed model using experimental data [11]; and (iii) evaluating the reliability of the product using reliability analysis

techniques (e.g., a reliability block diagram [12] and statistical inference [13]). The reliability model can be extended by incorporating the effects of operating conditions and environmental factors on product reliability. The reliability model not only provides valuable information about the reliability of a design, but also paves the way for reliability-based design optimization (RBDO). In RBDO, the techniques of reliability analysis and design optimization are integrated to develop reliability-based design methodologies that offer probabilistic approaches to engineering design. Reliability-based design attempts to find the optimum design of an engineered product that minimizes the cost and satisfies a target level of reliability, while accounting for uncertainty in parameters and design variables [14]. Once prototype units are built based on the optimal design derived from RBDO, reliability testing can be performed to gain more insights into the reliability of the design.

Task IV—*Performing Reliability Testing*: The design of a reliable product entails a carefully designed and executed test plan. The plan should place an emphasis on accelerated life testing (ALT) and highly accelerated life testing (HALT), both of which expose a product to environmental stresses (e.g., temperature, pressure, and loading) above what the product experiences in normal use, in order to stimulate failure to occur more quickly than in actual use. ALT/HALT serve as a formal process to measure and/or improve product reliability in a timely and cost-effective manner. They guarantee a proportional reduction in the time to market as well as greatly improved reliability [9]. Efforts should be devoted to planning robust reliability testing that takes into account various operating conditions using the design of experiment (DOE) technique. Reliability testing should be integrated with reliability-based design (Task III) in two aspects. First, all failures observed during reliability testing should be investigated and documented using the failure report and corrective action system (FRACAS), and the actual root cause of failure should be fed into reliability-based design as design feedback for reliability improvement. Second, the testing results should be used to assess and improve the validity of the reliability model developed and used in Task III. Note that, in addition to reliability testing, field return and warranty data can be an invaluable source of information in identifying and addressing potential reliability problems to improve the design.

Task V—*Controlling Production Quality*: During the manufacturing phase, the reliability engineer must conduct quality control by incorporating statistical process control (SPC). If the manufacturing process is not adequately tracked using SPC techniques, the relatively large uncertainty inherent in the manufacturing process is likely to cause manufacturing-induced unreliability. To reduce the magnitude of manufacturing uncertainty, use of the SPC technique is suggested. SPC uses statistical monitoring to measure and control the uncertainty in the manufacturing process. Integrating SPC into the manufacturing process requires the following four steps: (i) selecting key personnel (e.g., the quality control manager) for SPC skill training; (ii) conducting sensitivity analysis to identify critical process parameters that significantly affect the product quality; (iii) deploying hardware and software for SPC; and (iv) investigating the cause thoroughly if a large uncertainty is

observed. The SPC provides the manufacturing team with the ability to identify specific causes of large variations and to make process improvements for higher product reliability. In addition, through proper rework on products that are detected to have defects, the manufacturing line can output a larger number of qualified products given the same amount of cost. Thus, greater revenue can be achieved.

In the last decade or so, reliability engineering has been extending its domain of application to sensory-based health monitoring, diagnostics, and prognostics. A tremendous amount of research efforts have been devoted to this extension, and this stream of research is centered on utilizing sensory signals acquired from an engineered system to monitor the health condition and predict the remaining useful life of the system over its operational lifetime. This is treated as Task VI, and is one of the major focuses of this book.

This health information provides an advance warning of potential failures and a window of opportunity for implementing measures to avert these failures.

1.4 Outline of Following Chapters

This book focuses on exploring the fundamentals, implementation, and applications of probabilistic analysis and design methods for improving the reliability of engineered systems. Specifically, it aims to address, from the perspective of engineering design, the following questions: (i) How can an engineered system be designed with near-zero failure probability during product development? and (ii) How can failures be anticipated and prevented during system operation? The first question can be dealt with by conducting reliability-based design, which encompasses reliability analysis, testing, and design optimization (Tasks III and IV). Addressing this question is the focus of Chaps. 4–7 of this book. Chapter 4 presents the fundamentals of reliability analysis, including the concepts and formulations of time-independent and time-dependent reliability analyses. Chapter 5 introduces the state-of-the-art techniques for time-independent reliability analysis, including the first- or second order reliability methods (FORM/SORM), direct or smart Monte Carlo simulation (MCS), and the emerging stochastic response surface methods. Chapter 6 is devoted to explaining how to design a reliable engineered system by performing RBDO. Chapter 7 discusses several newly developed techniques for time-dependent reliability analysis.

The second question can be answered using sensor-based health monitoring, diagnostic, and prognostic techniques (Task VI). The development of these techniques has led to the emergence of a new discipline, prognostics and health management (PHM). In the last decade or so, PHM has been successfully applied to many engineered systems to assess the health conditions of these systems under actual operating conditions and to predict the remaining useful life of a system over its lifetime. This health information provides an advance warning of potential failures and a window of opportunity for implementing measures to avert these failures. Chapter 8 discusses the current state-of-the-art techniques and approaches

in this emerging discipline that enable optimal design of sensor networks for fault detection, effective extraction of health-relevant information from sensory signals, and robust prediction of remaining useful life.

References

1. Mobley, R. K. (2002). *An introduction to predictive maintenance* (2nd ed., Chap. 1). New York, NY: Elsevier.
2. Coleridge, S. T. (1983). Biographia literaria. In J. Engell & W. J. Bate (Eds.), *The collected works of Samuel Taylor Coleridge*. New Jersey, USA: Princeton University Press.
3. Saleh, J. H., & Marais, K. (2006). Highlights from the early (and pre-) history of reliability engineering. *Reliability Engineering & System Safety, 91*(2), 249–256.
4. Coppola, A. (1984). Reliability engineering of electronic equipment: A historical perspective. *IEEE Transactions on Reliability, 33*(1), 29–35.
5. Denson, W. (1998). The history of reliability prediction. *IEEE Transactions on Reliability, 47* (3-SP), 321–328.
6. WASH-1400. (1975). *Reactor safety study*. USA: US Nuclear Regulatory Commission.
7. Moranda, P. B. (1975). Prediction of software reliability during debugging. In *Proceedings of the Annual Reliability Maintenance Symposium* (pp. 327–332).
8. Zio, E. (2009). Reliability engineering: Old problems and new challenges. *Reliability Engineering & System Safety, 94*(2), 125–141.
9. O'Connor, P. D. T., Newton, D., & Bromley, R. (2002). *Practical reliability engineering* (4th ed.). West Sussex, England: Wiley.
10. Ireson, W. G., Coombs, C. F., & Moss, R. Y. (1996). *Handbook of reliability engineering and management* (2nd ed.). New York, NY: McGraw-Hill Professional.
11. Xiong, Y., Chen, W., Tsui, K.-L., & Apley, D. (2008). A better understanding of model updating strategies in validating engineering models. *Computer Methods in Applied Mechanics and Engineering, 198*(15–16), 1327–1337.
12. Kuo, W., & Zuo, M. J. (2002). *Optimal reliability modeling: Principles and applications*. Hoboken, NJ: Wiley.
13. Singpurwalla, N. D. (2006). *Reliability and risk: A bayesian perspective*. New York, NY: Wiley.
14. Haldar, A., & Mahadevan, S. (2000). *Probability, reliability, and statistical methods in engineering design*. New York, NY: Wiley.

Chapter 2
Fundamentals of Probability Theory

Probability theory is a mathematical discipline that investigates possible outcomes of repeated experiments and a long-run relative frequency of these outcomes. The word "probability" generally refers to the chance of a specific event occurring, taking values between zero (impossible) and one (certain). Probability theory enables the analysis of reliability, i.e. the probability that a system performance meets its marginal value (or requirement) under uncertainty at the very beginning of operation (time-independent reliability) or during its lifetime (time-dependent reliability). In this chapter, we briefly summarize fundamental probability theory with the aim of providing a sufficient background in probability to enable understanding and use of techniques and methods found in later chapters.

2.1 Probability Concepts

The basic setting of a probability model is a random experiment that produces different possible outcomes. In this setting, a probability model is formulated in a complete probability space $(\Omega, \mathcal{A}, \mathcal{P})$, where Ω is a sample space, $\mathcal{A}$ is an event field on Ω, and $\mathcal{P}$ is a probability measure function $\mathcal{P}$: $\mathcal{A} \rightarrow [0, 1]$. Depending on the setting of a random experiment, a sample space can be countably finite, countably infinite, or uncountably infinite. If we treat rolling a die once as an experiment and the resulting die number as the outcome, the sample space $\Omega = \{1, 2, 3, 4, 5, 6\}$, which is countably finite. If we treat rolling a die multiple times until the occurrence of an even number as an experiment and the resulting number of times rolled as the outcome, the sample space $\Omega = \{1, 2, \ldots, +\infty\}$, which is countably infinite. If we treat running an engineered system until its failure as an experiment, and the resulting life as the outcome, the sample space $\Omega = (0, +\infty)$, which is uncountably infinite. Another example of an uncountably infinite sample space is the fatigue test, where the sample data can be obtained to derive the physical quantities used in the strain-life equation, as shown in Fig. 2.1.

C. Hu et al., *Engineering Design under Uncertainty and Health Prognostics*,
Springer Series in Reliability Engineering,
https://doi.org/10.1007/978-3-319-92574-5_2

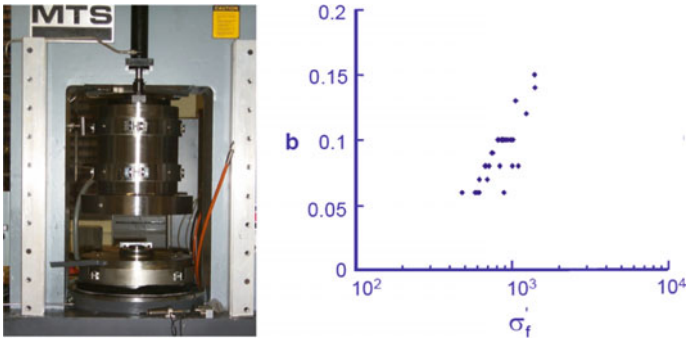


Fig. 2.1 Fatigue tests and sample data set

An event E, as a member of the event field $\mathcal{A}$ and a subset of Ω, can be defined as a collection of possible outcomes of a random experiment. Two special events exist: the whole sample space Ω is defined as the certain event; the empty set $\emptyset$ is called the null event. An event E is assigned with a probability $P(E)$ that satisfies the following property: $0 \leq P(E) \leq 1$. In probability theory, the probability $P(E)$ is the chance that the event E is true. In practice, $P(E)$ refers to the limiting value that the relative frequency of the occurrence of E will approach if the random experiment is repeated an infinite number of times under identical conditions. This statement is referred to as *the law of large numbers*. The convergence of the relative frequency can be demonstrated by repeating a random experiment of tossing a coin and computing the relative frequency of a result of "heads." The process is graphically shown in Fig. 2.2, where the relative frequency converges to 0.5 or the probability of "heads." It then becomes reasonable to treat the relative frequency at the time of any rolling, or in general, at any number of repeated experiments, as an estimate of $P(E)$.

With an either finite or countably infinite sample space Ω, every possible outcome s is assigned with a probability $p(s)$. If every possible outcome s is a real number, s then becomes a discrete random variable and we use a probability mass function to represent $p(s)$. In such cases, the probability of any event E, consisting of a collection of s, can be computed as

$$P(E) = \sum_{s \in E} p(s) \tag{2.1}$$

With the event E being a countable union of disjoint sets s, the relation above can be derived from the addition theorem for mutually exclusive events. This will be explained in detail in a subsequent section. On the other hand, if the sample space Ω is uncountably infinite and s represents a real value, s can then be treated as a continuous random variable. In this case, a probability density function (PDF) is used to represent $p(s)$.

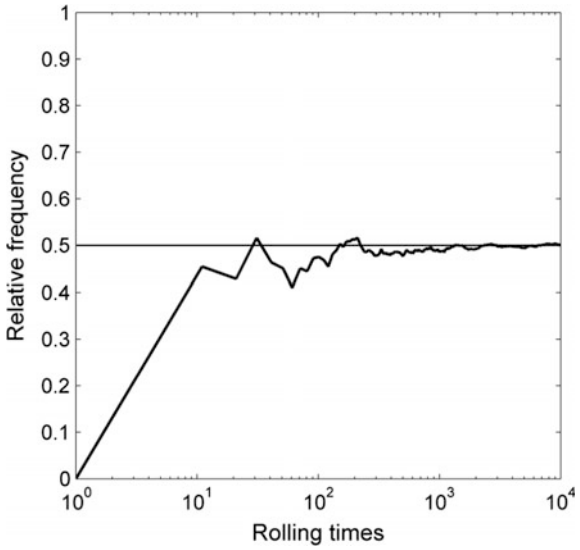


Fig. 2.2 Convergence of the relative frequency of heads when tossing a coin *ns* times

2.2 Axioms and Theorems of Probability Theory

This section presents the basic axioms and theorems of probability theory that form the theoretical basis for probability analysis.

2.2.1 Basic Axioms

As mentioned in Sect. 2.1, the probability measure function $\mathcal{P}$, defined on Ω, maps any event E in $\mathcal{A}$ to a real number. The following basic axioms apply to this real number:

Axiom 1: $P(E) \geq 0$ for any $E \subset \Omega$.

Axiom 2: $P(\Omega) = 1$.

Axiom 3: If $E_1, E_2, \ldots$ are mutually exclusive it follows that

$$P\left(\bigcup_{i=1}^{\infty} E_i\right) = \sum_{i=1}^{\infty} P(E_i) \tag{2.2}$$

Other useful rules that follow from Axioms 1–3 can be expressed as follows:

$$\begin{aligned} &P(\varnothing) = 0 \\ &P(E_1) \leq P(E_2) \text{ if } E_1 \subseteq E_2 \\ &P(\bar{E}_1) = 1 - P(E_1) \end{aligned} \tag{2.3}$$

The third rule of the equation above can be proven using Axioms 2 and 3. Applications of these axioms and rules lead to useful theorems for the calculation of the probability of events. These will be detailed in the next subsection.

2.2.2 Addition Theorems

Events E_1, E_2, …, E_n are mutually exclusive if the occurrence of any event excludes the occurrence of the other $n - 1$ events. In other words, any two events E_i and E_j in this set of events cannot occur at the same time, or are mutually exclusive. This can be mathematically expressed as $E_i \cap E_j = \emptyset$. For example, a battery cell could fail due to internal short circuit or over-discharge. The events E_1 = {failure due to internal short circuit} and E_2 = {failure due to over-discharge} can be treated as mutually exclusive due to the impossibility of the simultaneous occurrence of the two events. It follows from Axiom 3 (Sect. 2.2.1) that

$$P(E_1 \cup E_2) = P(E_1) + P(E_2) \tag{2.4}$$

In general, if events E_1, E_2, …, E_n are mutually exclusive, the following addition theorem applies:

$$P\left(\bigcup_{i=1}^{n} E_i\right) = \sum_{i=1}^{n} P(E_i) \tag{2.5}$$

With this knowledge of mutually exclusive events in place, let us now investigate a general case where the mutually exclusive assumption may not hold. The probability that event E_1, event E_2, or both events occur is computed based on the union of two events, expressed as

$$P(E_1 \cup E_2) = P(E_1) + P(E_2) - P(E_1 \cap E_2) \tag{2.6}$$

For the case of n events E_1, E_2, …, E_n, the probability that at least one of them occurs can be derived by generalizing Eq. (2.6) as

$$P\left(\bigcup_{i=1}^{n} E_i\right) = \sum_{j=1}^{n} (-1)^{j+1} P_j \text{ with } P_j = \sum_{1 \le i_1 < \cdots < i_j \le n} P\left(\bigcap_{k \in \{i_1, \ldots, i_j\}} E_k\right) \tag{2.7}$$

The equation above can be derived by first grouping $(n - 1)$ events as a single event, applying Eq. (2.6) to the resulting two events, and then repeating the same process $(n - 1)$ times.

Example 2.1 Suppose that among a set of 50 battery cells, 1 cell suffers from damage due to an internal short circuit and 2 cells suffer from damage due to overvoltage. If we randomly choose one cell from the set, what is the probability of getting a defective cell?

Solution
Since the events E_1 = {get a cell with an internal short circuit} and E_2 = {get a cell with an overvoltage} are mutually exclusive, we can compute the probability of their union using Eq. (2.4) as

$$\begin{aligned} P(E_1 \cup E_2) &= P(E_1) + P(E_2) \\ &= \frac{1}{50} + \frac{2}{50} = \frac{3}{50} \end{aligned}$$

2.2.3 Multiplication Theorem

Events E_1 and E_2 are independent if the occurrence of one event does not have any influence on that of the other. In such cases, the probability of a joint event can be computed as the multiplication of the probabilities of individual events, expressed as

$$P(E_1 \cap E_2) = P(E_1) \cdot P(E_2) \tag{2.8}$$

For a general case where events E_1 and E_2 may not be independent, the probability of the joint event can be expressed as

$$P(E_1 \cap E_2) = P(E_1) \cdot P(E_2|E_1) = P(E_2) \cdot P(E_1|E_2) \tag{2.9}$$

In the equations above, $P(E_1|E_2)$ and $P(E_2|E_1)$ are conditional probabilities that assume that E_2 and E_1 have occurred, respectively.

2.2.4 Total Probability Theorem

Assuming events E_1, E_2, …, E_n are mutually exclusive ($E_i \cap E_j = \emptyset$ for all $i \neq j$) and collectively exhaustive (the sum of probabilities of all events equals one), we then divide the probability of an arbitrary event E_A into the probabilities of n mutually exclusive events, expressed as

$$P(E_A) = \sum_{i=1}^{n} P(E_A \cap E_i) = \sum_{i=1}^{n} P(E_i)P(E_A|E_i) \tag{2.10}$$

We call this equation the total probability theorem. This theorem further leads to Bayes' theorem that is used to compute the posterior probability of an event as a function of the priori probability and likelihood. Bayes' theorem will be discussed in detail in Chap. 3.

2.3 Random Variables and Univariate Distribution Functions

A random variable is a function that maps events in the sample space Ω into outcomes in the real number space $\mathbb{R}$ where the outcomes can be real or integer, continuous or discrete, success or failure, etc. The random variable often written as $X: \Omega \rightarrow \mathbb{R}$, is useful in quantifying uncertainty mathematically. In what follows, we will introduce two types of random variables, namely discrete random variables and continuous random variables.

2.3.1 Discrete Random Variables

A discrete random variable X is a function that maps events in a sample space into to a finite or countably infinite set of real numbers. An example of a discrete random variable can be found in specimen tensile tests with 10 kN tensile force. If we repeat this test 100 times with each test employing 20 specimens, the number of failed specimens in each tensile test can be treated as a discrete random variable. This variable can only represent a finite set of discrete integer values between 0 and 20. In general, a discrete random variable can only represent a value at a finite or infinite set of discrete points. This means that the probability of such a variable can only be computed at these discrete points. The randomness in this variable is described using the so-called *probability mass function* (PMF), denoted as $p_X(x)$. Assuming that X can be any of a series of discrete values $x_1, x_2, \ldots, x_M$, we can then define the PMF of X as

$$p_X(x_k) = P(X = x_k), \text{ with } \sum_{k=1}^{M} p_X(x_k) = 1 \tag{2.11}$$

Note that the PMF is a discrete function that consists of a series of discrete values, as shown in Fig. 2.3a. The cumulative distribution function (CDF) of a discrete random variable can be computed by adding all PMFs together. The CDF takes the

form of a step function, as shown in Fig. 2.3b. Mathematically, the CDF can be expressed as

$$F_X(x) = P(X \leq x) = \sum_{x_k \leq x} p_X(x_k) \tag{2.12}$$

In the next chapter, we will introduce several commonly used discrete distributions.

Example 2.2 Let X be the number of heads that appear if a coin is tossed three times sequentially. Compute the PMF of X.

Solution

If H and T represent head and tail, respectively, the sample space for this experiment can be expressed as Ω = {HHH, HHT, HTH, THH, HTT, THT, TTH, TTT}. First, consider the event $\{X = 0\}$, which can be easily mapped to the outcome w = TTT. The probability of $\{X = 0\}$ or $p_X(0)$ can be computed as

$$p_X(0) = P(\{X = 0\}) = P(\{w = \text{TTT}\}) = \left(\frac{1}{2}\right)^3 = \frac{1}{8}$$

In the calculation above, we use the multiplication theorem for independent events, as introduced in Eq. (2.8). Similarly, we can compute the other PMF values

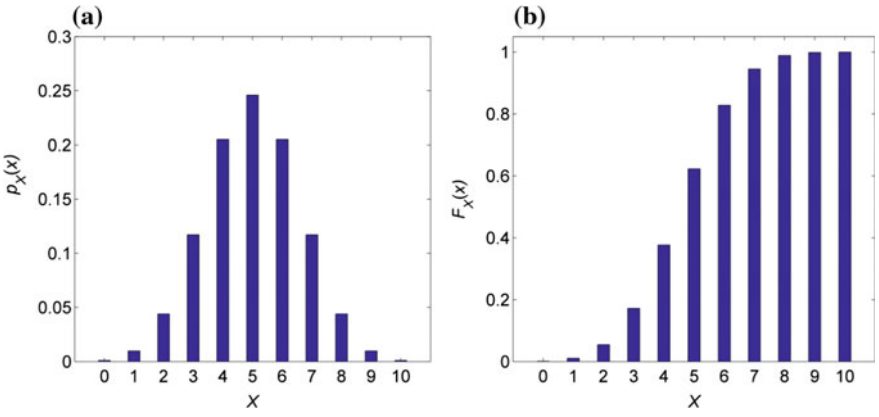


Fig. 2.3 An example of discrete random variables: **a** PMF; **b** CDF

$$p_X(1) = P(\{X = 1\}) = P(\{w \in \{\text{HTT, THT, TTH}\}\}) = 3\left(\frac{1}{2}\right)^3 = \frac{3}{8}$$

$$p_X(2) = P(\{X = 2\}) = P(\{w \in \{\text{THH, HTH, HHT}\}\}) = 3\left(\frac{1}{2}\right)^3 = \frac{3}{8}$$

$$p_X(3) = P(\{X = 3\}) = P(\{w = \text{HHH}\}) = \left(\frac{1}{2}\right)^3 = \frac{1}{8}$$

The computations of $p_X(1)$ and $p_X(2)$ use the addition theorem for mutually exclusive events and the multiplication theorem for independent events, as introduced earlier in the chapter.

2.3.2 Continuous Random Variables

Let us consider again the outcome of an experiment. This time the experiment is conducted to test an LED light bulb until it burns out. The random variable X represents the bulb's lifetime in hours. Since X can take any positive real value, it does not make sense to treat X as a discrete random variable. As a random variable that can represent continuous values, X should be treated as a continuous random variable, and its randomness can be modeled using a PDF $f_X(x)$. Note that the PDF at any point only provides information on density, not mass. Only integration (volume) of the PDF gives information on mass (probability). Thus, we become interested in computing the probability that the outcome of X falls into a specific interval $(x_1, x_2]$. This probability can be computed as

$$P(x_1 < X \leq x_2) = \int_{x_1}^{x_2} f_X(x)dx \tag{2.13}$$

Based on the equation above, we can calculate the CDF $F_X(x)$ by setting x_1 and x_2 to $-\infty$ and x, respectively. This can be expressed as

$$F_X(x) = P(X \leq x) = \int_{-\infty}^{x} f_X(x)dx \tag{2.14}$$

The PDF $f_X(x)$ has a relationship with the CDF $F_X(x)$ almost everywhere, expressed as

$$f_X(x) = \frac{dF_X(x)}{dx} \tag{2.15}$$

The relationship between the PDF $f_X(x)$ and CDF $F_X(x)$ of a continuous random variable is shown in Fig. 2.4, where it can be observed that Eq. (2.13) can be equivalently written as

$$P(x_1 < X \leq x_2) = F_X(x_2) - F_X(x_1) \tag{2.16}$$

To satisfy the three axioms in Sect. 2.2.1, a CDF $F_X(x)$ must possess the following properties:

(1) F is *non-decreasing*, i.e., $F_X(x_1) \geq F_X(x_2)$ whenever $x_1 > x_2$.
(2) F is *normalized*, i.e., $F_X(-\infty) = 0$, $F_X(+\infty) = 1$.
(3) F is *right-continuous*, i.e.,

$$\lim_{\varepsilon \downarrow 0} F_X(x+\varepsilon) = F_X(x) \tag{2.17}$$

Similarly, a PDF $f_X(x)$ must possess the following properties:

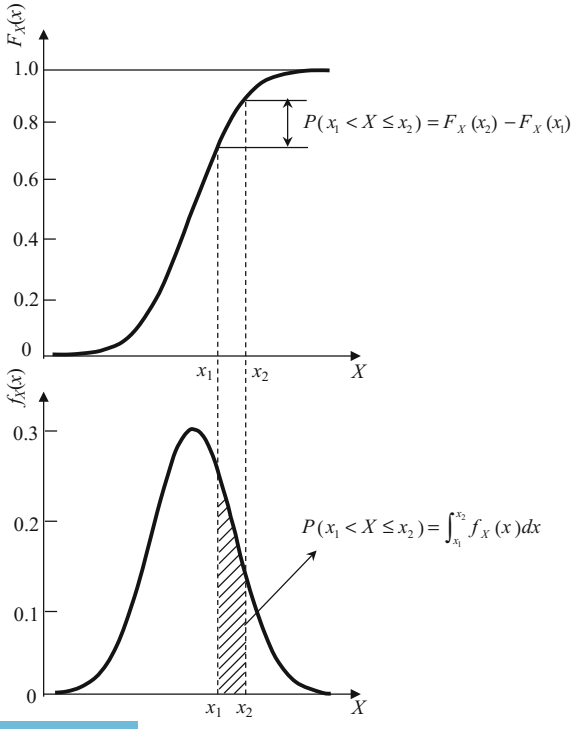


Fig. 2.4 Relationship between the PDF $f_X(x)$ and CDF $F_X(x)$ of a continuous random variable

(1) f is *non-negative*, i.e., $f_X(x) \geq 0$ everywhere.
(2) f is *normalized*, i.e.,

$$\int_{-\infty}^{+\infty} f_X(x)dx = 1 \tag{2.18}$$

Example 2.3 Assume that the remaining useful life (in days) of an engineered system follows the following distribution function

$$f_X(x) = \begin{cases} \frac{2x}{(1+x^2)^2}, & \text{for } x \geq 0 \\ 0, & \text{for } x < 0 \end{cases}$$

(1) Show that $f_X(x)$ satisfies the requirements of a PDF.
(2) Compute $P(3 \leq X \leq 5)$.
(3) If we define the reliability of this system as the probability of this system to operate for at least 5 days, calculate the reliability.

Solution

(1) It is clear that $f_X(x) \geq 0$ for all $x \in \mathbb{R}$. Then, an integral is computed

$$\int_{-\infty}^{+\infty} f_X(x)dx = \int_{0}^{+\infty} \frac{2x}{(1+x^2)^2}dx \overset{u=1+x^2}{=} \int_{1}^{+\infty} \frac{1}{u^2}du = 1$$

(2) Compute the probability $P(3 \leq X \leq 5)$ using Eq. (2.13) as

$$P(3 \leq X \leq 5) = \int_{3}^{5} \frac{2x}{(1+x^2)^2}dx \overset{u=1+x^2}{=} \int_{10}^{26} \frac{1}{u^2}du = -\frac{1}{u}\Big|_{10}^{26} = \frac{4}{65}$$

(3) The reliability can be written as $P(X \geq 5)$. This probability can be computed using Eq. (2.13) as

$$P(X \geq 5) = \int_{5}^{+\infty} \frac{2x}{(1+x^2)^2}dx \overset{u=1+x^2}{=} \int_{26}^{+\infty} \frac{1}{u^2}du = -\frac{1}{u}\Big|_{26}^{+\infty} = \frac{1}{26}$$

2.4 Statistical Moments of Random Variables

In Sect. 2.3, we learned that the randomness in a random variable can be exactly modeled using a distribution function. In engineering practice, we also need to characterize the statistical nature of a random variable with numerical parameters that can be estimated from available samples. Such numerical parameters include mean, standard deviation, skewness, kurtosis, etc.; these are known as statistical moments.

2.4.1 Mean

Let us first consider a discrete random variable X that can take a series of discrete values $x_1, x_2, \ldots, x_M$ with probabilities $p_X(x_1), p_X(x_2), \ldots, p_X(x_M)$. The mean μ_X, or expected value $E(X)$, can be defined as

$$E(X) = \sum_{k=1}^{M} x_k p_X(x_k) \tag{2.19}$$

Note that if we have a set of independent and identically distributed (i.i.d.) random samples, the arithmetic mean of these samples approaches the expected value in Eq. (2.19) as the number of samples approaches infinity. The proof of this statement can be derived based on the strong law of large numbers, which is omitted here. In reliability analysis, we are particularly interested in computing the expectation of a system's performance. A system's performance is often a function of several random variables. If we assume that the input X of a system performance function $g(X)$ is a discrete random variable, we can calculate the expected value of $g(X)$ as

$$E(g(X)) = \sum_{k=1}^{M} g(x_k) p_X(x_k) \tag{2.20}$$

Example 2.4 Consider the discrete random variable in Example 2.2. Compute its expected value.

Solution
Since X can represent any of the values 0, 1, 2 and 3, its expected value can be computed as

$$E(X) = \sum_{x=0}^{3} x \cdot p_X(x) = 0 \cdot \frac{1}{8} + 1 \cdot \frac{3}{8} + 2 \cdot \frac{3}{8} + 3 \cdot \frac{1}{8} = \frac{3}{2}$$

Example 2.5 Assume that the number X of failed power transformers per year follows the following PMF

$$p_X(x) = \frac{\lambda^x e^{-\lambda}}{x!}, \quad x = 0, 1, 2, \ldots$$

Calculate the mean number of failed power transformers per year.

Solution
The calculation of the mean value of X can be expressed as

$$\begin{aligned} E(X) &= \sum_{x=0}^{+\infty} x \cdot p_X(x) \\ &= \sum_{x=0}^{+\infty} x \cdot \frac{\lambda^x e^{-\lambda}}{x!} \\ &= \sum_{x=1}^{+\infty} \frac{\lambda^x e^{-\lambda}}{(x-1)!} \overset{l=x-1}{=} \lambda e^{-\lambda} \sum_{l=0}^{+\infty} \frac{\lambda^l}{l!} \\ &= \lambda e^{-\lambda} e^{\lambda} = \lambda \end{aligned}$$

Note: As we will see in Sect. 2.5, the PMF in this example is called a Poisson distribution, which is useful in modeling randomness in the number of events occurring in a fixed period of time under the assumptions of occurrence independence and a constant occurrence rate.

Next, let us consider a continuous random variable X with a PDF $f_X(x)$. In contrast to the discrete case where the expected value is a simple weighted-sum average, the mean μ_X, or expected value $E(X)$, in the continuous case is computed by performing integration as

$$E(X) = \int_{-\infty}^{+\infty} x f_X(x) dx \tag{2.21}$$

If we treat the integral above as a summation of the multiplications of x and its PDF value $f_X(x)$ over infinitesimally narrow intervals, the expected value of a continuous random variable can also be regarded as a weighted-sum average, which bears a resemblance to a discrete random variable. If we assume that the input X of a system performance function $g(X)$ is a continuous random variable, we can calculate the expected value of $g(X)$ as

$$E(g(X)) = \int_{-\infty}^{+\infty} g(x) f_X(x) dx \tag{2.22}$$

Example 2.6 Assume that a continuous random variable X follows the following distribution function

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]$$

Show that $E(X) = \mu$.

Solution
It is equivalent to show that $E(X - \mu) = 0$. The expectation can be computed as

$$\begin{aligned} E(X) &= \int_{-\infty}^{+\infty} (x-\mu) f_X(x) dx \\ &= \int_{-\infty}^{+\infty} (x-\mu) \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right] dx \\ &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} \frac{x-\mu}{\sigma} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right] dx \end{aligned}$$

If we replace the term $(x - \mu)/\sigma$ with a new variable z, we get

$$\begin{aligned} E(X) &= \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{+\infty} z \exp\left(-\frac{1}{2}z^2\right) dz \\ &= -\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2}z^2\right)\Big|_{-\infty}^{+\infty} = 0 \end{aligned}$$

Note: As we will see in Sect. 2.5, the PDF in this example is called a Gaussian or normal distribution—the most widely used continuous distribution.

In engineering practice, we often don't know the exact probability distribution of a random variable but instead we often only have a group of random samples x_1, x_2,

…, x_M. In such cases, the sample mean can be used as an approximation to the population mean. The sample mean is expressed as

$$\hat{\mu}_X = \frac{1}{M}\sum_{k=1}^{M} x_k \tag{2.23}$$

As the number of samples M approaches infinity, the sample mean approaches the population mean.

2.4.2 Variance

Similar to the mean, the variance is also an expectation. Letting X in Eq. (2.19) be $(X - \mu_X)^2$, we can then obtain the expression of the variance of a discrete random variable X as

$$\mathrm{Var}(X) = E\left[(X - \mu_X)^2\right] = \sum_{k=1}^{M} (x_k - \mu_X)^2 p_X(x_k) \tag{2.24}$$

The equation above suggests that the variance is the weighted-sum average of the squared deviation of X from its mean value μ_X and that the variance measures the dispersion of samples in a probabilistic manner.

Replacing X in Eq. (2.21) with $(X - \mu_X)^2$ gives the expression of the variance of a continuous random variable X as

$$\mathrm{Var}(X) = E\left[(X - \mu_X)^2\right] = \int_{-\infty}^{+\infty} (x - \mu_X)^2 f_X(x)dx \tag{2.25}$$

The standard deviation of X is defined as the square root of its variance, denoted by σ.

For a random variable with a non-zero mean, the computation is easier using a simplified formula for variance, specifically

$$\mathrm{Var}(X) = E\left(X^2\right) - (E(X))^2 \tag{2.26}$$

which essentially decomposes the calculation of the variance into the calculations of the expectation of square and of the square of expectation. When there is a larger number of random samples, x_1, x_2, …, x_M, the sample variance can be computed as

$$E(X) = \frac{1}{M-1}\sum_{k=1}^{M} (x_k - \hat{\mu}_X)^2 \tag{2.27}$$

Observe from Eqs. (2.24) and (2.25) that, for both discrete and continuous random variables, the variance is defined as the expectation of the squared deviation of the corresponding random variable from its mean value. This definition of variance can be generalized to the definition of the nth central moment of X, expressed as $E[(X - \mu_X)^n]$. The variance is equivalent to the 2nd central moment. In Sect. 2.4.3, we will introduce two higher-order moments, namely skewness and kurtosis, which are defined based on the 3rd and 4th central moments, respectively.

Example 2.7 Compute the variance of the continuous random variable X in Example 2.6.

Solution
According to Eq. (2.25), we can compute the variance of X as

$$\begin{aligned}\mathrm{Var}(X) &= E\left[(X-\mu)^2\right] = \int_{-\infty}^{+\infty} (x-\mu)^2 f_X(x)dx \\ &= \int_{-\infty}^{+\infty} (x-\mu)^2 \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right] dx \\ &= \frac{\sigma}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} \left(\frac{x-\mu}{\sigma}\right)^2 \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right] dx\end{aligned}$$

If we make the same replacement as in Example 2.6, we get

$$\begin{aligned}\mathrm{Var}(X) &= \frac{\sigma^2}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} z^2 \exp\left(-\frac{1}{2}z^2\right) dz \\ &= -\frac{\sigma^2}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} z \cdot d\left[\exp\left(-\frac{1}{2}z^2\right)\right] \\ &= \underbrace{-\frac{\sigma^2}{2\sqrt{2\pi}} z \exp\left(-\frac{1}{2}z^2\right)\Big|_{-\infty}^{+\infty}}_{0} + \sigma^2 \underbrace{\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} \exp\left(-\frac{1}{2}z^2\right) dz}_{1} \\ &= \sigma^2\end{aligned}$$

The fact that first term on the right hand side is zero is due to the faster decay of $\exp(-0.5z^2)$ than z. Detailed computation of the second term will be discussed in the next subsection.

2.4.3 Skewness and Kurtosis

The skewness of a random variable X, denoted by β_1, is defined as the normalized 3rd central moment, expressed as

$$\beta_1 = E\left[\left(\frac{X - \mu_X}{\sigma_X}\right)^3\right] \tag{2.28}$$

Skewness measures the asymmetry of the probability distribution of a random variable. In general, a probability distribution can be positively or negatively skewed (asymmetric), or evenly distributed (symmetric). A graphical comparison between positively and negatively skewed PDFs is given in Fig. 2.5. Observe that a positive skew (see Fig. 2.5a) means a longer tail on the right side of PDF and a larger area on the left side. Likewise, a negative skew (see Fig. 2.5b) means a longer tail on the left side of PDF and a larger area on the right side.

The kurtosis of a random variable X, denoted by β_2, is defined as the normalized 4th central moment, expressed as

$$\beta_2 = E\left[\left(\frac{X - \mu_X}{\sigma_X}\right)^4\right] \tag{2.29}$$

Kurtosis measures the portion of the variance that results from infrequent extreme deviations. Thus, for a random variable with high kurtosis, a large portion of its variance is due to infrequent extreme deviations.

The first four statistical moments are named mean, variance, skewness, and kurtosis, respectively. As will be discussed in later chapters, these moments provide information regarding the uncertainty of a system's performance for reliability analysis and, in some cases, can even help estimate the PDF of the system performance. Furthermore, the two lower-order moments (i.e., mean and variance) are important quantities used in design problems, such as reliability-based robust design optimization.

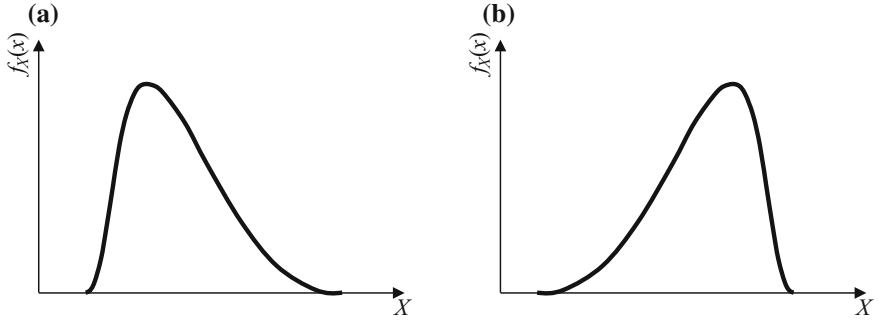


Fig. 2.5 Positively skewed PDF (**a**) versus negatively skewed PDF (**b**)

2.5 Commonly Used Univariate Distribution Functions

In this chapter's prior sections, we have introduced discrete and continuous random variables and presented fundamentals of the PMF/PDF and CDF for modeling randomness in these random variables. In what follows, commonly used univariate distribution functions will be introduced with an aim to enhance understanding of probability distributions as well as to provide preliminary information to lay the foundation for reliability analysis to be introduced in later chapters.

2.5.1 Discrete Probability Distributions

This section introduces three commonly used discrete distributions, namely binomial distribution, Poisson distribution, and geometric distribution.

Binomial Distribution

Recall Example 2.2 where a coin is tossed three times sequentially. The experiment in this example possesses two properties: (i) only two outcomes (head and tail) are possible in an experimental trial (tossing a coin); and (ii) repeated trials are conducted independently with each trial having a constant probability for each outcome (0.5 for head and 0.5 for tail). If a sequence of n experimental trials satisfies the two properties and the probability of occurrence of an outcome in each trial is p_0, the number X of occurrences of this outcome follows a binomial distribution, and its PMF $p_X(x)$ can be expressed as

$$p_X(x; n, p_0) = C(n, x) p_0^x (1 - p_0)^{n-x}, \quad x = 0, 1, 2, \ldots, n \tag{2.30}$$

where $C(n, x) = n!/[x!(n - x)!]$ is the binomial coefficient and can be interpreted as the number of different ways that the outcome occurs x times out of n trials. The equation above can be derived by summing the probabilities of $C(n, x)$ mutually exclusive events, of which each has a probability of occurrence $p_0^x(1 - p_0)^{n-x}$.

Example 2.8 Reconsider Example 2.2, except here the coin is tossed five times. Compute the probability of getting at least two heads in five tosses.

Solution
Let X denote the number of heads in five tosses, which follows a binomial distribution $p_X(x; 5, 0.5)$. The probability of $\{X \geq 2\}$ can be computed as

$$\begin{aligned} P(\{X \geq 2\}) &= 1 - P(\{X \leq 1\}) \\ &= 1 - P(\{X = 1\}) - P(\{X = 0\}) \\ &= 1 - C(5,1) \cdot 0.5^1 \cdot (1 - 0.5)^4 - C(5,0) \cdot 0.5^0 \cdot (1 - 0.5)^5 \\ &= 1 - 0.1563 - 0.0313 \\ &= 0.8125 \end{aligned}$$

This calculation uses the rule for the probability of complementary events, as outlined in Eq. (2.3).

Poisson Distribution

In Example 2.5, we presented the PMF of a discrete random variable X following the Poisson distribution. It takes the following form

$$p_X(x) = \frac{\lambda^x e^{-\lambda}}{x!}, \quad x = 0, 1, 2, \ldots \tag{2.31}$$

The Poisson distribution can be used to model the randomness in the number of events occurring in a fixed period of time under the assumptions of occurrence independence and a constant occurrence rate. We have seen from Example 2.5 that the mean of X is λ. It will be shown in Example 2.9 that the variance of X is also λ.

The Poisson distribution can be treated as a limiting case of the binomial distribution. In fact, as the number of trials $n \to +\infty$, the probability of occurrence $p_0 \to 0$, and $np_0 = \lambda$, the binomial distribution can be well represented by the Poisson distribution. This can be proved by setting $p_0 = \lambda/n$ in Eq. (2.30) and applying the limit $n \to +\infty$, expressed as

$$\begin{aligned} p_X(x; n, p_0) &= \lim_{n \to +\infty} \left[\frac{n!}{x!(n-x)!} \left(\frac{\lambda}{n}\right)^x \left(1 - \frac{\lambda}{n}\right)^{n-x} \right] \\ &= \lim_{n \to +\infty} \left[\frac{n(n-1)\cdots(n-x+1)}{n^x} \frac{\lambda^x}{x!} \left(1 - \frac{\lambda}{n}\right)^n \left(1 - \frac{\lambda}{n}\right)^{-x} \right] \\ &= \lim_{n \to +\infty} \left[1 \left(1 - \frac{1}{n}\right) \cdots \left(1 - \frac{x-1}{n}\right) \frac{\lambda^x}{x!} \left(1 - \frac{\lambda}{n}\right)^n \right] \\ &= \frac{\lambda^x}{x!} \lim_{n \to +\infty} \left[\left(1 - \frac{\lambda}{n}\right)^n \right] = \frac{\lambda^x e^{-\lambda}}{x!} \end{aligned} \tag{2.32}$$

Example 2.9 Reconsider Example 2.5. Find the variance of the number of failed power transformers per year.

Solution
According to Eq. (2.26), the variance of X can be computed as

$$\begin{aligned}\mathrm{Var}(X) &= E\left(X^2\right) - (E(X))^2 = \sum_{x=0}^{+\infty} x^2 \cdot \frac{\lambda^x e^{-\lambda}}{x!} - \lambda^2 \\ &= \sum_{x=1}^{+\infty} (x-1+1) \cdot \frac{\lambda^x e^{-\lambda}}{(x-1)!} - \lambda^2 \\ &= \sum_{x=2}^{+\infty} \frac{\lambda^x e^{-\lambda}}{(x-2)!} + \sum_{x=1}^{+\infty} \frac{\lambda^x e^{-\lambda}}{(x-1)!} - \lambda^2\end{aligned}$$

Applying $l = x - 2$ and $l = x - 1$ to the first and second terms, respectively, we get

$$\begin{aligned}\mathrm{Var}(X) &= \lambda^2 e^{-\lambda} \sum_{l=0}^{+\infty} \frac{\lambda^l}{l!} + \lambda e^{-\lambda} \sum_{l=0}^{+\infty} \frac{\lambda^l}{l!} - \lambda^2 \\ &= \lambda^2 + \lambda - \lambda^2 = \lambda\end{aligned}$$

Geometric Distribution

In a sequence of experimental trials, we are often interested in calculating how many trials have to be conducted until a certain outcome can be observed. For a Bernoulli sequence of experimental trials, the number X of trials conducted before the first occurrence of a certain outcome follows a geometric distribution as

$$p_X(x) = (1-p)^{x-1} p, \quad x = 1, 2, \ldots \tag{2.33}$$

where p is the probability of occurrence in each trial.

Example 2.10 The acceptance scheme for purchasing lots containing a large number of batteries is to test no more than 75 randomly selected batteries and to reject a lot if a single battery fails. Assume the probability of a single battery failure is 0.001.

(1) Compute the probability that a lot is accepted.
(2) Compute the probability that a lot is rejected on the 20th test.

(3) Compute the probability that a lot is rejected in less than 75 trials.

Solution

(1) The probability of acceptance is equal to the probability of no failure in 75 trials

$$P(\{\text{No Failure in 75 Trials}\}) = 0.999^{75} \approx 0.9925$$

(2) The probability that a lot is rejected on the 20th test can computed by using a geometric distribution with $X = 19$ and $p = 0.001$ as

$$P(\{\text{20th Failure}\}) = 0.999^{19} \cdot 0.001 \approx 0.0001$$

(3) The probability of rejection in less than 75 trials can be calculated through its complementary events (i.e., failure in the 75th trial and no failure in 75 trials) as

$$\begin{aligned} P(\{\text{Failure in Less Than 75 Trials}\}) &= 1 - P(\{\text{75th Failure}\}) - P(\{\text{No Failure}\}) \\ &= 1 - 0.999^{74} \cdot 0.001 - 0.999^{75} \approx 0.0074 \end{aligned}$$

2.5.2 *Continuous Probability Distributions*

In reliability analysis, continuous probability distributions are more often used than discrete probability distributions. This section presents three commonly used continuous distributions, namely normal distribution, Weibull distribution, and exponential distribution.

Normal Distribution

The most widely used continuous probability distribution is the normal or Gaussian distribution. The density function of a normal distribution can be expressed as

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right] \tag{2.34}$$

As demonstrated in Example 2.5, $\mu \in \mathbb{R}$ is the mean and $\sigma > 0$ is the standard deviation. Normal PDFs with different means and standard deviations are compared in Fig. 2.6. As σ increases, the width of the PDF increases and, to keep the total

area as one, the height decreases. The change of the mean only leads to the change of the center location of the PDF while the shape remains the same.

If $\mu = 0$ and $\sigma^2 = 1$, X follows a standard normal distribution, its PDF $f_X(x)$ becomes a standard normal density, and its CDF $F_X(x)$ becomes a standard normal CDF, denoted as $\Phi(x)$. If $\mu \neq 0$ or $\sigma^2 \neq 1$, we can use the substitution $z = (x - \mu)/\sigma$ to transform the original normal distribution into a standard normal distribution as

$$f_Z(z) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}z^2\right) \tag{2.35}$$

Next, let us verify that the normal PDF $f_X(x)$ satisfies the requirements of a PDF. Clearly, $f_X(x) > 0$ for all $x \in \mathbb{R}$. We then need to show that the integration of $f_X(x)$ is one. First, we use the substitution $z = (x - \mu)/\sigma$ to simplify the integration as

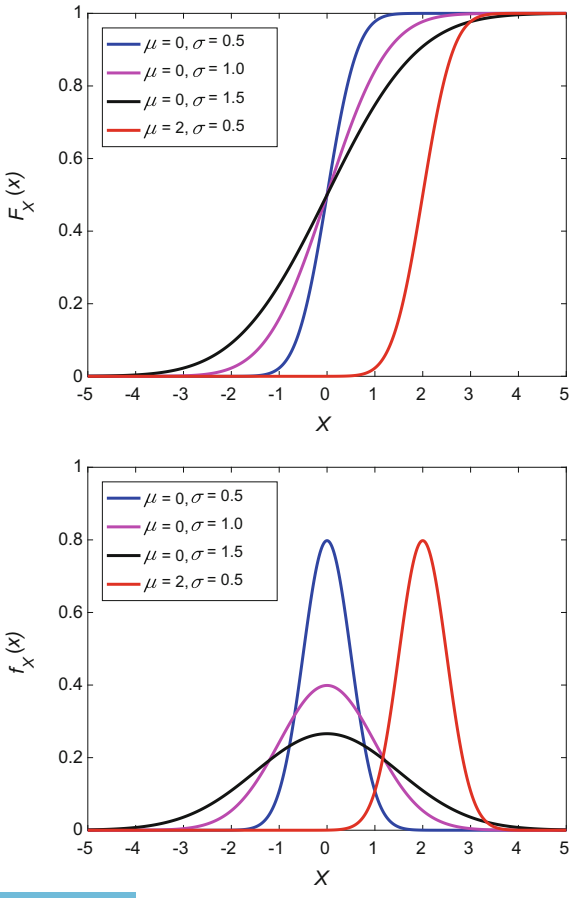


Fig. 2.6 Normal PDFs and the corresponding CDFs

$$\begin{aligned}\int_{-\infty}^{+\infty} f_X(x)dx &= \int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]dx \\ &= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{+\infty}\exp\left(-\frac{1}{2}z^2\right)dz\end{aligned} \tag{2.36}$$

Then the problem becomes the calculation of the following integral

$$I = \int_{-\infty}^{+\infty}\exp\left(-\frac{1}{2}z^2\right)dz \tag{2.37}$$

This integral can be converted to a double integral I^2 as

$$\begin{aligned}I^2 &= \left(\int_{-\infty}^{+\infty}\exp\left(-\frac{1}{2}x^2\right)dx\right)\left(\int_{-\infty}^{+\infty}\exp\left(-\frac{1}{2}z^2\right)dz\right) \\ &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\exp\left(-\frac{1}{2}\left(x^2+z^2\right)\right)dxdz\end{aligned} \tag{2.38}$$

Next, we change Cartesian x-z coordinates to polar r-θ coordinates with the substitutions $x = r\cos\theta$, $z = r\sin\theta$ and $dxdy = rdrd\theta$. In Cartesian coordinates, the integral is over the entire x-z plane. Equivalently, in polar coordinates, the integration is over the range of the radius r from 0 to $+\infty$ and the range of the angle θ from 0 to 2π. We then have

$$\begin{aligned}I^2 &= \left(\int_{-\infty}^{+\infty}\exp\left(-\frac{1}{2}x^2\right)dx\right)\left(\int_{-\infty}^{+\infty}\exp\left(-\frac{1}{2}z^2\right)dz\right) \\ &= \int_{0}^{+\infty}\int_{0}^{2\pi}\exp\left(-\frac{1}{2}r^2\right)rd\theta dr \\ &= 2\pi\int_{0}^{+\infty}\exp\left(-\frac{1}{2}r^2\right)rdr \\ &= -2\pi\exp\left(-\frac{1}{2}r^2\right)\Big|_0^{+\infty} = 2\pi\end{aligned} \tag{2.39}$$

This gives $I = \sqrt{2\pi}$ and the integral in Eq. (2.36) then becomes one. By calculating I, we solve the problem left in Example 2.5.

Lognormal Distribution

If the logarithm of a continuous random variable X follows a normal distribution, the random variable follows a lognormal distribution

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma x}\exp\left[-\frac{1}{2}\left(\frac{\ln x - \mu}{\sigma}\right)^2\right] \tag{2.40}$$

Here, $x > 0$, $\mu > 0$ and $\sigma > 0$. Note that μ and σ are two lognormal parameters, not the mean and standard deviation of X. The mean μ_X, and standard deviation σ_X of X, can be computed as

$$\begin{aligned} \mu_X &= \exp\left(\mu + \sigma^2/2\right) \\ \sigma_X^2 &= \left(e^{\sigma^2} - 1\right)\exp\left(2\mu + \sigma^2\right) \end{aligned} \tag{2.41}$$

Similar to the normal case, we can also transform a lognormal random variable to a standard normal variable by using the substitution $z = (\ln x - \mu)/\sigma$. We can then derive the CDF of a lognormal random variable as

$$\begin{aligned} F_X(x) &= P(X \le x) = P\left(\frac{\ln X - \mu}{\sigma} \le \frac{\ln x - \mu}{\sigma}\right) \\ &= \int_{-\infty}^{\frac{\ln x - \mu}{\sigma}} \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}z^2\right)dz \\ &= \Phi\left(\frac{\ln x - \mu}{\sigma}\right) \end{aligned} \tag{2.42}$$

Lognormal PDFs with different means and standard deviations are compared in Fig. 2.7. As σ_X increases, the width of the PDF increases and the height decreases. Unlike the normal distribution, the lognormal distribution can represent values in $(0, +\infty)$ and is unsymmetrical.

Example 2.11 A chemical plant superintendent has standing orders to shut down the process and make a readjustment whenever the pH of the final product falls below 6.90 or above 7.10. Assume that sample pH is lognormally distributed with a varying mean μ_X and a fixed standard deviation $\sigma_X = 0.05$.

(1) Determine the probability of readjusting when the process is operating as intended and $\mu_X = 7.0$.
(2) Determine the probability of failing to readjust (failing to notice the out-of-range value) when the process is too acidic and the mean pH is $\mu_X = 6.8$.

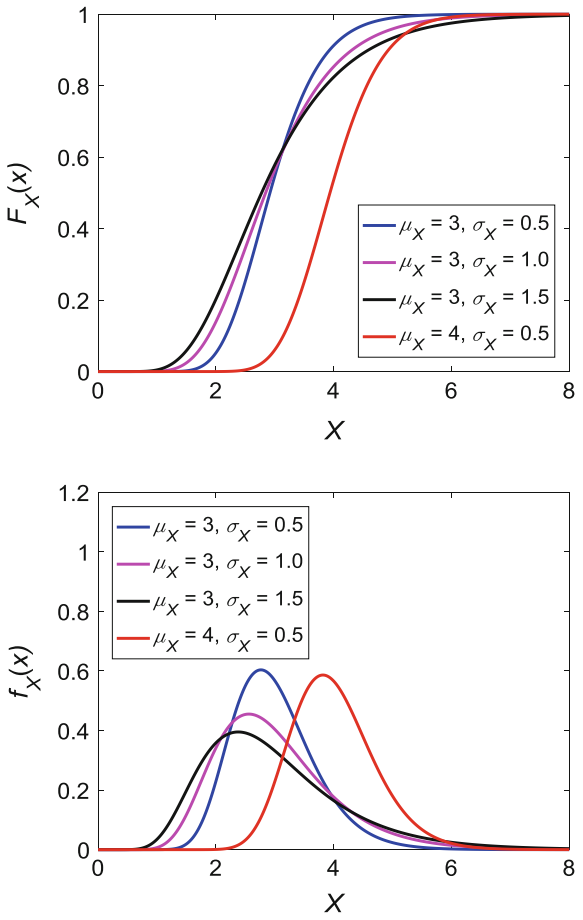


Fig. 2.7 Lognormal PDFs and the corresponding CDFs

Solution

(1) We first derive the lognormal parameters by inversely using Eq. (2.41) as

$$\mu = \ln\left(\mu_X^2\right)/\left(\sigma_X + \mu_X^2\right)^{1/2}$$
$$\sigma = \left(\ln\left(\sigma_X/\mu_X^2 + 1\right)\right)^{1/2}$$

The parameters are $\mu = 1.9454$ and $\sigma = 0.0319$. We then convert the critical pH values to the corresponding standard normal values as

$$z_1 = \frac{\ln 6.9 - 1.9454}{0.0319} = -0.4351,\ z_2 = \frac{\ln 7.1 - 1.9454}{0.0319} = 0.0160$$

Then the probability can be computed by using the normal CDF values as

$$\begin{aligned} P(x \leq 6.9, \text{ or } x \geq 7.1) &= P(z \leq -0.4351, \text{ or } z \geq 0.0160) \\ &= \Phi(-0.4351) + \Phi(-0.0160) \approx 0.8254 \end{aligned}$$

(2) We apply the same procedure to solve this question. First, we compute the lognormal parameters as $\mu = 1.9164$ and $\sigma = 0.0329$. We then compute the critical standard normal values as

$$z_1 = \frac{\ln 6.9 - 1.9164}{0.0329} = 0.4596, \; z_2 = \frac{\ln 7.1 - 1.9454}{0.0319} = 1.3281$$

Since failing to readjust means that pH is not in the readjustment range, the probability of failing to readjust can be expressed as

$$\begin{aligned} P(6.9 \leq x \leq 7.1) &= P(0.4596 \leq z \leq 1.3281) \\ &= \Phi(1.3281) - \Phi(0.4596) \approx 0.2308 \end{aligned}$$

Weibull Distribution

The Weibull distribution is a continuous probability distribution that is widely used to model the time-to-failure distribution of engineered systems. The PDF of a Weibull random variable X is defined as

$$f_X(x) = \lambda k(\lambda x)^{k-1} e^{-(\lambda x)^k} \tag{2.43}$$

Here, $x > 0$, $\lambda > 0$ is the scale parameter, and $k > 0$ is the shape parameter. Integrating the Weibull PDF results in a Weibull CDF of the following form

$$F_X(x) = 1 - e^{-(\lambda x)^k} \tag{2.44}$$

If the Weibull distribution is used to model the time-to-failure distribution, the failure rate can be computed as

$$h_X(x) = \frac{f_X(x)}{1 - F_X(x)} = \lambda k(\lambda x)^{k-1} \tag{2.45}$$

for the time $x > 0$. The failure rate is a constant over time if $k = 1$. In this case, the Weibull distribution becomes an exponential distribution. A value of $k > 1$

indicates a decreasing failure rate and a value of $k < 1$ indicates an increasing failure rate. Weibull PDFs with different scale and shape parameters are compared in Fig. 2.8. As λ increases, the PDF shrinks towards the y axis and the height increases. As can be seen in the plots, the Weibull distribution is able to model both "peaked" and "peakless" PDFs.

Exponential Distribution

As mentioned earlier, the Weibull distribution with $k = 1$ is an exponential distribution with the parameter λ. The PDF of an exponential distribution can be readily obtained by setting $k = 1$ in Eq. (2.43). It takes the following form

$$f_X(x) = \lambda e^{-\lambda x} \tag{2.46}$$

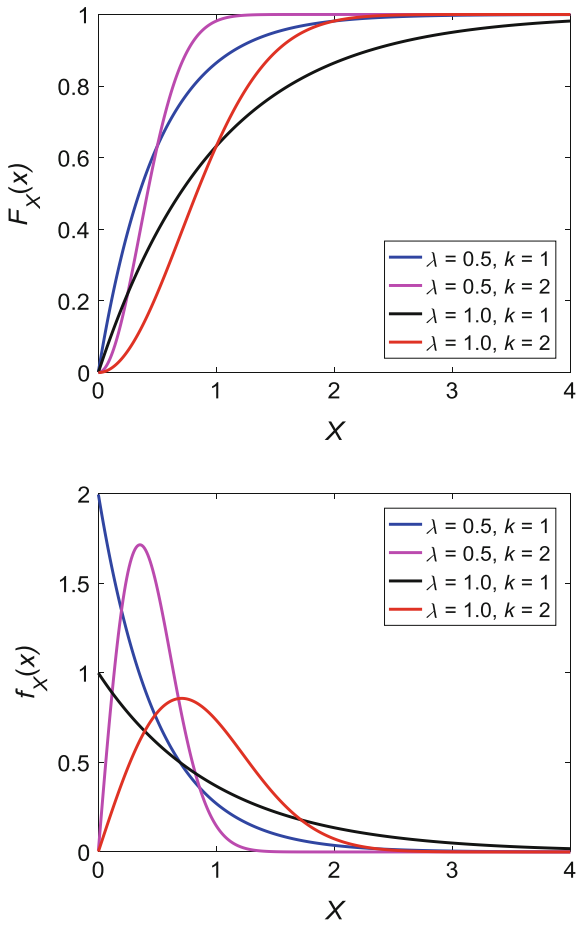


Fig. 2.8 Weibull PDFs and the corresponding CDFs

The CDF can be easily obtained as

$$F_X(x) = 1 - e^{-\lambda x} \tag{2.47}$$

The exponential distribution can be used to model the lifetime distribution of an engineered system with a constant failure rate. Exponential PDFs with different parameters are graphically compared in Fig. 2.9. As λ increases, the height increases and the width decreases.

Example 2.12 The lifespan of a certain type of device has an advertised failure rate of 0.01 per hour. The failure rate is constant and the exponential distribution applies.

(1) Compute the mean time-to-failure (MTTF).
(2) Compute the probability that at least 200 h pass before a failure is observed.

Solution

(1) The time-to-failure distribution is

$$f_T(t) = \lambda e^{-\lambda t} = 0.01 e^{-0.01t}$$

The MTTF can be computed by taking the expectation of T, expressed as

$$\begin{aligned} MTTF &= \int_0^{+\infty} t \cdot f_T(t) dt = \int_0^{+\infty} t \cdot \lambda e^{-\lambda t} dt = -\int_0^{+\infty} t de^{-\lambda t} \\ &= -te^{-\lambda t}|_0^{+\infty} + \int_0^{+\infty} e^{-\lambda t} dt \\ &= 0 + \frac{1}{\lambda} = \frac{1}{\lambda} = 100\,\text{h} \end{aligned}$$

(2) The probability of time-to-failure being at least 200 h can be computed as

$$\begin{aligned} P(t \geq 200) &= \int_{200}^{+\infty} f_T(t) dt = \int_{200}^{+\infty} \lambda e^{-\lambda t} dt \\ &= -e^{-\lambda t}|_{200}^{+\infty} = e^{-2} \approx 0.1353 \end{aligned}$$

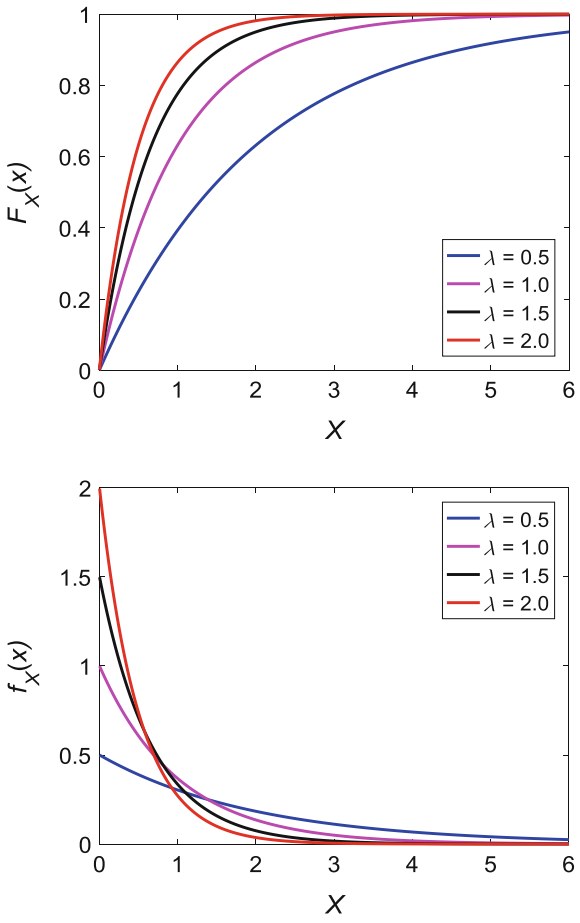


Fig. 2.9 Exponential PDFs and the corresponding CDFs

2.6 Random Vectors and Joint Distribution Functions

In the previous sections, we have focused our discussion on single random variables. In engineering practice, quite often two or more random variables are present. For example, in a two-dimensional rectangular plate, both the length and width can be modeled as random variables. Thus, it is important to study multiple random variables, which are often grouped as a single object, namely a random vector. The aim of this section is to introduce the probability model of a random vector, conditional probability, and independence. We will separately discuss both continuous and discrete random vectors that consist of two random variables.

2.6.1 Joint and Marginal Probability Distributions

Let us first consider the *discrete* case. The joint PMF of two discrete random variables X and Y is defined by

$$p_{XY}(x, y) = P(X = x, Y = y) \tag{2.48}$$

In this case, the joint CDF, denoted as $F_{XY}(x, y)$, can be defined in a similar fashion as the univariate CDF. It takes the following form

$$F_{XY}(x, y) = \sum_{y_i \le y} \sum_{x_i \le x} p_{XY}(x_i, y_j) \tag{2.49}$$

We can derive the marginal PMFs $p_X(x)$ and $p_Y(y)$ from the joint PMF $p_{XY}(x, y)$ as

$$\begin{aligned} p_X(x) &= \sum_j p_{XY}(x_i, y_j) \\ p_Y(y) &= \sum_i p_{XY}(x_i, y_j) \end{aligned} \tag{2.50}$$

Example 2.13 Find the marginal PMF $p_X(x)$ given the following joint PMF

$$p_{XY}(x, y) = \begin{cases} \frac{2[(x+1)/(x+2)]^y}{n(n+3)}, & \text{if } x = 0, 1, 2, \ldots, n-1,\ y \ge 0 \\ 0, & \text{otherwise} \end{cases}$$

Solution

For x = 0, 1, 2, …, n, we take the summation of y over the whole range

$$\begin{aligned} p_X(x) &= \sum_{y=-\infty}^{+\infty} p_{XY}(x, y) \\ &= \sum_{y=0}^{+\infty} \frac{2[(x+1)/(x+2)]^y}{n(n+3)} \\ &= \frac{2}{n(n+3)} \frac{1}{1-(x+1)/(x+2)} = \frac{2(x+2)}{n(n+3)} \end{aligned}$$

Thus

$$p_X(x) = \begin{cases} \frac{2(x+2)}{n(n+3)}, & \text{if } x = 0, 1, 2, \ldots, n-1, \\ 0, & \text{otherwise} \end{cases}$$

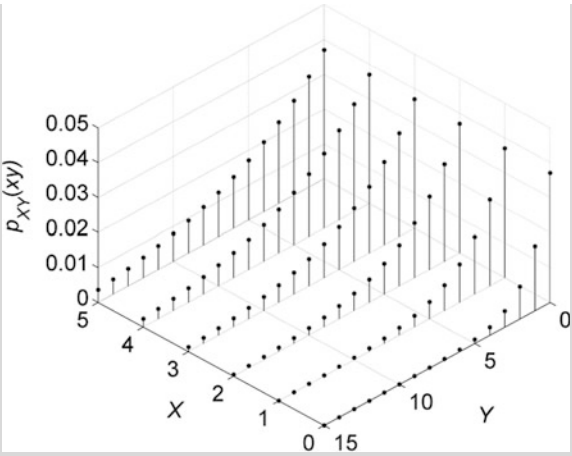


Fig. 2.10 Joint PMF in Example 2.13

We verify by induction that

$$\sum_{x=0}^{n-1} p_X(x) = \sum_{x=0}^{n-1} \frac{2(x+2)}{n(n+3)} = \frac{2}{n(n+3)} \frac{n(n+3)}{2} = 1$$

The joint PMF with $n = 5$ is illustrated in Fig. 2.10.

Next, let us consider the *continuous* case. The joint CDF of two continuous random variables X and Y is defined by

$$F_{XY}(x,y) = P(X \le x, Y \le y) = \int_{-\infty}^{y} \int_{-\infty}^{x} f_{XY}(\tau,\upsilon) d\tau d\upsilon \tag{2.51}$$

where $f_{XY}(x, y)$ is the joint PDF of X and Y. The joint PDF $f_{XY}(x, y)$ can also be obtained from the joint CDF as

$$f_{XY}(x,y) = \frac{\partial^2 F_{XY}(x,y)}{\partial x \partial y} = \frac{\partial^2 F_{XY}(x,y)}{\partial y \partial x} \tag{2.52}$$

An example joint PDF is shown in Fig. 2.11, where X and Y are jointly continuous. The line boundary between two surface areas with different line styles follows the shape of a conditional PDF, which will be introduced in the next section.

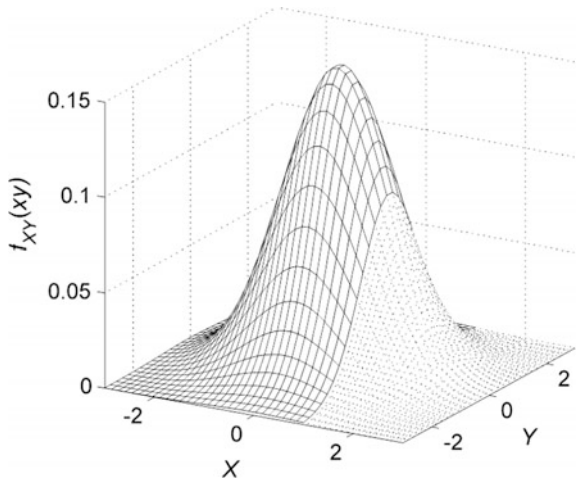


Fig. 2.11 An example of a joint continuous PDF

The integrations of the joint PDF $f_{XY}(x, y)$ with respect to X and Y give the marginal PDFs $f_Y(y)$ and $f_X(x)$, respectively, as

$$\begin{aligned} f_Y(y) &= \int_{-\infty}^{+\infty} f_{XY}(\tau, y) d\tau \\ f_X(x) &= \int_{-\infty}^{+\infty} f_{XY}(x, \upsilon) d\upsilon \end{aligned} \tag{2.53}$$

The equation above suggests that integrating out one random variable produces the marginal PDF of the other.

Example 2.14 Find the marginal PDFs $f_U(u)$ and $f_V(v)$ of two random variables u and v given the following joint PDF ($|\rho| < 1$)

$$\phi(u, v; \rho) = \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left[-\frac{1}{2(1-\rho^2)}\left(u^2 - 2\rho uv + v^2\right)\right]$$

Solution
The symmetry of the joint PDF indicates that the two random variables have the same marginal PDFs. It is then sufficient to only compute $f_U(u)$. To do so, we integrate out v as

$$
\begin{aligned}
f_U(u) &= \int_{-\infty}^{+\infty} \phi(u, v; \rho) dv \\
&= \int_{-\infty}^{+\infty} \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left[-\frac{1}{2(1-\rho^2)}\left(u^2(1-\rho^2) + (v-\rho u)^2\right)\right] dv \\
&= \frac{e^{-u^2/2}}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}\sqrt{1-\rho^2}} \exp\left[-\frac{1}{2(1-\rho^2)}(v-\rho u)^2\right] dv
\end{aligned}
$$

Making the substitution

$$w = \frac{v - \rho u}{\sqrt{1-\rho^2}}$$

gives the following

$$f_U(u) = \frac{e^{-u^2/2}}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}w^2\right) dw$$

We have shown that the integration is one. This gives the marginal PDF as

$$f_U(u) = \frac{e^{-u^2/2}}{\sqrt{2\pi}}$$

which is a univariate standard normal distribution. As will be discussed later, the joint PDF considered here is a bivariate normal PDF, which has been widely used for modeling two continuous random variables.

2.6.2 Conditional Probability and Independence

In engineering practice, we may encounter problems where the probability distribution of a random variable is dependent on that of another random variable. The modeling of such a relationship requires the use of a conditional probability. Recall

from Sect. 2.2.3 that the probability of a joint event can be expressed as the product resulting from the multiplication of the probabilities of an individual event and a conditional event. Then it follows that

$$P(E_1|E_2) = \frac{P(E_1 \cap E_2)}{P(E_2)} \tag{2.54}$$

In the equation above, $P(E_1|E_2)$ is the conditional probability, assuming that E_2 has occurred. If we apply this relationship to a discrete random variable, we then obtain a conditional PMF as

$$\begin{aligned} p_{X|Y}(x|y) &= P(X = x|Y = y) \\ &= \frac{P(X = x, Y = y)}{P(Y = y)} \\ &= \frac{p_{XY}(x, y)}{p_Y(y)} \end{aligned} \tag{2.55}$$

Similarly, $p_{Y|X}(y|x) = p_{XY}(x, y)/p_X(x)$. These formulae indicate that, for any fixed x and y, $p_{Y|X}(y|x)$ and $p_{X|Y}(x|y)$ share the same shapes as slices of $p_{XY}(x, y)$ with fixed x and y, respectively. Two discrete random variables are independent if and only if the following relationship holds

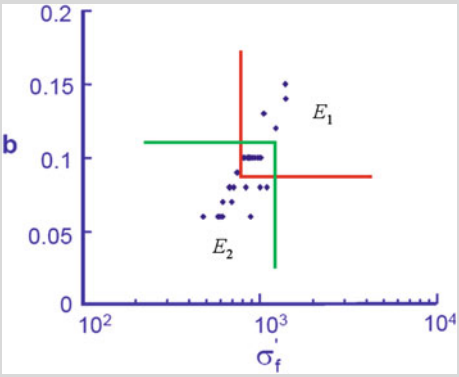
$$p_{XY}(x, y) = p_X(x)p_Y(y) \tag{2.56}$$

Under the assumption of independence, the conditional probability becomes simply the probability under no conditions: $p_{X|Y}(x|y) = p_X(x)$; $p_{Y|X}(y|x) = p_Y(y)$.

Example 2.15 Let us recall the example of fatigue tests. The sample measurements can be obtained for the physical quantities in the strain-life model below.

$$\frac{\Delta\varepsilon}{2} = \frac{\sigma_f'}{E}(2N_f)^b + \varepsilon_f'(2N_f)^c$$

Exercise: Assume that we have 20 measurements for the fatigue strength coefficient (σ_f') and exponent (b) used in the strain-life formula. Two events are defined as $E_1 = \{(X_1, X_2)|\ X_1 > 8 \times 10^2$ and $X_2 > 0.09\}$ and $E_2 = \{(X_1, X_2)|\ X_1 < 1.2 \times 10^3$ and $X_2 < 0.11\}$, which are graphically shown in the following figure



Note that $P(E_1) = 8/20 = 2/5$, $P(E_2) = 16/20 = 4/5$, $P(E_1 \cap E_2) = 4/20 = 1/5$. Find the conditioned probabilities $P(E_1|E_2)$ and $P(E_2|E_1)$..

Solution
From Eq. (2.54), we have

$$P(E_1|E_2) = \frac{P(E_1 \cap E_2)}{P(E_2)} = \frac{1/5}{4/5} = \frac{1}{4}$$
$$P(E_2|E_1) = \frac{P(E_1 \cap E_2)}{P(E_1)} = \frac{1/5}{2/5} = \frac{1}{2}$$

As shown in Fig. 2.10 (with $n = 5$), for any fixed x, the joint $p_{XY}(x, y)$ becomes a slice as a function of y, which shares the same shape as that of the above $p_{X|Y}(x|y)$.

Example 2.16 Find the conditional PMF $p_{Y|X}(y|x)$ given the joint PMF in Example 2.13.

Solution
Recall from Example 2.13 that

$$p_X(x) = \begin{cases} \frac{2(x+2)}{n(n+3)}, & \text{if } x = 0, 1, 2, \ldots, n-1, \\ 0, & \text{otherwise} \end{cases}$$

Thus, for $y \geq 0$,

$$p_{Y|X}(y|x) = \frac{p_{XY}(x,y)}{p_X(x)} = \frac{2[(x+1)/(x+2)]^y/[n(n+3)]}{[2(x+2)]/[n(n+3)]} = \frac{1}{x+2}\left(\frac{x+1}{x+2}\right)^y$$

Therefore,

$$p_{Y|X}(y|x) = \begin{cases} \frac{1}{x+2}\left(\frac{x+1}{x+2}\right)^y, & \text{if } y \geq 0 \\ 0, & \text{if } y < 0 \end{cases}$$

As shown in Fig. 2.10 (with $n = 5$), for any fixed x, the joint $p_{XY}(x, y)$ becomes a slice as a function of y, which shares the same shape as that of the above $p_{X|Y}(x|y)$.

For a continuous random variable, the conditional PDF can be expressed as

$$\begin{aligned} f_{X|Y}(x|y) &= \frac{f_{XY}(x,y)}{f_Y(y)} \\ f_{Y|X}(y|x) &= \frac{f_{XY}(x,y)}{f_X(x)} \end{aligned} \tag{2.57}$$

Two jointly continuous random variables X and Y are independent if and only if $f_{XY}(x, y) = f_X(x)f_Y(y)$. As shown in Fig. 2.11, a slice of $f_{XY}(x, y)$ for a fixed x (i.e., the line boundary between two surface areas) shares the same shape as the conditional PDF.

2.6.3 Covariance and Correlation Coefficient

In the case of a single random variable, we use its variance to describe the extent to which samples deviate from the mean of this variable. In the case of two random variables, we need to measure how much the two variables deviate together, or the cooperative deviation. For that purpose, we define the covariance between two random variables X and Y as

$$\text{Cov}(X, Y) = E[(X - \mu_X)(Y - \mu_Y)] \tag{2.58}$$

We further derive this definition as

$$\begin{aligned}\mathrm{Cov}(X,Y) &= E(XY - \mu_X Y - X\mu_Y + \mu_X\mu_Y) \\ &= E(XY) - \mu_X\mu_Y = E(XY) - E(X)E(Y)\end{aligned} \tag{2.59}$$

Based on the definition of covariance, we define the correlation coefficient as

$$\rho_{XY} = \frac{\mathrm{Cov}(X,Y)}{\sigma_X\sigma_Y} = E\left[\left(\frac{X-\mu_X}{\sigma_X}\right)\left(\frac{Y-\mu_Y}{\sigma_Y}\right)\right] \tag{2.60}$$

It is important to note that the correlation coefficient satisfies $|\rho_{XY}| \leq 1$ and measures a linear dependence between two random variables. $|\rho_{XY}| = 1$ in the case of a perfect linear relationship between X and Y; and $|\rho_{XY}| = 0$ if and only if X and Y are uncorrelated. Clearly, if $E[XY] = E[X]E[Y]$, X and Y are uncorrelated.

If the two variables are independent, they are correlated and $\rho_{XY} = 0$. However, the converse may not be true since the correlation coefficient only measures a linear dependence between two variables. It is thus fair to say that the condition of independence is stronger than that of being uncorrelated. Examples of correlated and dependent, uncorrelated and dependent, and uncorrelated and independent random variables are shown in Fig. 2.12.

2.6.4 Bivariate Normal Distribution

The bivariate normal or Gaussian distribution is a widely used joint distribution that can be easily extended to the multivariate case. In order to define a general bivariate normal distribution, let us first define a standard bivariate normal distribution as

$$\phi(u,v;\rho) = \frac{1}{2\pi\sqrt{1-\rho^2}}\exp\left[-\frac{1}{2(1-\rho^2)}\left(u^2 - 2\rho uv + v^2\right)\right] \tag{2.61}$$

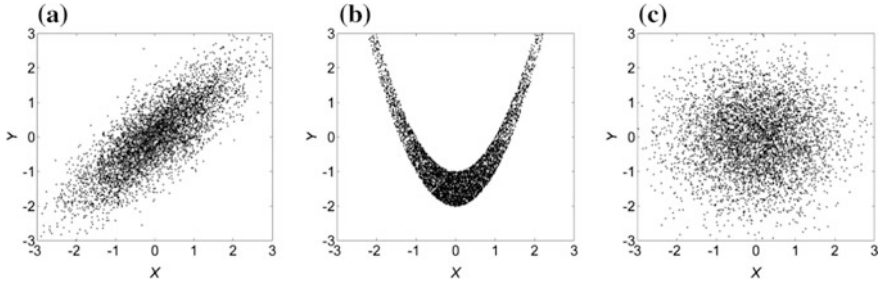


Fig. 2.12 Scatter plots of random samples from correlated and dependent (**a**), uncorrelated and dependent (**b**), and uncorrelated and independent (**c**) random variables

where u and v are jointly continuous standard normal variables, and $|\rho| < 1$ is the correlation coefficient. Recall from Example 2.14 the following integration

$$\int_{-\infty}^{+\infty} \phi(u, v; \rho) dv = \frac{e^{-u^2/2}}{\sqrt{2\pi}} \tag{2.62}$$

which, after taking another integration, becomes

$$\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \phi(u, v; \rho) dv du = \int_{-\infty}^{+\infty} \frac{e^{-u^2/2}}{\sqrt{2\pi}} du \tag{2.63}$$

This integral of a standard normal PDF has been shown to be one. Until now, we have shown that the double integral of the bivariate normal PDF in Eq. (2.61) equals one, which means the probability that the pair (u, v) falls into any point on the whole 2D plane is one.

A bivariate standard normal PDF surface for $\rho = 0$ is shown in Fig. 2.13a. Observe from the figure that the surface exhibits a perfect circular symmetry. In other words, for all (u, v) combinations on a circle with a certain radius, the standard normal PDF takes the same values. This can be more clearly seen from the PDF contour in Fig. 2.13b. Next, let us take one step further to derive Eq. (2.61) under $\rho = 0$ as

$$\begin{aligned} \phi(u, v; \rho = 0) &= \frac{1}{2\pi} \exp\left[-\frac{1}{2}\left(u^2 + v^2\right)\right] \\ &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{u^2}{2}\right) \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{v^2}{2}\right) \end{aligned} \tag{2.64}$$

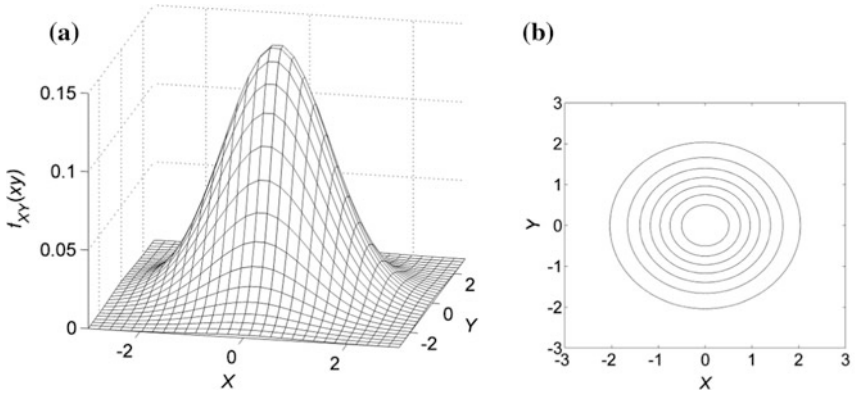


Fig. 2.13 Bivariate standard normal PDF surface (**a**) and PDF contour (**b**): $\rho = 0$

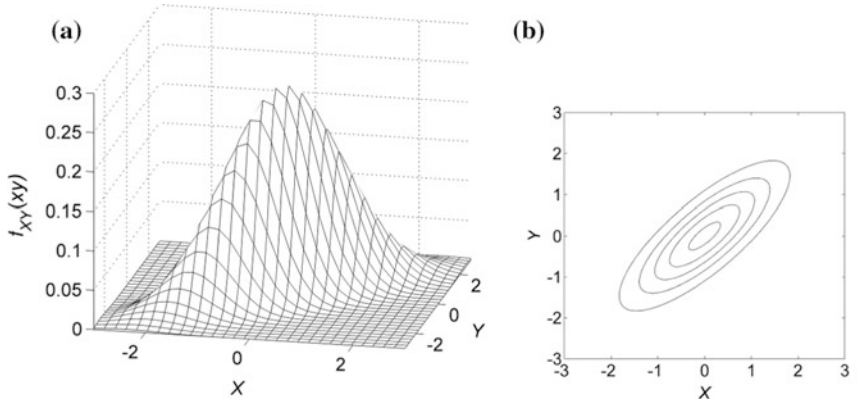


Fig. 2.14 Bivariate standard normal PDF surface (**a**) and PDF contour (**b**): $\rho = 0.8$

This means that the joint PDF is a product of two marginal standard normal PDFs. We then note that u and v are independent if $\rho = 0$. We further note that, if $\rho \neq 0$, the joint PDF is not separable. Thus, jointly continuous standard normal variables are independent if and only if their correlation coefficient $\rho = 0$. The joint PDF surface for $\rho = 0.8$ is plotted in Fig. 2.14a and the corresponding PDF contour is plotted in Fig. 2.14b. Note that the circles on which the joint PDF takes constant values now become ellipses whose axes are the 45° and 135° diagonal lines.

Based on the bivariate standard normal PDF, we can easily define a general bivariate normal PDF with the mean values μ_X and μ_Y, the standard deviations σ_X and σ_Y, and the correlation coefficient ρ_X as

$$\begin{aligned} f_{XY}(x,y) &= \frac{1}{\sigma_X \sigma_Y} \phi\left(\frac{x-\mu_X}{\sigma_X}, \frac{y-\mu_Y}{\sigma_Y}; \rho\right) \\ &= \frac{\exp\left\{-\frac{1}{2(1-\rho_{XY}^2)}\left[\left(\frac{x-\mu_X}{\sigma_X}\right)^2 - 2\rho_{XY}\left(\frac{x-\mu_X}{\sigma_X}\right)\left(\frac{y-\mu_Y}{\sigma_Y}\right) + \left(\frac{y-\mu_Y}{\sigma_Y}\right)^2\right]\right\}}{2\pi\sigma_X\sigma_Y\sqrt{1-\rho_{XY}^2}} \end{aligned} \tag{2.65}$$

By integrating out unneeded variables, we can show that the marginal PDFs $f_X(x)$ and $f_Y(y)$, respectively, follow normal distributions $N(\mu_X, \sigma_X^2)$ and $N(\mu_Y, \sigma_Y^2)$. The bivariate normal distribution can be generalized for an N-dimensional random vector $\mathbf{X}$: $\Omega \to \mathbb{R}^N$. The joint CDF and PDF for an N-dimensional random vector $\mathbf{X}$ are written as

$$\begin{aligned} &\text{Joint CDF:} \quad F_{\mathbf{X}}(\mathbf{x}) = P\left(\cap_{i=1}^{n}\{X_i \leq x_i\}\right) \\ &\text{Joint PDF:} \quad f_{\mathbf{X}}(\mathbf{x}) = \frac{\partial^n}{\partial x_1 \cdots \partial x_n} F_{\mathbf{X}}(\mathbf{x}) \end{aligned} \tag{2.66}$$

A multivariate normal random vector is distributed as

$$f_{\mathbf{X}}(\mathbf{x}) = (2\pi)^{-\frac{n}{2}}|\boldsymbol{\Sigma}_{\mathbf{X}}|^{-\frac{1}{2}}\exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_{\mathbf{X}})^T\boldsymbol{\Sigma}_{\mathbf{X}}^{-1}(\mathbf{x}-\boldsymbol{\mu}_{\mathbf{X}})\right] \tag{2.67}$$

where $\boldsymbol{\mu}_{\mathbf{X}}$ and $\boldsymbol{\Sigma}_{\mathbf{X}}$ are the mean vector and covariance matrix of $\mathbf{X}$, respectively.

2.7 Exercises

2.1 A drawer contains 10 pairs of loose, unpaired socks. Three pairs are black, 4 are gray, and 3 are red. Answer the following questions:

(1) If you remove 2 of the socks, what is the probability that one is black and the other is gray?
(2) If you remove 2 of the socks, what is the probability that they will match?

2.2 Recall the fatigue test example. The sample measurements can be obtained for the physical quantities in the damage model below.

$$\frac{\Delta\varepsilon}{2} = \frac{\sigma_f'}{E}(2N_f)^b + \varepsilon_f'(2N_f)^c$$

Consider a set of 30 measurement data (see Table 2.1) for the fatigue ductility coefficient (ε_f') and exponent (c) used in the strain-life formula. Answer the following questions:

(1) Compute the sample means and variances of the fatigue ductility coefficient (ε_f') and exponent (c), respectively.
(2) Construct the covariance matrix and find the coefficient of correlation using the data set given in Table 2.1.

2.3 In a cellular phone company, LCD fracture failures are commonly experienced. To gain a good understanding of the LCD fracture failure, the company performed a dent test on 50 LCD modules. The test data is given in

Table 2.1 Data for the fatigue ductility coefficient and exponent

ε_f'	c	ε_f'	c	ε_f'	c	ε_f'	c	ε_f'	c
0.022	0.289	0.253	0.466	0.539	0.630	0.989	0.694	1.611	0.702
0.071	0.370	0.342	0.531	0.590	0.621	1.201	0.690	1.845	0.760
0.146	0.450	0.353	0.553	0.622	0.653	1.304	0.715	1.995	0.759
0.185	0.448	0.354	0.580	0.727	0.635	1.388	0.717	2.342	0.748
0.196	0.452	0.431	0.587	0.729	0.645	1.392	0.716	3.288	0.821
0.215	0.460	0.519	0.655	0.906	0.703	1.426	0.703	6.241	0.894

Table 2.2 where the test data identifies the failure displacement (d_f) and failure force (F_f). Answer the following questions:

(1) Compute the sample means and variances of the failure displacement (d_f) and failure force (F_f).
(2) Construct the covariance matrix and find the coefficient of correlation using the data set given in Table 2.2.

2.4 During a manufacturing process that lasts 10 days, 15 units are randomly sampled each day from the production line to check for defective units. Based on historical information, it is known that the probability of a defective unit is 0.05. Any time that two or more defectives are found in the sample of 15, the process of that day is stopped.

(1) What is the probability that, of the 10 total days, 3 days will have the production stopped?
(2) Given (1), what is the probability that the process is stopped in the first 2 days?

2.5 Customers arrive in a certain store according to a Poisson process with a rate of $\lambda = 4$ per hour (the number of customers arriving at any time interval with a length t follows a Poisson distribution with the parameter λt). Given that the store opens at 9:00 am, answer the following questions.

(1) What is the probability that exactly 1 customer has arrived by 9:30 am?
(2) Given (1), what is the probability that a total of 5 customers have arrived by 11:30 am?

2.6 Let X and Y be two discrete random variables that take any of the values 1, 2, or 3. Their joint PMF $p_{XY}(x, y)$ is given by the following matrix, with $p_{XY}(x, y)$ being the element on the xth row and the yth column

Table 2.2 Failure displacement/force data (from the LCD module dent test)

d_f	F_f	d_f	F_f	d_f	F_f	d_f	F_f	d_f	F_f
1.0105	2.0428	1.1680	1.9648	1.2717	1.8874	1.2233	2.2746	1.0946	2.7343
0.6915	2.6993	0.6809	2.5530	0.6093	2.4002	1.1010	2.6674	1.0367	1.8956
1.4959	1.9897	0.6728	1.9722	0.6436	2.5901	0.9569	1.8998	1.4014	2.4851
0.8989	2.7379	1.2995	1.3366	1.2011	2.7858	0.6554	2.7567	1.3191	2.1417
0.8676	2.7248	1.4146	1.7351	1.1431	1.9160	1.3022	2.4822	1.2609	1.3805
0.6558	2.0972	1.0804	2.7867	1.0735	2.5905	0.9038	2.5498	0.6462	2.3304
0.8684	2.4429	0.6982	2.4016	1.2567	2.3779	1.1471	2.2240	0.6656	2.0282
0.6417	2.0810	1.3432	2.3986	1.2365	2.2008	1.2671	2.7754	0.6797	1.9377
1.0549	1.9513	0.9043	2.0410	1.3032	2.5604	0.6943	2.3596	1.3185	2.3303
1.2853	2.1653	0.8646	2.1482	0.8592	2.0957	0.7151	2.7484	1.4487	2.0789

$$\begin{bmatrix} 0 & 0 & 1/8 \\ 1/2 & 1/4 & 0 \\ 0 & 0 & 1/8 \end{bmatrix}$$

Answer the following questions.

(1) Find the marginal PMFs $p_X(x)$ and $p_Y(y)$.
(2) Compute the probability $P(X < Y)$.
(3) Determine whether X and Y are independent.
(4) Compute the conditional probability $P(X = 2|Y = 2)$.

2.7 Given the joint density function of two random variables X and Y

$$f_{XY}(x,y) = \begin{cases} 4xy, & \text{if } 0 < x, y < 1 \\ 0, & \text{otherwise} \end{cases}$$

Answer the following questions.

(1) Find the marginal distributions of X and Y.
(2) Compute the conditional probability $P(0 < X < 0.5|Y = 0.25)$.

2.8 Given the joint density function of two random variables X and Y

$$f_{XY}(x,y) = \begin{cases} \frac{2}{3}(x+2y), & \text{if } 0 < x, y < 1 \\ 0, & \text{otherwise} \end{cases}$$

Answer the following questions:

(1) Find the marginal distributions of X and Y.
(2) Compute the means and variances of X and Y.
(3) Compute the correlation coefficient between X and Y.

2.9 Let the random variables X and Y be independent with the densities being $f_X(x)$ and $f_Y(y)$, respectively. Assume that Y is a positive random variable. Define a new random variable as $Z = \ln(Y)/X$. Answer the following questions:

(1) Find the PDF of Z.
(2) Given that the expectations of X and Y being $E(X)$ and $E(Y)$, respectively, compute $E(Z)$.

2.10 Suppose that X and Y follow a standard bivariate normal distribution with the correlation coefficient being ρ. Answer the following questions:

(1) Prove that the covariance of X and Y is ρ.
(2) Find the conditional PDFs $f_{X|Y}(x|y)$ and $f_{Y|X}(y|x)$.

Chapter 3
Statistical Data Analysis

In Chap. 2, we discussed the basic concepts of probability, discrete and continuous random variables, univariate and joint probability distributions, conditional probability, and independence. A reasonably complete and self-sustained treatment has been given to these topics, which form a theoretical basis for practical statistical data analysis. This chapter discusses statistical analysis based on available sample data. Specifically, it introduces statistical procedures to determine an appropriate probability distribution for a random variable based on a limited set of sample data. Reliability analysis that will be discussed in subsequent chapters often requires these procedures. In this chapter, discussion of statistical data analysis will be separately conducted on two different branches of techniques: (i) conventional statistical methods (graphical methods and statistical hypothesis tests) and (ii) Bayesian statistics.

3.1 Conventional (or Frequentist) Statistical Methods

In many engineering design problems, uncertainty from various sources (e.g., material properties, geometric tolerances, and operation conditions) is modeled using random variables. The randomness of the variables is represented by probability distributions, which are often unknown. In most situations, it is impossible or impractical to observe the entire population of engineered products. For example, it is prohibitively time-consuming and expensive to test the compressive strength of all Aluminum–Lithium alloy parts. Thus, we often rely on a given sample of observations (or sample data) from the population to infer a probability distribution that best represents the population. In such cases, we need to develop and validate an answer to an essential question: Which distribution best represents the randomness conveyed by the sample of observations? Under the context of conventional statistical inference, graphical methods (e.g., histograms and probability plots) and/or quantitative methods (e.g., sample statistics, maximum

C. Hu et al., *Engineering Design under Uncertainty and Health Prognostics*,
Springer Series in Reliability Engineering,
https://doi.org/10.1007/978-3-319-92574-5_3

likelihood estimation, and statistical tests) can be used to derive such a probability distribution. These two categories of methods are separately discussed in this section.

3.1.1 Graphical Methods

Graphical display of sample data enables visual examination of the data, which often gives insights useful for choosing an appropriate probability distribution to model the uncertainty of the population from which the data are sampled. Two graphical methods, namely histograms and probability plotting, have proved their usefulness in statistical data analysis.

Histogram

A histogram visually shows the frequency distribution of a given sample of observations. A histogram can be constructed using the following steps:

Step 1: Specify the number of bins n_b based on the number of observations M, subdivide the range of data into n_b equal intervals, and specify the boundaries of these intervals.
Step 2: Count the number of observations falling into each interval as the frequency in the interval and, if needed, calculate the normalized frequency by dividing the observed frequency in the interval by the total number of observations M.
Step 3: Plot the histogram by drawing, above each interval, a rectangle whose width is the length of the interval and whose height is the frequency (or normalized frequency) corresponding to the interval.

The number of bins n_b is very critical in constructing an informative histogram. Generally speaking, a choice of n_b between 5 and 20 often gives satisfactory results in engineering practice. An empirical square root relationship between n_b and the number of observations M (i.e., $n_b = M^{1/2}$) can be used as a general guideline in determining the number of bins.

We can practice the above-mentioned three steps using the comprehensive strength data in Table 3.1. The data were obtained from testing 80 Aluminum–Lithium alloy specimens. In what follows, we demonstrate the process of creating a histogram in a step-by-step manner.

Step 1: Using the empirical square root relationship, we find $n_b = M^{1/2} = 80^{1/2} \approx 9$. Therefore, the raw data are subdivided into 9 equal intervals. The minimum and maximum strength values, 76 and 245, are then rounded off to 70 and 250, respectively. Thus, the total length of the intervals is 180 with 9 bins, which results in the width of each bin being 20.
Step 2: The number of observations falling into each interval and the resultant frequency distribution are then counted and summarized, as shown in Table 3.2.

Table 3.1 Compressive strength data (unit: psi) from 80 Aluminum–Lithium alloy specimens

105	221	183	186	121	181	180	143
97	154	153	174	120	168	167	141
245	228	174	199	181	158	176	110
163	131	154	115	160	208	158	133
207	180	190	193	194	133	156	123
134	178	76	167	184	135	229	146
218	157	101	171	165	172	158	169
199	151	142	163	145	171	148	158
160	175	149	87	160	237	150	135
196	201	200	176	150	170	118	149

Table 3.2 Distributions of frequency and normalized frequency for compressive strength data

Interval (psi)	Frequency	Normalized frequency
70–90	2	2/80 = 0.0250
90–110	3	0.0375
110–130	6	0.0750
130–150	14	0.1750
150–170	22	0.2750
170–190	17	0.2125
190–210	10	0.1250
210–230	4	0.0500
230–250	2	0.0250

The normalized frequency is also computed by dividing the frequency by the total number of observations.

Step 3: The histogram is graphed, as shown in Fig. 3.1a, using the frequency values in Table 3.2. Also plotted is the normalized histogram, shown in Fig. 3.1b, whose total area is approximately 1.0.

A histogram graphically informs an engineer of the properties (e.g., central tendency, dispersion, skewness) of the distribution of the sample data. These properties often give insights into the choice of a probability distribution to model the uncertainty of the population. As can be seen from Fig. 3.1, the histogram appears to be symmetric and bell-shaped, which provides evidence that the normal distribution is a good choice to represent the population of compressive strength measurements. The good match between the histogram and the normal fit strengthens the validity of this choice. Detailed information regarding how to fit a normal distribution and how to quantitatively validate this fit will be discussed in subsequent sections.

The histogram in Fig. 3.1 possesses a single mode where the maximum value is taken. This type of probability distribution is referred to as a *unimodal* distribution. In engineering practice, we may also encounter probability distributions with more than one mode. An example of a bimodal distribution is shown in Fig. 3.2. The

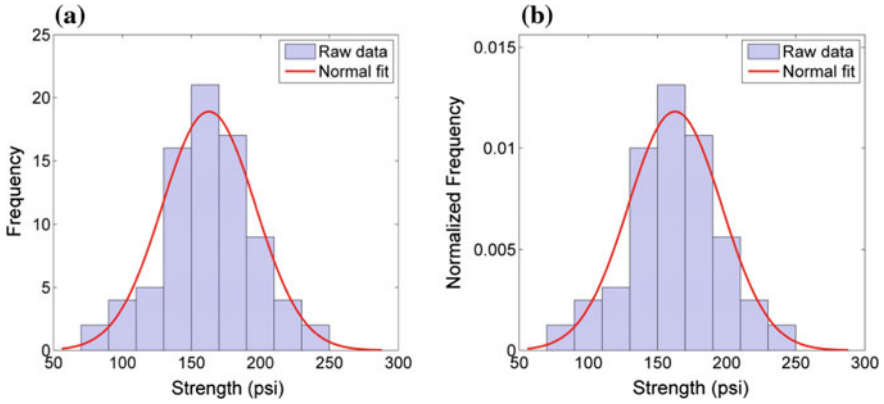


Fig. 3.1 Histogram (**a**) and normalized histogram (**b**) of compressive strength

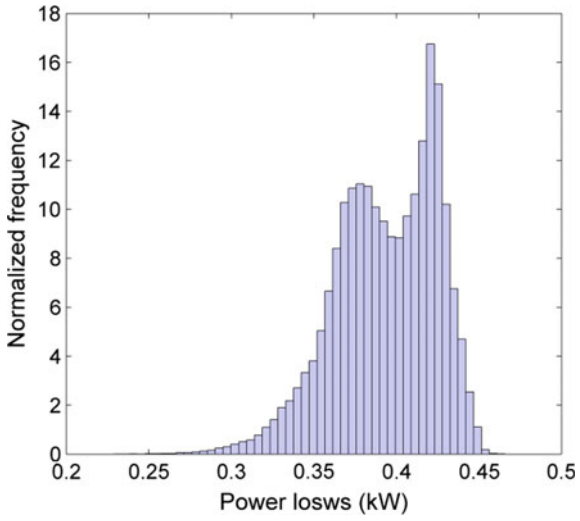


Fig. 3.2 An example bimodal probability distribution

random variable is the power loss due to the friction between the piston ring and cylinder liner, oil consumption, blow-by, and/or liner wear rate in a V6 gasoline engine. Compared to a unimodal distribution, a bimodal distribution is more complicated and more challenging to analyze. Reliability analysis involving this type of PDF will be discussed in detail in Chap. 5.

Example 3.1 The reliability of the Aluminum–Lithium alloy is defined as the probability that its compressive strength exceeds 110 psi. Estimate this reliability based on the histogram in Fig. 3.1a.

Solution
In conventional statistical inference, probability is interpreted as normalized frequency. Thus, the reliability can be estimated by computing the normalized frequency of finding a sample alloy specimen whose comprehensive strength is larger than 110 psi, expressed as

$$R = P(E > 110\,\text{psi}) = 1 - P(E \leq 110\,\text{psi}) = 1 - \frac{2+3}{80} = \frac{15}{16}$$

Probability Plotting

As mentioned earlier, the visual display of the sample distribution provided by a histogram provides insights into which probability distribution offers a reasonable representation of the uncertainty in the sample data. However, for a problem with sample data of a small to moderate size, the histogram might give a misleading indication on the underlying distribution. In such cases, the underlying distribution can be better identified by plotting the sample data along the x-axis and the corresponding empirical cumulative probability values along the y-axis. This graphical method is referred to as *probability plotting*. It assumes a hypothesized distribution and uses a *probability paper* that is constructed according to the hypothesized distribution. Probability paper is commonly used for the normal, lognormal, and Weibull distributions. If the empirical cumulative curve follows a linear line on the probability paper, it can be concluded that the underlying distribution conforms to the hypothesis.

The process of constructing a probability plot consists of three steps. First, the sample of observations is sorted in ascending order. The sorted observations are denoted as $x_1, x_2, \ldots, x_M$. Next, the empirical cumulative probability values are computed for x_i as $(i - 0.5)/M$, $i = 1, 2, \ldots, M$. Finally, the empirical cumulative probability values are plotted versus the sample values on the probability paper corresponding to a hypothesized distribution. If the hypothesized distribution correctly represents the uncertainty of the data, the plotted points should form an approximately straight line. The further the points deviate from a straight line, the greater the indication of a departure from the hypothesized distribution. It is usually a subjective decision to determine whether the data plot forms a straight line.

This process can be illustrated using the compressive strength data in Table 3.1. Let us, for example, investigate whether the data follow a normal distribution by using the normal probability plot. First, we arrange the data in ascending order and compute the empirical cumulative probability values, as summarized in Table 3.3. Then, we plot $(i - 0.5)/M$ and x_i on a normal probability paper, as shown in Fig. 3.3a. A straight line is also plotted to help judge whether the data plot follows a

Table 3.3 Empirical cumulative probability values for compressive strength data

x_i	i	$(i-0.5)/M$	x_i	i	$(i-0.5)/M$	x_i	i	$(i-0.5)/M$	x_i	i	$(i-0.5)/M$
76	1	0.0063	145	21	0.2562	163	41	0.5062	181	61	0.7562
87	2	0.0187	146	22	0.2687	163	42	0.5188	183	62	0.7688
97	3	0.0313	148	23	0.2813	165	43	0.5313	184	63	0.7813
101	4	0.0437	149	24	0.2938	167	44	0.5437	186	64	0.7937
105	5	0.0563	149	25	0.3063	167	45	0.5563	190	65	0.8063
110	6	0.0688	150	26	0.3187	168	46	0.5687	193	66	0.8187
115	7	0.0813	150	27	0.3312	169	47	0.5813	194	67	0.8313
118	8	0.0938	151	28	0.3438	170	48	0.5938	196	68	0.8438
120	9	0.1063	153	29	0.3563	171	49	0.6062	199	69	0.8562
121	10	0.1187	154	30	0.3688	171	50	0.6188	199	70	0.8688
123	11	0.1313	154	31	0.3812	172	51	0.6312	200	71	0.8812
131	12	0.1437	156	32	0.3937	174	52	0.6438	201	72	0.8938
133	13	0.1563	157	33	0.4063	174	53	0.6563	207	73	0.9063
133	14	0.1688	158	34	0.4188	175	54	0.6687	208	74	0.9187
134	15	0.1812	158	35	0.4313	176	55	0.6813	218	75	0.9313
135	16	0.1938	158	36	0.4437	176	56	0.6937	221	76	0.9437
135	17	0.2062	158	37	0.4562	178	57	0.7063	228	77	0.9563
141	18	0.2188	160	38	0.4688	180	58	0.7188	229	78	0.9688
142	19	0.2313	160	39	0.4813	180	59	0.7312	237	79	0.9812
143	20	0.2437	160	40	0.4938	181	60	0.7438	245	80	0.9938

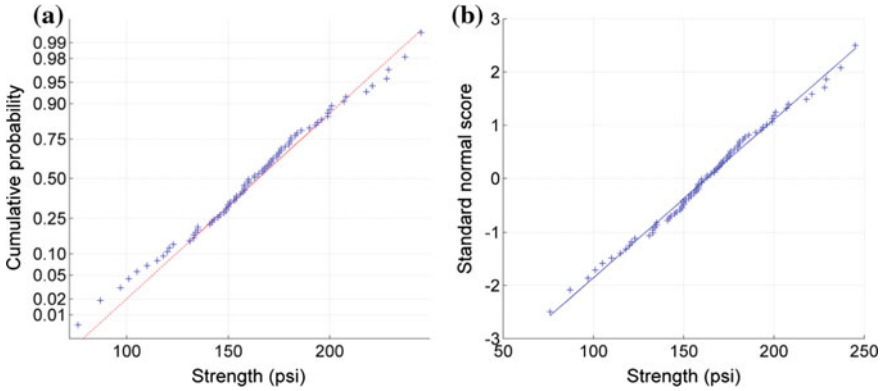


Fig. 3.3 Normal probability plots of compressive strength data using cumulative probability values (**a**) and standard normal scores (**b**)

linear line. Since all the points appear to lie around the straight line, it can be concluded that the compressive strength follows, at least approximately, a normal distribution.

It is worth noting that we can also build a probability plot on a standard two dimensional graph paper. This can be done by transforming the CDF values of x_i to the corresponding standard normal scores z_i. This transformation can be expressed as

$$z_i(x_i) = \Phi^{-1}(P(Z \leq z_i)) = \Phi^{-1}\left(\frac{i - 0.5}{M}\right) \tag{3.1}$$

where Φ^{-1} is the inverse standard normal CDF. The standard normal scores z_i are plotted against the sample values x_i in Fig. 3.3b. This normal probability plot makes the empirical cumulative curve linear by transforming the cumulative probabilities of x_i, while the one in Fig. 3.3a does so by adjusting the scale of the y-axis. Note that these two plots are virtually equivalent.

3.1.2 *Quantitative Methods*

Quantitative methods for estimating the underlying distribution of the population can be categorized into two groups: parameter estimation and hypothesis testing. We will discuss two methods for parameter estimation: (i) *method of moments* and (ii) *maximum likelihood estimation* (*MLE*). The discussion of hypothesis testing will be focused on *goodness-of-fit* tests.

Method of Moments

As mentioned in Sect. 2.4, mean, variance, skewness, and kurtosis are the first four statistical moments. These moments not only provide information about the characteristics (e.g., center, dispersion, and shape) of the PDF of a random variable X, but the moments can also be used to estimate the distributional parameters of the PDF. The central idea behind the method of moments (or sample statistics) is that the distributional parameters of a PDF can be estimated by equating the population moments (derived from the functional form of the PDF) to the corresponding sample moments (derived from a sample of observations). A set of such equations with the distributional parameters as the unknowns can be solved to produce estimates of these distributional parameters. The detailed procedure is listed as follows:

Step 1: Equate the first population moment $E(X)$ to the first sample moment (sample mean).
Step 2: Equate the second population moment $E[(X - \mu)^2]$ to the second sample moment (sample variance).

Step 3: Continue equating the third and higher-order population moments $E[(X-\mu)^k]$, $k = 3, 4,\ldots$, with the corresponding sample moments, until the number of equations equals that of distributional parameters.
Step 4: Solve the equations from Steps 1–3 to obtain the method of moments estimators of the parameters.

Most distributions have two distributional parameters, and the mean and variance suffice to yield the estimates of the distributional parameters for these distributions. The first two population moments, mean and variance, can be expressed as functions of distributional parameters for several important probability distributions, as shown in Table 3.4.

Example 3.2 Suppose we have a set of random samples $x_1, x_2, \ldots, x_M$ from a normal distribution with the PDF as

$$f_X(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]$$

Find the method of moments estimators of the distributional parameters μ and σ.

Solution
The first population moment (population mean) is

$$E(X) = \mu$$

The second population moment (population variance) is

$$\mathrm{Var}(X) = E\left[(X-\mu)^2\right] = \sigma^2$$

Again, since the normal distribution has two distributional parameters, we only need two equations. Equating the first population moment with the corresponding sample moment gives

$$E(X) = \mu = \frac{1}{M}\sum_{i=1}^{M} x_i$$

Next, equating the second theoretical moment about the mean with the corresponding sample moment gives:

$$E\left[(X-\mu)^2\right] = \sigma^2 = \frac{1}{M}\sum_{i=1}^{M}(x_i-\hat{\mu})^2$$

Table 3.4 Mean and standard deviation as functions of distributional parameters

	Distribution	Probability mass/density function (PMF/PDF)	Moments as functions of parameters
Discrete	Bernoulli	$p_X(x;p_0) = p_0^x(1-p_0)^{1-x}$ $x = 0, 1$	$\mu_X = p_0$ $\sigma_X^2 = p_0(1-p_0)$
	Binomial	$p_X(x;n,p_0) = C(n,x)p_0^x(1-p_0)^{n-x}$ $x = 0, 1, 2, \ldots, n$	$\mu_X = np_0$ $\sigma_X^2 = np_0(1-p_0)$
	Poisson	$p_X(x) = \frac{\lambda^x e^{-\lambda}}{x!}$ $x = 0, 1, 2, \ldots$	$\mu_X = \lambda,\ \sigma_X^2 = \lambda$
	Geometric	$p_X(x) = (1-p)^{x-1}p$ $x = 1, 2, \ldots$	$\mu_X = 1/p$ $\sigma_X^2 = (1-p)/p^2$
Continuous	Normal	$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]$	$\mu_X = \mu,\ \sigma_X^2 = \sigma^2$
	Lognormal	$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma x}\exp\left[-\frac{1}{2}\left(\frac{\ln x - \mu}{\sigma}\right)^2\right]$ $x > 0$	$\mu_X = \exp(\mu + \sigma^2/2)$ $\sigma_X^2 = \left(e^{\sigma^2} - 1\right)\exp(2\mu + \sigma^2)$
	Weibull	$f_X(x) = \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} e^{-\left(\frac{x}{\lambda}\right)^k}$ $x \geq 0$	$\mu_X = \lambda\Gamma\left(1 + \frac{1}{k}\right)$ $\sigma_X^2 = \lambda^2\Gamma\left(1 + \frac{2}{k}\right) - \mu_X^2$
	Exponential	$f_X(x) = \lambda e^{-\lambda x}$ $x \geq 0$	$\mu_X = \frac{1}{\lambda},\ \sigma_X^2 = \frac{1}{\lambda^2}$

Finally, we need to solve the two equations for the two parameters. In this particular case, the equations appear to have already been solved for μ and σ^2. We can thus obtain the method of moments estimators of the parameters as

$$\hat{\mu} = \frac{1}{M}\sum_{i=1}^{M} x_i$$

$$\hat{\sigma}^2 = \frac{1}{M}\sum_{i=1}^{M} (x_i - \hat{\mu})^2$$

Maximum Likelihood Estimation

Compared to the method of moments, the maximum likelihood estimation (MLE) is a more general and widely used method used to estimate distributional parameters. The basic idea of this method is to optimize the parameters such that a likelihood function is maximized.

Let us first consider the case of a continuous probability distribution. Assume that we have a set of random samples x_1, x_2, …, x_M from the PDF $f_X(x; \theta)$ of a continuous random variable X, where θ is an unknown distributional parameter. The goal here is to find a good point estimator of θ using the random samples x_1, x_2, …, x_M. It is reasonable that such an estimator of θ should be the value of θ that maximizes the probability (or likelihood) of obtaining the set of random samples. The likelihood function in this case can be viewed as a joint probability density function, expressed as

$$L(\theta) = f_X(x_1; \theta) f_X(x_2; \theta) \cdots f_X(x_M; \theta) = \prod_{i=1}^{M} f_X(x_i; \theta) \tag{3.2}$$

For convenience, we can use the logarithm of the likelihood function, namely the log-likelihood function, expressed as

$$\ln L(\theta) = \sum_{i=1}^{M} \ln f_X(x_i; \theta) \tag{3.3}$$

The maximum likelihood estimator of θ that maximizes the likelihood function above can be obtained by equating the first-order derivative of this function to zero. In cases where multiple distributional parameters need to be estimated, the likelihood function becomes a multivariate function of the unknown distributional parameters. We can find the maximum likelihood estimators of these parameters by equating the corresponding partial derivatives to zeros and solving the resultant set of equations.

Example 3.3 Suppose we have a set of random samples $x_1, x_2, \ldots, x_M$ from a normal distribution with the PDF as

$$f_X(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]$$

Find the maximum likelihood estimators of the distributional parameters μ and σ.

Solution
First, the likelihood function of the random samples is derived as

$$\begin{aligned} L(\mu,\sigma) &= \prod_{i=1}^{M} f_X(x_i;\mu,\sigma) = \prod_{i=1}^{M}\frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{1}{2}\left(\frac{x_i-\mu}{\sigma}\right)^2\right] \\ &= \left(2\pi\sigma^2\right)^{-n/2}\exp\left[-\frac{1}{2\sigma^2}\sum_{i=1}^{M}(x_i-\mu)^2\right] \end{aligned}$$

To make the computation more convenient, we derive the log-likelihood function as

$$\ln L(\mu,\sigma) = -\frac{n}{2}\ln\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2}\sum_{i=1}^{M}(x_i-\mu)^2$$

Next, we compute the partial derivatives of the log-likelihood function with respect to μ and σ, expressed as

$$\begin{aligned} \frac{\partial \ln L(\mu,\sigma)}{\partial\mu} &= \frac{1}{\sigma^2}\sum_{i=1}^{M}(x_i-\mu) \\ \frac{\partial \ln L(\mu,\sigma)}{\partial\sigma} &= -\frac{M}{2\sigma^2} + \frac{1}{2\sigma^4}\sum_{i=1}^{M}(x_i-\mu)^2 \end{aligned}$$

Finally, we equate the derivatives to zero and solve the resultant set of equations to obtain the maximum likelihood estimators

$$\begin{aligned} \hat{\mu} &= \frac{1}{M}\sum_{i=1}^{M}x_i \\ \hat{\sigma}^2 &= \frac{1}{M}\sum_{i=1}^{M}(x_i-\hat{\mu})^2 \end{aligned}$$

Note that the maximum likelihood estimators are the same as the estimators from the method of moments (see Example 3.2).

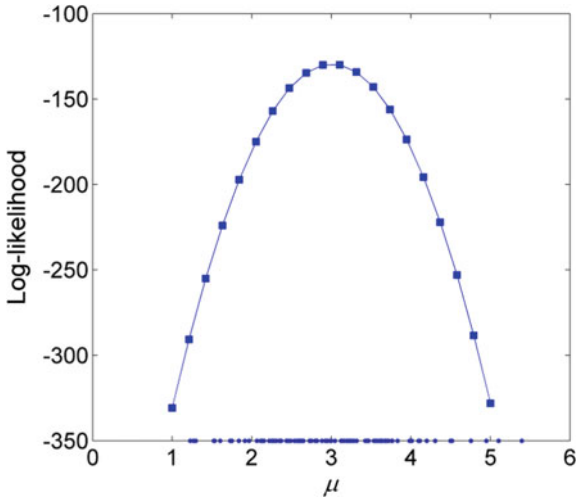


Fig. 3.4 Log-likelihood of 100 random samples from a normal distribution

Using this information, we can graphically demonstrate the basic idea of the MLE. First, we randomly generate 100 samples from a normal distribution with $\mu = 3.0$ and $\sigma = 1.0$. We then fix σ at 1.0 and plot the log-likelihood values against μ in Fig. 3.4. Observe that the log-likelihood function has a maximum value around $\mu = 3.0$, which indicates that the maximum likelihood estimator of μ is approximately equal to its true value.

For the case of a discrete probability distribution, the likelihood function of the samples becomes the probability of obtaining the samples $x_1, x_2, \ldots, x_M$, expressed as

$$L(\theta) = P(X_1 = x_1, X_2 = x_2, \ldots, X_M = x_M) = \prod_{i=1}^{M} p_{X_i}(x_i; \theta) \tag{3.4}$$

where $p_X(x; \theta)$ is the PMF of the discrete random variable and θ is the unknown distributional parameter. Observe that the maximum likelihood estimator of the parameter of a discrete distribution maximizes the probability of obtaining the sample values $x_1, x_2, \ldots, x_M$.

Example 3.4 Suppose the samples $x_1, x_2, \ldots, x_M$ are randomly drawn from a Bernoulli distribution with the PMF as

$$p_X(x; p_0) = \begin{cases} p_0^x (1 - p_0)^{1-x} & x = 0, 1 \\ 0 & \text{otherwise} \end{cases}$$

Find the maximum likelihood estimator of the distributional parameter p_0.

Solution
First, we derive the log-likelihood function of the random samples as

$$\ln L(p) = \sum_{i=1}^{M} \ln p_X(x; p_0) = \ln p_0 \sum_{i=1}^{M} x_i + \ln(1 - p_0)\left(M - \sum_{i=1}^{M} x_i\right)$$

Next, we take the first-order derivative of the likelihood function above as

$$\frac{d \ln L(p_0)}{dp_0} = \frac{1}{p_0} \sum_{i=1}^{M} x_i - \frac{1}{1 - p_0}\left(M - \sum_{i=1}^{M} x_i\right)$$

Finally, we equate the derivative to zero and solve the resultant equation to obtain the maximum likelihood estimator

$$\hat{p}_0 = \frac{1}{M} \sum_{i=1}^{M} x_i$$

Note that the estimator is the same as the estimator from the method of moments (see Table 3.4).

Chi-Square Goodness-of-Fit Tests

The parameter estimation methods discussed earlier are applicable to problems where the type of the underlying probability distribution of the population is known. In engineering practice, we often encounter problems where the types of the distributions are unknown. In such cases, we need to determine how well a sample of observations fit an assumed distribution. It is simple and quick to use probability plotting to determine how closely a set of sample data fit a certain type of distribution, but this graphical method requires subjective judgment on the linearity of the plotted curve on probability paper. Thus, an objective and quantitative way to test the goodness-of-fit is needed. In what follows, we describe a statistical test procedure for this purpose that is based on the chi-square distribution, namely the *chi-square test*.

The chi-square test is a statistical test that addresses how well an assumed distribution reflects a set of sample data, i.e., how close the observed values are to those that would be expected under the assumed distribution. The chi-square test is defined for the following hypotheses:

H_0 The data follow an assumed distribution.
H_1 The data do not follow the assumed distribution.

Suppose we have M random samples from a population whose distribution is unknown. We follow the same steps as in plotting a histogram to arrange the samples in n_b equally spaced intervals (or bins). Let O_i and E_i denote the observed and expected frequencies for the ith interval, respectively. The expected frequency is calculated by

$$E_i = M \cdot \left[F_X\left(x_i^u\right) - F_X\left(x_i^l\right)\right] \tag{3.5}$$

where $F_X(x)$ is the CDF of the hypothesized distribution being tested, and x_i^u and x_i^l are respectively the upper and lower limits of the ith interval. It follows that the test statistic

$$\chi^2 = \sum_{i=1}^{n_b} \frac{(O_i - E_i)^2}{E_i} \tag{3.6}$$

approximately follows a chi-square distribution with the degrees of freedom $v = n_b - k - 1$, where k denotes the number of distributional parameters of the hypothesized distribution. A high value of χ^2 suggests the observations have a high degree of deviation from the assumed distribution, which casts doubt on the null hypothesis. The null hypothesis is usually rejected when the value of χ^2 is larger than the 95th percentile. This corresponds to a significance level $\alpha = 0.05$, which requires the hypothesized distribution be acceptable for at least 95 out of 100 sets of random samples. The hypothesis $\mathbf{H_0}$, in favor of the assumed distribution, should be rejected if

$$\chi^2 = \sum_{i=1}^{n_b} \frac{(O_i - E_i)^2}{E_i} > \chi^2_{\alpha, n_b - k - 1} \tag{3.7}$$

where $\chi^2_{\alpha, n_b - k - 1}$ is the chi-square critical value with $n_b - k - 1$ degrees of freedom and the significance level α.

The chi-square goodness-of-fit test can be applied to both continuous and discrete distributions. Since this test involves the use of binned data (i.e., data categorized into classes), the value of the chi-square test statistic varies depending on how the data are binned. Furthermore, this test requires a reasonably large number [1] of observations in order for the chi-square approximation to be valid.

Example 3.5 Use the chi-square goodness-of-fit test to determine whether the compressive strength data of the Aluminum–Lithium alloy in Table 3.1 follow a normal distribution ($\alpha = 0.05$).

[1] A rough rule of thumb concerning the sample size is that there should be at least 50 samples in order for the chi-square approximation to be valid. A more rigorous approach to sample size determination involves studying the power of a hypothesis test [1].

Solution

The distributional parameters, μ and σ, of the hypothesized normal distribution can be directly obtained from the sample data as $\mu_X = 162.66$, $\sigma_X = 33.77$. We then follow Table 3.2 to divide the sample data into intervals and pool end intervals with frequencies less than 5 with neighboring intervals to make the frequencies at least 5. The distribution of observed and expected frequencies is shown in the table below. We then execute the hypothesis testing step-by-step.

Interval (psi)	Observed frequency O_i	Expected frequency E_i	$(O_i - E_i)^2/E_i$
70–110	5	4.51	0.0523
110–130	6	8.58	0.7771
130–150	14	14.97	0.0627
150–170	22	18.57	0.6331
170–190	17	16.39	0.0227
190–210	10	10.29	0.0081
210–250	6	6.05	0.0005
Total	80	80	1.5565

1. Specify the null and alternative hypotheses as

   H_0 The compressive strength data follow a normal distribution.
   H_1 The compressive strength data do not follow a normal distribution.

2. The significance level $\alpha = 0.05$.
3. The chi-square test statistic is defined as

$$\chi^2 = \sum_{i=1}^{n_b} \frac{(O_i - E_i)^2}{E_i}$$

4. The degree of freedom is computed as $v = n_b - k - 1 = 7 - 2 - 1 = 4$.
5. The critical value is $\chi^2_{0.05,4} = 9.49$.
6. The test statistic can be computed from a χ^2 look-up table as $\chi^2 = 1.56$.
7. Since $\chi^2 = 1.56 < \chi^2_{0.05,4} = 9.49$, we cannot reject $\mathbf{H_0}$. We cannot find sufficient evidence for the hypothesis that the strength data are not normally distributed. The p-value is computed as $p = 0.9065$.

Kolmogorov-Smirnov Goodness-of-Fit Tests

Another goodness-of-fit test, the *Kolmogorov-Smirnov (K-S) test*, is also quite useful in engineering practice, particularly in reliability analysis. The K-S test is used to statistically test if sampled data come from a hypothesized distribution, using the following hypotheses:

H_0 The data follow an assumed distribution.
H_1 The data do not follow the assumed distribution.

The basic idea of this test is to compare the empirical CDF $S_X(x)$ from the sample data with the theoretical CDF $F_X(x)$ from the hypothesized probability distribution. First, let us look at how to define an empirical CDF. Suppose we have M ordered random samples $x_1 \leq x_2 \leq \cdots \leq x_M$ from the underlying distribution of a population. The empirical CDF $S_X(x)$ at a value x is defined as the proportion of sample points less than or equal to x, expressed as

$$S_X(x) = \frac{1}{M}\sum_{i=1}^{M} I(x_i \leq x) \tag{3.8}$$

where $I(\cdot)$ is an indicator function that takes the value 1 if $x_i \leq x$, and 0 otherwise. As shown in Fig. 3.5, the empirical CDF is essentially a step function that exhibits a step increase of $1/M$ at each sample point. Now that we have the definition of the empirical CDF, the K-S distance can then be defined as the maximum distance between the empirical and theoretical CDF curves, expressed as

$$D_{KS} = \max_{1 \leq i \leq M} |S_X(x_i) - F_X(x_i)| \tag{3.9}$$

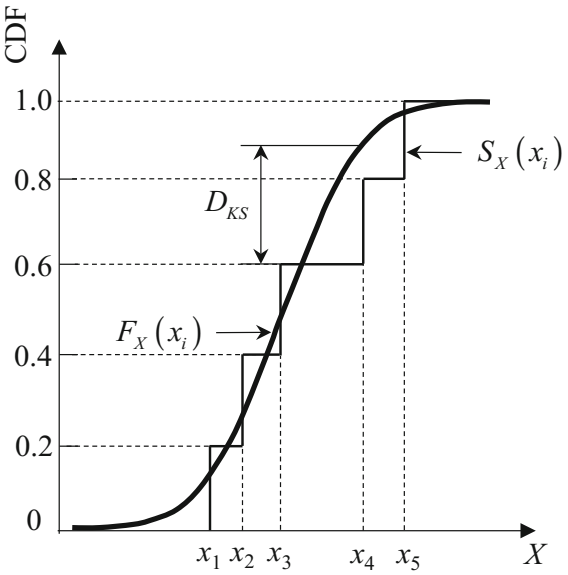


Fig. 3.5 K-S test statistic with five random samples

Here, the K-S distance D_{KS} is the test statistic whose probability distribution depends only on the sample size M. With a specified significance level α, we can obtain the critical value from a K-S look-up table. The null hypothesis in favor of the assumed distribution should be rejected if

$$D_{KS} > D_{KS,\alpha,M} \tag{3.10}$$

where $D_{KS,\alpha,M}$ is the K-S critical value with the sample size M and the significance level α.

A distinctive feature of the K-S test is that, unlike the chi-square test, it does not require a large sample size to make the assumption about the validity of the distribution of the test statistic. However, the K-S test is only applicable to continuous distributions. More details regarding these two statistical tests can be found in [2] and [3], respectively.

Example 3.6 Use the K-S goodness-of-fit test to determine whether the compressive strength data of the Aluminum–Lithium alloy in Table 3.1 can be represented by a normal distribution ($\alpha = 0.05$).

Solution

As mentioned in Example 3.5, the distributional parameters of the hypothesized normal distribution can be obtained from the samples as $\mu_X = 162.66$, $\sigma_X = 33.77$. Based on Eq. (3.8), we can compute the empirical CDF $S_X(x)$, which is plotted along with the theoretical CDF in Fig. 3.6.

We then execute the hypothesis testing step-by-step.

1. Specify the null and alternative hypotheses as

   H_0 The compressive strength data follow a normal distribution.
   H_1 The compressive strength data do not follow a normal distribution.

2. The significance level $\alpha = 0.05$.
3. The K-S test statistic is defined as

$$D_{KS} = \max_{1 \le i \le M} |S_X(x_i) - F_X(x_i)|$$

4. The critical value is $D_{KS,0.05,4} = 0.1521$.
5. The value of the test statistic can be computed as $D_{KS} = 0.0561$.
6. Since $D_{KS} = 0.0561 < D_{KS,0.05,4} = 0.1521$, we cannot reject H_0. Thus, we cannot find sufficient evidence for the assumption that the strength data do

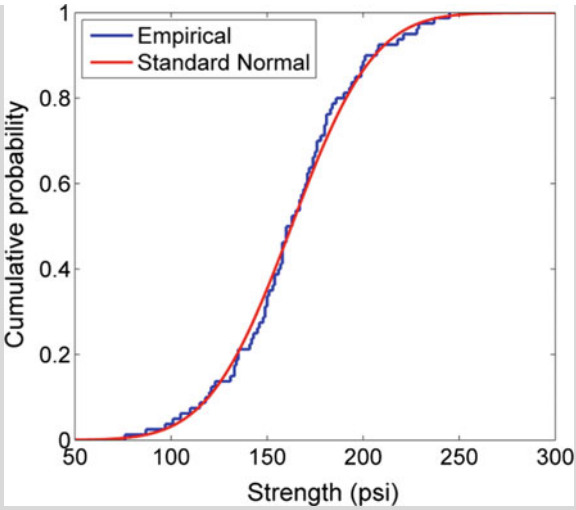


Fig. 3.6 Empirical and theoretical CDFs of compressive strength data

not follow a normal distribution. The p-value can be computed as $p = 0.9507$. The conclusion is the same as that reached with the chi-square test in Example 3.5.

3.2 Bayesian Statistics

We have discussed methods of statistical inference that view probability as normalized frequency and that rely on a set of samples randomly drawn from a population to estimate the probability distribution of the population. All of these methods have been developed using what statisticians would call a *frequentist approach*. In this section, we revisit parameter estimation using a different approach, namely *Bayesian inference*. This approach utilizes prior information in conjunction with the sample information for parameter estimation.

3.2.1 Bayes' Theorem

Bayes' theorem (also known as Bayes' rule or Bayes' law) is developed based on the definition of conditional probability. If A and B denote two stochastic events, $P(A|B)$ denotes the probability of A conditional on B. Bayes' theorem relates the conditional and marginal probabilities of A and B as

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} \tag{3.11}$$

The theorem simply states that a conditional probability of A given B is equal to the conditional probability of B given event A, multiplied by the marginal probability of A and divided by the marginal probability of B [4]. It is straightforward to derive Bayes' theorem in Eq. (3.11) based on the probability rules introduced in Chap.2. From the multiplication theorem, we know that $P(A,B) = P(A|B) \cdot P(\mathrm{B})$, and similarly, $P(B,A) = P(B|A) \cdot P(A)$. Since $P(A,B) = P(B,A)$, the right sides of these two equations equal each other, which gives $P(A|B) \cdot P(\mathrm{B}) = P(A|B)P(A)$. Moving $P(A)$ from the right-hand side to the left-hand side leaves us with Eq. (3.11).

The terms in Bayes' theorem are defined as follows:

- $P(A)$ is the prior probability or marginal probability of A. The prior probability can be treated as the subjective probability that expresses our belief prior to the occurrence of B. It is "prior" in the sense that it does not take into account any information about B.
- $P(B)$ is the marginal probability of B, and acts as a normalizing constant. Based on the total probability theorem, this quantity can be computed as the sum of the conditional probabilities of B under all possible (mutually exclusive) events A_k (included in a set $\mathbf{S}_A$) in the sample space. This can be mathematically expressed for a discrete sample space as:

$$P(B) = \sum_{A_i \in \mathbf{S}_A} P(B|A_i)P(A_i) \tag{3.12}$$

- $P(B|A)$ is the conditional probability of B given the prior information about A.
- $P(A|B)$ is the conditional probability of A given B. It is also called the posterior probability of A given B, since it depends upon the specified value of B. The posterior probability expresses our degree of belief on the probability of A after the occurrence of B.

Example 3.7 Assume there are three doors (D_1, D_2, and D_3); behind two of the doors are goats and behind the third door is a new car. The three doors are equally likely to have the car. Thus, the probability of getting the car by picking each door at the beginning of the game is simply 1/3. After you have picked a door, say D_1, instead of showing you what is behind that door, Monty opens another door, say D_2, which reveals a goat. At this point, Monty gives you the opportunity to switch the door from D_1 to D_3. What should you do, given that Monty is trying to let you get a goat?

Solution

The question is whether the probability of getting the car by picking the door D_1 is the same as that by picking D_3, or mathematically, whether

$P(D_1|D_{2,Monty}) = P(C|D_{2,Monty}) = 0.5$. The process of computing these two conditional probabilities is laid out as follows:

1. The prior probabilities read $P(D_1) = P(D_2) = P(D_3) = 1/3$.
2. We have some useful conditional probabilities $P(D_{2,Monty}|D_1) = 1/2$, $P(D_{2,Monty}|D_1) = 0$, and $P(D_{2,Monty}| D_3) = 1$.
3. We can compute the probabilities of joint events as $P(D_{2,Monty}, D_1) = 1/2 \times 1/3 = 1/6$, $P(D_{2,Monty}, D_1) = 0$, and $P(D_{2,Monty}, D_3) = 1 \times 1/3 = 1/3$.
4. Finally, with the denominator computed as $P(D_{2,Monty}) = 1/6 + 0 + 1/3 = 1/2$, we then find that $P(D_1|D_{2,Monty}) = 1/3$, $P(D_3|D_{2,Monty}) = 2/3$.

Since the conditional probability of revealing the car by picking the door D_3 is higher than that by picking the door D_1, you should switch to D_3 for a better chance of getting the car.

An example that looks at vibration-based fault diagnosis of rolling element bearings can help to make this theorem more concrete. Assume that the diagnostic test has a detection rate of 95%[2] and a false alarm rate of 30%.[3] Suppose a bearing unit, which has operated for approximately 2 years, is diagnosed as defective by the test. We would then like to know the probability that the bearing unit is in fact defective, given the positive test result, that is, P(defective|test+). In this example, event A (the true condition of the bearing unit) has two possible outcomes A_k: A_1 = defective and A_2 = not defective. Additionally, given the detection and false alarm rates, we know the conditional probabilities of a positive test result under these events: P(test +|defective) = 0.90 and P(test +|not defective) = 0.30. Using this conditional information, together with some prior information about the probability of becoming defective after 2 years of operation, Bayes' theorem offers a prescription for estimating the posterior probability that the bearing unit is defective. The prior information we need, $P(A)$ or P(defective), is the marginal probability that the bearing unit is defective, not knowing anything beyond the fact that the unit has operated in the field for around 2 years. The term "prior" simply means that the information exists prior to the diagnostic test. Historical data obtained from the same type of bearing suggest that the probability of becoming defective after 2 years of operation is approximately 0.20. With the prior and conditional information, we can now compute the posterior probability $P(A|B) \equiv P$(defective|test+), as

[2]A detection rate of 90% means that the test correctly detects defective cases 90% of the time.

[3]A false alarm rate of 30% means that, in cases where a bearing unit is not defective, the test produces an alarm, suggesting the detection of a fault 30% of the time.

$$P(\text{defective} \mid \text{test}+) = \frac{P(\text{test}+ \mid \text{defective}) \cdot P(\text{defective})}{P(\text{test}+ \mid \text{defective})P(\text{defective}) + P(\text{test}+ \mid \text{not defective})P(\text{not defective})} \tag{3.13}$$

Combining Eqs. (3.11) and (3.12) gives rise to the formula above. Filling in the prior and conditional probabilities yields:

$$P(\text{defective} \mid \text{test}+) = \frac{0.90 \cdot 0.20}{0.90 \cdot 0.20 + 0.30 \cdot 0.80} \approx 0.4286 \tag{3.14}$$

Thus, the probability that the unit is defective conditional on the positive test result is 0.4286. Since this probability is an estimated probability after the data (from the diagnostic test) is observed, it is termed a posterior probability.

An important contribution of Bayes' theorem is that it provides a rule on how to update or revise a *prior* belief to a *posterior* belief; this lies at the core of Bayesian inference. In the bearing example, the reliability or quality engineer may choose to repeat the diagnostic test (i.e., conduct a second test). After the second test, the engineer can use the posterior probability of being defective ($P = 0.429$) in Eq. (3.14) as the new prior P(defective). By doing so, the engineer has updated the prior probability of being defective to reflect the result of the first test. If the second test still gives a positive result, the updated posterior probability of being defective can be computed as:

$$P(\text{defective} \mid \text{test}+) = \frac{0.90 \cdot 0.0.4286}{0.90 \cdot 0.4286 + 0.30 \cdot 0.5714} \approx 0.6923 \tag{3.15}$$

With the second positive test result, we obtain an increase in the posterior probability from 0.4286 to 0.6923, which means that the added test result (positive) has increased our belief that the bearing unit might be defective. If the engineer continues to repeat the test and observe a positive result in each of the repeated tests, these repeated tests will yield the posterior probabilities shown in Table 3.5.

Bayesian statistics stems from the concept of repeating a test and recomputing the posterior probability of interest based on the results of the repeated testing. In the context of reliability analysis, the Bayesian approach begins with a prior probability of the system success event, and updates this prior probability with new data to obtain a posterior probability. The posterior probability can then be used as a prior probability in subsequent analysis. This may be an appropriate strategy for

Table 3.5 Posterior probabilities of being defective after repeated tests with positive results

Test number	1	2	3	4	5	6	7	8	9	10
Posterior probability	0.4286	0.6923	0.8710	0.9529	0.9838	0.9945	0.9982	0.9994	0.9998	0.9999

reliability analysis in some engineering applications. In this method, we continuously gather testing data to evaluate the reliability of an engineered system. We do not start with the same estimate each time we attempt to analyze the reliability, because previous testing data provide us with a priori information concerning the reliability of the system.

3.2.2 Bayesian Inference

In Bayesian statistics, the quantities in Bayes' theorem in Eq. (3.11) are typically expressed in the form of probability distributions rather than point probabilities. In the bearing example, we assume the prior probability of being defective after 2 years of operation is a point probability of exactly 0.20. However, the uncertainty in bearing parameters and design variables, as well as our imperfect knowledge of the reliability of the bearing population, give rise to a certain degree of unit-to-unit variation in this prior probability. Thus, it is unreasonable to use a precise point value to represent this probability. Instead, a probability distribution should be used for the prior defect probability to capture our uncertainty about its true value. Similarly, the point values for the conditional probabilities should be replaced with probability distributions to represent our uncertainty about their true values. The inclusion of the prior and conditional probability distributions eventually produces a posterior probability distribution that is no longer a single quantity. This posterior distribution combines the positive result observed from the diagnostic test with the prior probability distribution to produce an updated posterior distribution that expresses our knowledge of the probability that the bearing unit is defective.

Let us now express Bayes' theorem in terms of continuous probability distributions. Let X be a continuous random variable with a PDF $f(x, \theta)$, where θ is the distributional parameter (e.g., the mean and standard deviation of a normally distributed variable). The goal of Bayesian inference is to represent prior uncertainty of a distributional parameter with a probability distribution and to update this probability distribution with newly acquired data. The updating procedure yields a posterior probability distribution of the parameter. This perspective is in contrast with frequentist inference, which relies exclusively on the data as a whole, with no reference to prior information. From the Bayesian point of view, the parameter θ is interpreted as a realization of a random variable Θ with a PDF $f_{\Theta}(\theta)$. Based on Bayes' theorem, the posterior distribution of Θ, given a new observation x, can be expressed as

$$f_{\Theta|X}(\theta|x) = \frac{f_{X|\Theta}(x|\theta) \cdot f_{\Theta}(\theta)}{f_X(x)} \tag{3.16}$$

where $f_{\Theta|X}(\theta|x)$ is the posterior distribution of the parameter Θ, $f_{X|\Theta}(x|\theta)$ is the so-called sampling density of the observation x, $f_{\Theta}(\theta)$ is the prior distribution of Θ, and $f_X(x)$ is the marginal probability of x. The sampling density $f_{X|\Theta}(x|\theta)$ is

proportional to the likelihood function and only differs by a multiplicative constant that makes the integral of the density over its domain equal to 1. The marginal probability $f_X(x)$ can be treated as the marginal likelihood of x and acts as a normalizing constant to make the posterior density a proper PDF. Since the sampling density is proportional to the likelihood function, and the denominator simply acts as a scaling constant that makes the posterior density a proper PDF, Bayesian inference in Eq. (3.16) is often expressed as "posterior is proportional to likelihood times prior," or in a mathematical form as, "posterior $\propto$ likelihood $\times$ prior." Let us now consider a Bayesian normal inference model as one example to illustrate the inference procedure.

Example 3.8 Suppose that we have a set of random samples $\mathbf{x} = \{x_1, x_2, \ldots, x_M\}$ drawn from the normal PDF $f_X(x; \mu, \sigma)$ of a random variable X, where the mean μ is unknown and the standard deviation σ is known. Assume that the prior distribution of μ, $f_M(\mu)$, is a normal distribution with its mean u and variance τ^2. Determine the posterior distribution of μ, $f_{M|X}(\mu|\mathbf{x})$, given the random observations $\mathbf{x}$.

Solution
First, we compute the conditional probability of obtaining $\mathbf{x}$, given μ, as

$$f_{X|M}(\mathbf{x}|\mu) = \prod_{i=1}^{M} \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\left(\frac{x_i - \mu}{\sigma}\right)^2\right]$$
$$= \left(2\pi\sigma^2\right)^{-n/2} \exp\left[-\frac{1}{2\sigma^2}\sum_{i=1}^{M}(x_i - \mu)^2\right]$$

Next, we compute the numerator in Eq. (3.16) as

$$f_{X|M}(\mathbf{x}|\mu) f_M(\mu) = \left(2\pi\sigma^2\right)^{-n/2}\left(2\pi\tau^2\right)^{-1/2} \exp\left[-\frac{1}{2\sigma^2}\sum_{i=1}^{M}(x_i - \mu)^2 - \frac{1}{2\tau^2}(\mu - u)^2\right]$$
$$= K_1(x_1, \ldots, x_M, \sigma, u, \tau) \exp\left[-\left(\frac{M}{2\sigma^2} + \frac{1}{2\tau^2}\right)\mu^2 + \left(\frac{M\bar{x}}{\sigma^2} + \frac{u}{\tau^2}\right)\mu\right]$$

We then attempt set up a square inside the exponent as

$$f_{X,M}(\mathbf{x}, \mu) = K_2(x_1, \ldots, x_M, \sigma, u, \tau) \exp\left[-\frac{1}{2}\left(\frac{M}{\sigma^2} + \frac{1}{\tau^2}\right)\left(\mu - \frac{\frac{M\bar{x}}{\sigma^2} + \frac{u}{\tau^2}}{\frac{M}{\sigma^2} + \frac{1}{\tau^2}}\right)^2\right]$$
$$= K_2(x_1, \ldots, x_M, \sigma, u, \tau) \exp\left[-\frac{1}{2}\left(\frac{M}{\sigma^2} + \frac{1}{\tau^2}\right)\left(\mu - \frac{M\tau^2\bar{x} + \sigma^2 u}{M\tau^2 + \sigma^2}\right)^2\right]$$

Since, as previously mentioned, the denominator $f_{\mathbf{X}}(x_1, x_2, \ldots, x_M)$ in Eq. (3.16) is a scaling constant and does not vary with μ, we then derive the posterior distribution of μ as

$$f_{M|X}(\mu|\mathbf{x}) = K_3(x_1, \ldots, x_M, \sigma, u, \tau) \exp\left[-\frac{1}{2}\left(\frac{M}{\sigma^2} + \frac{1}{\tau^2}\right)\left(\mu - \frac{M\tau^2\bar{x} + \sigma^2 u}{M\tau^2 + \sigma^2}\right)^2\right]$$

Clearly, this is a normal distribution with the mean and variance as

$$\hat{u} = \frac{M\tau^2\bar{x} + \sigma^2 u}{M\tau^2 + \sigma^2}, \ \hat{\tau}^2 = \left(\frac{M}{\sigma^2} + \frac{1}{\tau^2}\right)^{-1} = \frac{\sigma^2\tau^2}{M\tau^2 + \sigma^2}$$

Observe that the Bayesian posterior estimate of μ is essentially a weighted-sum of the sample mean $\bar{x}$ and the prior mean u. In contrast, the maximum likelihood estimator is simply the sample mean (see Example 3.3). Note also that, as the number of samples M approaches infinity, the Bayesian estimate becomes equal to the maximum likelihood estimator, since the sample data tend to have a predominant influence over the prior information. However, in cases where there is a small sample size, the prior information plays an important role, especially when the prior variance τ^2 is small (or, in other words, when the prior information is precise).

As can be observed in Example 3.8, Bayesian inference and the MLE provide essentially the same estimate if we have an extremely large sample size. In engineering practice, however, we often have very limited sample data due to the expense and time demands associated with obtaining the data. In such cases, the MLE may not give an accurate or even reasonable estimate. In contrast, Bayesian inference gives a better estimate if we assume a reasonable prior distribution. The term "reasonable" means that the prior assumption is at least consistent with the underlying distribution of the population. If there is no such consistency, Bayesian inference may give an erroneous estimate due to the misleading prior information.

Another important observation we can make from Example 3.8 is that the posterior distribution shares the same form (i.e., normal distribution) with the prior distribution. In such cases, we say that the prior is *conjugate* to the likelihood. If we have a conjugate prior, the posterior distribution can be obtained in an explicit form. Conjugacy is desirable in Bayesian inference, because using conjugate priors/likelihoods with known forms significantly eases the evaluation of the posterior probability. Looking back at Example 3.8, we note that the normal (or Gaussian) family is conjugate to itself (or self-conjugate): if the likelihood function is normal, choosing a normal prior ensures that the posterior is also normal. Other conjugate Bayesian inference models include the binomial inference, exponential inference, and Poisson inference. Among these inference models, the binomial inference is the

most widely used. Consider a binomial sequence of n experimental trials with x occurrences of an outcome whose probability of occurrence p_0 is unknown. Assume a beta prior $B(a, b)$ for the unknown binomial probability p_0, expressed as

$$f_{P_0}(p_0) = \frac{\Gamma(a,b)}{\Gamma(a)\Gamma(b)} p_0^{a-1}(1-p_0)^{b-1} \tag{3.17}$$

Based on the form of the binomial distribution, the likelihood function can be expressed as

$$L(x; n, p_0) = C(n,x)p_0^x(1-p_0)^{n-x} \propto p_0^x(1-p_0)^{n-x} \tag{3.18}$$

We can then obtain the posterior distribution of p_0, expressed as

$$f_{P_0|X}(p_0|x) = \frac{\Gamma(x+a, n+b-x)}{\Gamma(x+a)\Gamma(n+b-x)} p_0^{x+a-1}(1-p_0)^{n+b-x-1} \tag{3.19}$$

The posterior distribution follows the same form (i.e., the beta distribution) as the prior distribution, which suggests that the beta prior is conjugate to the binomial likelihood. In Example 3.9, we demonstrate this conjugate inference model with a simple reliability analysis problem.

Example 3.9 Suppose we want to quantify the reliability of a product by conducting a sequence of 10 repeated tests. The product passes 8 of these tests and fails at the other two. We assume a beta prior $B(4, 4)$ for the probability of success (or reliability) p_0 in each test. Compute the posterior distribution of p_0, given the reliability testing data.

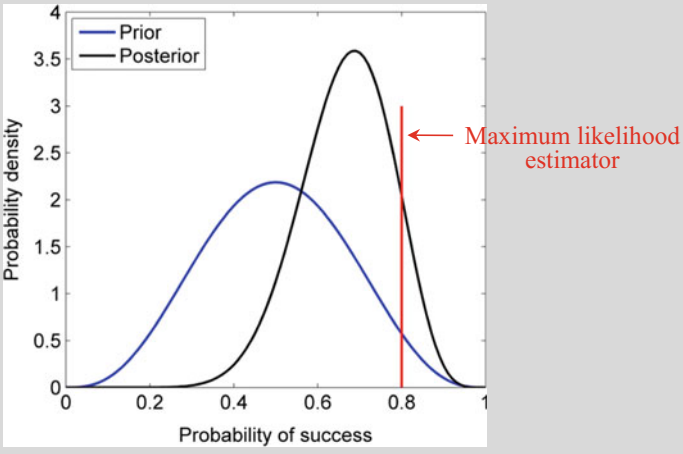


Fig. 3.7 Prior and posterior distributions for Example 3.9

Solution

The parameters in this example take the following values: $a = 4$, $b = 4$, $x = 8$, $n = 10$. The posterior distribution of p_0 can be obtained, according to Eq. (3.19), as $B(x + a, n + b - x)$, or $B(12, 6)$. The prior and posterior distributions of p_0 are plotted in Fig. 3.7. The figure shows that the posterior distribution results—from a combination of the prior information and the testing data (evidence)—lie between the prior distribution and the maximum likelihood estimator (which exclusively relies on the testing data).

In many engineering problems, the conjugacy condition does not hold, and explicit solutions of posterior distributions cannot be readily obtained through simple mathematical manipulations. In such cases, we are left to draw random samples from posterior distributions to approximate the distributions. A commonly used simulation method for drawing samples from a posterior distribution is referred to as *Markov chain Monte Carlo* (*MCMC*), in which two important sampling techniques, namely the Metropolis–Hastings algorithm and Gibbs sampling, are often used. An in-depth discussion of these techniques is beyond the scope of this book. Readers are recommended to refer to [3] for detailed information.

In Bayesian updating, Bayesian inference expressed in Eq. (3.16) is often performed iteratively over time. In other words, after observing the initial set of testing data, Bayesian inference is performed to obtain the resulting posterior probability, and this posterior probability can then be treated as a prior probability for computing a new posterior probability as the next set of testing data becomes available. Figure 3.8 shows the overall procedure of Bayesian updating for a distributional parameter Θ. In each updating iteration, Bayesian inference is performed with the most "up-to-date" prior information and the most recent data. The posterior density of Θ after one iteration becomes the prior density for the next iteration. The capability of continuous updating is an attractive feature of Bayesian statistics that is useful for parameter estimation with evolving data sets or random variables.

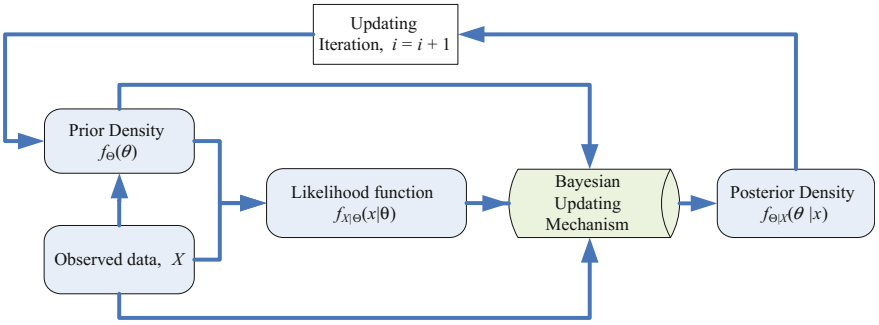


Fig. 3.8 Flowchart of Bayesian updating

3.3 Exercises

3.1 Recall Problem 2.2 in Chap. 2. Answer the following question based on the 30 sample data obtained from the fatigue tests described in that problem.

(1) Use normal, Weibull, and lognormal distributions. Find the most suitable parameters of the three distributions for the fatigue ductility coefficient (ε_f') and exponent (c) using the MLE method.
(2) Find the most suitable distribution for the data set (ε_f', c) using the chi-square goodness-of-fit test.
(3) Verify the results using the graphical methods described in the chapter (a histogram and a probability plot).

3.2 Recall Problem 2.3 in Chap. 2. Answer the following question based on the 50 sample data obtained from LCD module dent tests.

(1) Use normal, Weibull, and uniform distributions. Find the most suitable parameters of the three distributions for the failure displacement (d_f) and failure force (F_f) using the MLE method.
(2) Find the most suitable distributions for the data set (d_f, F_f) using the chi-square goodness-of-fit test.
(3) Verify the results using probability plots.

3.3 Suppose that we are interested in identifying the probability distribution for the number of cars passing through the main gate of the University of Maryland per minute. The data have been collected by a group of students and are shown in Table 3.6.

Table 3.6 Observed frequency distribution for the number of cars

Vehicles per minute	Observed frequency	Vehicles per minute	Observed frequency
40	14	53	102
41	24	54	96
42	57	55	90
43	111	56	81
44	194	57	73
45	256	58	64
46	296	59	61
47	378	60	59
48	250	61	50
49	185	62	42
50	171	63	29
51	150	64	18
52	110	65	15

(1) Determine whether or not a Poisson distribution is an appropriate probability distribution for the data ($\alpha = 0.05$).
(2) Verify the results using an exponential probability plot.

3.4 Suppose it is desired to estimate the failure rate of an electronic component. For this purpose, a test has been performed on 100 components; time-to-failure (TTF) data are summarized in Table 3.7. Answer the following questions:

(1) Construct a histogram of TTF.
(2) Find a probability distribution model $f_T(t)$ and its parameters for the TTF data (use the MLE method for parameter estimation and the K-S goodness-of-fit test for distribution selection).
(3) Attempt to update the TTF mean value (θ) with aggregation of 100 TTF data using Bayesian inference. Assume that the TTF follows a normal distribution with standard deviation of $\sigma = 315.16$ and that the prior distribution $f_\Theta(\theta)$ of θ follows a normal distribution with the mean $u = 1750.0$ and standard deviation $\tau = 500$.

Table 3.7 Data for 100 electronic components' time-to-failure (TTF) [minutes]

1703.2	1071.4	2225.8	1826.5	1131.0	2068.9	1573.5	1522.1	1490.7	2226.6
1481.1	2065.1	1880.9	2290.9	1786.4	1867.2	1859.1	1907.5	1791.8	1871.0
1990.4	2024.1	1688.6	1962.7	2191.7	1841.0	1814.1	1918.1	2237.5	1396.8
1692.8	707.2	2101.3	2165.4	1975.2	1961.6	2116.7	1373.0	1798.8	2248.4
1872.3	1597.8	1865.1	742.8	1436.7	1380.8	2258.2	1960.0	2182.8	1772.7
2003.6	1589.4	1988.3	1874.9	1859.0	2051.9	1763.0	1854.6	1974.7	2289.9
1945.7	1774.8	1579.6	1430.5	1855.0	1757.9	1029.3	1707.2	1864.7	1964.8
1719.4	1565.2	1736.8	1759.4	1939.4	2065.7	2258.5	2292.8	1452.5	1692.2
2120.7	1934.8	999.4	1919.9	2162.4	2094.9	2158.2	1884.2	1748.7	2260.3
1040.8	1535.0	1283.4	2267.7	2100.3	2007.9	2499.8	1902.9	1599.6	1567.5

Table 3.8 Data for 100 cutting tools' time-to-failure (TTF) [minutes]

839.3	838.8	959.3	950.5	873.9	948.3	859.8	898.7	903.6	1031.6
852.4	891.5	965.0	856.1	739.0	895.3	916.8	921.7	1093.3	863.7
965.0	927.1	888.6	918.4	1025.4	811.3	960.4	826.9	875.2	980.7
905.6	982.7	892.0	928.4	918.7	1071.5	824.1	743.9	915.0	1064.0
753.3	787.6	836.0	941.7	951.7	791.8	949.1	874.6	975.8	948.2
1046.2	817.6	939.5	850.7	809.7	936.6	1040.8	947.1	857.9	901.4
952.3	848.5	999.0	1007.7	915.8	931.3	907.1	966.3	810.8	771.2
776.4	913.7	1003.7	978.0	1035.7	1065.8	1107.6	766.0	772.6	999.1
870.0	1007.6	877.7	709.8	958.1	874.1	846.0	746.2	994.0	954.7
916.6	1054.9	917.4	812.6	963.4	1017.5	1122.9	865.1	938.8	837.5

3.5 Suppose it is desired to estimate the failure rate of a machine cutting tool. A test can be performed to estimate its failure rate. The failure times in minutes are shown in Table 3.8. Answer the following questions:

(1) Construct a histogram of TTF.
(2) Find a probability distribution model $f_T(t)$ and its parameters for the TTF data (use the MLE method for parameter estimation and the K-S goodness-of-fit test for distribution selection).
(3) Attempt to update the TTF mean value (θ) with aggregation of 100 TTF data using Bayesian inference. Assume that the TTF follows a normal distribution with a standard deviation of $\sigma = 80$ and that the prior distribution $f_\Theta(\theta)$ of θ follows a normal distribution with the mean $u = 1000$ and the standard deviation $\tau = 100$.

3.6 The TTF of a machine has an exponential distribution with parameter λ. Assume that the prior distribution for λ is exponential with a mean of 100 h. We have five observed TTF from five machines; the average TTF is 1200 h.

(1) Compute the posterior distribution of λ based on our observations.
(2) Based on the posterior distribution, what proportion of the machines will fail after 1000 h?

References

1. Lenth, R. V. (2001). Some practical guidelines for effective sample size determination. *The American Statistician, 55*(3), 187–193.
2. http://www.itl.nist.gov/div898/handbook/eda/section3/eda35e.htm
3. http://www.itl.nist.gov/div898/handbook/eda/section3/eda35g.htm
4. Lynch, S. (2010). *Introduction to applied Bayesian statistics and estimation for social scientists*. New York, NY: Springer.

Chapter 4
Fundamentals of Reliability Analysis

Failures of engineered systems (e.g., vehicle, aircraft, and material) lead to significant maintenance/quality-control costs, human injuries, and fatalities. Examples of such system failures can be found in various engineering fields: the Chernobyl disaster in Russia (1986), the collapse of the I-35 W Mississippi River Bridge in the U.S. (2007), the explosion of a compressed natural gas (CNG) bus in the Republic of Korea (2010), and the lithium-ion battery fire/smoke on Boeing 787 Dreamliners in the U.S. and Japan (2013). Many system failures can be traced back to various difficulties in evaluating and designing complex systems under highly uncertain manufacturing and operational conditions. One of the greatest challenges in design of an engineered system is to ensure high reliability and maintainability of the system during its life-cycle. Our attempt to address this challenge begins with the discussion of the fundamentals of reliability analysis. This discussion will be separately conducted for time-independent and time-dependent reliability analyses, with an aim to facilitate more in-depth discussions in later chapters.

4.1 Definition of Reliability

The formal definition of reliability is the probability of an engineered system to perform its required function under prescribed conditions (for a specified period). The use of parentheses in this definition indicates the existence of two types of reliabilities, namely time-independent reliability and time-dependent reliability. *Time-independent reliability* is typically used when designing an engineered system for the very beginning of the system's life-cycle (i.e., it generally does not consider health degradation that occurs during system operation). *Time-dependent reliability* is often used to assess the operational reliability of an engineered system through the system's life-cycle (i.e., time-dependent reliability analysis takes into account health degradation that occurs during system operation).

C. Hu et al., *Engineering Design under Uncertainty and Health Prognostics*,
Springer Series in Reliability Engineering,
https://doi.org/10.1007/978-3-319-92574-5_4

4.1.1 Time-Independent Reliability

Time-independent reliability is defined as the probability that the actual performance of an engineered system meets the required or specified design performance under various uncertainty sources (e.g., material properties, geometric tolerances, loading conditions). This definition is often used in the design of civil structural systems, mechanical systems, and aerospace systems. In order to formulate the time-independent reliability in a mathematical framework, random variables are often used to model uncertainty sources. The time-independent reliability can then be formulated as

$$R(\mathbf{X}) = P(G(\mathbf{X}) \leq 0) = 1 - P(G(\mathbf{X}) > 0) \tag{4.1}$$

where the random vector $\mathbf{X} = (X_1, X_2, \ldots, X_N)^{\mathrm{T}}$ models uncertainty sources, such as material properties, geometric tolerances, and loading conditions; $G(\mathbf{X})$ is a system performance function, and the system success event is $E_{sys} = \{G(\mathbf{X}) \leq 0\}$. The uncertainty of the vector $\mathbf{X}$ further propagates and leads to the uncertainty in the system performance function G. In reliability analysis, equating the system performance function G to zero, i.e., $G = 0$, gives us the so-called limit-state function, which separates the safe region $G(\mathbf{X}) \leq 0$ from the failure region $G(\mathbf{X}) > 0$. Depending on the specific problems, a wide variety of system performance functions can be defined to formulate time-independent reliabilities. The most well-known example is the safety margin between the strength and load of an engineered system, which will be discussed in Sect. 4.2. The concept of time-independent reliability analysis in a two-dimensional case is illustrated in Fig. 4.1. The dashed lines represent the contours of the joint PDF of the two random variables X_1 (manufacturing tolerance) and X_2 (operational factors). The basic idea of reliability analysis is to compute the probability that $\mathbf{X}$ is located in the safe region $\{G \leq 0\}$.

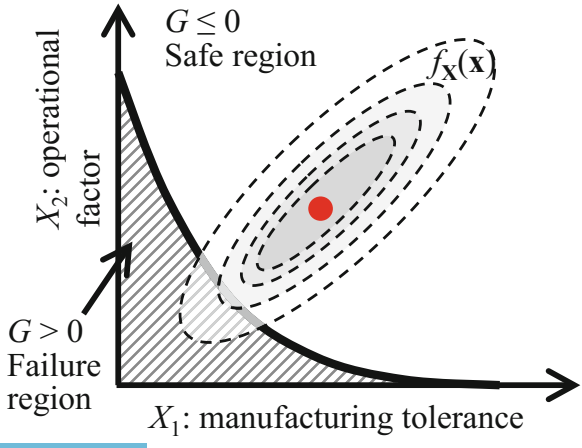


Fig. 4.1 Schematic of time-independent reliability analysis in a two-dimensional space. Reprinted (adapted) with permission from Ref. [1]

4.1.2 Time-Dependent Reliability

Unlike time-independent reliability, time-dependent reliability takes into account the life-cycle health degradation of an engineered system and is defined according to the failure-free operating time of the system. Formally, time-dependent reliability can be defined as the probability that the actual life (or failure-free operating time) of an engineered system exceeds the required or specified design life. For any specific time t, the time-dependent reliability can be formulated as

$$R_T(t) = P(T(\mathbf{X}) > t) = 1 - P(T(\mathbf{X}) \leq t) = 1 - F_T(t) \tag{4.2}$$

where $\mathbf{X}$ is the random vector representing engineering uncertainty factors, and the time-to-failure (TTF) $T(\mathbf{X})$ of the system is defined as the time when the system's performance function (or health condition) is worse than a predefined critical value. The equation above indicates that time-dependent reliability analysis requires modeling of the underlying TTF distribution. This can be done by using a wide variety of parametric probability distributions; the most commonly used are exponential distribution, Weibull distribution, and normal distribution. The reliability functions under these distributions will be discussed in Sect. 4.3.

4.2 Reliability Function (Time-Independent)

Consider the most well-known performance function, i.e., the safety margin between the strength S of an engineered system and the load L on this system. This performance function takes the following form

$$G = L - S \tag{4.3}$$

The strength S and load L are random in nature and their randomness can be characterized by two PDFs $f_S(s)$ and $f_L(l)$, respectively. Under the assumption of normal distributions, these two PDFs are plotted in Fig. 4.2a. The probability of failure depends on the intersection (shaped) area of the two PDFs, where the load on the system might exceed its strength. Let μ_S and μ_L, respectively, denote the means of S and L; let σ_S and σ_L, respectively, denote the standard deviations of S and L. We can then compute the mean and standard deviation of the normally distributed performance function G as

$$\begin{aligned} \mu_G &= \mu_L - \mu_S \\ \sigma_G &= \sqrt{\sigma_L^2 + \sigma_S^2 - 2\rho_{LS}\sigma_L\sigma_S} \end{aligned} \tag{4.4}$$

Under the assumption of statistical independence between S and L, we can compute the reliability based on the standard normal CDF as

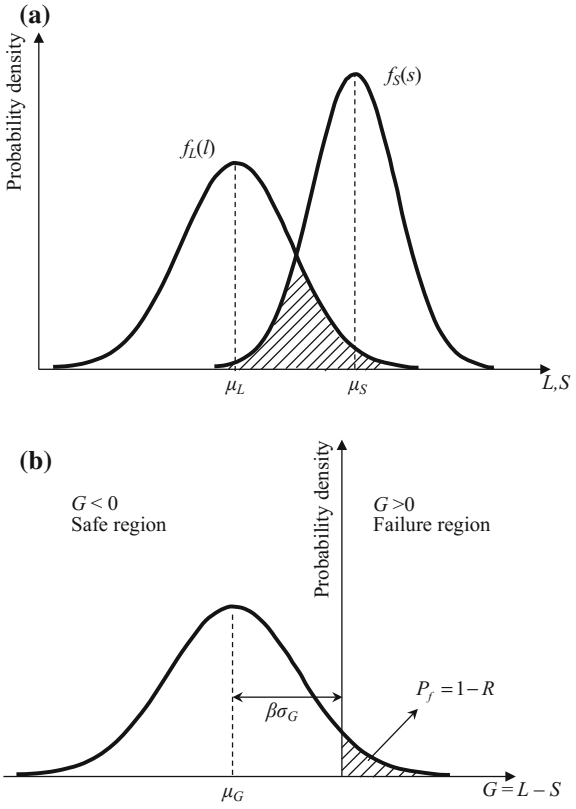


Fig. 4.2 PDFs of load and strength (**a**) and of the system performance function G (**b**)

$$R = P(G \le 0) = P\left(\frac{G - (\mu_L - \mu_S)}{\sqrt{\sigma_L^2 + \sigma_S^2}} \le \frac{0 - (\mu_L - \mu_S)}{\sqrt{\sigma_L^2 + \sigma_S^2}}\right) = P\left(Z \le \frac{\mu_S - \mu_L}{\sqrt{\sigma_L^2 + \sigma_S^2}}\right) = \Phi\left(\frac{\mu_S - \mu_L}{\sqrt{\sigma_L^2 + \sigma_S^2}}\right) \tag{4.5}$$

Based on the equation above and our intuition, we can reduce the intersection area and thus increase the reliability through either of the following two strategies:

- Increase the relative distance between the two means: As the relative distance $\mu_S - \mu_L$ between the means increases, the numerator of the standard normal value in Eq. (4.5) increases. Accordingly, the standard normal CDF value, or reliability, increases.
- Decrease the variances of two variables S and L: Reduction in either variance leads to a reduction in the denominator of the standard normal value in Eq. (4.5). The decrease of the denominator results in an increase of the standard normal CDF value, or reliability.

The PDF of the system performance function is plotted in Fig. 4.2b, where the probability of failure, or one minus reliability, is indicated by the shaded area. We note that the distance between the mean performance function (safety margin) and the limit state $G = 0$ is equal to the standard deviation σ_G multiplied by a factor β. In reliability analysis, this factor is named the reliability index, and is expressed as

$$\beta = \Phi^{-1}(R) = \frac{\mu_S - \mu_L}{\sqrt{\sigma_L^2 + \sigma_S^2}} \tag{4.6}$$

The reliability index provides an alternative measure of reliability from the perspective of a standard normal distribution. In Chap. 5, we will see that the reliability index is a very useful measure for reliability assessment.

Example 4.1 Suppose that the coefficients of variation for the load and strength are $\rho_L = 0.2$ and $\rho_S = 0.1$, respectively. Assume both variables follow normal distributions. Determine the ratio of means μ_S/μ_L required to achieve a reliability no less than 0.99.

Solution

According to the Appendix in Chap. 5, the reliability index corresponding to a 0.99 reliability is $\beta = 2.3263$. The reliability index can be mathematically expressed as

$$\beta = \frac{\mu_S - \mu_L}{\sqrt{\sigma_L^2 + \sigma_S^2}} = \frac{\mu_S/\mu_L - 1}{\sqrt{\rho_L^2 + (\mu_S/\mu_L)^2 \rho_S^2}}$$

By substituting $\kappa = \mu_S/\mu_L$ (note that $\kappa > 1$), we find

$$\beta = \frac{\kappa - 1}{\sqrt{\rho_L^2 + \kappa^2 \rho_S^2}}$$

The equation above can be written as a quadratic equation with κ as the only unknown, expressed as

$$\left(1 - \beta^2 \rho_S^2\right)\kappa^2 - 2\kappa + \left(1 - \beta^2 \rho_L^2\right) = 0$$

Solving this equation gives us two solutions

$$\begin{aligned}\kappa &= \frac{2 \pm \left[4 - 4\left(1 - \beta^2 \rho_S^2\right)\left(1 - \beta^2 \rho_L^2\right)\right]^{1/2}}{2\left(1 - \beta^2 \rho_S^2\right)} \\ &= \frac{2 \pm [4 - 4(1 - 2.3263^2 0.1^2)(1 - 2.3263^2 0.2^2)]^{1/2}}{2(1 - 2.3263^2 0.1^2)} \\ &= 0.5193 \text{ or } 1.5951\end{aligned}$$

Since $0.5193 < 1$ it does not satisfy $\kappa > 1$; thus, the final solution is 1.5951.

Let us next consider a general case where the normality assumption may not hold. To calculate the reliability, we can perform a two-dimensional integration as

$$R = P(L - S \leq 0) = \int_0^{+\infty} \left(\int_s^{+\infty} f_L(l)dl \right) f_S(s)ds \tag{4.7}$$

Observe that the integration above is performed over the safe region Ω^S that is defined as $\Omega^S = \{\mathbf{x} = (L, S)^T: G(\mathbf{x}) \leq 0\}$. The integration above can thus be equivalently expressed as

$$R = \iint_{\Omega^S} f_L(l) f_S(s) dl ds = \iint_{\Omega^S} f_X(\mathbf{x}) d\mathbf{x} \tag{4.8}$$

where $f_{\mathbf{X}}(\mathbf{x})$ denotes the joint PDF of the vector $\mathbf{X}$. Now let us further generalize this calculation to any multi-dimensional random vector whose joint PDF may or may not be separable. In this general case, the time-independent reliability can be formulated as a multi-dimensional integration of the joint PDF over a safe region

$$R = \int \cdots \int_{\Omega^S} f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} \tag{4.9}$$

where $f_{\mathbf{X}}(\mathbf{x})$ denotes the joint PDF of this random vector, and the safe region Ω^S is defined as $\Omega^S = \{\mathbf{x} : G(\mathbf{x}) \leq 0\}$.

In engineering practice, however, it is extremely difficult, if not impossible, to perform multi-dimensional numerical integration when the performance function involves a large number of random input variables. The search for efficient computational procedures to perform this multi-dimensional integration has resulted in a variety of numerical and simulation methods, such as first- and second-order reliability methods (FORM/SORM), direct or smart Monte Carlo simulation (MCS), the dimension reduction (DR) method, the stochastic spectral method, and the stochastic collocation method. These methods will be introduced in Chap. 5.

Example 4.2 Given the joint density function of two random variables X and Y

$$f_{XY}(x, y) = \begin{cases} \frac{6-x-y}{8}, & \text{if } 0 < x < 2,\ 2 < y < 4 \\ 0, & \text{otherwise} \end{cases}$$

and the performance function of an engineered system $G(X, Y) = 2X - Y$, compute the reliability R of this system.

Solution

Here, the safe region Ω^S can be defined as $\Omega^S = \{(X, Y): G(X, Y) \leq 0\} = \{(X, Y): 2X \leq Y\}$. Given the joint density function, this is equivalent to $\Omega^S = [\{X: 0 < X < 1\} \cap \{Y: 2 < Y < 4\}] \cup [\{X: 1 \leq X < 2\} \cap \{Y: Y \geq 2X\}]$. Reliability can be computed by performing a two-dimensional integration over this safe region Ω^S, expressed as

$$\begin{aligned} R &= \int_0^1 \int_2^4 f_{XY}(x,y)dydx + \int_1^2 \int_{2x}^4 f_{XY}(x,y)dydx \\ &= \int_0^1 \int_2^4 \frac{6-x-y}{8}dydx + \int_1^2 \int_{2x}^4 \frac{6-x-y}{8}dydx \\ &= \int_0^1 \frac{3-x}{4}dx + \int_1^2 \frac{(x-2)^2}{2}dx = \frac{3}{4} - \frac{x^2}{8}\Big|_0^1 + \frac{(x-2)^3}{6}\Big|_1^2 \\ &= \frac{19}{24} \end{aligned}$$

4.3 Reliability Function (Time-Dependent)

Most electrical, electronic, and mechanical systems deteriorate during use as a result of elevated operating temperatures, chemical changes, mechanical wear, fatigue, overloading, and for a variety of other reasons. Failures of such systems may eventually occur as a result of this deterioration. Time-dependent reliability considers the life-cycle health degradation of an engineered system and is defined based on the failure-free operating time of the system.

4.3.1 Reliability and Failure Rate Functions

Given the TTF distribution $f_T(t)$ of an engineered system, time-dependent reliability can be formulated as

$$R_T(t) = P(T(\mathbf{X}) > t) = \int_t^{+\infty} f_T(t)dt = 1 - F_T(t) \tag{4.10}$$

The expected TTF, known as the mean-time-to-failure (MTTF), can be defined as

$$MTTF = E[T] = \int_0^\infty \tau f_T(\tau) d\tau \tag{4.11}$$

In a life test of replaceable units, the mean of all the sample units approaches the MTTF as the number of tested units approaches infinity. By equating $f_T(t)$ to the negative derivative of $R_T(t)$, we can derive another useful formula of MTTF as

$$\begin{aligned} MTTF &= -\int_0^\infty \tau \frac{\partial R_T(\tau)}{\partial \tau} d\tau = -[tR_T(t)]_0^\infty + \int_0^\infty R(\tau) d\tau \\ &= \int_0^\infty R_T(\tau) d\tau \end{aligned} \tag{4.12}$$

Insights can be gained into failure mechanisms by examining the behavior of the so-called *failure rate*. The failure rate, denoted by $h(t)$, can be derived from the reliability and the TTF distribution. Let $h(t)\Delta t$ be the conditional probability that the system will fail at some time $t < T < t + \Delta t$ given that it has not yet failed at $T = t$. This conditional probability can be expressed as

$$\begin{aligned} h(t)\Delta t = P\{T < t + \Delta t | T > t\} &= \frac{P\{(T > t) \cap (T < t + \Delta t)\}}{P\{T > t\}} \\ = \frac{P\{t < T < t + \Delta t\}}{R(t)} &= \frac{f_T(t)\Delta t}{R(t)} \end{aligned} \tag{4.13}$$

which gives the failure rate (or hazard) function as

$$h(t) = \frac{f_T(t)}{R(t)} \tag{4.14}$$

The failure rate of a large population of statistically independent system units can often be represented by a bathtub curve (see Fig. 4.3). It consists of three distinct parts:

- Burn-in failures: The first part starts with a high "infant mortality" failure rate, which rapidly decreases over time. These failures can be attributed to manufacturing defects in materials and components.
- Constant failure rate: The second part is the constant failure rate, known as random failures.
- Wear-out failures: The third part shows an increasing failure rate over time due to health degradation (e.g., corrosion, fatigue, wear) of the system units.

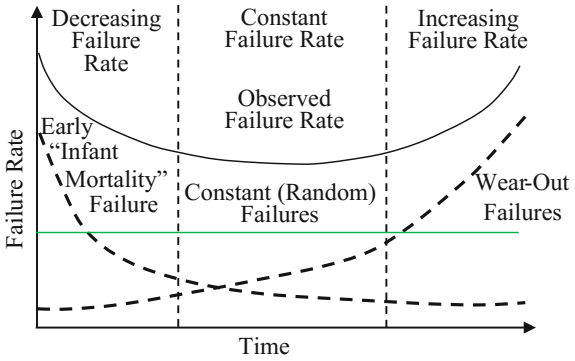


Fig. 4.3 Bathtub curve for the failure rate (or hazard) function

The bathtub shape of the failure rate curve can be attributed to the existence of both defective and non-defective units in the population. In the first part, the failure rate curve is dominated by the initially defective units (burn-in failures) whose failure rate rapidly decreases until these defective units fail. In the second part, the failure rate remains at a low constant since initially defective units have failed and the remaining units are not yet experiencing wear-out failures. In the third part, the non-defective units become dominant but the units begin to wear out over time, leading to an increasing failure rate.

There are a handful of parametric models that have successfully served as population models for failure times arising from a wide range of products and failure mechanisms. Sometimes there are probabilistic arguments based on the physics of the failure mode that tend to justify the choice of the model. Other times, the model is used solely because of its empirical success in fitting actual failure data. The next section discusses some popular parametric models.

4.3.2 Parametric Time-to-Failure Distributions

This section presents the most commonly used probability distributions for modeling the TTF distribution.

Exponential Distribution

With the assumption of a constant failure rate, the TTF follows an exponential distribution with only one unknown parameter, expressed as

$$f_T(t) = \lambda e^{-\lambda t} \tag{4.15}$$

for $t \geq 0$. The exponential distribution is the simplest among all life distributions. The reliability function can be easily obtained as

$$R_T(t) = e^{-\lambda t} \tag{4.16}$$

The MTTF is the mean value of the exponential random variable t, i.e., MTTF = $1/\lambda$. As shown in Eq. 4.12, the MTTF can also be computed by integrating the reliability function from zero to infinity as

$$\begin{aligned} MTTF &= \int_0^{+\infty} R_T(\tau)d\tau = \int_0^{+\infty} e^{-\lambda\tau}d\tau \\ &= -\frac{1}{\lambda}e^{-\lambda\tau}\Big|_0^{+\infty} = \frac{1}{\lambda} \end{aligned} \tag{4.17}$$

The failure rate (or hazard) function can be easily computed as $h(t) = 1/\lambda$. The reliability functions and failure rate functions with different parameters are graphically compared in Fig. 4.4. As λ increases, the reliability decreases more rapidly over time.

Example 4.3 Suppose the TTF of an engineered system follows an exponential distribution with the PDF $f_T(t) = \lambda\exp(-\lambda t)$, for $t \geq 0$, and $f_T(t) = 0$, for t < 0. Find the conditional probability that the system will fail after time $a + b$, given that the system does not fail before time a.

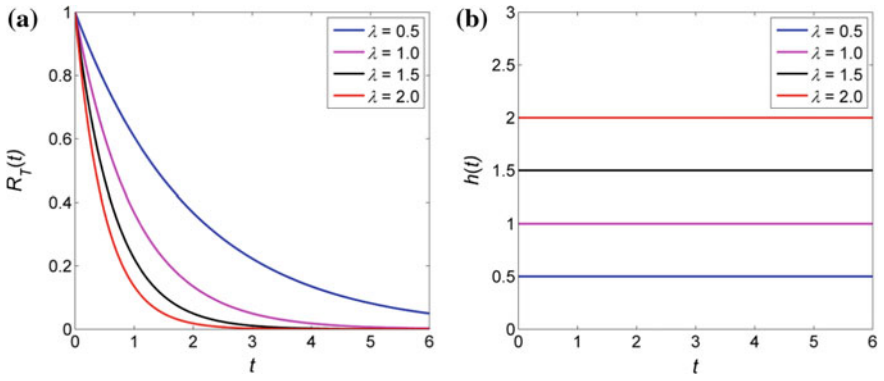


Fig. 4.4 Reliability functions (**a**) and failure rate functions (**b**) for exponential distribution

Solution
This conditional probability can be computed as

$$P(t > a+b|t > a) = \frac{P(t > a+b)}{P(t > a)} = \frac{R_T(t = a+b)}{R_T(t = a)}$$
$$= \frac{e^{-\lambda(a+b)}}{e^{-\lambda a}} = e^{-\lambda b}$$

Note that this conditional reliability equals the unconditional reliability at time t = b, i.e., $R_T(t = b)$. This suggests that, if the TTF of an engineered system follows an exponential distribution, the future failure behavior of this system does not depend on how long it has already reliably operated. This phenomenon is known as the memoryless property of the exponential distribution.

Weibull Distribution

A generalization of the exponential distribution is the Weibull distribution, which can model a constant, decreasing, or increasing failure rate function. The density of a Weibull TTF distribution can be expressed as

$$f_T(t) = \frac{k}{\lambda}\left(\frac{t}{\lambda}\right)^{k-1} e^{-\left(\frac{t}{\lambda}\right)^k} \tag{4.18}$$

Here, $t > 0$, $\lambda > 0$ is the scale parameter, and $k > 0$ is the shape parameter. The reliability function can be computed by subtracting the CDF from 1 as

$$R_T(t) = 1 - \int_0^t f_T(\tau)d\tau = e^{-\left(\frac{t}{\lambda}\right)^k} \tag{4.19}$$

The MTTF is the mean value of the exponential random variable t, expressed as

$$MTTF = \frac{\Gamma(1+1/k)}{\lambda} \tag{4.20}$$

The failure rate function can be easily computed as

$$h(t) = \frac{k}{\lambda}\left(\frac{t}{\lambda}\right)^{k-1} \tag{4.21}$$

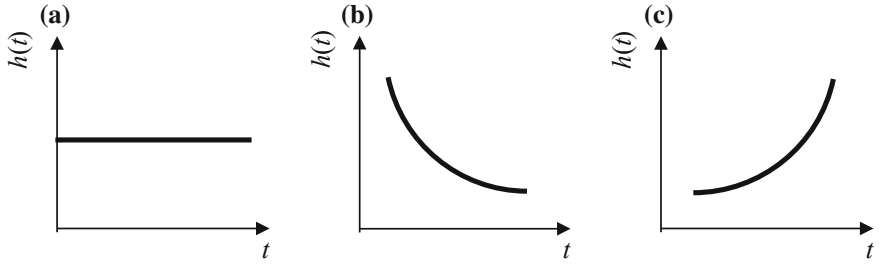


Fig. 4.5 Three cases of a failure rate: $k = 1$ (**a**); $k > 1$ (**b**); $k < 1$ (**c**)

From the equation above, we can see that the temporal trend of the failure rate depends on the shape parameter k. This dependence is illustrated in Fig. 4.5. When $k = 1$, the failure rate is a constant and the Weibull distribution reduces to the exponential model with MTTF = $1/\lambda$. When $k < 1$, the failure rate decreases monotonically and can be used to represent the case of infant mortality. When $k > 1$, the failure rate increases monotonically and can be used to represent the case of wear-out aging. As mentioned before, the Weibull distribution with $k = 1$ is an exponential distribution with the parameter λ. Note that, for $k > 4$, the Weibull distribution becomes symmetric and bell-shaped, like the curve of a normal distribution.

The reliability functions and failure rate functions with different parameters are compared in Fig. 4.6. As λ increases, the reliability decreases more slowly over time. For $k = 1$, the failure rates remain constant over time. For $k > 1$, the failure rates increase monotonically over time.

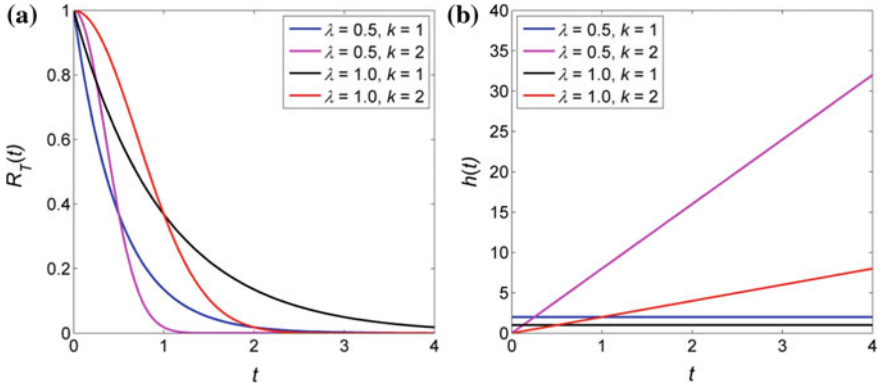


Fig. 4.6 Reliability functions (**a**) and failure rate functions (**b**) for a Weibull distribution

Example 4.4 Assume that the failure rate function of a solid-state power unit takes the following form (t: hrs)

$$h(t) = 0.003\left(\frac{t}{500}\right)^{0.5}$$

Answer the following questions.

(1) Compute the reliability of this power unit at $t = 50$ h.
(2) If the unit has operated for 50 h, compute the probability that it will operate for another 50 h.

Solution

(1) The form of the failure rate function indicates that the TTF of this power unit follows a Weibull distribution. From the failure rate function, we can easily derive the two distributional parameters as $\lambda = 500$ and $k = 1.5$. Then, the reliability at $t = 50$ h can be computed as

$$R_T(50) = e^{-\left(\frac{50}{\lambda}\right)^k} = e^{-\left(\frac{50}{500}\right)^{1.5}} \approx 0.9689$$

(2) The conditional probability that the unit fails after 100 h, given that it has not failed before 50 h, can be computed as

$$P(t > 100|t > 50) = \frac{P(t > 100)}{P(t > 50)} = \frac{R_T(t = 100)}{R_T(t = 50)} \approx 0.9438$$

Normal Distribution

Another widely used TTF distribution is the normal, or Gaussian, distribution. The density of this distribution can be expressed as

$$f_T(t; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2\right] \tag{4.22}$$

where $\mu \geq 0$ is the mean or MTTF, and $\sigma > 0$ is the standard deviation of the TTF. The reliability function can be expressed in terms of the standard normal CDF as

$$R_T(t) = \int_t^{+\infty} \frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2\right]d\tau = 1 - \Phi\left(\frac{t-\mu}{\sigma}\right) \tag{4.23}$$

The failure rate function is

$$h(t) = f_T(t;\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2\right]\left[1-\Phi\left(\frac{t-\mu}{\sigma}\right)\right]^{-1} \tag{4.24}$$

Examples for modeling TTF with normal distributions include the useful life of the tread of a tire and the wear-out time of the cutting edge of a machine tool. The reliability functions and failure-rate functions with different parameters are graphically compared in Fig. 4.7. Two observations can be made: (i) as μ increases, the reliability curve shifts to the right; (ii) the shape of the reliability curve is determined by σ, and a decrease of σ leads to the compression of the curve along the center line $t = \mu$.

4.3.3 Remarks on Time-Dependent Reliability

The main task in time-dependent reliability analysis is to accurately estimate the TTF distribution of an engineered system. The classical approach to this estimation is to fit a probability distribution to the TTF data (censored and uncensored) obtained through accelerated life testing (ALT), other reliability testing in the lab, and/or through performance trends observed in products in the field. This classical approach is capable of providing an instantaneous reliability estimate for an engineered system unit based on the degradation characteristics of other similar units. However, this approach only provides a population-wise reliability estimate that takes the same value for the entire population of units. In engineering practice, we are often interested in assessing the reliability of each individual unit under its actual use conditions to determine the advent of a failure and mitigate potential risk throughout the life-cycle. That is to say, we need a unit-wise, time-dependent

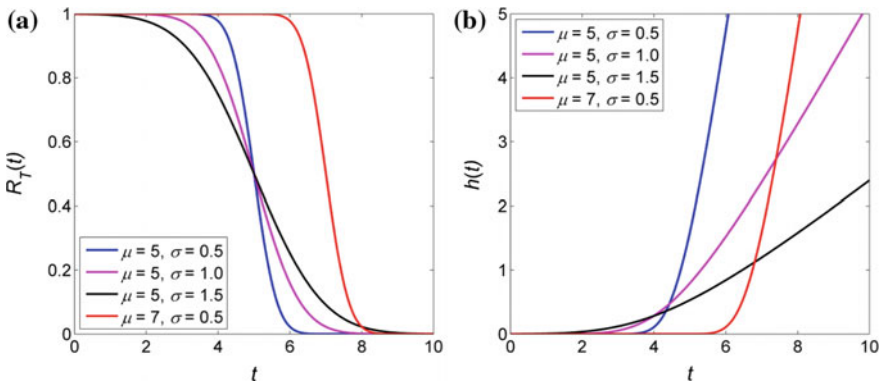


Fig. 4.7 Reliability functions (**a**) and failure rate functions (**b**) for a normal distribution

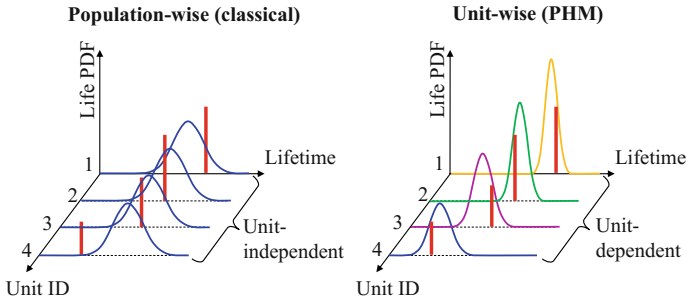


Fig. 4.8 Population- and unit-wise time-dependent reliability analyses

reliability analysis method (see Fig. 4.8 for the difference between population- and unit-wise reliability analysis) to ensure high operational reliability of an engineered system throughout its life-cycle. To overcome the limitations of classical reliability analysis, prognostics and health management (PHM) has recently emerged as a key technology to (i) evaluate the current health condition (health monitoring) and (ii) predict the future degradation behavior (health prognostics) of an engineered system throughout its life-cycle. This emerging discipline will be discussed in Chap. 8 of this book.

4.4 Exercises

4.1 Consider a cantilever beam-bar system with an ideal elastic-plastic cantilever beam supported by an ideal rigid-brittle bar, with a load applied at the mid-point of the beam, as shown in Fig. 4.9.
Suppose that a failure mode consists of two failure events: the formation of a hinge at the fixed point of the beam (failure event $\bar{E}_1$), followed by the fracture of the brittle bar (failure event $\bar{E}_2$). The two safety events can be expressed as:

$$E_1 = \{X, L, M | 3LX/8 - M \leq 0\},$$
$$E_2 = \{X, L, M, T | LX - M - 2LT \leq 0\}$$

The statistical information of the independent input random variables X and M is given in Table 4.1. The two dimensions L and T are assumed to be deterministic: $L = 5.0$ and $T = 1000$.

(1) Assuming $\mu_X = 60$, compute the reliabilities R_1 and R_2 corresponding to E_1 and E_2.
(2) Determine the value of μ_X so that the minimum value of R_1 and R_2 is greater than 0.90.

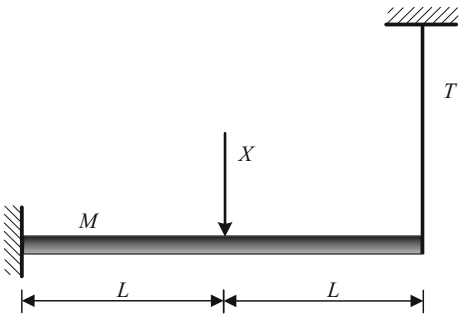


Fig. 4.9 Cantilever beam-bar system subjected to a vertical load. Reprinted (adapted) with permission from Ref. [2]

Table 4.1 Statistical information of random variables for Problem 4.1

Random variable	M	X
Distribution	Normal	Normal
Mean	150	μ_X
Standard deviation	30	20

(3) Assuming that X and M have a correlation coefficient $\rho_{XM} = 0.90$ and that $\mu_X = 60$, compute the reliabilities R_1 and R_2 corresponding to E_1 and E_2.

4.2 A cantilever beam is shown in Fig. 4.10. The length of the beam is 100 in. The width and thickness are represented by w and t, respectively. The free end of the beam is subjected to two transverse loads X and Y along the orthogonal directions, as shown in Fig. 4.10.
The stress limit-state function is expressed as

$$G = \sigma(X, Y) - S = \left(\frac{600}{wt^2}Y + \frac{600}{w^2t}X\right) - S = 0$$

where S is the yield strength, and w and t are the design parameters (fixed in this problem: $w = 2$ in and $t = 1$ in). S, X, and Y are independent random variables whose means and standard deviations are summarized in Table 4.2.

(1) Calculate the mean and standard deviation of $\sigma(X, Y)$.
(2) Compute the time-independent reliability, defined as $P(G \leq 0)$.

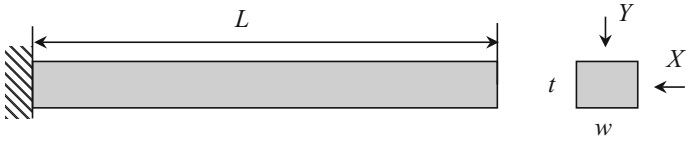


Fig. 4.10 Cantilever beam subjected to end transverse loads

Table 4.2 Statistical information for the random variables in Problem 4.2

Random variables	X [lb]	Y [lb]	S [psi]
Distribution	Normal	Normal	Normal
Mean	500	1000	400,000
Standard deviation	100	100	20,000

(3) Assume that the mean of S degrades in the following manner: $\mu_S(t) = 400{,}000 - 100t$ while the variation of S remains unchanged. Develop a time-dependent reliability function.

4.3 Recall Problem 3.4 in Chap. 3. Suppose the target life of the electronic component is set to 2000 min. $R(t) = P(T > 2000 \text{ min})$. Answer the following questions:

(1) Construct a reliability function based on the identified probability distribution of TTF.
(2) Determine the MTTF, the standard deviation of TTF, and the hazard function.
(3) Compare the reliability estimates (i) from n_f/N according to the TTF data and (ii) from the constructed reliability function ($t = 2000$ min), where n_f is the number of failed components and N (=100) is the total number of components.

4.4 Recall Problem 3.5 in Chap. 3. Suppose the target life of the machine cutting tool is set to 800 min, i.e., $R(t) = P(T > 800 \text{ min})$. Answer the following questions:

(1) Construct a reliability function.
(2) Determine the MTTF, the standard deviation of TTF, and the hazard function.
(3) Compare the reliability estimates (i) from n_f/N according to the TTF data and (ii) from the constructed reliability function ($t = 800$ min), where n_f is the number of failed components and N (=100) is the total number of components.

4.5 Suppose that 100 identical components are tested for 1000 h. Based on the historical testing data, we assume that the failure rate is a constant and that MTTF = 400 h. Answer the following questions.

(1) Estimate the expected number of components that will fail between 100 and 200 h.
(2) What is the conditional probability of failure of a component given that this component hasn't failed during the first 400 h?
(3) How many more components will fail if it is known that 30 components have failed during the first 200 h?

References

1. Hu, C., Wang, P., & Youn, B. D. (2015). Advances in system reliability analysis under uncertainty. In *Numerical methods for reliability and safety assessment* (pp. 271–303). Cham: Springer.
2. Song, J., & Der Kiureghian, A. (2003). Bounds on system reliability by linear programming. *Journal of Engineering Mechanics, 129*(6), 627–636.

Chapter 5
Reliability Analysis Techniques (Time-Independent)

Reliability analysis under uncertainty, which assesses the probability that a system's performance (e.g., fatigue, corrosion, fracture) meets its marginal value while taking into account various uncertainty sources (e.g., material properties, loads, geometries), has been recognized as having significant importance in product design and process development. However, reliability analysis in many engineering problems has been a challenging task due to the overwhelmingly large computational burden. To resolve the computational challenges, a variety of numerical and simulation techniques have been developed during the last two decades. This chapter is devoted to providing an in-depth discussion of these developments with the aim of providing insights into their relative merits and limitations.

5.1 Overview of Reliability Analysis Techniques

As discussed earlier, time-independent reliability can be formulated as a multi-dimensional integration of the joint PDF $f_{\mathbf{X}}(\mathbf{x})$ over a safe region

$$R = \int \cdots \int_{\Omega^S} f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} \tag{5.1}$$

where $\mathbf{X} = (X_1, X_2, \ldots, X_N)^{\mathrm{T}}$ denotes an N-dimensional random vector that models uncertainty sources, such as material properties, loads, and geometric tolerances; $f_{\mathbf{X}}(\mathbf{x})$ denotes the joint PDF of this random vector; the safe region Ω^{S} is defined by the limit-state function as $\Omega^{\mathrm{S}} = \{\mathbf{X}: G(\mathbf{X}) \leq 0\}$; and $G(\mathbf{X})$ is a system performance (or response) function.

Neither analytical multi-dimensional integration nor direct numerical integration is computationally feasible for large-scale engineering problems where the numbers of random variables are relatively large (e.g., a finite element model with over 20

C. Hu et al., *Engineering Design under Uncertainty and Health Prognostics*,
Springer Series in Reliability Engineering,
https://doi.org/10.1007/978-3-319-92574-5_5

parameters). The search for efficient computational procedures to estimate the reliability has resulted in a variety of numerical and simulation methods. In general, these methods can be categorized into four groups: (i) expansion methods; (ii) most probable point (MPP)-based methods; (iii) sampling methods; and (iv) stochastic response surface methods. In what follows, we will provide an overview of these methods.

Expansion methods obtain the second-moment statistics of the performance function based on the first- or second-order Taylor series expansion of this function at the mean values of the input random variables [1]. Reliability can be computed by assuming that the performance function follows a normal distribution. It can be inferred, therefore, that expansion methods involve two approximations, specifically, (i) the first-order (linear) or second-order (quadratic) approximation of the performance function at the mean values and (ii) the normal approximation to the PDF of the performance function. These approximations lead to the fact that these methods are only applicable for engineering problems with relatively small input uncertainties and weak output nonlinearities.

Among the many reliability analysis methods, the first- or second-order reliability methods (FORM [2] or SORM [3, 4]), which are MPP-based methods, are most commonly used. FORM/SORM use the first- or second-order Taylor expansion to approximate a limit-state function at the most probable failure point (MPP) where the limit-state function separates failure and safe regions of a product (or process) response. Some major challenges of FORM/SORM include (i) it is very expensive to build the probability density function (PDF) of the response and (ii) structural design can be expensive when employing a large number of responses.

Sampling methods include direct or smart Monte Carlo simulation (MCS) [5–9]. Assuming that we know the statistical information (PDFs or PMFs) of the input random variables, direct MCS generally involves the following three steps:

Step 1: Randomly generate a large number of samples based on the PDFs or PMFs of the random inputs.

Step 2: Evaluate the performance function at each of the random samples. Simulations or experiments need to be conducted to obtain the performance function values. Upon the completion of this step, a large number of random values (or realizations) of the performance function can be obtained.

Step 3: Extract the probabilistic characteristics of the performance function, including statistical moments, reliability, and PDF, from the random realizations obtained in Step 2.

Although direct MCS [5] produces accurate results for reliability analysis and allows for relative ease in implementation, it demands a prohibitively large number of simulation runs. Thus, it is often used for the purpose of benchmarking in reliability analysis. To alleviate the computational burden of direct MCS, researchers have developed various smart MCS methods, such as (adaptive)

importance sampling methods [6–8] and the enhanced MCS method with optimized extrapolation [9]. Despite improved efficiency over direct MCS, these methods are still computationally expensive.

The stochastic response surface method (SRSM) is an emerging technique for reliability analysis under uncertainty. As opposed to the deterministic response surface method (RSM), whose input variables are deterministic, SRSM employs random variables as its inputs. The aim of SRSM is to alleviate the computational burden required for accurate uncertainty quantification (i.e., quantifying the uncertainty in the performance function) and reliability analysis. This is achieved by constructing an explicit multi-dimensional response surface approximation based on function values given at a set of sample points. Generally speaking, uncertainty quantification and reliability analysis through SRSM consists of the following steps:

Step 1: Determine an approximate functional form for the performance function.
Step 2: Evaluate the parameters of the functional approximation (or the stochastic response surface) based on the function values at a set of sample points.
Step 3: Conduct MCS or numerical integration based on the functional approximation to obtain the probabilistic characteristics (e.g., statistical moments, reliability, and PDF) of the performance function.

The current state-of-the-art SRSMs for uncertainty quantification include the dimension reduction (DR) method [10–12], the stochastic spectral method [13–15], and the stochastic collocation method [16–19].

5.2 Expansion Methods

Recall the simple performance function discussed in Chap. 4, where the safety margin between the strength S of an engineered system and the load L on this system is defined as the performance function. Under the assumption of normal distributions for S and L, the performance function G also follows a normal distribution. By further assuming statistical independence between S and L, we can compute the reliability based on the standard normal CDF of the following form

$$R = \Phi\left(\frac{\mu_S - \mu_L}{\sqrt{\sigma_L^2 + \sigma_S^2}}\right) \tag{5.2}$$

The reliability of the engineered system is estimated based on the first two statistical moments (mean and standard deviation). Here, we only consider a simple performance function, which is a linear combination of two normally distributed random variables. In fact, this idea of reliability analysis using the first two statistical moments can be generalized to cases where the functions are in a nonlinear form.

We begin with the first-order Taylor series expansion of the performance function $G(\mathbf{X})$ at the mean value, expressed as

$$\begin{aligned} G(\mathbf{X}) &\approx G(\boldsymbol{\mu}_{\mathbf{X}}) + \sum_{i=1}^{N} \frac{\partial G(\boldsymbol{\mu}_{\mathbf{X}})}{\partial X_i} \left(X_i - \mu_{X_i}\right) \\ &= a_1 X_1 + \cdots + a_N X_N + b \end{aligned} \tag{5.3}$$

This can be rewritten in a vector form as $G(\mathbf{X}) = \mathbf{a}^{\mathrm{T}}\mathbf{X} + b$, where $\mathbf{a} = [a_1, a_2, \ldots, a_N]^{\mathrm{T}}$ contains the first-order partial derivatives of G with respect to input random variables and is called a sensitivity vector of G. We can then obtain the first-order approximate mean and variance of G as

$$\mu_G = E[G] \approx E\left[\mathbf{a}^{\mathrm{T}}\mathbf{X} + b\right] = \mathbf{a}^{\mathrm{T}}\boldsymbol{\mu}_{\mathbf{X}} + b \tag{5.4}$$

and

$$\begin{aligned} \sigma_G^2 &= E\left[(G - \mu_G)^2\right] = E\left[(G - \mu_G)(G - \mu_G)^{\mathrm{T}}\right] \\ &\approx E\left[\left(\mathbf{a}^{\mathrm{T}}\mathbf{X} + b - \mathbf{a}^{\mathrm{T}}\boldsymbol{\mu}_{\mathbf{X}} - b\right)\left(\mathbf{a}^{\mathrm{T}}\mathbf{X} + b - \mathbf{a}^{\mathrm{T}}\boldsymbol{\mu}_{\mathbf{X}} - b\right)^{\mathrm{T}}\right] \\ &= \mathbf{a}^{\mathrm{T}} E\left[(\mathbf{X} - \boldsymbol{\mu}_{\mathbf{X}})(\mathbf{X} - \boldsymbol{\mu}_{\mathbf{X}})^{\mathrm{T}}\right]\mathbf{a} \\ &= \mathbf{a}^{T}\boldsymbol{\Sigma}_{\mathbf{X}}\mathbf{a} \end{aligned} \tag{5.5}$$

where $\boldsymbol{\Sigma}_{\mathbf{X}}$ is the covariance matrix of $\mathbf{X}$. Under the assumption of normality for the performance function, the reliability can be computed based on the first two statistical moments as

$$R = \Phi\left(-\frac{\mu_G}{\sigma_G}\right) \tag{5.6}$$

Note that the formula above gives the exact reliability only if the performance function is a linear function of normally distributed random variables. However, it is rare, in engineering practice, to encounter an engineered system whose performance function is a simple linear combination of normally distributed random variables. It is more likely that the performance function is of a nonlinear form and that some of the random variables are non-normally distributed. In such cases, the first-order expansion method leads to an inaccurate reliability estimate that often contains a large error.

Example 5.1 A cantilever beam is shown in Fig. 5.1.

Assume that the vertical loads P_1 and P_2 follow normal distributions with means 1000 and 500 lb, and standard deviations 100 and 50 lb, respectively. In addition, assume that the bending moment is deterministic (fixed at

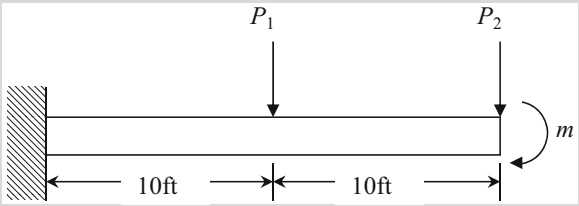


Fig. 5.1 Cantilever beam subjected to vertical loads and a bending moment

10,000 lb-ft). Further assume that P_1 and P_2 are uncorrelated. Answer the following questions.

(1) Compute the mean, standard deviation, and coefficient of variation (COV) of the maximum moment at the fixed end using the first-order expansion method.
(2) Compute the reliability with an allowable moment m_a = 33,000 lb–ft.

Solution

(1) At the fixed end, the maximum moment can be expressed as

$$m_{\max} = 10P_1 + 20P_2 + 10000 = \mathbf{a}^{\mathrm{T}}\mathbf{X} + b$$

where $\mathbf{a} = [10, 20]^{\mathrm{T}}$, b = 10,000, and $\mathbf{X} = [P_1, P_2]^{\mathrm{T}}$. Note that the mean vector $\boldsymbol{\mu}_{\mathbf{X}} = [1000, 50]^{\mathrm{T}}$, and the covariance matrix $\boldsymbol{\Sigma}_{\mathbf{X}} = [100^2, 0; 0, 50^2]$, where the semicolon denotes the separation between two adjacent rows in a matrix. We first compute the mean as

$$\begin{aligned}\mu_{m_{\max}} &= \mathbf{a}^{\mathrm{T}}\boldsymbol{\mu}_{\mathbf{X}} + b = [10 \quad 20]\begin{bmatrix}1000\\500\end{bmatrix} + 10000\\ &= 30,000 \text{ lb}-\text{ft}\end{aligned}$$

We can then compute the variance as

$$\begin{aligned}\sigma^2_{m_{\max}} &= \mathbf{a}^T\Sigma_{\mathbf{X}}\mathbf{a} = [10 \quad 20]\begin{bmatrix}100^2 & 0\\ 0 & 50^2\end{bmatrix}\begin{bmatrix}10\\20\end{bmatrix}\\ &= 2 \times 10^6\,[\text{lb}-\text{ft}]^2\end{aligned}$$

Taking the square root of the variance gives us the standard deviation as

$$\sigma_{m_{\max}} = \sqrt{2 \times 10^6} \approx 1414.2 \text{ lb}-\text{ft}$$

Next, we can compute the coefficient of variation (COV) of $m_{\max}$ as

$$\text{COV} \equiv \frac{\sigma_{m_{\max}}}{\mu_{m_{\max}}} = \frac{1414.2}{30000} \approx 0.047 \text{ or } 4.7\%$$

(2) Finally, the reliability can be computed as

$$\begin{aligned} R &= P(m_{\max} \le m_a) \\ &= P\left(\frac{m_{\max} - \mu_{m_{\max}}}{\sigma_{m_{\max}}} \le \frac{m_a - \mu_{m_{\max}}}{\sigma_{m_{\max}}}\right) \\ &= P\left(Z \le \frac{33{,}000 - 30{,}000}{1414} = 2.12\right) \\ &= \Phi(2.12) = 98.3\% \end{aligned}$$

Note that the performance function considered in this example is a simple linear function of normally distributed random variables P_1 and P_2. Therefore, the reliability estimate obtained from the first-order expansion method is exact.

From what has been discussed, we can see that the linearization of a performance function at the mean values of the random variables enables estimation of the mean and variance of the performance function. However, the estimates may contain large errors if second- and/or high-order expansion terms are significant. A more accurate approximation can be realized by the second-order Taylor series expansion of the performance function $G(\mathbf{X})$. This expansion involves quadratic (second-order) terms and can be expressed as

$$\begin{aligned} G(\mathbf{X}) \approx G(\boldsymbol{\mu}_{\mathbf{X}}) &+ \sum_{i=1}^{N} \frac{\partial G(\boldsymbol{\mu}_{\mathbf{X}})}{\partial X_i}\left(X_i - \mu_{X_i}\right) \\ &+ \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \frac{\partial^2 G(\boldsymbol{\mu}_{\mathbf{X}})}{\partial X_i \partial X_j}\left(X_i - \mu_{X_i}\right)\left(X_j - \mu_{X_j}\right) \end{aligned} \tag{5.7}$$

We can then obtain the second-order approximate mean and variance of G as

$$\mu_G = E[G] \approx G(\boldsymbol{\mu}_{\mathbf{X}}) + \frac{1}{2}\sum_{i=1}^{N} \frac{\partial^2 G(\boldsymbol{\mu}_{\mathbf{X}})}{\partial X_i^2}\sigma_{X_i}^2 \tag{5.8}$$

and

$$\sigma_Y^2 = E\left[(G-\mu_G)^2\right] \approx \sum_{i=1}^{N}\left(\frac{\partial G(\boldsymbol{\mu}_{\mathbf{X}})}{\partial X_i}\right)^2 \sigma_{X_i}^2 + \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{\partial^2 G(\boldsymbol{\mu}_{\mathbf{X}})}{\partial X_i \partial X_j}\sigma_{X_i}^2\sigma_{X_j}^2 \quad (5.9)$$

Under the assumption of normality for the performance function, the reliability can then be computed using Eq. (5.6).

5.3 MPP-Based Methods

The expansion methods discussed earlier suffer from the following two drawbacks: (i) these methods only utilize the first two statistical moments of the input random variables while ignoring the distribution information of these variables; and (ii) the mean point of the input random variables is treated as the reference point for building a linear or quadratic approximation of the performance function, which may lead to a large error in estimating a high reliability (or a low probability of failure). As an attempt to overcome the drawbacks of the expansion methods, Hasofer and Lind proposed the first-order reliability method (FORM) [2] in 1974. Since then, the attempts to improve FORM have resulted in more advanced MPP-based methods, including the second-order reliability method (SORM). We will briefly review these two most-well-known MPP-based methods.

5.3.1 First-Order Reliability Method (FORM)

The basic idea of FORM is to linearize the performance function $G(\mathbf{X})$ at the most probable failure point on the limit-state surface $G(\mathbf{X}) = 0$, or the MPP in the transformed **U**-space. The **U**-space is composed of independent standard normal variables **U** that are transformed from the input random variables **X** in the original **X**-space. Compared to the expansion methods, the two distinctive features of FORM are (i) the transformation **T** of input random variables to the standard normal space and (ii) the use of the MPP as the reference point for the linearization of the performance function. For a normal random variable X_i with mean μ_{Xi} and standard deviation σ_{Xi}, transformation T can be simply defined as

$$U_i = T(X_i) = \frac{X - \mu_{X_i}}{\sigma_{X_i}}, \quad i = 1, 2, \ldots, N \quad (5.10)$$

In a general case, the transformation formula can be derived based on the CDF mapping as

$$F_{X_i}(X_i) = \Phi(U_i) \tag{5.11}$$

where F_{Xi} and Φ are the CDFs of X_i and U_i, respectively. Note that, unlike the expansion methods, FORM utilizes the distribution information of input random variables to transform these variables to standard normal random variables. The transformation of a uniformly distributed random variable X_i to the corresponding standard normal random variable U_i is illustrated in Fig. 5.2. Observe that the one-to-one mapping between the CDFs ensures the one-to-one mapping between the values of the original and transformed variables. The transformations of five of the most commonly used types of probability distributions (i.e., normal, lognormal, Weibull, Gumbel, and uniform) are presented in Table 5.1.

Through the transformation, the performance function $G(\mathbf{X})$ in the original $\mathbf{X}$-space is mapped onto an equivalent function $G(\mathbf{U}) = G(\mathbf{T}(\mathbf{X}))$ in the transformed $\mathbf{U}$-space. The transformation of a performance function involving two normally distributed input variables is graphically presented in Fig. 5.3. In the $\mathbf{U}$-space, the

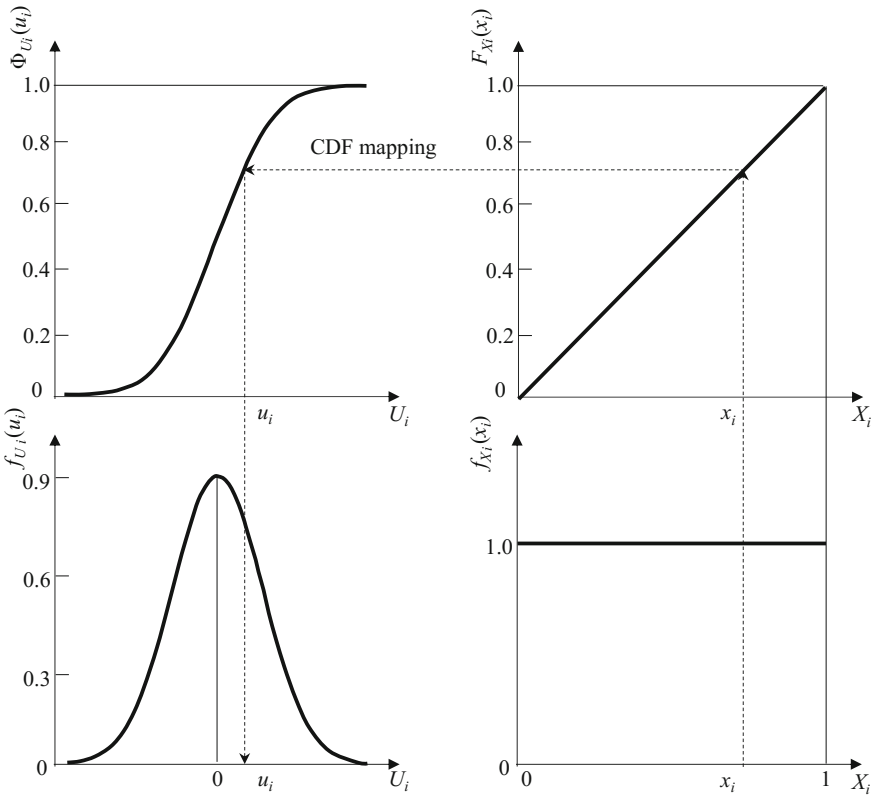


Fig. 5.2 Transformation of a uniform random variable to a standard normal random variable

Table 5.1 Probability distribution and its transformation between X- and U-space

Distribution	Probability density function (PDF)	Transformation
Normal	$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]$	$X = \mu + \sigma U$
Lognormal	$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma x}\exp\left[-\frac{1}{2}\left(\frac{\ln x - \mu}{\sigma}\right)^2\right]$ $x > 0$	$X = \exp(\mu + \sigma U)$ $\sigma_X^2 = \left(e^{\sigma^2} - 1\right)\exp\left(2\mu + \sigma^2\right)$
Weibull	$f_X(x) = \lambda k(\lambda x)^{k-1} e^{-(\lambda x)^k}$ $x \geq 0$	$X = \frac{1}{\lambda}[-\ln(\Phi(-U))]^{1/k}$
Gumbel	$f_X(x) = \alpha\exp\left[-\alpha(x-v) - e^{-\alpha(x-v)}\right]$, $-\infty \leq x \leq +\infty$	$X = a + (b-1)\Phi(U)$
Uniform	$f_X(x) = \frac{1}{b-a}$, $a \leq x \leq b$	$X = a + (b-1)\Phi(U)$

MPP **u*** denotes the point on the failure surface which has the minimum distance to the origin. This distance is called the reliability index, expressed as

$$\beta = \|\mathbf{u}^*\| = \left[\sum_{i=1}^{N}\left(\mathbf{u}_i^*\right)^2\right]^{1/2} \tag{5.12}$$

The reliability R can then be computed as (when $R \geq 0.5$)

$$R = \Phi(\beta) \tag{5.13}$$

As can be seen from Fig. 5.3, as the minimum distance β between the failure surface and the origin becomes larger, the area of the safe region becomes larger, while the area of the failure region becomes smaller. This indicates a lesser chance that the random samples will be located in the failure region, which means a lower

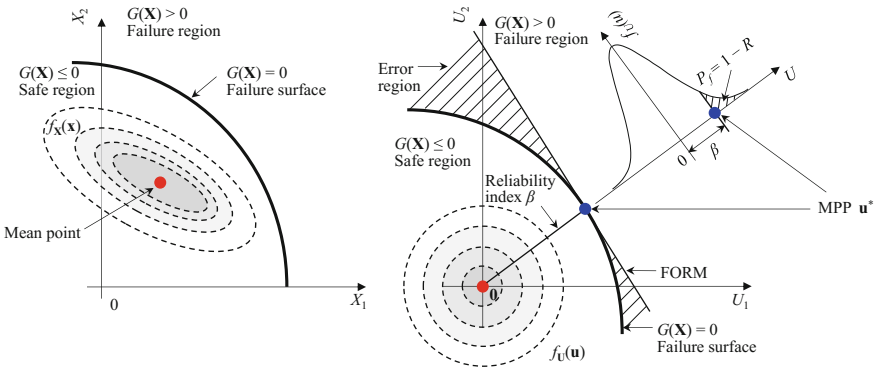


Fig. 5.3 Transformation from **X**-space to **U**-space and FORM approximation

probability of failure and thus higher reliability. Thus, the reliability index β is a good measure of reliability. Note that this measure gives an exact reliability value only if the failure surface in the **U**-space is of a linear form. In most engineering problems, however, the failure surface in the **U**-space is a nonlinear function, either due to nonlinearity in the performance function or due to nonlinear transformation from a non-normal distribution to a normal distribution. For the case in Fig. 5.3, the failure surface exhibits a nonlinear form in the **U**-space due to the nonlinearity in the performance function, and as a consequence, the FORM overestimates the reliability, as indicated by the shaded error region.

The remaining—however, the most critical—task is to search for the MPP. Mathematically, this task can be formulated as an optimization problem with one equality constraint in the **U**-space, expressed as

$$\begin{aligned} &\text{Minimize} \quad \|\mathbf{U}\| \\ &\text{Subject to} \quad G(\mathbf{U}) = 0 \end{aligned} \tag{5.14}$$

where the optimum point on the failure surface is the MPP $\mathbf{u}^*$. The MPP search generally requires an iterative optimization scheme based on the gradient information of the performance function. Among the many MPP search algorithms, the most widely used is the Hasofer-Lind and Rackwitz-Fiessler (HL-RF) method, due to its simplicity and efficiency. The HL-RF method consists of the following iterative steps:

Step 1: Set the number of iterations $k = 0$ and the initial MPP estimate $\mathbf{u} = \mathbf{u}^{(0)}$ that corresponds to the mean values of **X**.

Step 2: Transform $\mathbf{u}^{(k)}$ to $\mathbf{x}^{(k)}$ using Eq. (5.11). Compute the performance function $G(\mathbf{u}^{(k)}) = G(\mathbf{x}^{(k)})$ and its partial derivatives with respect to the input random variables in the **U**-space as

$$\nabla_{\mathbf{U}} G\left(\mathbf{u}^{(k)}\right) = \left[\frac{\partial G}{\partial U_1}, \frac{\partial G}{\partial U_2}, \ldots, \frac{\partial G}{\partial U_N}\right]_{\mathbf{U}=\mathbf{u}^{(k)}} \tag{5.15}$$

Step 3: Update the search point at the current iteration as

$$\begin{aligned} \mathbf{u}^{(k+1)} &= \left(\mathbf{u}^{(k)} \cdot \mathbf{n}^{(k)} - \frac{G\left(\mathbf{u}^{(k)}\right)}{\|\nabla_{\mathbf{U}} G(\mathbf{u}^{(k)})\|}\right)\mathbf{n}^{(k)} \\ &= \left[\mathbf{u}^{(k)} \cdot \nabla_{\mathbf{U}} G\left(\mathbf{u}^{(k)}\right) - G\left(\mathbf{u}^{(k)}\right)\right] \frac{\nabla_{\mathbf{U}} G\left(\mathbf{u}^{(k)}\right)}{\|\nabla_{\mathbf{U}} G(\mathbf{u}^{(k)})\|^2} \end{aligned} \tag{5.16}$$

where $\mathbf{n}^{(k)}$ is the normalized steepest ascent direction of $G(\mathbf{U})$ at $\mathbf{u}^{(k)}$, expressed as

$$\mathbf{n}^{(k)} = \frac{\nabla_{\mathbf{U}} G\left(\mathbf{u}^{(k)}\right)}{\left\|\nabla_{\mathbf{U}} G\left(\mathbf{u}^{(k)}\right)\right\|} \tag{5.17}$$

Step 4: Repeat Steps 2 and 3 until the convergence of $\mathbf{u}$.

The updating process in the HL-RF method is graphically shown in Fig. 5.4. The process starts with the initial MPP estimate $\mathbf{u}^{(0)}$ where a linear approximation is constructed based on the vector of partial derivative from Step 2. A perpendicular to the linear approximation line is then constructed through the origin. The updated point $\mathbf{u}^{(1)}$ is identified as the point on the perpendicular whose distance from the origin equals $[\mathbf{u}^{(0)} \cdot \mathbf{n}^{(0)} - G\left(\mathbf{u}^{(0)}\right) / ||\nabla_{\mathbf{U}} G(\mathbf{u}^{(0)})||]$, i.e., the magnitude of the unit vector $\mathbf{n}^{(0)}$, as shown in Eq. (5.16). This completes the first iteration. In each of the following iterations, the method constructs a linear approximation to the contour of the performance function at the most recent point, and then locates the updated point on the perpendicular of the linear approximation based on Eq. (5.16). This process continues until the convergence of the search.

Observe that the MPP search algorithm requires computation of the first-order derivatives of $G(\mathbf{U})$. The derivatives can be evaluated using the finite difference method, which uses the difference between the original and perturbed values of $G(\mathbf{U})$ to compute the first-order sensitivities. Since we need to compute both the original and the perturbed responses, any iteration during the MPP search requires $N + 1$ function evaluations. If we have k iterations, the total number of function evaluations will be $k(N + 1)$. This suggests that the FORM is first-order efficient.

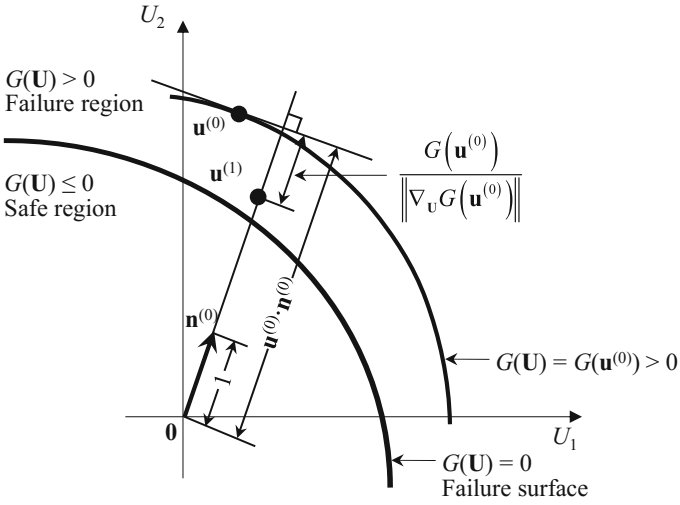


Fig. 5.4 Procedure for the HL-RF method in search of MPP

Example 5.2 Consider the following performance function

$$G(X_1, X_2) = 1 - \frac{80}{X_1^2 + 8X_2 + 5}$$

where X_1 and X_2 each follow a normal distribution with the mean 4 and the standard deviation 0.6. Find the MPP and compute the reliability with the FORM.

Solution

The first iteration in the HL-RF method is detailed as follows:

Step 1: Set the number of iterations k = 0 and the initial values $\mathbf{u}$ = (0,0).

Step 2: Transform $\mathbf{u}^{(0)}$ to $\mathbf{x}^{(0)}$ with Eq. (5.11). $\mathbf{x}^{(0)}$ = (4,4). Compute the performance function $G(\mathbf{x}^{(k)})$ as

$$G(X_1, X_2) = 1 - \frac{80}{X_1^2 + 8X_2 + 5} = 1 - \frac{80}{4^2 + 8 \times 4 + 5} \approx -0.5049$$

and the partial derivatives as

$$\frac{\partial G}{\partial U_1} = \frac{\partial G}{\partial X_1}\frac{\partial X_1}{\partial U_1} = \frac{160 X_1 \sigma_{X_1}}{\left(X_1^2 + 8X_2 + 5\right)^2} = \frac{160 \times 4 \times 0.6}{(4^2 + 8 \times 4 + 5)^2} \approx 0.1367$$
$$\frac{\partial G}{\partial U_2} = \frac{\partial G}{\partial X_2}\frac{\partial X_2}{\partial U_2} = \frac{640 \sigma_{X_2}}{\left(X_1^2 + 8X_2 + 5\right)^2} = \frac{640 \times 0.6}{(4^2 + 8 \times 4 + 5)^2} \approx 0.1367$$

Step 3: Update the search point at the current iteration as

$$\begin{aligned}\mathbf{u}^{(1)} &= \left[\mathbf{u}^{(0)} \cdot \nabla_{\mathbf{U}} G\left(\mathbf{u}^{(0)}\right) - G\left(\mathbf{u}^{(0)}\right)\right] \frac{\nabla_{\mathbf{U}} G\left(\mathbf{u}^{(0)}\right)}{\left\|\nabla_{\mathbf{U}} G(\mathbf{u}^{(0)})\right\|^2} \\ &= [(0,0) \cdot (0.1367, 0.1367) - (-0.5094)] \frac{(0.1367, 0.1367)}{0.1367^2 + 0.1367^2} \\ &= (1.8633, 1.8633).\end{aligned}$$

Table 5.2 Iteration history for MPP search in Example 5.2

Iteration	U_1	U_2	β	$G(\mathbf{U})$	$\frac{\partial G}{\partial U_1}$	$\frac{\partial G}{\partial U_2}$
0	0.0000	0.0000	0.0000	−0.5094	0.1367	0.1367
1	1.8633	1.8633	2.6351	−0.1090	0.0944	0.0738
2	2.7774	2.1707	3.5251	−0.0059	0.0860	0.0607
3	2.9225	2.0630	3.5772	0.0001	0.0863	0.0600
4	2.9366	2.0416	3.5766	0.0000	0.0864	0.0600
Optimum	2.9366	2.0416	3.5766	0.0000	0.0864	0.0600

The results for all the iterations needed to find the MPP are summarized in Table 5.2. Finally, the reliability is estimated as the standard normal CDF value of β, which is 0.9998.

5.3.2 *Second-Order Reliability Method (SORM)*

As can be seen in Fig. 5.3, FORM constructs a first-order or linear approximation to the limit-state function at the MPP. If $G(\mathbf{U})$ is nonlinear, FORM either over- or under-estimates the reliability since it ignores the curvature of the nonlinear limit state. The resulting error (the shaded area in Fig. 5.3) can be large if $G(\mathbf{U})$ is highly nonlinear (i.e., when the failure surface exhibits a large curvature). In such cases, SORM, which takes into account the curvature information in the limit-state function approximation, is more desirable. The comparison between FORM and SORM is graphically shown in Fig. 5.5. A nonlinear approximation to the failure surface through use of SORM results in greater accuracy in the reliability estimate. This accuracy improvement is achieved by utilizing more information and, specifically, the second-order derivatives with respect to the input random variables.

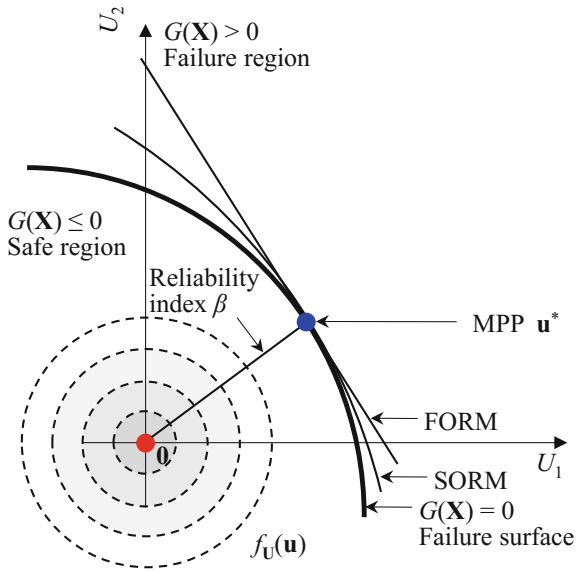


Fig. 5.5 FORM and SORM approximations

Breitung's SORM approximation of the probability of failure can be expressed in an explicit form as [3]

$$p_f = \Phi(-\beta) \prod_{i=1}^{N-1} (1 + \beta\kappa_i) \tag{5.18}$$

where κ_i, $i = 1, 2, \ldots, N$, are the principal curvatures of $G(\mathbf{U})$ at the MPP, and β is the reliability index. Clearly, upon completion of the FORM computation, an extra computational task is needed to find the principal curvatures κ_i. This task can be completed in two steps, which are listed as follows:

Step 1: Rotate the standard normal variables U_i (in the **U**-space) to a set of new standard normal variables Y_i (in the **Y**-space), of which the last variable Y_N shares the same direction with the unit gradient vector of $G(\mathbf{U})$ at the MPP. To do so, we generate an orthogonal rotation matrix **R**, which can be derived from a simple matrix $\mathbf{R}_0$, expressed as

$$\mathbf{R}_0 = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial G(\mathbf{u}^*)/\partial u_1}{|\nabla G(\mathbf{u}^*)|} & \frac{\partial G(\mathbf{u}^*)/\partial u_2}{|\nabla G(\mathbf{u}^*)|} & \cdots & \frac{\partial G(\mathbf{u}^*)/\partial u_N}{|\nabla G(\mathbf{u}^*)|} \end{bmatrix} \tag{5.19}$$

where the last row consists of the components of the unit gradient vector of the limit-state function at the MPP. Next, the orthogonal matrix **R** can be obtained by orthogonalizing $\mathbf{R}_0$ using the Gram-Schmidt algorithm. In the rotated **Y**-space, the **U**-space, second-order approximation to the limit-state function at the MPP can be expressed as

$$G(\mathbf{Y}) \approx -Y_N + \beta + \frac{1}{2}(\mathbf{Y} - \mathbf{Y}^*)^{\mathrm{T}} \mathbf{R}\mathbf{D}\mathbf{R}^{\mathrm{T}} (\mathbf{Y} - \mathbf{Y}^*) \tag{5.20}$$

where **D** is the second-order Hessian matrix of the size N by N; and $\mathbf{Y}^* = [0, 0, \ldots, \beta]^{\mathrm{T}}$ is the MPP in the **Y**-space.

Step 2: Compute the principal curvatures κ_i as the $N - 1$ eigenvalues of an $(N - 1) \times (N - 1)$ matrix **A** of the following form

$$\mathbf{A} = \frac{\mathbf{R}\mathbf{D}\mathbf{R}^{\mathrm{T}}}{|\nabla G(\mathbf{u}^*)|} \tag{5.21}$$

The second-order approximation can then be rewritten as

$$G(\mathbf{U}) = -U_N + \beta + \frac{1}{2}\sum_{i=1}^{N-1} \kappa_i U_i^2 \tag{5.22}$$

Finally, Breitung's SORM formula in Eq. (5.18) can be used to compute the probability of failure or reliability. Besides Breitung's formula, another popular and more accurate SORM formulation is given by Tvedt [4].

5.4 Sampling Methods

During the last several decades, sampling methods have played an important role in advancing research in reliability analysis. These methods generally involve generation of random samples of input random variables, deterministic evaluations of the performance function at these random samples, and post-processing to extract the probabilistic characteristics (e.g., statistical moments, reliability, and PDF) of the performance function. In this section, direct Monte Carlo simulation (MCS), the most crude (yet widely used) sampling method, is briefly introduced. Following this brief introduction, we introduce a smart MCS method that borrows ideas from MPP-based methods, namely the importance sampling method.

5.4.1 Direct Monte Carlo Simulation

The term "Monte Carlo" was originally used as a Los Alamos code word by Ulam and von Neumann, who worked on stochastic simulations to achieve better atomic bombs. Since then, the word has been widely used in articles and monographs and MCS has been applied to a wide variety of scientific disciplines. The basic idea behind MCS is to approximate an underlying distribution of a stochastic function and the associated probabilistic characteristics (e.g., mean, variance, and higher-order moments) by computing the function values at simulated random samples.

To introduce this concept, let us rewrite the multi-dimensional integration in Eq. (4.9) for reliability analysis with an indicator function as

$$R = \int \cdots \int_{\Omega^S} f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} = \int \cdots \int_{\Omega} I_{\Omega^S}(\mathbf{x}) f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} = E\left(I_{\Omega^S}(\mathbf{x})\right) \tag{5.23}$$

where $I[\cdot]$ is an indicator function of the safe or fail state such that

$$I_{\Omega^S}(\mathbf{x}) = \begin{cases} 1, & \mathbf{x} \in \Omega^S \\ 0, & \mathbf{x} \in \Omega \backslash \Omega^S \end{cases} \tag{5.24}$$

Observe that the reliability can be computed as the expectation of an indicator function over the entire domain of $\mathbf{X}$. According to the law of large numbers, we can obtain a good approximation of this expectation by employing a sufficiently large number of random samples. Specifically, the direct MCS consists of four sequentially executed steps (assuming statistical independence between random variables):

Step 1: Specify the number of random samples M. For each random variable X_i, generate M "pseudo random numbers" v_{i1}, v_{i2}, …, v_{iM} from a standard uniform distribution (between 0 and 1). Then obtain M random vectors $\mathbf{v}_1$, $\mathbf{v}_2$, …, $\mathbf{v}_M$ with $\mathbf{v}_j = [v_{1j}, v_{2j}, \ldots, v_{Nj}]^{\mathrm{T}}$.

Step 2: Transform v_{ij} ($i = 1, 2, \ldots, N$, $j = 1, 2, \ldots, M$) to x_{ij} using the CDF mapping method shown in Eq. (5.11) and illustrated in Fig. 5.2. The mathematical expression of this transformation can be written as

$$x_{ij} = F_{X_i}^{-1}(v_{ij}) \tag{5.25}$$

For example, if an input random variable X follows a uniform distribution between a and b, its CDF can be written as $F_X(x) = (x - a)/(b - a)$ for $a \leq x \leq b$, which in turn gives us the transformation as $x = a + (b - a)v$. If we have two standard normal random variables X_1 and X_2, the transformation for either can be expressed as $x_i = \Phi^{-1}(v_i)$. The scatter plots of 1000 randomly generated samples for two standard uniform variables and the transformed random samples for the corresponding standard normal variables are shown in Fig. 5.6.

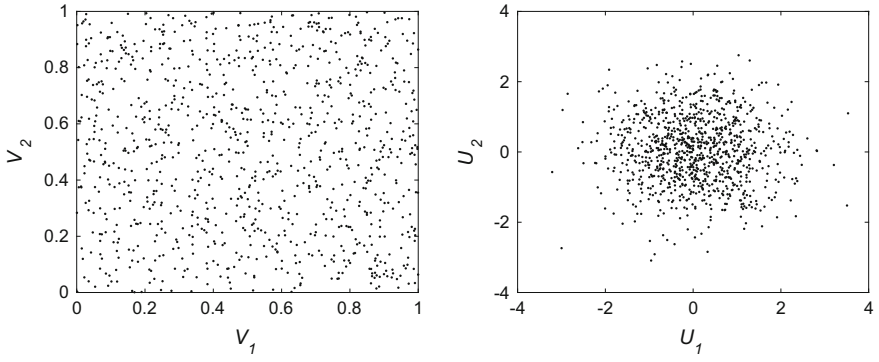


Fig. 5.6 Standard uniform samples (**a**) and the corresponding standard normal samples (**b**)

Step 3: Evaluate the values of the performance function $G(\mathbf{X})$ at the random samples $\mathbf{x}_j$. This step requires M function evaluations. Upon the completion of this step, we obtain M random function values $G(\mathbf{x}_j)$, for $j = 1, 2, \ldots, M$, which consist of rich statistical information of the performance function.

Step 4: Extract the probabilistic characteristics of $G(\mathbf{X})$, including statistical moments, reliability, and PDF, from the random function values. For example, the reliability can be estimated by

$$R = E\left(I_{\Omega^s}(\mathbf{x})\right) \approx \frac{1}{M}\sum_{j=1}^{M} I_{\Omega^s}\left(\mathbf{x}_j\right) \tag{5.26}$$

In addition, any rth-order moment of $G(\mathbf{X})$ can be calculated as

$$\gamma_r = \int \cdots \int_{\Omega} G^r(\mathbf{x}) f_{\mathbf{X}}(\mathbf{x})\, d\mathbf{x} = E(G^r(\mathbf{x})) \approx \frac{1}{M}\sum_{j=1}^{M} G^r\left(\mathbf{x}_j\right) \tag{5.27}$$

Random sampling allows for the derivation of all probabilistic characteristics (e.g., statistical moments, reliability, and PDF) of the performance function. This is different from MPP-based methods, which are only capable of estimating the reliability.

Reliability analysis using direct MCS is graphically illustrated in Fig. 5.7. A set of random samples of X_1 and X_2 are categorized into two groups, failed samples and safe samples, separated by the failure surface (or the limit state function) $G(\mathbf{X}) = 0$. The reliability is computed as the proportion of the safe samples over all random samples.

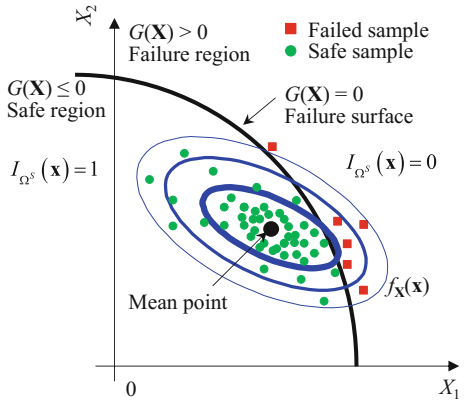


Fig. 5.7 Concept of reliability analysis using direct MCS

So far, it looks straightforward to obtain a reliability estimate using direct MCS. However, this estimate does raise a concern from the precision perspective. Intuitively, we would conjecture that precision depends on the number of samples M and, for a small M, the reliability estimate might be subject to large variance or low precision. We would further conjecture that, according to the law of large numbers, the reliability estimate would approach the true value as M approaches infinity. Indeed, all of these conjectures are correct; thus, we need to have a mathematical model to rigorously express the important quantity underlying our conjectures. This important quantity is the variance of the reliability estimate, expressed as [20]

$$\sigma_P = \sqrt{\frac{P(1-P)}{M}} \tag{5.28}$$

where P is the probability estimate (reliability or probability of failure) derived from direct MCS, and M is the number of MCS samples. To clearly reflect the variation in the probability estimate, we can compute the error bounds with a 95% confidence level. The error bound with a $100(1-\alpha)\%$ confidence can be computed as [20]

$$\varepsilon_S = z_{1-\alpha/2}\sigma_P = z_{1-\alpha/2}\sqrt{\frac{P(1-P)}{N_S}} \tag{5.29}$$

where $z_{1-\alpha/2}$ is the $100(1-\alpha/2)$th percentile of the standard normal distribution. For a 95% confidence level, $\alpha = 0.05$ and $z_{1-\alpha/2} = 1.96$.

Example 5.3 Recall Exercise 4.2 where a cantilever beam is subjected to two end transverse loads X and Y in orthogonal directions. Ten sets of random samples generated for direct MCS are summarized in Table 5.3. Obtain the reliability estimate using direct MCS and discuss the precision of the estimate.

Table 5.3 Summary of random samples for direct MCS in Example 5.3

Sample ID	X [lb]	Y [lb]	R [psi]	Sample ID	X [lb]	Y [lb]	R [psi]
1	614	853	448,480	6	566	850	407,240
2	674	1036	385,540	7	512	944	402,470
3	528	1158	359,820	8	415	1067	391,870
4	483	887	415,130	9	458	903	409,390
5	553	1057	383,340	10	484	1124	400,800

Solution
First, evaluate the values of the performance function at all ten sample points as −100,645, 26,390, 66,645, −76,745, 16,635, −67,505, −42,500, −9655, −69,700 and 8805. The number of safe samples is 6 and the reliability estimate from direct MCS is $R = 6/10 = 0.60$. Since direct MCS only employs a small number of samples ($M = 10$), the estimate may have a large variation, which is computed as

$$\sigma_P = \sqrt{\frac{R(1-R)}{M}} = \sqrt{\frac{0.6 \cdot 0.4}{10}} \approx 0.155$$

This indicates that the MC estimate contains a large variation that can be fully attributed to the small sample size. We can then compute the 95% confidence interval of the reliability estimate as [0.2962, 0.9038].

5.4.2 *Importance Sampling Method*

As can be observed from Eq. (5.28), direct MCS generally requires a large number M of samples to obtain a sufficiently small variance in the reliability estimate, especially in cases of high reliability (or low probability of failure). To alleviate this computational burden and reduce the variance in the reliability estimate, researchers have developed a wide variety of smart MCS methods, among which the most popular one is the *importance sampling method* [6–8]. The basic idea of importance sampling is to assign more sample points to the regions that have more impact on the probability of failure. If these regions are treated with greater importance by sampling more frequently, the variance in the resulting reliability or probability estimate can be reduced. Therefore, one of the most important elements in importance sampling is to choose an appropriate sampling distribution that encourages random samples to be placed in these regions. An example of such a sampling distribution is shown in the **U**-space in Fig. 5.8, where the sampling distribution centers at the MPP in the standard normal space. For a given number of random samples, more points are assigned to the failure region by the importance sampling method than by direct MCS; because direct MCS sampling distribution uses the means of random inputs (or the origin in the **U**-space) as the center.

To examine this, start with the formula for the probability of failure, expressed as

$$p_f = \int \cdots \int_{\Omega} I_{\Omega^F}(\mathbf{x}) f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} \approx \frac{1}{M} \sum_{j=1}^{M} I_{\Omega^F}(\mathbf{x}_j) \tag{5.30}$$

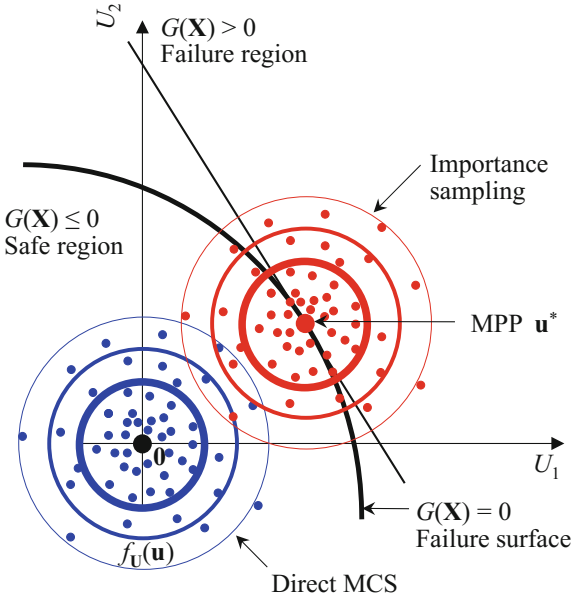


Fig. 5.8 Comparison between direct MCS and importance sampling

where the failure domain Ω^F is defined by the limit-state function $\Omega^F = \{\mathbf{X}: G(\mathbf{X}) > 0\}$. If we define a new sampling distribution $h_{\mathbf{X}}(\mathbf{x})$, such as the one in Fig. 5.8, we can rewrite the probability of failure as

$$p_f = \int \cdots \int_{\Omega} \left[I_{\Omega^F}(\mathbf{x}) \frac{f_{\mathbf{X}}(\mathbf{x})}{h_{\mathbf{X}}(\mathbf{x})} \right] h_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} p_f \approx \frac{1}{M} \sum_{j=1}^{M} I_{\Omega^F}(\mathbf{x}_j) \frac{f_{\mathbf{X}}(\mathbf{x}_j)}{h_{\mathbf{X}}(\mathbf{x}_j)} \tag{5.31}$$

It can be expected that, in the new sampling distribution, a random sample point will have a greater chance of the indicator value being one or, in other words, a greater chance of falling in the failure region. Therefore, we can expect a larger number of failure samples, which leads to a smaller variance in the estimate of the probability of failure.

5.5 Stochastic Response Surface Methods

Stochastic response surface methods (SRSMs) are capable of alleviating the computational burden required by sampling methods, while still maintaining comparable accuracy. This section introduces three state-of-the art SRSMs for uncertainty quantification and reliability analysis, namely the dimension reduction (DR) method [10–12], the stochastic spectral method [13–15], and the stochastic collocation method [16–19].

5.5.1 Dimension Reduction (DR) Method

The search for efficient computational procedures to deal with high-dimensional problems has led to the development of the *dimension reduction* (*DR*) method [10, 11, 21, 22]. Outside of the engineering design field, the DR method is widely known as the high-dimensional model representation (HDMR) method, which was originally developed for efficient multivariate model representation in chemical system modeling [21, 23–26]. This method approximates a multi-dimensional response function with a set of component functions with increasing numbers of random variables from a constant to multidimensional functions. For a system response with negligible high-order variate interaction, the HDMR method enables an efficient, yet accurate, formulation of this response function with low-order component functions (usually second-order or bivariate are sufficient). In fact, the responses of most practical physical systems are significantly affected by only low-order interactions of the input random variables. Depending on the way in which the component functions are determined, HDMR methods can be categorized into two types: ANOVA-HDMR and Cut-HDMR [21]. ANOVA-HDMR exactly follows the analysis of variance (ANOVA) procedure, and is useful for measuring the contributions of the variance of each component function to the output variance [27]. However, the multi-dimensional integrations involved in ANOVA-HDMR make this expansion computationally unattractive. On the other hand, Cut-HDMR expansion exactly represents the response function in the hyperplane that passes through a reference point in the input random space. This expansion does not require multi-dimensional integrations and is computationally much more efficient than ANOVA-HDMR. It is fair to say that the DR method is essentially Cut-HDMR designated for the purpose of reliability analysis. Specialized versions of this method include the *univariate dimension reduction* (*UDR*) method that simplifies one multi-dimensional response function to multiple one-dimensional component functions [10, 21] and the *bivariate dimension reduction* (*BDR*) method that simplifies one multi-dimensional response function to multiple one- and two-dimensional integrations [11, 22].

Fundamentals of Dimension Reduction (DR)

In the classical ANOVA decomposition, an N-dimensional, real-valued smooth stochastic response can be decomposed in a hierarchical and convergent manner as [11, 23]

$$G(\mathbf{x}) = G_0 + \sum_{i=1}^{N} G_i(X_i) + \sum_{1 \le i_1 < i_2 \le N} G_{i_1 i_2}(X_{i_1}, X_{i_2}) + \cdots + \sum_{1 \le i_1 < \cdots < i_s \le N} G_{i_1 \cdots i_S}(X_{i_1}, \ldots, X_{i_s}) + \cdots + G_{1 \cdots N}(X_1, \ldots, X_N) \tag{5.32}$$

Here, G_0 is a constant acting as the zeroth-order component function that represents the mean effect; $G_i(X_i)$ is a univariate function acting as the first-order component function that expresses the individual effect of x_i on the response $G(\mathbf{X})$, $G_{i_1 i_2}(X_{i_1}, X_{i_2})$ is a bivariate function acting as the second-order component function that describes the interactive effects of X_{i_1} and X_{i_2} on the response; the higher order terms give the interactive effects of increasing numbers of input random variables acting together to contribute to the response; and the last term accounts for any residual dependence of all of the input random variables cooperatively locked together to affect the response. Once we suitably determine all of the important component functions, the resulting decomposed model can be used as a stochastic response surface model to efficiently compute the response.

If we define a set of dimensional indices $\mathbf{u} \subseteq \mathcal{D}$ where $\mathcal{D} = \{1, \ldots, N\}$ denotes a set of all dimensional indices, we can obtain a more compact notation of the generalized dimension decomposition, expressed as [28]

$$G(\mathbf{X}) = \sum_{\mathbf{u} \in \mathcal{D}} G_{\mathbf{u}}(\mathbf{X}_{\mathbf{u}}) \tag{5.33}$$

Here, $G_{\mathbf{u}}$ denotes a $|\mathbf{u}|$-dimensional component function whose random dimensions correspond to the dimensional indices belonging to $\mathbf{u}$, where $|\mathbf{u}|$ is the number of indices in the set $\mathbf{u}$. For example, if we have $\mathbf{u} = \{1, 2, 4\}$, then $G_{\mathbf{u}} = G_{124}(X_1, X_2, X_4)$.

The component functions can be obtained by defining an appropriate product measure and an error functional and then minimizing this error functional [21, 23]. An efficient way is to choose the measure as the Dirac measure at a reference point $\boldsymbol{\mu}_{\mathbf{X}}$, leading to the Cut-HDMR decomposition as [21, 28]

$$\begin{aligned} G(\mathbf{X}) = G_0^C + \sum_{i=1}^{N} G_i^C(X_i) + \sum_{1 \le i_1 < i_2 \le N} G_{i_1 i_2}^C(X_{i_1}, X_{i_2}) \\ + \cdots + \sum_{1 \le i_1 < \cdots < i_s \le N} G_{i_1 \cdots i_S}^C(X_{i_1}, \ldots, X_{i_s}) + \cdots + G_{1 \cdots N}^C(X_1, \ldots, X_N) \end{aligned} \tag{5.34}$$

or in a more compact manner as

$$G(\mathbf{X}) = \sum_{\mathbf{u} \in \mathcal{D}} G_{\mathbf{u}}^C(\mathbf{X}_{\mathbf{u}}) \tag{5.35}$$

where the component functions are explicitly given as

$$\begin{aligned} G_0^C &= G(\boldsymbol{\mu}_{\mathbf{X}}), \\ G_i^C &= G(\mathbf{X})|_{\mathbf{X}=\boldsymbol{\mu}_{\mathbf{X}}\backslash X_i} - G_0^C, \\ G_{i_1 i_2}^C &= G(\mathbf{X})|_{\mathbf{X}=\boldsymbol{\mu}_{\mathbf{X}}\backslash\left(X_{i_1}, X_{i_2}\right)} - G_{i_1}^C - G_{i_2}^C - G_0^C, \\ &\cdots \end{aligned} \tag{5.36}$$

Here, the notation $\mathbf{x} = \boldsymbol{\mu}_{\mathbf{X}}\backslash X_i$ denotes the vector $\mathbf{X}$ with its components other than X_i being set equal to the corresponding components of the reference vector $\boldsymbol{\mu}$. A general recursive formula for the component functions can be derived as [21]

$$G_{\mathbf{u}}^C(\mathbf{X}_{\mathbf{u}}) = G(\mathbf{X})|_{\mathbf{X}=\boldsymbol{\mu}_{\mathbf{X}}\backslash\mathbf{X}_{\mathbf{u}}} - \sum_{\mathbf{v}\subset\mathbf{u}} G_{\mathbf{v}}^C(\mathbf{X}_{\mathbf{v}}) \tag{5.37}$$

and can also be more conveniently expressed as [29]

$$G_{\mathbf{u}}^C(\mathbf{X}_{\mathbf{u}}) = \sum_{\mathbf{v}\subseteq\mathbf{u}} (-1)^{|\mathbf{u}|-|\mathbf{v}|} G(\mathbf{X})|_{\mathbf{X}=\boldsymbol{\mu}_{\mathbf{X}}\backslash\mathbf{X}_{\mathbf{v}}} \tag{5.38}$$

where the notation $\mathbf{X} = \boldsymbol{\mu}_{\mathbf{X}}\backslash\mathbf{X}_{\mathbf{u}}$ denotes the vector $\mathbf{X}$ with its components other than those indices that belong to the set $\mathbf{u}$ being set equal to the corresponding components of the reference vector $\boldsymbol{\mu}$.

It is worth noting that we can derive the Cut-HDMR formulae from a Taylor series expansion of the response function at the reference point $\boldsymbol{\mu}_{\mathbf{x}}$ as [11]

$$\begin{aligned} G(\mathbf{X}) = G(\boldsymbol{\mu}_{\mathbf{X}}) &+ \sum_{j=1}^{\infty} \frac{1}{j!} \sum_{i=1}^{N} \frac{\partial^j G}{\partial X_i^j}(\boldsymbol{\mu}_{\mathbf{X}})\left(X_i - \mu_{X_i}\right)^j \\ &+ \sum_{j_1, j_2 \geq 1}^{\infty} \frac{1}{j_1! j_2!} \sum_{1 \leq i_1 < i_2 \leq N} \frac{\partial^{j_1+j_2} G}{\partial X_{i_1}^{j_1} \partial X_{i_2}^{j_2}}(\boldsymbol{\mu}_{\mathbf{X}})\left(X_{i_1} - \mu_{X_{i_1}}\right)^{j_1}\left(X_{i_2} - \mu_{X_{i_2}}\right)^{j_2} + \cdots \end{aligned} \tag{5.39}$$

We can see that any component function in the Cut-HDMR expansion accounts for an infinite number of Taylor series terms containing the same set of random variables as that component function. For example, the univariate decomposed component function $G_i^C(X_i)$ in Eq. (5.34) contains the univariate terms with X_i of any order in the Taylor series expansion, and so on. Thus, the dimension decomposition of any order in Eq. (5.34) should not be viewed as a Taylor series expansion of the same order, nor do they represent a limited degree of nonlinearity in $g(\mathbf{x})$. In fact, the dimension decomposition provides greater accuracy than a Taylor series expansion of the same or even higher order. In particular, the residual error in a univariate approximation to a multidimensional integration of a system response over a symmetric domain was reported to be far less than that of a second-order Taylor expansion method for probability analysis [11]. We also note that, to construct the dimension decomposition of a response function, or the Cut-HDMR, we

need to first define a reference point $\boldsymbol{\mu}_{\mathbf{X}} = (\mu_{X_1}, \mu_{X_2}, \ldots, \mu_{X_N})$ in the input random space. Regarding this issue, the work by Sobol [27] suggests that it is optimum to define the reference point as the mean value of the input random variables. Thus, this study will employ the mean point of the random inputs as the reference point.

The responses of most practical physical systems are significantly affected by only low-order interactions (usually up to the second-order) of the input random variables; the high-order interactions of these variables are often very weak. In these systems, a few lower-order component functions are sufficient to capture the response uncertainty. These considerations led to two well-known versions of Cut-HDMR, namely the univariate dimension reduction (UDR) method [10] and the bivariate dimension reduction (BDR) method [11]. Considering the component functions in Eq. (5.34), looking only up to the first-order yields the univariate decomposed response, expressed as

$$G_{\mathrm{U}}(\mathbf{X}) = G_0^C + \sum_{i=1}^{N} G_i^C(X_i) \tag{5.40}$$

Replacing the component functions with the formulae in Eq. (5.36) gives us the UDR formulation, expressed as

$$G_{\mathrm{U}}(\mathbf{X}) = \sum_{i=1}^{N} G(\mathbf{X})|_{\mathbf{x}=\boldsymbol{\mu}_{\mathbf{X}}\setminus X_i} - (N-1)G(\boldsymbol{\mu}_{\mathbf{X}}) \tag{5.41}$$

For example, if a response function $G(\mathbf{X})$ has three input random variables X_1, X_2, and X_3, the univariate decomposed response can be expressed as

$$\begin{aligned} G_{\mathrm{U}}(X_1, X_2, X_3) = {} & G\left(X_1, \mu_{X_2}, \mu_{X_3}\right) + G\left(\mu_{X_1}, X_2, \mu_{X_3}\right) \\ & + G\left(\mu_{X_1}, \mu_{X_2}, X_3\right) - 2G\left(\mu_{X_1}, \mu_{X_2}, \mu_{X_3}\right) \end{aligned} \tag{5.42}$$

Limiting the Cut-HDMR at the second-order gives us the bivariate decomposed response, expressed as

$$G_{\mathrm{B}}(\mathbf{X}) = G_0^C + \sum_{i=1}^{N} G_i^C(X_i)G(\mathbf{X}) + \sum_{1 \le i_1 < i_2 \le N} G_{i_1 i_2}^C(X_{i_1}, X_{i_2}) \tag{5.43}$$

Substituting the component functions with the formulae in Eq. (5.36) gives us the BDR formulation, expressed as

$$\begin{aligned} G_{\mathrm{B}}(\mathbf{X}) = {} & \sum_{1 \le i_1 < i_2 \le N} G(\mathbf{X})|_{\mathbf{x}=\boldsymbol{\mu}_{\mathbf{X}}\setminus\left(X_{i_1}, X_{i_2}\right)} \\ & - (N-2)\sum_{i=1}^{N} G(\mathbf{X})|_{\mathbf{x}=\boldsymbol{\mu}_{\mathbf{X}}\setminus X_i} + \frac{(N-1)(N-2)}{2} G(\boldsymbol{\mu}_{\mathbf{X}}) \end{aligned} \tag{5.44}$$

For the same response function $G(\mathbf{X})$ with three input random variables X_1, X_2, and X_3, the bivariate decomposed response can be expressed as

$$\begin{aligned} G_{\mathrm{U}}(X_1, X_2, X_3) = {} & G(X_1, X_2, \mu_{X_3}) + G(X_1, \mu_{X_2}, X_3) + G(\mu_{X_1}, X_2, X_3) \\ & - G(X_1, \mu_{X_2}, \mu_{X_3}) - G(\mu_{X_1}, X_2, \mu_{X_3}) - G(\mu_{X_1}, \mu_{X_2}, X_3) \\ & + G(\mu_{X_1}, \mu_{X_2}, \mu_{X_3}) \end{aligned} \tag{5.45}$$

To further predict the reliability or PDF of the response, the decomposed component functions need to be integrated or interpolated, followed by the use of a PDF generation technique (in the case of integration [11]) or the use of direct MCS (in the case of interpolation [22]). In what follows, the procedure for numerical interpolation is discussed in detail.

Numerical Interpolation for Component Function Approximation

Consider the UDR formula in Eq. (5.41) where the univariate component function can be approximated with function values at a set of univariate sample points, expressed as

$$\begin{aligned} G(\mathbf{X})|_{\mathbf{x}=\boldsymbol{\mu}_{\mathbf{X}} \setminus X_i} &= G(\mu_{X_1}, \ldots, \mu_{X_{i-1}}, X_i, \mu_{X_{i+1}}, \ldots, \mu_{X_N}) \\ &= \sum_{j=1}^{m} a_j(X_i) \cdot G\left(\mu_{X_1}, \ldots, \mu_{X_{i-1}}, x_i^{(j)}, \mu_{X_{i+1}}, \ldots, \mu_{X_N}\right) \end{aligned} \tag{5.46}$$

where m is the number of univariate sample points ($X_i = x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(m)}$), and $a_j(X_i)$ is the jth interpolation basis function. A widely used interpolation basis function is called the *Lagrange polynomial*. In Lagrange interpolation, a_j^i has the following form

$$a_j(X_i) = \frac{\prod_{k=1, k\neq j}^{m} \left(X_i - x_i^{(k)}\right)}{\prod_{k=1, k\neq j}^{m} \left(x_i^{(j)} - x_i^{(k)}\right)} \tag{5.47}$$

Repeating this interpolation for all univariate component functions in Eq. (5.36), we then have an explicit function approximation for the response function, expressed as

$$G_{\mathrm{U}}(\mathbf{X}) = \sum_{i=1}^{N} \sum_{j=1}^{m} a_j(X_i) \cdot G\left(\mu_{X_1}, \ldots, \mu_{X_{i-1}}, x_i^{(j)}, \mu_{X_{i+1}}, \ldots, \mu_{X_N}\right) - (N-1) G(\boldsymbol{\mu}_{\mathbf{X}}) \tag{5.48}$$

If we use the same number of sample points m for the Lagrange interpolation of all univariate component functions, we then need $(m-1)N + 1$ function evaluations for the UDR. An empirical sample point distribution for the UDR when $m = 3$ is

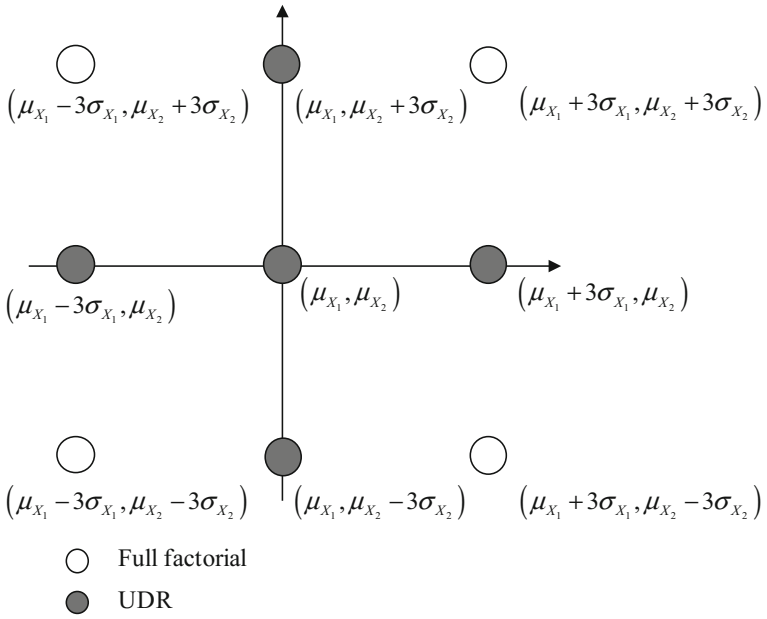


Fig. 5.9 Empirical sample point distribution for UDR (m = 3)

shown in Fig. 5.9. Also shown is the full factorial (or tensor-product) design (the number of sample points being m^N) without the use of UDR. It is apparent that, compared to the full factorial design, UDR achieves a significant reduction in the number of sample points.In the case of the BDR, the Lagrange interpolation of the bivariate component function can be expressed as

$$\begin{aligned} G(\mathbf{X})|_{\mathbf{x}=\boldsymbol{\mu}_{\mathbf{X}}\setminus(X_{i_1},X_{i_2})} &= G\left(\mu_{X_1},\ldots,\mu_{X_{i_1-1}},X_{i_1},\mu_{X_{i_1+1}},\ldots,\mu_{X_{i_2-1}},X_{i_2},\mu_{X_{i_2+1}},\ldots,\mu_{X_N}\right) \\ &= \sum_{j_2=1}^{m}\sum_{j_2=1}^{m} a_{j_1}(X_{i_1})a_{j_2}(X_{i_2})\cdot \\ &\quad G\left(\mu_{X_1},\ldots,\mu_{X_{i_1-1}},x_{i_1}^{(j_1)},\mu_{X_{i_1+1}},\ldots,\mu_{X_{i_2-1}},x_{i_1}^{(j_2)},\mu_{X_{i_2+1}},\ldots,\mu_{X_N}\right) \end{aligned} \tag{5.49}$$

Repeating this interpolation for all bivariate component functions in Eq. (5.44) gives us an explicit formula for the bivariate approximation of the response function, expressed as

$$
\begin{aligned}
G_{\mathrm{B}}(\mathbf{X}) = & \sum_{1 \le i_1 < i_2 \le N} \sum_{j_2=1}^{m} \sum_{j_2=1}^{m} a_{j_1}(X_{i_1}) a_{j_2}(X_{i_2}) \cdot \\
& G\left(\mu_{X_1}, \ldots, \mu_{X_{i_1-1}}, x_{i_1}^{(j_1)}, \mu_{X_{i_1+1}}, \ldots, \mu_{X_{i_2-1}}, x_{i_1}^{(j_2)}, \mu_{X_{i_2+1}}, \ldots, \mu_{X_N}\right) \\
& - (N-2) \sum_{i=1}^{N} \sum_{j=1}^{m} a_j(X_i) \cdot G\left(\mu_{X_1}, \ldots, \mu_{X_{i-1}}, x_i^{(j)}, \mu_{X_{i+1}}, \ldots, \mu_{X_N}\right) \\
& + \frac{(N-1)(N-2)}{2} G(\boldsymbol{\mu}_{\mathbf{X}})
\end{aligned}
\tag{5.50}
$$

Since we have $N(N-1)/2$ bivariate combinations and we need $(m-1)^2$ sample points for each bivariate combination (excluding the $m-1$ univariate sample points), the number of sample points for computing the bivariate component functions is $N(N-1)(m-1)^2/2$. Therefore, the total number of sample points required by the BDR is $N(N-1)(m-1)^2/2 + (m-1)N + 1$. Similarly, we can apply Lagrange interpolation to other versions of the DR method involving third- and higher-order component functions.

Monte Carlo Simulation for Uncertainty Quantification

Once Lagrange interpolation is completed for all component functions in the UDR or BDR, an approximate function $\hat{G}$ of the original response function G can be obtained by interpolation using Lagrange polynomials at a set of sample points. Thus, any probabilistic characteristics of $G(\mathbf{x})$, including statistical moments, reliability, and PDF, can be easily estimated by performing direct MCS. For example, any rth moment can be calculated as

$$
\begin{aligned}
\beta_r &\cong \int \hat{G}^r(\mathbf{x}) f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} \\
&= E\left(\hat{G}^r(\mathbf{x})\right) = \lim_{M \to \infty} \frac{1}{M} \sum_{j=1}^{M} \hat{G}^r\left(\mathbf{x}_j\right)
\end{aligned}
\tag{5.51}
$$

where β_{r} is the rth moment of the performance function $G(\mathbf{X})$; $f_{\mathbf{X}}(\mathbf{x})$ is the joint PDF; $\mathbf{x}_j$ is the jth realization of $\mathbf{X}$; and M is the sample size. For reliability estimation, we can define an approximate safe domain for the performance function g as

$$
\hat{\Omega}^S = \left\{\mathbf{x} : \hat{G}(\mathbf{x}) \le 0\right\} \tag{5.52}
$$

Therefore, the reliability R can also be estimated by performing MCS as

$$R \cong \int I_{\hat{\Omega}^S}(\mathbf{x}) f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} = E\left(I_{\hat{\Omega}^S}(\mathbf{x})\right) = \lim_{ns\to\infty} \frac{1}{M} \sum_{j=1}^{M} I_{\hat{\Omega}^S}(\mathbf{x}_j) \tag{5.53}$$

where $I[\cdot]$ is an indicator function of a "safe" or "fail" state such that

$$I_{\hat{\Omega}^S}(\mathbf{x}_j) = \begin{cases} 1, & \mathbf{x}_j \in \hat{\Omega}^S \\ 0, & \mathbf{x}_j \in \Omega \backslash \hat{\Omega}^S \end{cases} \tag{5.54}$$

It should be noted that the MCS performed here employs the explicit interpolation $\hat{G}$ instead of the original performance function G and is thus inexpensive. It is also noted that the approximation of the response function over the input domain allows for the derivation of any probabilistic characteristics (e.g., statistical moments, reliability, and PDF) based on the same set of sample points. This is desirable, especially in reliability-based robust design problems where both moment estimation and reliability analysis are required [30–32].

MATLAB Code for UDR-based SRSM

A 99-line MATLAB code that implements a UDR-based SRSM is provided in the Appendix. The method in the code first uses the UDR to decompose the multidimensional performance function into multiple one-dimensional univariate functions and then employs cubic spine interpolation to approximate the one-dimensional univariate functions. The two-step process results in a stochastic response surface, with which the MCS is then applied to obtain the full probabilistic characteristics (e.g., statistical moments, reliability, and PDF) of the performance function.

5.5.2 *Stochastic Spectral Method*

The *stochastic spectral method* [13] is an emerging technique for reliability analysis of complex engineering problems. This method uses a number of response samples and generates a stochastic response surface approximation with multi-dimensional polynomials over a random space. Once the explicit response surface is constructed, MCS is often used for reliability analysis due to its convenience. The most popular stochastic spectral method is the *polynomial chaos expansion* (*PCE*) method. The fundamentals and computational procedures of PCE are detailed next.

Fundamentals of PCE

The original Hermite polynomial chaos, also called the homogeneous chaos, was derived from the original theory of Wiener [14] for the spectral representation of

Table 5.4 Types of random inputs and corresponding generalized polynomial chaos basis

	Random variable	Polynomial chaos	Support
Continuous	Gaussian	Hermite	$(-\infty, +\infty)$
	Gamma (Exponential)	Generalized Laguerre (Laguerre)	$[0, +\infty)$
	Beta	Jacobi	[a, b]
	Uniform	Legendre	[a, b]
Discrete	Poisson	Charlier	{0, 1, …}
	Binomial	Krawtchouk	{0, 1, …, N}
	Negative binomial	Meixner	{0, 1, …}
	Hypergeometric	Hahn	{0, 1, …, N}

any second-order stochastic response in terms of Gaussian random variables. To improve the expansion convergence rate, Xiu and Karniadakis [15] extended the method, under the Askey polynomial scheme, to non-Gaussian random variables (e.g., gamma, uniform, and beta). The types of random variables and the corresponding orthogonal polynomial families are listed in Table 5.4. In the finite dimensional random space Ω, a second-order stochastic response G, can be expanded in a convergent series of generalized polynomial chaos basis as

$$\begin{aligned} G(\mathbf{X}) = c_0\Gamma_0 + \sum_{i_1=1}^{\infty} c_{i_1}\Gamma_1\left(\zeta_{i_1}(\mathbf{X})\right) \\ + \sum_{i_1=1}^{\infty}\sum_{i_2=1}^{i_1} c_{i_1 i_2}\Gamma_2\left(\zeta_{i_1}(\mathbf{X}), \zeta_{i_2}(\mathbf{X})\right) \\ + \sum_{i_1=1}^{\infty}\sum_{i_2=1}^{i_1}\sum_{i_3=1}^{i_2} c_{i_1 i_2 i_3}\Gamma_3\left(\zeta_{i_1}(\mathbf{X}), \zeta_{i_2}(\mathbf{X}), \zeta_{i_3}(\mathbf{X})\right) + \cdots \end{aligned} \tag{5.55}$$

where $\Gamma_n\left(\zeta_{i_1}(\mathbf{X}), \zeta_{i_2}(\mathbf{X}), \ldots, \zeta_{i_n}(\mathbf{X})\right)$ denotes the n-dimensional Askey-chaos of order n in terms of the random variables $\left\{\zeta_{i_1}, \zeta_{i_2}, \ldots, \zeta_{i_n}\right\}$. According to the Cameron-Martin theorem [33], the polynomial chaos expansion in Eq. (5.55) converges in the L_2 sense (the mean square sense).

For the purpose of notational convenience, Eq. (5.55) is often rewritten as

$$G(\mathbf{X}) = \sum_{i=0}^{\infty} s_i \Psi_i(\boldsymbol{\zeta}(\mathbf{X})), \quad \boldsymbol{\zeta} = \{\zeta_1, \zeta_2, \ldots\} \tag{5.56}$$

where there exists a one-to-one mapping between the polynomial basis functions Γ_n and Ψ_i, and the PCE coefficients s_i and $c_{i_1,\ldots,i_r}$.

If the random variables $\boldsymbol{\zeta}$ follow the standard normal distribution, the following expression can be used to obtain the corresponding univariate Hermite polynomials with an order p:

$$\Psi_p(\zeta) = (-1)^p e^{\zeta^2/2} \frac{\partial^p e^{-\zeta^2/2}}{\partial^p \zeta} \tag{5.57}$$

Using the expression above, we can easily derive the first five Hermite polynomials, expressed as

$$\Psi_0 = 1,\ \Psi_1 = \zeta,\ \Psi_2 = \zeta^2 - 1,\ \Psi_3 = \zeta^3 - 3\zeta,\ \Psi_4 = \zeta^4 - 6\zeta^2 + 3 \tag{5.58}$$

These polynomials are plotted in Fig. 5.10. Observe that higher-order Hermite polynomials generally exhibit higher degrees of nonlinearity.

The univariate Hermite polynomials serve as the foundation for constructing the multi-dimensional Hermite polynomials by taking tensor products. To do so, we first define a multi-index $\mathbf{p} = \{p_1, p_2, \ldots, p_N\}$ whose ith element i_k is the polynomial order corresponding to the ith standard normal variable ζ_i. We then define the modulus of the multi-index p as

$$|\mathbf{p}| = \sum_{i=1}^{N} p_i \tag{5.59}$$

This definition enables us to construct multi-dimensional Hermite polynomials of an order p as

$$\Psi_p^N(\boldsymbol{\zeta}) = \prod_{i=1}^{N} \Psi_{p_i}^1(\zeta_i) \tag{5.60}$$

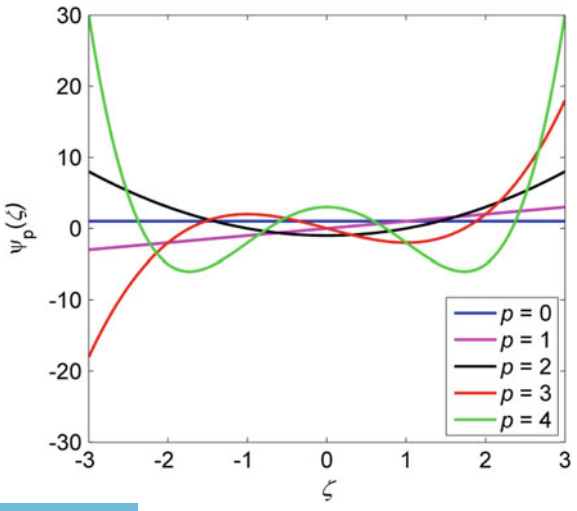


Fig. 5.10 First five Hermite polynomials ($p \leq 4$)

satisfying $|\mathbf{p}| = p$. For example, let us construct two-dimensional Hermite polynomials of the orders $|\mathbf{p}| \leq 3$. All multi-indices that satisfy this condition are listed as follows:

$$\mathbf{p} = \left[\begin{array}{c|cc|ccc|cccc} 0 & 1 & 0 & 2 & 1 & 0 & 3 & 2 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 2 & 0 & 1 & 2 & 3 \end{array}\right] \tag{5.61}$$

where the first and second rows contain the values for p_1 and p_2, respectively, and the vertical lines separate a lower-order $|\mathbf{p}|$ from the adjacent higher-order $|\mathbf{p}| + 1$. Accordingly, the third-order PCE has the following form as

$$\begin{aligned} G(\mathbf{X}) = \underbrace{s_0}_{|\mathbf{p}|=0} &+ \underbrace{s_1\zeta_1(\mathbf{X}) + s_2\zeta_2(\mathbf{X})}_{|\mathbf{p}|=1} + \underbrace{s_3\left(\zeta_1^2(\mathbf{X}) - 1\right) + s_4\zeta_1(\mathbf{X})\zeta_2(\mathbf{X}) + s_5\left(\zeta_2^2(\mathbf{X}) - 1\right)}_{|\mathbf{p}|=2} \\ &+ s_6\left(\zeta_1^3(\mathbf{X}) - 3\zeta_1(\mathbf{X})\right) + s_7\left(\zeta_1^2(\mathbf{X}) - 1\right)\zeta_2(\mathbf{X}) \\ &\underbrace{+ s_8\zeta_1(\mathbf{X})\left(\zeta_2^2(\mathbf{X}) - 1\right) + s_9\left(\zeta_2^3(\mathbf{X}) - 3\zeta_2(\mathbf{X})\right)}_{|\mathbf{p}|=3} \end{aligned} \tag{5.62}$$

The response surfaces of these multi-dimensional polynomials are plotted in Fig. 5.11, where we again see that higher-order polynomials exhibit higher degrees of nonlinearity.

The orthogonality of the Askey-chaos can be expressed as

$$E\left[\Psi_i\Psi_j\right] = \delta_{ij}E\left[\Psi_i^2\right] \tag{5.63}$$

where δ_{ij} is the Kronecker's delta and $E[\cdot]$ is the expectation operator. This property is very useful in computing the PCE coefficients, as will be discussed later. In engineering practice, it is impractical to consider an infinite summation in Eq. (5.56) and we often truncate the expansion up to a specific order p. All N-dimensional polynomials of orders not exceeding p result in the truncated PCE as follows (with P denoting the number of unknown PCE coefficients):

$$G(\mathbf{x}) = \sum_{i=0}^{P-1} s_i\Psi_i(\boldsymbol{\zeta}), \quad \mathbf{x} = \{x_1, x_2, \ldots, x_N\}, \quad \boldsymbol{\zeta} = \{\zeta_1, \zeta_2, \ldots \zeta_N\} \tag{5.64}$$

In the summation above, the number of unknown PCE coefficients P is

$$P = \binom{N+p}{p} = \frac{(N+p)!}{N!p!} \tag{5.65}$$

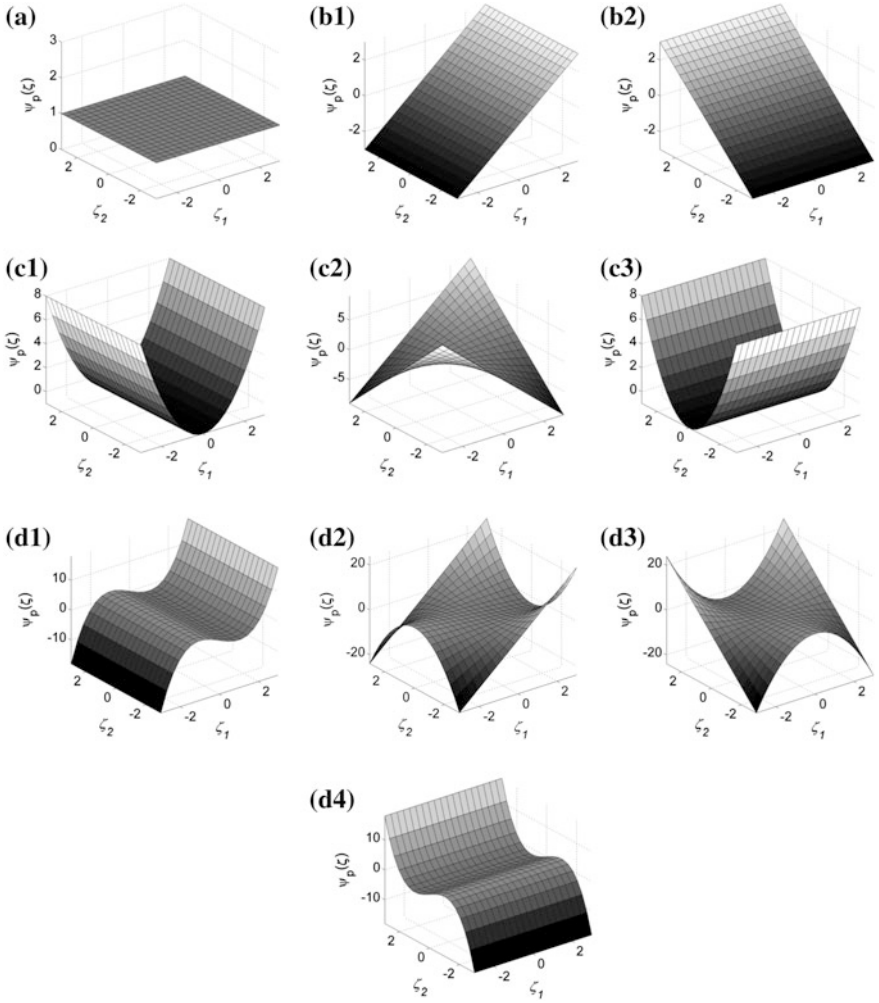


Fig. 5.11 Response surfaces of two-dimensional Hermite polynomials: **a** $|\mathbf{p}| = 0$; **b** $|\mathbf{p}| = 1$; **c** $|\mathbf{p}| = 2$; **d** $|\mathbf{p}| = 3$

Determination of PCE Coefficients

Our interest is in reliability analysis for the performance function G with random inputs $\mathbf{X}$. Since the uncertainty of a stochastic response G can be fully characterized by the PCE coefficients in Eq. (5.56), an efficient and accurate numerical procedure to compute the coefficients is essential for reliability analysis. In general, there are two non-intrusive methods, the projection method and the regression method; these methods only require the response values at a set of sample points.

Projection method

Based on the orthogonality of the polynomial chaos, the *projection method* [34, 35] can be used as a non-intrusive approach to compute the expansion coefficients of a response. Pre-multiplying both sides of Eq. (5.56) by $\Psi_j(\boldsymbol{\zeta})$ and taking the expectation gives the following equation

$$E\left[G(\mathbf{X})\Psi_j(\boldsymbol{\zeta})\right] = E\left[\sum_{i=0}^{\infty} s_i \Psi_i(\boldsymbol{\zeta})\Psi_j(\boldsymbol{\zeta})\right] \tag{5.66}$$

Due to the orthogonality of the polynomial chaos, Eq. (5.66) takes the form

$$s_j = \frac{E\left[G(\mathbf{X})\Psi_j(\boldsymbol{\zeta})\right]}{E\left[\Psi_j^2(\boldsymbol{\zeta})\right]} \tag{5.67}$$

In this expression, the denominator is readily obtained in an analytical form, while the numerator may require a multi-dimensional integration. This integration may be accomplished by full tensorization of a one-dimensional Gaussian quadrature [35], crude MCS [36], or through a Smolyak sparse grid [16]. The relative merits and disadvantages of these approaches are discussed below:

Approach 1: The full tensorization of a one-dimensional Gaussian quadrature exhibits fast convergence for smooth integrands. However, the computational cost grows exponentially with the dimension N: $M = M_1^N$, which is known as the "curse of dimensionality." Here, M denotes the total number of function evaluations and M_1 denotes the number of one-dimensional quadrature points. To prevent large integration errors, M_1 should be at least equal to the PCE order p.
Approach 2: Crude MCS is robust and has a convergence rate that is asymptotically independent of the dimension N [37]. However, the convergence is very slow (as $1/\sqrt{M}$). Thus, accurate results require a large number of function evaluations, which may incur an intolerable computational burden, especially for complex engineered systems that are computationally intensive.
Approach 3: The sparse grid collocation based on the Smolyak algorithm [16] offers an alternative way for multidimensional integration [38]. Compared with a fully tensorized quadrature, it also achieves fast convergence for smooth integrands but with much lower computational cost. Recently, adaptive algorithms [17, 39] have been developed that further reduce the computational cost. However, the sparse grid collocation methods still cannot fully resolve the difficulty induced by the "curse of dimensionality."

Regression method

The *regression method* is another non-intrusive approach that can be used to compute the PCE coefficients. The basic idea is to conduct a least-square regression of the exact performance function with respect to the basis functions $\Psi_j(\boldsymbol{\zeta})$, for

$j = 1, 2, \ldots, P-1$, in Eq. (5.64). Let us assume that we have a set of M regression points, $\boldsymbol{\zeta}_1, \boldsymbol{\zeta}_2, \ldots, \boldsymbol{\zeta}_M$, in the standard input space (e.g., standard normal space, standard uniform space) and correspondingly a set of M regression points, $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M$, in the original input space. The regression method computes the PCE coefficients by minimizing the mean square error (MSE) as

$$\mathbf{S} = \arg\min \frac{1}{M} \sum_{j=1}^{M} \left[G(\mathbf{x}_i) - \sum_{i=0}^{P-1} s_i \Psi_i(\boldsymbol{\zeta}_j) \right]^2 \tag{5.68}$$

Let us define an $M \times P$ information matrix $\boldsymbol{\Psi}$ whose elements are defined as $\boldsymbol{\Psi}_{ij} = \Psi_j(\boldsymbol{\zeta}_i)$, and an M-element response vector $\mathbf{G} = [G(\mathbf{x}_1), G(\mathbf{x}_2), \ldots, \mathrm{G}(\mathbf{x}_M)]\mathrm{T}$. Then, the solution to the minimization problem in (5.68) can be expressed as

$$\mathbf{S} = \left(\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi}\right)^{-1} \cdot \boldsymbol{\Psi}^{\mathrm{T}} \cdot \mathbf{G} \tag{5.69}$$

Now the remaining question is how to select the regression points. Let us first consider a univariate case ($N = 1$) where the PCE is truncated up to the degree p (clearly, $P = p + 1$). The optimum regression points are given by the roots of the orthogonal polynomial of the order $p + 1$, that is $\{r_1, r_2, \ldots, r_{p+1}\}$. For example, the optimum regression points for a third-order PCE ($p = 3$) with Hermite polynomials can be obtained by equating the fourth-order Hermite polynomial to zero, that is $\zeta^4 - 6\zeta^2 + 3 = 0$, which gives us the following points: −2.3344, −0.7420, 0.7420 and 2.3344. For a general multivariate case ($N > 1$), an optimum set of regression points can be obtained by applying the tensor-product formula to the univariate points $\{r_1, r_2, \ldots, r_{p+1}\}$, which can be explicitly expressed as

$$\mathbf{r}_k = (r_{i_1}, r_{i_2}, \ldots, r_{i_N}), \quad 1 \le i_1 \le i_2 \le \cdots \le i_N \le p+1 \tag{5.70}$$

for $k = 1, 2, \ldots, (p + 1)^N$. Note that, in cases of high dimensions (N) or high orders (p), the computational cost of a full tensor-product formula becomes intolerably expensive. The research efforts to address this issue have resulted in the idea of selecting a subset of the full tensor-product points [40, 41]. One way is to choose the first $(N - 1)P$ roots with the smallest Euclidean distances to the origin [40]. Another way is to sort the tensor-product points according to increasing Euclidean distances and adaptively add the point into the information matrix until $\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi}$ becomes invertible, which was reported to give less than $(N - 1)P$ regression points [41]. It should also be noted that, for very high expansion orders (e.g., above 20), the regression method with even full tensor-product points can encounter numerical instability, i.e., the term $\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi}$ is ill-conditioned. In such cases, we need to rely on the projection method to compute the PCE coefficients.

Recently, to estimate small failure probability, shifted and windowed Hermite polynomial chaos were proposed to enhance the accuracy of a response surface in the failure region [42]. Although the PCE method is considered to be accurate, the

primary drawback of the method is the curse of dimensionality, which substantially increases the computational cost as the number of random variables increases.

Example 5.4 This example examines Fortini's clutch, shown in Fig. 5.12. This problem has been extensively used in the field of tolerance design [43, 44]. As shown in Fig. 5.12, the overrunning clutch is assembled by inserting a hub and four rollers into the cage.

The contact angle y between the vertical line and the line connecting the centers of two rollers and the hub, is expressed in terms of the independent component variables, x_1, x_2, x_3, and x_4 as follows:

$$y(\mathbf{x}) = \arccos\left(\frac{x_1 + 0.5(x_2 + x_3)}{x_4 - 0.5(x_2 + x_3)}\right)$$

The statistical information of the random variables is summarized in Table 5.5. The limit-state function was defined as $G = y - c$, where c specifies a limit-state value and is set as 5° in this example.

Compute the statistics of the limit-state function and the reliability $R = \Pr(y - c \leq 0)$ using the first- and second-order PCEs and compare the results with those from direct MCS.

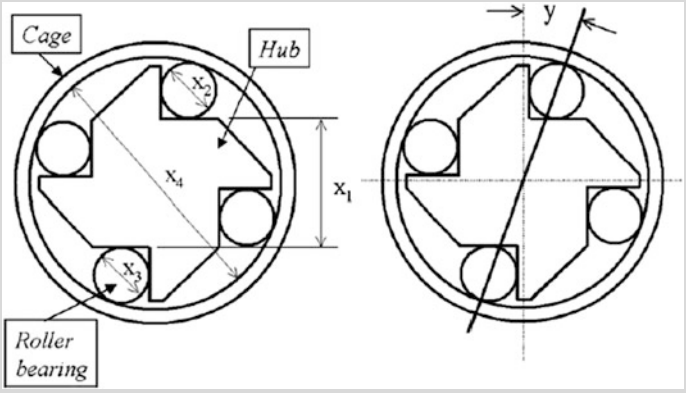


Fig. 5.12 Fortini's clutch. Reprinted (adapted) with permission from Ref. [44]

Table 5.5 Statistical information of the random variables in Example 5.4

Component	Distribution type	Mean (mm)	Std. dev. (mm)
x_1	Normal	55.29	0.0793
x_2	Normal	22.86	0.0043
x_3	Normal	22.86	0.0043
x_4	Normal	101.60	0.0793

Solution

Let us first compute the expansion coefficients for the first-order PCE. We solve $\zeta^2 - 1 = 0$ for the roots of the second-order Hermite polynomial, which gives us $r_1 = -1$ and $r_2 = 1$. We then obtain $2^4 = 16$ tensor-product regression points $\boldsymbol{\zeta}_j = (\zeta_{1,j}, \zeta_{1,j}, \zeta_{1,j}, \zeta_{1,j})$, for $j = 1, 2, \ldots, 16$. Using these, we construct the information matrix of the following form:

$$\boldsymbol{\Psi} = \begin{bmatrix} 1 & \zeta_{1,1} & \zeta_{2,1} & \zeta_{3,1} & \zeta_{4,1} \\ 1 & \zeta_{1,2} & \zeta_{2,2} & \zeta_{3,2} & \zeta_{4,2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \zeta_{1,16} & \zeta_{2,16} & \zeta_{4,16} & \zeta_{4,16} \end{bmatrix}$$

and the response matrix of the following form:

$$\mathbf{G} = \begin{bmatrix} G(\mathbf{x}_1(\boldsymbol{\zeta}_1)) \\ G(\mathbf{x}_2(\boldsymbol{\zeta}_2)) \\ \vdots \\ G(\mathbf{x}_{16}(\boldsymbol{\zeta}_{16})) \end{bmatrix}$$

Next, we conduct a least square regression using Eq. (5.69) to obtain the PCE coefficients and construct the PCE model as

$$\hat{G}_{|\mathbf{p}|=2} = 0.03465 - 0.00831\zeta_1 - 0.00045\zeta_2 - 0.00045\zeta_3 + 0.00826\zeta_4$$

Similarly, we can go through the same process and construct the second-order PCE model as

$$\begin{aligned} \hat{G}_{|\mathbf{p}|=3} = {} & 0.03466 - 0.00843\zeta_1 - 0.00046\zeta_2 - 0.00046\zeta_3 + 0.00837\zeta_4 \\ & - 0.00030\left(\zeta_1^2 - 1\right) + 0.00000\left(\zeta_2^2 - 1\right) + 0.00000\left(\zeta_3^2 - 1\right) \\ & - 0.00031\left(\zeta_4^2 - 1\right) - 0.00003\zeta_1\zeta_2 - 0.00003\zeta_1\zeta_3 + 0.00060\zeta_1\zeta_4 \\ & + 0.00000\zeta_2\zeta_3 + 0.00003\zeta_2\zeta_4 + 0.00003\zeta_3\zeta_4 \end{aligned}$$

The PDF approximations of the first- and second-order PCEs and direct MCS with 1,000,000 samples are compared in Fig. 5.13. Observe that the PDF approximation is improved from the first-order to the second-order through inclusion of the second-order orthogonal Hermite polynomials in the PCE model.

Table 5.6 summarizes the probability analysis results for the first- and second-order PCEs, and compares these with direct MCS. The second-order PCE produces more accurate results than its first-order counterpart but requires more computational effort (in terms of original performance function evaluations). The second-order PCE gives comparable accuracy to direct

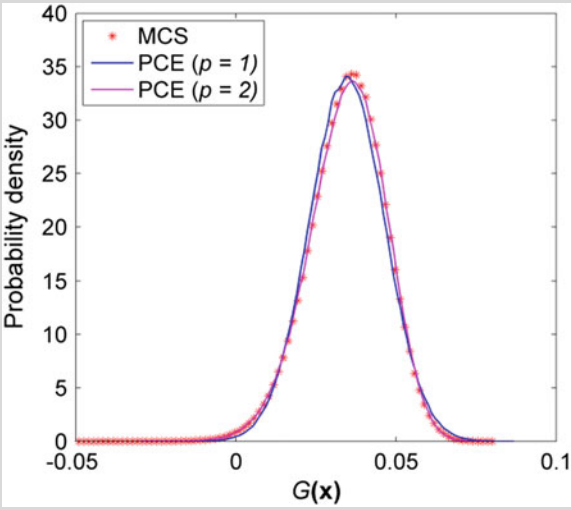


Fig. 5.13 PDF approximations for the Fortini's clutch example

Table 5.6 Probability analysis results in Example 5.4

	MCS	PCE (p = 1)	PCE (p = 2)
Mean (rad)	0.0346	0.0346	0.0346
Std. dev. (rad)	0.0118	0.0117	0.0119
Skewness	−0.3159	0.0044	−0.3062
Kurtosis	3.2936	2.9939	3.1232
Pr(y < 5 deg)	0.004856	0.001533	0.004484
No. FE	1,000,000	16	81

MCS but is far more efficient, which suggests that a PCE model with a sufficiently high order can be used as a better alternative to direct MCS for reliability.

5.5.3 *Stochastic Collocation Method*

Great attention has been paid to the *stochastic collocation method* for approximating a multi-dimensional random function due to its strong mathematical foundation and its ability to achieve fast convergence for interpolation construction. This method is another SRSM that approximates a multi-dimensional random function using function values given at a set of collocation points. In the stochastic collocation

method, the great improvement in reducing the curse of dimensionality in numerical integration was accomplished by Smolyak [16], who introduced the concept of the sparse grid. Compared to a full grid, the sparse grid achieves the same accuracy level for integration and interpolation but with a much smaller number of collocation points. The *dimension-adaptive tensor-product* (*DATP*) quadrature method introduced the concept of the generalized sparse grid and leveraged the dimensional importance indicated by an error estimator to adaptively refine the collocation points for efficient multi-dimensional integration [17]. Compared to the conventional sparse grid interpolation, the generalized sparse grid interpolation (i.e., the dimension-adaptive tensor-product interpolation) achieves a substantially higher convergence rate by detecting important dimensions and placing more collocation points in those dimensions. Klimke [18] further developed this work for hierarchical interpolation by using either piecewise multi-linear basis functions or Lagrangian polynomials. In this method, all dimensions in the random space are not considered of equal importance and an adaptive sampling scheme automatically detects the highly nonlinear dimensions and adaptively refines the collocation points in those dimensions. Recently, a further improvement of the adaptive capability of the DATP method was achieved in [45], where the authors introduced the concept of the directional sparse grid (DSG) and developed an asymmetric dimension-adaptive tensor-product (ADATP) method to detect both dimensional and directional importance; this has promising applications in reliability analysis. In this section, we present the stochastic collocation methods using the tensor-product grid, the conventional and generalized sparse grids, the hierarchical interpolation scheme using multivariate hierarchical basis functions, and the DSG.

In what follows, we will model the N-dimensional real random variables $\mathbf{X} = (x^1, x^2, \ldots, x^N)^{\mathrm{T}}$ in a complete probability space $(\Omega, \mathcal{A}, \mathcal{P})$, where Ω is a sample space, $\mathcal{A}$ is a σ-algebra on Ω, and $\mathcal{P}$ is a probability measure function $\mathcal{P}$: $\mathcal{A} \rightarrow [0, 1]$. Then, the probability density function (PDF) of the random variable x^i defines a probability mapping $f_i(x^i)$: $\Pi_i \rightarrow \mathbb{R}^+$, where the support Π_i is a one-dimensional random space of x^i. Note that, in this section, the superscript i is used to denote the index of the random variable; this differs from our previous discussions. Under the assumption of statistical independence, the probabilistic characteristics of the random variables $\mathbf{x}$ can then be completely defined by the joint PDF $f_{\mathbf{X}}(\mathbf{x}) = f_1(x^1) \cdot f_2(x^2) \cdot \cdots \cdot f_N(x^N)$ with the support of $\Pi = \Pi_1 \cdot \Pi_2 \cdot \cdots \cdot \Pi_N$. Since the construction of an interpolation in the stochastic collocation method often requires a specially bounded support $\Gamma = [0, 1]^N$ of the random variables $\mathbf{x}$, we first truncate any unbounded one-dimensional random space Π_i (e.g., in the case of a Gaussian random variable) to a bounded one $\Gamma_i^* = [c_i, d_i]$ that achieves a nearly full coverage of Π_i and then map any truncated one-dimensional support $[c_i, d_i]$ to $[0, 1]$, resulting in a bounded hypercube $\Gamma = [0, 1]^N$. Let $g(\mathbf{x})$ denote a smooth, measurable performance function on $(\Omega, \mathcal{A})$, which can be treated as a one-to-one mapping between the transformed N-dimensional random space and one-dimensional space G: $[0, 1]^N \rightarrow \mathbb{R}$. In general, the performance function $G(\mathbf{X})$ cannot be analytically obtained, and the function evaluation of G for a given input $\mathbf{x}$ requires expensive computer simulation. Therefore, it is important to employ a numerical method for reliability analysis

that is capable of producing accurate probabilistic characteristics of $g(\mathbf{x})$ with an acceptably small (i.e., computationally efficient) number of function evaluations.

Classical Stochastic Collocation: Tensor-Product Grid

The stochastic collocation method basically approximates the performance function G using N-dimensional interpolating functions with performance function values at a finite number of collocation points $\boldsymbol{\Theta} = \{\mathbf{x}_j | \mathbf{x}_j \in \Gamma, j = 1, \ldots, M_T\}$. Suppose that we can obtain the performance function value $G(\mathbf{x}_j)$ at each collocation point $\mathbf{x}_j$. We then aim to build an interpolation or surrogate model of the original performance function g by using the linear combinations of these function values $G(\mathbf{x}_j)$. The sampling process to construct this interpolation can be accomplished by using a tensor-product grid, a conventional sparse grid based on the Smolyak algorithm [16], or a generalized sparse grid based on the dimension-adaptive tensor-product algorithm [17]. We begin by constructing the interpolation using a tensor-product grid, or the tensor-product of one-dimensional interpolation formulas.

In the one-dimensional case ($N = 1$), we can construct the following one-dimensional interpolation

$$U^i(g) = \sum_{j=1}^{m_i} a_j^i \cdot G\left(x_j^i\right) \tag{5.71}$$

with a set of support nodes

$$X^i = \left\{ x_j^i | x_j^i \in [0,1], \quad j = 1, 2, \ldots, m_i \right\} \tag{5.72}$$

where $i \in \mathbb{N}$ is the interpolation level, $a_j^i \in C([0,1])$ is the jth interpolation nodal basis function, x_j^i is the jth support node, and m_i is the number of support nodes in the interpolation level i. Note that we use the superscript i to denote the interpolation level during the development of the stochastic collocation method. Two widely used nodal basis functions are the piecewise multi-linear basis function and the Lagrange polynomial. Here, we briefly describe the fundamentals of piecewise multi-linear basis functions. To achieve faster error decay, the Clenshaw-Curtis grid with equidistant nodes is often used for piecewise multi-linear basis functions [18]. In the case of a univariate interpolation ($N = 1$), the support nodes are defined as

$$\begin{aligned} m_i &= \begin{cases} 1 & \text{if } i = 1 \\ 2^{i-1} + 1, & \text{if } i > 1 \end{cases} \\ x_j^i &= \begin{cases} \frac{j-1}{m_i - 1} & \text{for } j = 1, \ldots, m_i \text{ if } m_i > 1 \\ 0.5 & \text{for } j = 1, \ldots, m_i \text{ if } m_i = 1 \end{cases} \end{aligned} \tag{5.73}$$

The resulting set of the points fulfills the nesting property $X^i \subset X^{i+1}$ that is very useful for the hierarchical interpolation scheme detailed later. Then, the univariate

piecewise multi-linear basis functions, supported by the Clenshaw-Curtis grid, can be expressed as [18]

$$a_j^i = 1 \quad \text{for } i = 1$$
$$a_j^i = \begin{cases} 1 - (m_i - 1) \cdot \left| x - x_j^i \right|, & \text{if } \left| x - x_j^i \right| < 1/(1 - m_i) \\ 0, & \text{otherwise} \end{cases} \tag{5.74}$$

for $i > 1$. More detailed information on the one-dimensional interpolation can be found in [18].

Applying the sequence of formulas in Eq. (5.71) to the original performance function G in a nested form for all N dimensions, we can easily derive the tensor-product of multiple one-dimensional interpolation formulas as the following multi-dimensional interpolation formula

$$\left(U^{i_1} \otimes \cdots \otimes U^{i_N}\right)(G) = \sum_{j_1=1}^{m_1} \cdots \sum_{j_N=1}^{m_N} \left(a_{j_1}^{i_1} \otimes \cdots \otimes a_{j_N}^{i_N}\right) \cdot G\left(x_{j_1}^{i_1}, \ldots, x_{j_N}^{i_N}\right) \tag{5.75}$$

where the superscript i_k, $k = 1, \ldots, N$, denotes the interpolation level along the kth dimension, U^{i_k} are the interpolation functions with the interpolation level i_k along the kth dimension, and the subscript j_k, $k = 1, \ldots, N$, denotes the index of a given support node in the kth dimension. The number of function evaluations required by the tensor-product formula reads

$$M_T = m_1 \cdot m_2 \cdots m_N \tag{5.76}$$

Suppose that we have the same number of collocation points in each dimension, i.e., $m_1 = m_2 = \cdots = m_N \equiv m$, and that the total number of tensor-product collocation points is $M_T = m^N$. Even if we only have three collocation points ($m = 3$) in each dimension, this number ($M_T = 3^N$) still grows very quickly as the number of dimensions is increased (e.g., $M_T = 3^{10} \sim 6 \times 10^4$, for $N = 10$). Thus, we need more efficient sampling schemes than the tensor-product grid to reduce the amount of computational effort for multi-dimensional interpolation. The search for such sampling schemes has resulted in sparse grid methods, the fundamentals of which are briefly introduced in subsequent sections.

Smolyak Algorithm: Conventional Sparse Grid

Compared to the classical tensor-product algorithm, the Smolyak algorithm achieves an order of magnitude reduction in the number of collocation points, while maintaining the approximation quality of the interpolation by imposing an inequality constraint on the summation of multi-dimensional indices [16]. This inequality leads to special linear combinations of tensor-product formulas such that the interpolation error remains the same as for the tensor-product algorithm.

The Smolyak formulas $A(q, N)$ are special linear combinations of tensor-product formulas. Using tensor-products of one-dimensional interpolation functions, the Smolyak algorithm constructs a sparse, multi-dimensional interpolation, expressed as

$$A_{q,N}(G) = \sum_{q-N+1 \le |\mathbf{i}| \le q} (-1)^{q-|\mathbf{i}|} \cdot \binom{N-1}{q-|\mathbf{i}|} \cdot \left(U^{i_1} \otimes \cdots \otimes U^{i_N}\right)(G) \tag{5.77}$$

where $\mathbf{i} = (i_1, \ldots, i_N)$ is the multi-index, and $|\mathbf{i}| = i_1 + \cdots + i_N$. The formula above indicates that the Smolyak algorithm builds the multi-dimensional interpolation by considering one-dimensional functions of interpolation levels $i_1, \ldots, i_N$ under the constraint that the sum of these interpolation levels lies within the range $[q - N + 1, q]$. Figure 5.14 shows an example of two-dimensional ($N = 2$) nodes derived from a sparse grid $A_{4,2}$ with $|\mathbf{i}| \le 4$ and from a tensor-product grid based on the same one-dimensional points. Observe that the number of points in the sparse grid is significantly smaller than that in the tensor-product grid. The 2D Clenshaw-Curtis grids for different levels of resolutions specified by different q values are plotted in Fig. 5.15.

With the incremental interpolant, $\Delta^i = U^i - U^{i-1}$, $U^0 = 0$, the Smolyak formulas can be equivalently written as

$$\begin{aligned} A_{q,N}(G) &= \sum_{|\mathbf{i}| \le q} \left(\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_N}\right)(G) \\ &= A_{q-1,N}(g) + \sum_{|\mathbf{i}| = q} \left(\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_N}\right)(G) \end{aligned} \tag{5.78}$$

The formulas above suggest that the Smolyak algorithm improves the interpolation by utilizing all of the previous interpolation formulas $A_{q-1,N}$ and the current incremental interpolant with the order q. If we select the sets of support nodes in a nested fashion (i.e., $X^i \subset X^{i+1}$) to obtain recurring points (e.g., the Clenshaw-Curtis grid) when extending the interpolation level from i to $i + 1$, we only need to compute function values at the differential grids that are unique to X^{i+1},

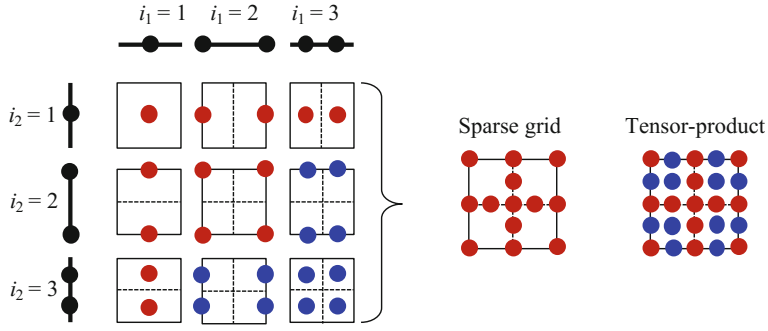


Fig. 5.14 Comparison of a sparse grid and a tensor-product grid (sparse grid for $|\mathbf{i}| \le 4$, tensor-product grid based on the same one-dimension points)

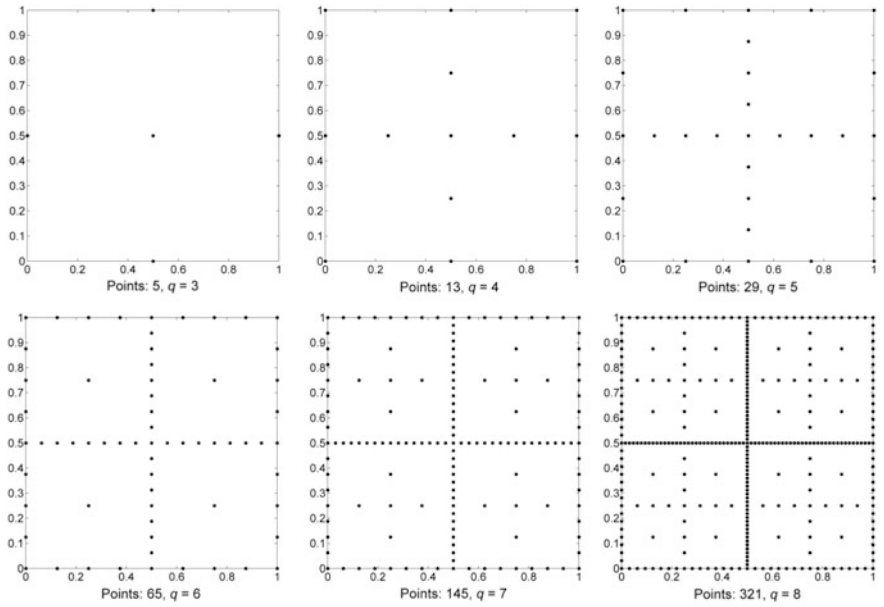


Fig. 5.15 Clenshaw-Curtis grids with different levels of resolution

$X_{\Delta}^{i+1} = X^{i+1} \backslash X^{i}$. In such cases, to build a sparse multi-dimensional interpolation with the order q, we only need to compute function values at the nested sparse grid

$$\begin{aligned} H_{q,N} &= \bigcup_{|\mathbf{i}| \le q} \left(X_{\Delta}^{i_1} \times \cdots \times X_{\Delta}^{i_N} \right) = H_{q-1,N} \cup \Delta H_{q,N} \\ \Delta H_{q,N} &= \bigcup_{|\mathbf{i}| = q} \left(X_{\Delta}^{i_1} \times \cdots \times X_{\Delta}^{i_N} \right) \end{aligned} \tag{5.79}$$

where $\Delta H_{q,N}$ denotes the grid points required to increase an interpolation order from $q - 1$ to q.

Although the Smolyak algorithm greatly reduces the number of collocation points for the multi-dimensional interpolation compared to the tensor-product algorithms, there is still a possibility that the number of function evaluations can be further reduced in cases where the performance function exhibits different degrees of nonlinearity in the stochastic dimensions. To achieve such a reduction, one must adaptively detect the dimensions with higher degrees of nonlinearity and assign more collocation points to those dimensions. This can be accomplished by using the dimension-adaptive tensor-product algorithm, which is detailed in the next subsection.

Dimension-Adaptive Tensor-Product Algorithm: Generalized Sparse Grid

For a given interpolation level l, the conventional sparse grid requires the index set $\mathbf{I}_{l,N} = \{\mathbf{i} \mid |\mathbf{i}| \le l + N\}$ to build the interpolation $A(l + N, N)$. If we loosen the

admissibility condition on the index set, we can construct the index set of the generalized sparse grid [17]. An index set $\mathbf{I}$ is called admissible if for all $\mathbf{i} \in \mathbf{I}$,

$$\mathbf{i} - \mathbf{e}_k \in \mathbf{I} \quad \text{for } 1 \leq k \leq N, \; i_k > 1 \tag{5.80}$$

Here, $\mathbf{e}_k$ is the kth unit vector. This admissibility condition still satisfies the telescopic property of the incremental interpolant $\Delta^i = U^i - U^{i-1}$. Thus, we can take advantage of the previous interpolation to construct a better interpolation by just sampling the differential grids that are unique to the finer interpolation, as shown in Eqs. (5.78) and (5.79). In each step of the algorithm, an error indicator is assigned to each multi-index $\mathbf{i}$. The multi-index $\mathbf{i}_t$ with the largest estimated error is selected for adaptive refinement, since it is possible that a larger error reduction can be achieved. The admissible indices in the forward neighborhood of $\mathbf{i}_t$ are added to the index set $\mathbf{I}$. The forward neighborhood of an index $\mathbf{i}$ can be defined as

$$\mathbf{I}_{\mathrm{F}}(\mathbf{i}) = \{\mathbf{i} + \mathbf{e}_k, \quad 1 \leq k \leq N\} \tag{5.81}$$

In each step, the newly added indices are called active indices and grouped as an active index set $\mathbf{I}_{\mathrm{A}}$; whereas, those indices whose forward neighborhood has been refined are called old indices and grouped as an old index set $\mathbf{I}_{\mathrm{O}}$. The overall index set $\mathbf{I}$ is comprised of the active and old index sets: $\mathbf{I} = \mathbf{I}_A \cup \mathbf{I}_{\mathrm{O}}$.

Note that in the dimension-adaptive algorithm, the generalized sparse grid construction allows for adaptive detection of the important dimensions, and thus a more efficient refinement compared to the conventional sparse grid interpolation. However, in engineering practice, not only different dimensions but also two opposite directions (positive and negative) within one dimension often demonstrate a large difference in response nonlinearity. In such cases, it is desirable to place more points in the direction with higher nonlinearity. The dimension-adaptive algorithm may not be appropriate for this purpose.

Hierarchical Interpolation Scheme

For dimension-adaptive interpolation, the hierarchical interpolation scheme provides a more convenient way for error estimation than the nodal interpolation scheme [18]. Here, we start by deriving the hierarchical interpolation formulae in the case of the univariate interpolation, which takes advantage of the nested characteristic of grid points (i.e., $X^i \subset X^{i+1}$). Recall the incremental interpolant, $\Delta^i = U^i - U^{i-1}$. Based on Eq. (5.71) and $U^{i-1}(G) = U^i(U^{i-1}(G))$, we can write [18]

$$\begin{aligned} \Delta^i(G) &= U^i(G) - U^i\left(U^{i-1}(G)\right) \\ &= \sum_{x_j^i \in X^i} a_j^i \cdot G\left(x_j^i\right) - \sum_{x_j^i \in X^i} a_j^i \cdot U^{i-1}(G)\left(x_j^i\right) \\ &= \sum_{x_j^i \in X^i} a_j^i \cdot \left(G\left(x_j^i\right) - U^{i-1}(G)\left(x_j^i\right)\right) \end{aligned} \tag{5.82}$$

Because for all $x_j^i \in X^{i-1}, G(x_j^i) - U^{i-1}(G)(x_j^i) = 0$, Eq. (5.82) can be rewritten as

$$\Delta^i(G) = \sum_{x_j^i \in X_\Delta^i} a_j^i \cdot \left(G\left(x_j^i\right) - U^{i-1}(G)\left(x_j^i\right) \right) \tag{5.83}$$

Because $X^i \subset X^{i+1}$, the number of grid points in X_Δ^i reads

$$m_\Delta^i = m^i - m^{i-1} \tag{5.84}$$

By denoting the jth element of X_Δ^i by x_j^i, Eq. (5.83) can be rewritten as

$$\Delta^i(G) = \sum_{j=1}^{m_\Delta^i} a_j^i \cdot \underbrace{\left(G\left(x_j^i\right) - U^{i-1}(G)\left(x_j^i\right) \right)}_{w_j^i} \tag{5.85}$$

Here, w_j^i is defined as the hierarchical surplus, which indicates the interpolation error of a previous interpolation at the node x_j^i of the current interpolation level i. The bigger the hierarchical surpluses, the larger the interpolation errors. For smooth performance functions, the hierarchical surpluses approach zero as the interpolation level goes to infinity. Therefore, the hierarchical surplus can be used as a natural candidate for error estimation and control. Figure 5.16 shows the comparison between the hierarchical and nodal basis functions with a piecewise linear spline and a Clenshaw-Curtis grid [18, 45]. Figure 5.17 illustrates the comparison between the hierarchical and nodal interpolation. Based on the Smolyak formula in Eq. (5.78), a multivariate hierarchical interpolation formula can be obtained as [18, 45].

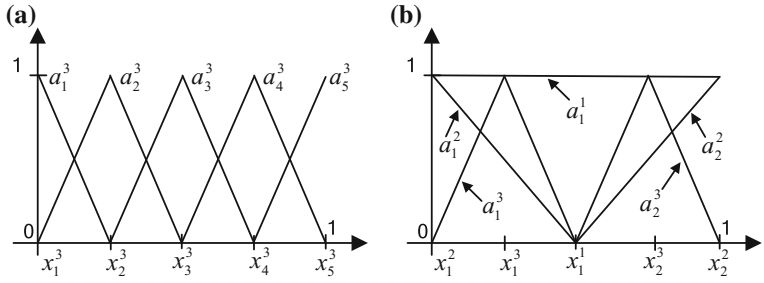


Fig. 5.16 Nodal basis functions $a_j^3, x_j^3 \in X^3$ (**a**) and hierarchical basis functions a_j^i with the support nodes $x_j^i \in X_\Delta^i$, $i = 1, 2, 3$ (**b**) for the Clenshaw-Curtis grid. Reprinted (adapted) with permission from Ref. [45]

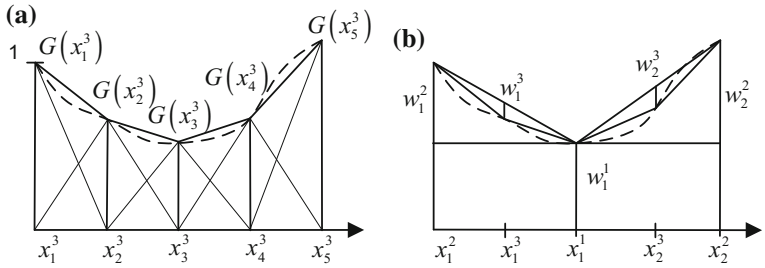


Fig. 5.17 Nodal (**a**) and hierarchical (**b**) interpolations in 1D. Reprinted (adapted) with permission from Ref. [45]

$$\begin{aligned} A_{q,N}(G) &= A_{q-1,N}(G) + \Delta A_{q,N}(G) \\ &= A_{q-1,N}(G) \\ &\quad + \sum_{|\mathbf{i}|=q} \sum_{\mathbf{j}} \underbrace{\left(a_{j_1}^{i_1} \otimes \cdots \otimes a_{j_N}^{i_N}\right)}_{a_{\mathbf{j}}^{\mathbf{i}}} \cdot \underbrace{\left(G\left(x_{j_1}^{i_1}, \ldots, x_{j_N}^{i_N}\right) - A_{q-1,N}(G)\left(x_{j_1}^{i_1}, \ldots, x_{j_N}^{i_N}\right)\right)}_{w_{\mathbf{j}}^{\mathbf{i}}} \end{aligned} \tag{5.86}$$

Asymmetric Dimension-Adaptive Tensor-Product Method

A recent development directed at enhancing the adaptive feature of the dimension-adaptive algorithm is the concept of the *directional sparse grid* (*DSG*) [45], which allows consideration of both directional and dimensional importance in multidimensional interpolation. For construction of the DSG, a conventional index i in the case of the univariate interpolation is decomposed into positive and negative directional indices (DIs) as $\mathbf{I}^D = \{i^+, i^-\}$. The positive (negative) DI i^+ (i^-) corresponds to the kth level interpolation points whose values are larger (smaller) than the value (0.5) of the center grid point. For the multivariate case ($N > 1$), we obtain a tensor-product formula of DI sets for a multi-index $\mathbf{i}$ as

$$\mathbf{I}^D = \mathbf{I}_1^D \times \cdots \times \mathbf{I}_N^D \tag{5.87}$$

where, $\mathbf{I}_k^D = \left\{i_k^+, i_k^-\right\}$, $1 \le k \le N$. Here, the forward neighborhood of a multi-dimensional DI $\mathbf{i}^d \in \mathbf{I}^D$ is defined as the N indices $\left\{\mathbf{i}^d + \mathbf{e}_k^{+/-}\right\}$, $1 \le k \le N$, and the sign of the kth directional unit vector $\mathbf{e}_k^{+/-}$ is the same as that of the kth element i_k^d of $\mathbf{i}^d$. An example of the tensor-product grid and a DSG in a two-dimensional space ($N = 2$) is shown in Fig. 5.18. Observe that the DI in a DSG divides the original index space into four quadrants; this division allows for an adaptive refinement of the quadrature points in these quadrants.

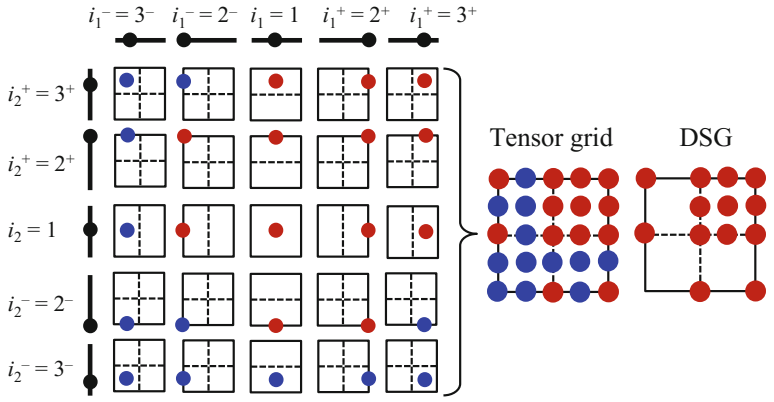


Fig. 5.18 Comparison of tensor-product (tensor) grid and a DSG in a two-dimensional space

Based on the proposed concepts of the DI and DSG, the overall procedure of ADATP interpolation is briefly summarized in Table 5.7. The relative error indicator used in the interpolation scheme can be defined for a DI $\mathbf{i}$ as

$$\varepsilon_r(\mathbf{i}) = \frac{1}{(G_{\max} - G_{\min})M_{\mathbf{i}}} \sum_{\mathbf{j}} \left| w_{\mathbf{j}}^{\mathbf{i}} \right| \tag{5.88}$$

where $w_{\mathbf{j}}^{\mathbf{i}}$ are the hierarchical surpluses of the collocation points $\mathbf{X}^{\mathbf{i}} = X_{\Delta}^{i_1} \times \cdots \times X_{\Delta}^{i_N}$, with $\mathbf{j} = (j_1, \ldots, j_N)$, $j_k = 1, \ldots, m_{\Delta}^{i_k}$, $1 \le k \le N$, and $M_{\mathbf{i}} = m_{\Delta}^{i_1} \cdot m_{\Delta}^{i_2} \cdot \cdots \cdot m_{\Delta}^{i_N}$. Note that, for simplicity, $\mathbf{i} = (i_1, \ldots, i_N)$ is used instead of $\mathbf{i}^d = \left(i_1^d, \ldots, i_N^d\right)$ to denote a multi-dimensional DI and that the term "index" in the

Table 5.7 Procedure for ADATP interpolation. Reprinted (adapted) with permission from Ref. [45]

Step 1	Set an initial interpolation level l $(q - N) = 0$; set the initial old index set $\mathbf{I}_O = \emptyset$ and the initial active index set $\mathbf{I}_A = \{\mathbf{i}\}$, where the initial active DI $\mathbf{i}$ = (1, …, 1) is the center point (0.5, …, 0.5); set an initial relative error indicator $\varepsilon_r(\mathbf{i}) = 1$
Step 2	Select a trial index set $\mathbf{I}_T$ (from $\mathbf{I}_A$) with the error indicator greater than a relative error threshold value ε_C; move the active index set $\mathbf{I}_A$ to the old index set $\mathbf{I}_O$. If $\mathbf{I}_T = \emptyset$, go to Step 7
Step 3	Select and remove the trial index $\mathbf{i}_t$ with the largest error indicator from $\mathbf{I}_T$; if none, go to Step 6. If the number of the collocation points M exceeds the maximum number $M_{\max}$, go to Step 7
Step 4	Generate the forward neighborhood $\mathbf{I}_F$ of $\mathbf{i}_t$ and add $\mathbf{I}_F$ to the active index set $\mathbf{I}_A$
Step 5	Compute the hierarchical surplus of each new added point based on the collocation points in the old index set and compute the error indicator of each active index. Go to Step 3
Step 6	Set an interpolation level $l = l + 1$ and go to Step 2
Step 7	Construct an explicit interpolation $\hat{G}$ of the performance function G

description of the ADATP method refers to the DI. Under the scheme of asymmetric sampling, it is expected that the error decay is at least as fast as that of the DATP interpolation.

Once the asymmetric dimension-adaptive sampling procedure is completed, an approximate function $\hat{G}$ of the original performance function G can be obtained by interpolation using the hierarchical basis functions at the collocation points. Thus, any probabilistic characteristics of $G(\mathbf{x})$, including statistical moments, reliability, and PDF, can be easily estimated by performing MCS, as described in Sect. 5.3.1.

Example 5.5 This example utilizes the V6 gasoline engine problem introduced by Lee [46]. The performance function considered in this example is the power loss (PL) due to the friction between the piston ring and the cylinder liner, oil consumption, blow-by, and liner wear rate. A ring/liner subassembly simulation model was used to compute the PL. The simulation model has four input parameters, ring surface roughness x_1, liner surface roughness x_2, linear Young's modulus x_3, and linear hardness x_4. Of the four total inputs, the first two, ring surface roughness x_1 and linear surface roughness x_2, were treated as random inputs following normal distributions with mean 4.0 and 6.119 μm, respectively, and with unit variance. The other two inputs, linear Young's modulus x_3 and linear hardness x_4, were treated as deterministic inputs fixed at 80 GPa and 240 BHV, respectively. It has been shown in [46] that the PL has a bimodal PDF.

Compare the accuracy of the UDR, PCE, and ADATP methods in reproducing the PDF of the PL, and the accuracy of FORM, UDR, PCE, and ADATP methods in estimating the reliability defined as $P(\text{PL} \leq 0.3)$.

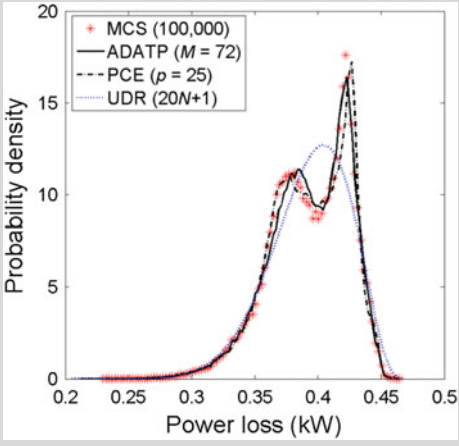


Fig. 5.19 PDF approximations for the V6 engine example. Reprinted (adapted) with permission from Ref. [45]

Table 5.8 Probability analysis results for the V6 engine example . Reprinted (adapted) with permission from Ref. [45]

	ADATP	MCS	PCE ($p = 25$)	20 $N + 1$ UDR	FORM
Mean (kW)	0.3935	0.3935	0.3934	0.3935	–
Std. dev. (kW)	0.0311	0.0310	0.0311	0.0314	–
Skewness	−0.6062	−0.5883	−0.5742	−0.5393	–
Kurtosis	3.0567	3.0828	3.0566	3.0974	–
$P(\text{PL} \leq 0.3)$	0.0055	0.0054 (± 0.0005[a])	0.0057	0.0048	0.0057
No. FE	72	100,000	625	41	15

[a]Error bounds computed with a 95% confidence level

Solution

To predict the bimodal shape of the PDF, the ADATP method uses $\varepsilon_C = 0.005$, $M_{\max} = 70$, and cubic Lagrange splines [47] as the hierarchical basis functions. Figure 5.19 shows the PDF approximations of the 25th order PCE with a fully tensorized Gauss–Hermite quadrature ($m_I = 25$), the UDR method, the ADATP method, and MCS. Both the ADATP and the PCE methods provide reasonably accurate approximations of the irregularly shaped PDF, while the UDR method fails to represent the irregular shape of this PDF. The probability analysis results shown in Table 5.8 suggest that the number of function evaluations of the ADATP method is much smaller than that of the PCE method with a fully tensorized Gaussian quadrature. In this example, the FORM requires the smallest number of function evaluations while still producing a good reliability estimate. The small error produced by FORM is due to the nonlinearity of the power loss function. However, FORM cannot be used for cases that require the construction of a complete PDF and subsequent uncertainty propagations.

5.6 Exercises

5.1 Consider the cantilever beam-bar system in Problem 4.1 (see Fig. 4.9) in Chap. 4. Suppose that a failure mode consists of two failure events: the formation of a hinge at the fixed point of the beam (event $\bar{E}_1$), followed by the formation of another hinge at the midpoint of the beam (event $\bar{E}_3$). The two safety events can be expressed as:

Table 5.9 Statistical information of random variables for Problem 5.1

Random variable	M	X	L
Distribution	Normal	Normal	Normal
Mean	150	μ_X	5.0
Standard deviation	30	20	0.05

$$E_1 = \{X, L, M | 3LX/8 - M \leq 0\},$$
$$E_3 = \{X, L, M | LX/3 - M \leq 0\}$$

The statistical information of the independent input random variables X, L, and M is given in Table 5.9.

(1) Assuming $\mu_X = 60$, compute the reliabilities R_1 and R_3 corresponding to E_1 and E_3 using the following methods, and compare the results in terms of accuracy and efficiency.

(a) Method 1: Direct MCS with 1 million random samples.
(b) Method 2: FORM (HL-RF).
(c) Method 3: UDR with the number of univariate sample points $m = 5$.
(d) Method 4: Second-order ($p = 2$) PCE with the expansion coefficients computed using the regression method.

(2) Assuming $\mu_X = 10$, re-compute the reliabilities R_1 and R_3 corresponding to E_1 and E_3 using the methods listed in (1) and compare the results in terms of accuracy and efficiency.

5.2 Recall Problem 4.2 in Chap. 4. Suppose S, X, Y, w, and t are independent random variables whose means and standard deviations are summarized in Table 5.10.

(1) Determine the PDF (or CDF) of G and estimate its reliability, defined as $P(G \leq 0)$, using (a) direct MCS (with 1 million random samples), (b) the first-order expansion method and (c) the DR-based method (UDR with the number of univariate sample points $m = 5$). Discuss and explain your conclusions.

(2) Assume that the mean of S degrades in the following manner: $\mu_S(t) = 400{,}000 - 100t$ while the variation of S remains unchanged. Estimate the time-dependent reliabilities at $t = 0, 100, 200, \ldots, 1000$ using the three methods in (1) and plot these estimates against time for all the methods.

Table 5.10 Statistical information for the random variables in Problem 5.2

Random variables	X [lb]	Y [lb]	S [psi]	w [in]	t [in]
Distribution	Normal	Normal	Normal	Normal	Normal
Mean	500	1000	400,000	2	1
Standard deviation	100	100	20,000	0.02	0.01

5.3 Consider the following simply supported beam subject to a uniform load, as illustrated in Fig. 5.20. Suppose $L = 5$ m, and two random variables EI (X_1) and w (X_2) are independent and follow normal distributions with means and standard deviations summarized in Table 5.11.
The maximum deflection of the beam is shown as

$$Y(X_1, X_2) = -\frac{5X_2L^4}{384X_1}$$

Failure is defined as when $Y < y_c = -3 \times 10^{-3}$ m. Determine the PDF (or CDF) of the maximum deflection and estimate its reliability using (a) direct MCS (with 1 million random samples), (b) the first-order expansion method, and (c) the MPP-based method (FORM with HL-RF). Discuss and explain your conclusions.

5.4 Consider a mathematical function

$$G(\mathbf{X}) = \exp\left[\sum_{k=1}^{N=5}(X_k - 1)^2 - \sum_{k=3}^{N=5} X_k^2 X_{k-1}^2 X_{k-2}^2\right]$$

where the five random variables X_1–X_5 are assumed to be statistically independent and uniformly distributed between 0 and 1. Compute the mean μ_G and standard deviation σ_G of $G(\mathbf{X})$ using the following methods, and compare the results in terms of accuracy and efficiency.

(a) Method 1: Direct MCS with 1 million random samples.
(b) Method 2: UDR based on Gauss-Legendre quadrature [48] with the number of one-dimensional quadrature points $m_I = 5$.

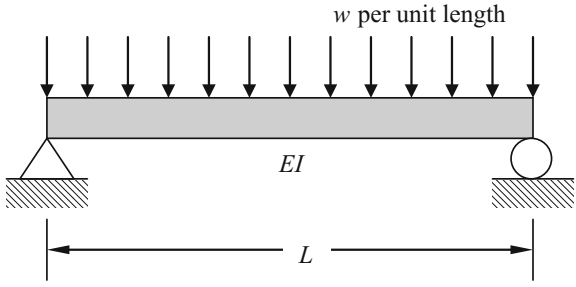


Fig. 5.20 Simply supported beam subject to a uniform load

Table 5.11 Statistical information of random variables for Problem 5.3

Random variable	X_1 [N m^2]	X_2 [N/m]
Distribution	Normal	Normal
Mean	30,000,000	10,000
Standard deviation	100,000	1000

(c) Method 3: BDR based on fully tensorized Gauss-Legendre quadrature [48] with the number of one-dimensional quadrature points $m_I = 5$.
(d) Method 4: ADATP with the relative error threshold $\varepsilon_C = 0.10$, the maximum number of collocation points $M_{max} = 120$, and cubic Lagrange splines as the hierarchical basis functions.

Appendix: A 99-Line MATLAB Code for UDR-Based SRSM

```
% A 99-LINE UDR-BASED SRSM CODE WRITTEN BY HU C., Wang P., AND YOUN B.D.  %
function UDR_RS()
clear all; close all;
u = [0.4  0.4];              %% Mean vector of random variables
s = [0.01 0.01];             %% Standard deviation vector


%======================== Generate MCS samples ========================%
ns = 5000000;                %% Number of MCS samples
nv = length(u);              %% Number of input random variables
xs = zeros(nv,ns);           %% Initialization of MCS sample vector
for k = 1:nv
    xs(k,:) = normrnd(u(k),s(k),1,ns);
end

%====================== Obtain Univariate Samples ======================%
[output,input,gg] = UDR_sampling(u,s);

%================= Compute Univariate Response Surface =================%
% Step 1: obtain univariate component function values
uniComp = zeros(nv,ns);
for k = 1:nv
   uniComp(k,:) = interp1(input(k,:),output(k,:),xs(k,:),'spline');
end
% Step 2: combine univariate responses with UDR formula
zz = squeeze(uniComp(:,:));
response_URS = sum(zz,1)-(nv-1)*gg;

%================ Compute True Responses by Direct MCS =================%
response_true = findresponse(xs);

%================== Conduct Reliability Analysis =======================%
rel_URS = length(find(response_URS < 0))/ns
```

```
rel_true = length(find(response_true < 0))/ns

%=============== Plot Probability Density Functions ==================%
% Plot PDF computed from univariate response surface
figure('DefaultAxesFontSize',16)
D = response_URS;
[cn,xout] = hist(D,100);
sss = sum(cn);
unit=(xout(100)-xout(1))/99;
for k = 1:100
    cn(k)=cn(k)/sss/unit;
end
plot(xout,cn,'k-'); hold on;
clear D cn xout;

% Plot true PDF from direct MCS
D = response_true;
[cn,xout] = hist(D,100);
sss = sum(cn);
unit=(xout(100)-xout(1))/99;
for k = 1:100
   cn(k)=cn(k)/sss/unit;
end
plot(xout,cn,'r*'); hold on;
clear D cn xout;
legend('UDR-SRSM','MCS');
xlabel('{\itG}({\bfX})'); ylabel('Probability density');

%=============== Obtain Univariate Samples ==========================%
function [output,input,gg] = UDR_sampling(u,s)
u_loc = [-3.0,-1.5,0,1.5,3.0]; %% Sample locations: [u+/-3.0s, u+/-
1.5s, u]
nv = length(u);              %% Dimension of the problem
m = length(u_loc);           %% Number of samples along each dimension
input = zeros(nv,m);
for k = 1:nv
   % Identify sample location
   input(k,:) = u(k) + u_loc*s(k);

   % Get response values
   xx = u;
   for kk = 1:m
```

```
      xx(k) = input(k,kk);
      if isequal(k,1) && isequal(xx,u)  %% Avoid re-evaluating mean value
         output(k,kk) = findresponse(xx);
         gg = output(k,kk);
      elseif ~isequal(k,1) && isequal(xx,u)
         output(k,kk) = gg;
      else
         output(k,kk) = findresponse(xx);
      end
   end
end
%=================== Define Performance Function ======================%
function response = findresponse(xx)
if isvector(xx) == 1
   response = 0.75*exp(-0.25*(9*xx(1)-2)^2-0.25*(9*xx(2)-2)^2)....
      +0.75*exp(-(9*xx(1)+1)^2/49-(9*xx(2)+1)/10)....
      +0.50*exp(-0.25*(9*xx(1)-7)^2-0.25*(9*xx(2)-3)^2)....
      -0.20*exp(-(9*xx(1)-4)^2-(9*xx(2)-7)^2) - 0.6;
else
   response = 0.75*exp(-0.25*(9*xx(1,:)-2).^2-0.25*(9*xx(2,:)-2).
^2)....
      +0.75*exp(-(9*xx(1,:)+1).^2/49-(9*xx(2,:)+1)/10)....
      +0.50*exp(-0.25*(9*xx(1,:)-7).^2-0.25*(9*xx(2,:)-3).^2)....
      -0.20*exp(-(9*xx(1,:)-4).^2-(9*xx(2,:)-7).^2) - 0.6;
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%% Notes to the Code %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                         %
% This is a compact code for reliability analysis based on the stochastic %
% response surface method (SRSM) which first uses the univariate          %
% dimension reduction (UDR) to decompose the multidimensional performance %
% function into multiple one-dimensional univariate functions and then    %
% employs the cubic spine interpolation to approximate the one-           %
% dimensional univariate functions. One mathematical example is solved    %
% for demonstration purpose.                                              %
%                                                                         %
% VARIABLE DEFINITION:                                                    %
% nv: number of random variables; ns: number of MCS samples;              %
% u: mean vector of random variables; s: standard deviation vector;       %
% uniComp: univariate component function values                           %
%                                                                         %
% FUNCTION DEFINITION                                                     %
% UDR_RS(): main function                                                 %
% UDR_sampling(): identify locations of univariate samples                %
% findresponse(): define performance function                             %
%                                                                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

References

1. Haldar, A., & Mahadevan, S. (2000). *Probability, reliability, and statistical methods in engineering design*. New York: Wiley.
2. Hasofer, A. M., & Lind, N. C. (1974). Exact and invariant second-moment code format. *Journal of Engineering Mechanics Division ASCE, 100,* 111–121.
3. Breitung, K. (1984). Asymptotic approximations for multinormal integrals. *Journal of Engineering Mechanics, ASCE, 110*(3), 357–366.
4. Tvedt, L. (1984). *Two second-order approximations to the failure probability. Section on structural reliability*. Hovik, Norway: A/S vertas Research.
5. Rubinstein, R. Y. (1981). *Simulation and the Monte Carlo method*. New York: Wiley.

6. Fu, G., & Moses, F. (1988). Importance sampling in structural system reliability. In *Proceedings of ASCE Joint Specialty Conference on Probabilistic Methods*, Blacksburg, VA (pp. 340–343).
7. Au, S. K., & Beck, J. L. (1999). A new adaptive importance sampling scheme for reliability calculations. *Structural Safety, 21*(2), 135–158.
8. Hurtado, J. E. (2007). Filtered importance sampling with support vector margin: A powerful method for structural reliability analysis. *Structural Safety, 29*(1), 2–15.
9. Naess, A., Leira, B. J., & Batsevych, O. (2009). System reliability analysis by enhanced Monte Carlo simulation. *Structural Safety, 31*(5), 349–355.
10. Rahman, S., & Xu, H. (2004). A univariate dimension-reduction method for multi-dimensional integration in stochastic mechanics. *Probabilistic Engineering Mechanics, 19,* 393–408.
11. Xu, H., & Rahman, S. (2004). A generalized dimension-reduction method for multi-dimensional integration in stochastic mechanics. *International Journal of Numerical Methods in Engineering, 61,* 1992–2019.
12. Youn, B. D., Zhimin, X., & Wang, P. (2007). Eigenvector Dimension Reduction (EDR) method for sensitivity-free probability analysis. *Structural and Multidisciplinary Optimization, 37,* 13–28.
13. Ghanem, R. G., & Spanos, P. D. (1991). *Stochastic finite elements: A spectral approach*. New York: Springer.
14. Wiener, N. (1938). The homogeneous chaos. *American Journal of Mathematics, 60*(4), 897–936.
15. Xiu, D., & Karniadakis, G. E. (2002). The Wiener-Askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing, 24*(2), 619–644.
16. Smolyak, S. (1963). Quadrature and interpolation formulas for tensor product of certain classes of functions. *Soviet Mathematics—Doklady, 4*, 240–243.
17. Grestner, T., & Griebel, M. (2003). Dimension-adaptive tensor-product quadrature. *Computing, 71*(1), 65–87.
18. Klimke, A. (2006). *Uncertainty modeling using fuzzy arithmetic and sparse grids* (Ph.D. thesis). Universität Stuttgart, Shaker Verlag, Aachen.
19. Ganapathysubramanian, B., & Zabaras, N. (2007). Sparse grid collocation schemes for stochastic natural convection problems. *Journal of Computational Physics, 225*(1), 652–685.
20. Law, A. M., & Kelton, W. D. (1982). *Simulation modeling and analysis*. New York: McGraw-Hill.
21. Rabitz, H., Alis, O. F., Shorter, J., & Shim, K. (1999). Efficient input–output model representations. *Computer Physics Communications, 117*(1–2), 11–20.
22. Xu, H., & Rahman, S. (2005). Decomposition methods for structural reliability analysis. *Probabilistic Engineering Mechanics, 20,* 239–250.
23. Rabitz, H., & Alis, O. F. (1999). General foundations of high dimensional model representations. *Journal of Mathematical Chemistry, 25*(2–3), 197–233.
24. Alis, O. F., & Rabitz, H. (2001). Efficient implementation of high dimensional model representations. *Journal of Mathematical Chemistry, 29*(2), 127–142.
25. Li, G., Rosenthal, C., & Rabitz, H. (2001). High dimensional model representations. *Journal of Physical Chemistry A, 105,* 7765–7777.
26. Li, G., Wang, S. W., & Rabitz, H. (2001). High dimensional model representations generated from low dimensional data samples—I: Mp-Cut-HDMR. *Journal of Mathematical Chemistry, 30*(1), 1–30.
27. Sobol, I. M. (2003). Theorems and examples on high dimensional model representations. *Reliability Engineering and System Safety, 79*(2), 187–193.
28. Griebel, M., & Holtz, M. (2010). Dimension-wise integration of high-dimensional functions with applications to finance. *Journal of Complexity, 26*(5), 455–489.
29. Kuo, F. Y., Sloan, I. H., Wasilkowski, G. W., & Wozniakowski, H. (2010). On decompositions of multivariate functions. *Mathematics of Computation, 79,* 953–966.

30. Youn, B. D., Choi, K. K., & Yi, K. (2005). Reliability-based robust design optimization using the performance moment integration method and case study of engine gasket-sealing problem. In *Proceedings of SAE 2005 World Congress*, Detroit, MI, United States.
31. Youn, B. D., Choi, K. K., & Yi, K. (2005). Performance Moment Integration (PMI) method for quality assessment in reliability-based robust design optimization. *Mechanics Based Design of Structures and Machines, 33,* 185–213.
32. Lee, S. H., Chen, W., & Kwak, B. M. (2009). Robust design with arbitrary distributions using Gauss-type quadrature formula. *Structural and Multidisciplinary Optimization, 39*(3), 227–243.
33. Cameron, R. H., & Martin, W. T. (1947). The orthogonal development of nonlinear functionals in series of Fourier–Hermite functionals. *Annals of Mathematics, 48,* 385–392.
34. Le Maître, O. P., Knio, O. M., Najm, H. N., & Ghanem, R. G. (2001). A stochastic projection method for fluid flow—I. Basic formulation. *Journal of Computational Physics, 173,* 481–511.
35. Le Maître, O. P., Reagan, M., Najm, H. N., Ghanem, R. G., & Knio, O. M. (2002). A stochastic projection method for fluid flow—II. Random process. *Journal of Computational Physics, 181,* 9–44.
36. Field, R. V. (2002). Numerical methods to estimate the coefficients of the polynomial chaos expansion. In *Proceedings of the 15th ASCE Engineering Mechanics Conference*.
37. Xiu, D. (2009). Fast numerical methods for stochastic computations: A review. *Communications in Computational Physics, 5,* 242–272.
38. Gerstner, T., & Griebel, M. (1998). Numerical integration using sparse grids. *Numerical Algorithms, 18*(3), 209–232.
39. Ma, X., & Zabaras, N. (2009). An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations. *Journal of Computational Physics, 228,* 3084–3113.
40. Berveiller, M. (2005). *Stochastic finite elements: Intrusive and non intrusive methods for reliability analysis* (Ph.D. thesis). Universite' Blaise Pascal, Clermont-Ferrand.
41. Sudret, B. (2008). Global sensitivity analysis using polynomial chaos expansions. *Reliability Engineering & System Safety, 93,* 964–979.
42. Paffrath, M., & Wever, U. (2007). Adapted polynomial chaos expansion for failure detection. *Journal of Computational Physics, 226,* 263–281.
43. Creveling, C. M. (1997). *Tolerance design: A handbook for developing optimal specification*. Cambridge, MA: Addison-Wesley.
44. Wu, C. C., Chen, Z., & Tang, G. R. (1998). Component tolerance design for minimum quality loss and manufacturing cost. *Computers in Industry, 35,* 223–232.
45. Hu, C., & Youn, B. D. (2011). An asymmetric dimension-adaptive tensor-product method for reliability analysis. *Structural Safety, 33*(3), 218–231.
46. Lee, S. H., & Chen, W. (2009). A comparative study of uncertainty propagation methods for black-box-type problems. *Structural and Multidisciplinary Optimization, 37*(3), 239–253.
47. Kvasov, B. I. (2000). *Methods of shape-preserving spline approximation*. Singapore: World Scientific Publications Co., Inc.
48. Abramowitz, M., & Stegun, I. A. (1972). *Handbook of mathematical functions* (9th ed.). New York: Dover Publications, Inc.

Chapter 6
Time-Dependent Reliability Analysis in Design

Engineering analysis and design methods have advanced to enable improvements in the reliability of engineered systems by considering various sources of uncertainty associated with these systems. Because one of the primary concerns in product design is ensuring a high level of system reliability throughout the product lifecycle, the ability to deal with the time-dependent probabilistic constraints necessary to carry out time-dependent reliability analysis in design is of vital importance in practical engineering design applications. Here, time-dependent reliability is defined as the probability that the time-dependent probabilistic constraints will be satisfied throughout the designed lifetime. For example, reliability that considers corrosion and fatigue is often called time-dependent reliability [1] because both corrosion and fatigue are time-dependent. In practical applications, time-dependent reliability is more desirable and provides more sophisticated system performance measures, as compared to traditional time-independent reliability, because it considers system performance changes in the time domain throughout the product's lifecycle. Different from time-independent reliability analysis, time-dependent reliability analysis involves limit state functions that change with time; thus, time-dependent reliability analysis necessitates the availability of a family of instantaneous limit states over the designed lifetime of the system. It is difficult for existing time-independent probability analysis approaches to handle the time dependency of the limit state functions; therefore, it is more challenging to calculate time-dependent reliability and carry out system design with time-dependent probabilistic constraints, as compared to time-independent cases.

This chapter introduces methods for time-dependent reliability analysis. Section 6.1 provides an overview of time-dependent reliability analysis. Sections 6.2–6.5 introduce four different methods for conducting time-dependent reliability analysis, namely simulation-based methods, extreme-value based methods, composite limit state based methods, and upcrossing rate based methods, respectively. The nested extreme response surface (NERS) approach—an extreme-value based method for time-dependent reliability analysis—is introduced in Sect. 6.3. The composite limit state methods and outcrossing rate methods are also

C. Hu et al., *Engineering Design under Uncertainty and Health Prognostics*,
Springer Series in Reliability Engineering,
https://doi.org/10.1007/978-3-319-92574-5_6

briefly discussed, and related references are provided for further study. Section 6.6 provides concluding thoughts. Exercise problems are provided in Sect. 6.7.

6.1 Overview of Time-Dependent Reliability Analysis

For static time-independent reliability analysis, the limit state function $G(\mathbf{X})$ is generally used, where the vector $\mathbf{X}$ represents random input variables with a joint probability density function (PDF) $f_{\mathbf{X}}(\mathbf{x})$. The probability of failure can be defined based on the limit state function as

$$P_f = P(G(\mathbf{X}) > 0) = \int \cdots \int_{G(\mathbf{X}) > 0} f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} \tag{6.1}$$

When time-dependent characteristics are taken into account in reliability analysis, the system response in general can be accordingly described by a random process $S(\mathbf{X}, t)$, which not only depends on the random input variables $\mathbf{X}$ but also is a function of time t. In this way, the time-dependent limit state can be described as

$$G(\mathbf{X}, t) = S(\mathbf{X}, t) - S_0 \tag{6.2}$$

where S_0 represents the marginal value. For example, for a roller clutch, the performance function $S(\mathbf{X}, t)$ could be defined as the hoop stress generated in the clutch operation, while the limit state function $G(\mathbf{X}, t)$ can be defined accordingly as the difference between the threshold value of the hoop stress R and $S(\mathbf{X}, t)$. By setting $G(\mathbf{X}, t) \leq 0$, we constrain the hoop stress generated in the clutch operation from going beyond the threshold. If t_l is the lifetime of interest, the probability of failure within the lifetime $[0, t_l]$ can be described based on the stress strength reliability model as

$$P_f(0, t_l) = P(\exists t \in [0, t_l], G(\mathbf{X}, t) > 0) \tag{6.3}$$

It is very difficult to evaluate $P_f(0, t_l)$, as it requires numerical evaluation of a multidimensional integration of a random process over time. In the literature, different methods have been developed to deal with this challenge, including both simulation-based methods and analytical approaches. Similar to time-independent reliability analysis, Monte Carlo simulation (MCS) can be employed to estimate the time-dependent reliability. However, MCS suffers from prohibitively high costs, as a large number of sample points must be evaluated over time. Therefore, the MCS method may not feasibly be used in practical engineering applications. To address this challenge, two basic categories of approaches have been developed in the literature for time-dependent reliability analysis—extreme performance based approaches [2–4] and first-passage based approaches [5–9]. The extreme

performance approaches use the extreme value of the performance function under consideration, and a failure occurs if the extreme value over a designed time interval is greater than a given threshold value. The first-passage approaches consider the first time instant when the performance of interest exceeds or falls below a threshold; this requires calculation of the "outcrossing rate" as the likelihood that the performance exceeds or falls below the threshold. In recent literature, a composite limit state (CLS) based method has also been developed to compute the cumulative probability of failure based on the Monte Carlo simulation (MCS). This approach constructs a global CLS to transform time-dependent reliability analysis into a time-independent analysis by combining all instantaneous limit states of each time interval in a series. In the rest of this chapter, these different methods will be introduced.

6.2 Simulation-Based Methods for Time-Dependent Reliability Analysis

In this section, the Monte Carlo simulation (MCS) method for time-dependent reliability analysis is first introduced; an example is then used to demonstrate the concept.

6.2.1 Procedure for Time-Dependent Reliability Analysis Using MCS

MCS is a computational algorithm that relies on repeated random sampling to obtain numerical results. To conduct time-dependent reliability analysis using the MCS method, the following general steps can be followed:

Step 1 *Random sample point generation*
In this step, a large number of random input samples are generated by drawing from probability distributions based on the randomness of system inputs. For time-dependent reliability analysis, random sample points are generated for random variables and parameters in a manner like that used to generate sample points for time-independent reliability analysis.

Step 2 *System performance evaluation*
In this step, the system performance function is evaluated over time at each Monte Carlo sample point, as generated in step 1. For time-dependent reliability analysis, the limit state function is dependent on time; thus, the evaluation of the limit state function at a given sample point $\mathbf{x}^*$ will generally provide a one-dimensional function $G(\mathbf{x}^*, t)$.

Step 3 *Sample point classification*
In this step, each Monte Carlo sample point is classified into one of two categories: the *failure* category or the *safe* category, respectively, based upon the system performance evaluation results. For time-dependent reliability analysis, for a given sample point $\mathbf{x}^*$, if the limit state function $G(\mathbf{x}^*, t)$ is larger than zero at any time t within $[0, t_l]$, $\mathbf{x}^*$ will be classified into the failure category; otherwise, $\mathbf{x}^*$ will be classified into the safe category.

Step 4 *Reliability Estimation*
In this step, the reliability is estimated based on the results of the sample point classification in step 3. For time-dependent reliability analysis, the reliability value is estimated based on the results of sample point classification considering the time-dependent limit state function. The reliability value is the ratio of the total number of sample points in the safe category over the total number of MCS samples.

Figure 6.1 shows the procedure for time-dependent reliability analysis using the MCS method. A mathematical example is employed to demonstrate time-dependent reliability analysis using the MCS approach.

6.2.2 Example of Time-Dependent Reliability Analysis Using MCS

The time-dependent limit state function $G(\mathbf{X}, t)$ is given by

$$G(\mathbf{X}, t) = 20 - X_1^2 X_2 + 5X_1 t - (X_2 + 1)t^2 \tag{6.4}$$

where t represents the time variable that varies within [0, 5], and X_1 and X_2 are normally distributed random variables: $X_1 \sim N(3.5,\ 0.3^2)$ and $X_2 \sim N(3.5,\ 0.3^2)$. The time-dependent probability of failure, $P(G(\mathbf{X}, t)>0)$, can then be calculated as detailed below.

Following the procedure shown in Fig. 6.1, the following steps can be used to estimate the time-dependent reliability of this example using the MCS method. We

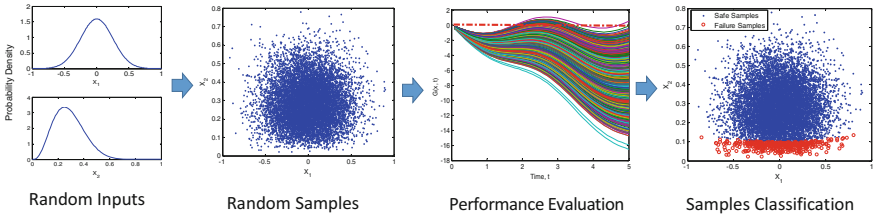


Fig. 6.1 Procedure for time-dependent reliability analysis using the MCS method

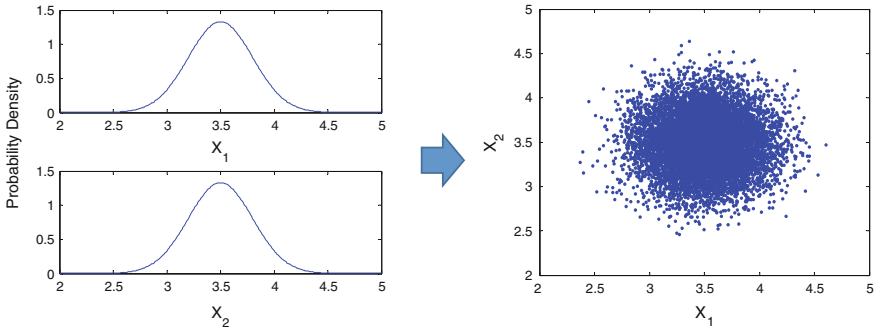


Fig. 6.2 Generating random sample points for X_1 and X_2 from their PDFs

first generate random sample points of X_1 and X_2 by drawing samples from the probability density functions (PDFs). Here, a total number of 10,000 samples are generated for X_1 and X_2, respectively. Figure 6.2 shows the random sample points generated for X_1 and X_2.

The limit sate function $G(\mathbf{X}, t)$, as shown in Eq. (6.4), is then evaluated at the random sample points generated in Step 1. As $G(\mathbf{X}, t)$ is time-dependent, evaluation of the limit state function at a given sample point $\mathbf{x}^*$ will generally provide a one-dimensional function $G(\mathbf{x}^*, t)$.

Figure 6.3a shows the evaluation of $G(\mathbf{X}, t)$ over time at several sample points, including the point $\mathbf{x} = [3.5, 3.5]$.

Considering the failure event as defined in Eq. (6.3), a sample point can be classified as a failure sample point if the limit state function $G(\mathbf{X}, t)$ goes beyond zero with the specified time interval. As shown in Fig. 6.3b, the limit state function has been evaluated at four different sample points over time. In the figure, the two read curves indicate two failure sample points; whereas, the two blue curves correspond to two safe samples. Following the classification procedure shown in

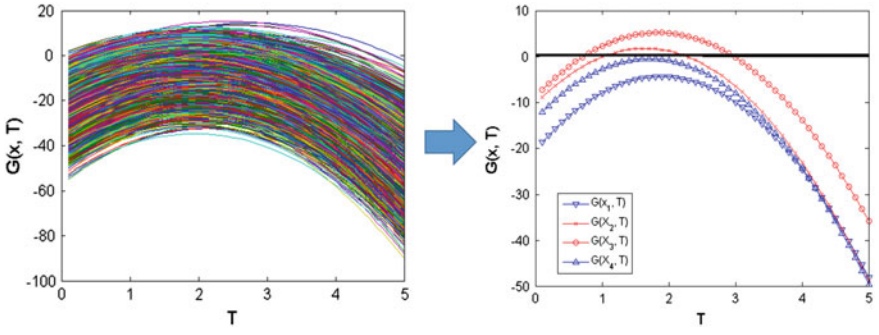


Fig. 6.3 Evaluation of system performance functions over time (**a**), and determination of failure sample points (**b**)

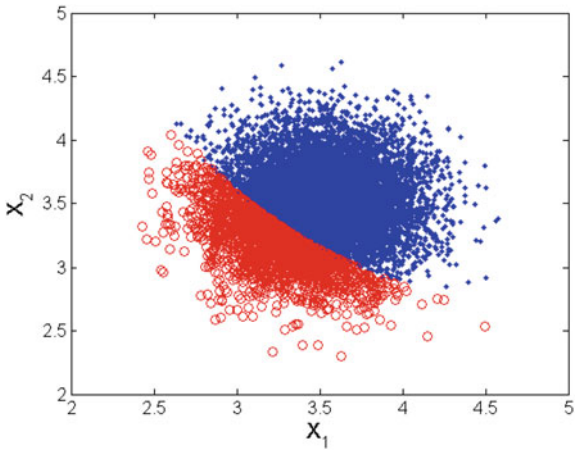


Fig. 6.4 Classification of random sample points based on the evaluation of system performance functions

Fig. 6.1, all Monte Carlo samples can be accordingly classified into either failure samples or safe samples, as seen in the classification results shown in Fig. 6.4.

Based on the classification of the sample points, 81.60% of sample points are classified to be safe within the time interval [0, 5]. Thus, based on the MCS method, the time-dependent reliability is estimated as 81.60%.

6.3 Extreme Value-Based Methods

This section presents the details about one of the extreme value-based methods for time-dependent reliability analysis, namely, the *nested extreme response surface* (NERS) approach [10–13]. Section 6.3.1 introduces the concept of the *nested time prediction model* (NTPM). Section 6.3.2 details how the NERS approach can be used to efficiently construct the NTPM. A mathematic example follows in Sect. 6.3.3.

6.3.1 Concept of a Nested Time Prediction Model

For a system of interest, the performance function $S(\mathbf{X}, t)$, as shown in Eq. (6.2), may change over time due to time-dependent loading conditions and/or component deterioration. Figure 6.5 shows the random realizations of two different types of system responses, where the solid and dashed lines represent monotonically and non-monotonically increasing performance, respectively. If $S(\mathbf{x}, t)$ increases or decreases monotonically over time, the extreme response will generally occur at the time interval boundary, where the probability of failure will also approach its maximum value. For this type of time-dependent reliability constraint, the reliability analysis needs to be carried out only at the time interval boundary, and the optimum

design derived from the reliability-based design optimization (RBDO) can guarantee the reliability requirements being satisfied over the entire time domain of interest. However, the situation is more complicated when the system response $S(\mathbf{X}, t)$ is a non-monotonic function, as shown in Fig. 6.5. In this case, it is critical that the reliability analysis is carried out at the instantaneous time when the extreme response of the performance function is obtained.

The time that leads to the extreme response of the system performance function varies with different designs $\mathbf{x}$; thus, a response surface of the time with respect to system design variables can be determined as

$$T = f(\mathbf{X}) : \left\{ \max_{t} S(\mathbf{x}, t), \forall \mathbf{x} \in \mathbf{X} \right\}. \tag{6.5}$$

For example, take a time-dependent limit state function $G(\mathbf{X}, t) = 20 - X_1^2 X_2 + 5X_1 t - (X_2 + 1)t^2$. For any given design $\mathbf{x}$: $\{x_1 \in [0, 10], x_2 \in [0, 10]\}$, there is an instantaneous time t: $\{t \in [0, 25]\}$ that maximizes $G(\mathbf{x}, t)$. To find out this correspondence, we let the derivative of G with respect to t equal zero and obtain the functional relationship between t and $\mathbf{x}$ that maximizes the $G(\mathbf{x}, t)$ as $t = 5x_1/(2x_2 + 2)$. Figure 6.6 shows this functional relationship, where x_1 and x_2 are two design variables and the z-axis represents time t. As shown, the extreme response of time varies with design variables x_1 and x_2. For example, for a particular realization of system design, $x_1 = 4$, $x_2 = 1$, the system approaches the maximum response at time $t = 5$; while for another particular realization of system design, where $x_1 = 5$, $x_2 = 0$, the system reaches its maximum response at time $t = 12.5$. The response surface of the time T here is defined as the nested time prediction model in the NERS approach. The inputs of the NTPM are design variables and parameters of interest, which can be either random or deterministic; whereas, the output is the time when the system response approaches its extreme value. With the NTPM, the time-dependent reliability analysis problem can be converted to a time-independent one, and thus the existing advanced reliability analysis approach, such as the first-order reliability method (FORM), the second-order reliability method (SORM), or dimension-reduction methods [14–16], can be conveniently used. In design optimization problems with time-dependent probabilistic constraints, the NTPM can be completely nested in the design process to convert the time-dependent reliability analysis

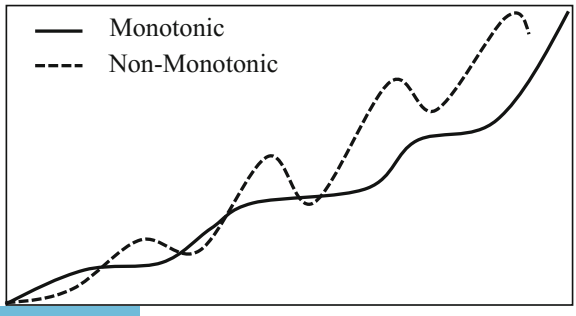


Fig. 6.5 Examples of time-dependent performance functions

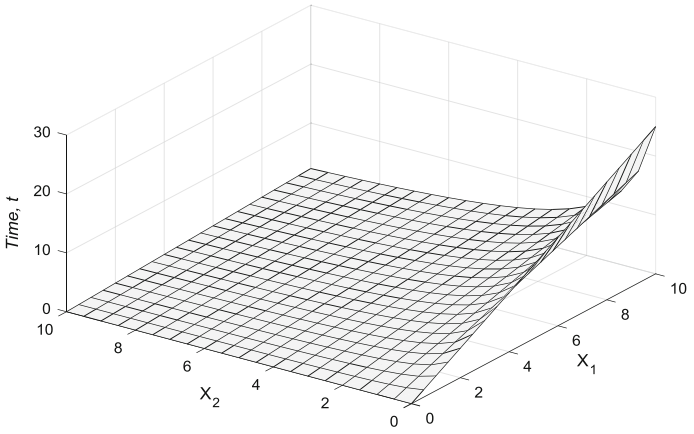


Fig. 6.6 The extreme response surface of time

into a time-independent one by estimating the extreme time responses for any given system design.

Although the NTPM facilitates time-dependent reliability analysis, it can be very challenging to efficiently develop a high-fidelity time-prediction model. First, analytical forms of time-dependent limit states are usually not available in practical design applications; consequently, NTPM must be developed based on limited samples. Second, because the samples for developing NTPM require extreme time responses over the design space, it is of vital importance that these responses be efficiently extracted to make the design process computationally affordable. Third, NTPM must be adaptive in performing two different roles: making predictions of extreme time responses for reliability analysis, and including extra sample points to improve the model itself at necessary regions during the iterative design process. The following section presents the NERS methodology, which addresses the three aforementioned challenges.

6.3.2 The NERS Methodology

In this subsection, we introduce one of the extreme value-based methods that can be used to tackle time-dependent reliability analysis and design problems, namely the *nested extreme response surface* (NERS) approach. The key to this approach is to effectively build the NTPM in the design space of interest, which can then be used to predict the time when the system response will approach its extreme value [10–13]. The NERS methodology is comprised of three major techniques within three consecutive steps: (1) efficient global optimization (EGO) for extreme time response identification, (2) construction of a kriging-based NTPM, and (3) adaptive time response prediction and model maturation. The first step, EGO, is employed to

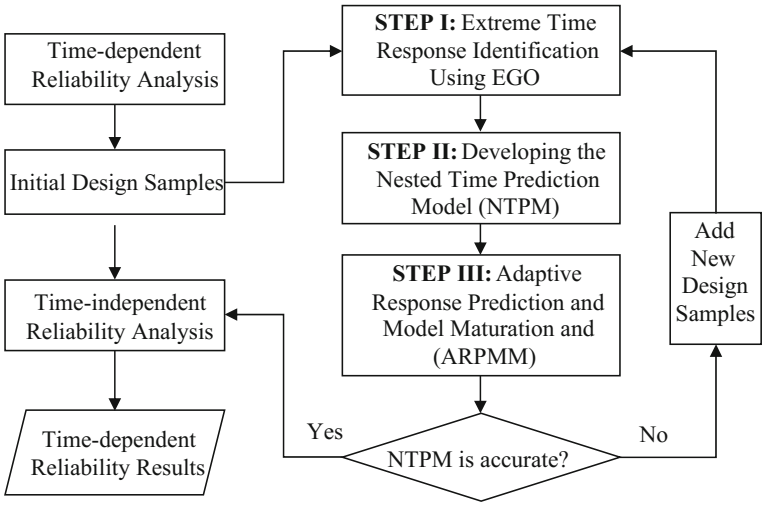


Fig. 6.7 The procedure of time-dependent reliability analysis using the NERS approach

efficiently extract a certain amount of extreme time response samples, which are then used for the development of NTPM in the second step. Once the kriging-based NTPM for extreme time responses is established, adaptive response prediction and model maturation mechanisms are used to assure the prediction accuracy and efficiency by autonomously enrolling new sample points when needed during the analysis process. The NERS methodology is outlined in the flowchart shown in Fig. 6.7, and the three aforementioned key techniques are explained in detail in the remainder of this section.

A. *Efficient Global Optimization for Extreme Time Response Identification*

For reliability analysis with time-dependent system responses, it is critical to efficiently compute the extreme responses of the limit state function and effectively locate the corresponding time when the system response approaches its extreme value. For a given system design, the response of the limit state function is time-dependent and could be either a monotonic or non-monotonic one-dimensional function with respect to time. Here, the Efficient Global Optimization (EGO) technique [17] can be employed to efficiently locate the extreme system response and the corresponding time when the extreme response is approached, mainly because it is capable of searching for the global optimum when dealing with a non-monotonic limit state, while at the same time assuring excellent computational efficiency. In this subsection, we focus on introducing the application of the EGO technique for extreme time response identification. Discussion of the EGO technique itself is omitted, because more detailed information about this technique can be obtained from references [18–20].

In order to find the global optimum time that leads to the extreme response of the limit state function, the EGO technique generates a one-dimensional stochastic process model based on existing sample responses over time. Stochastic process models have been widely used for function approximation; more information on these models can be found in the references [21, 22]. In this study, the response of the limit state function over time for a particular design point is expressed by a one-dimensional stochastic process model in EGO with a constant global mean as

$$F(t) = \mu + e(t) \tag{6.6}$$

where μ is the global model representing the function mean, and $e(t)$ is a stochastic process with mean zero and variance σ_e^2. The covariance between $e(t)$ at two different points t_i and t_j is defined as $Cov(e(t_i), e(t_j)) = \sigma_e^2 R_c(t_i, t_j)$, in which the correlation function is given by

$$R_c(t_i, t_j) = \exp\left(-a\left|t_i - t_j\right|^b\right) \tag{6.7}$$

where a and b are unknown model hyperparameters. Based on a set of initial samples, $F(t_1)$, $F(t_2)$, …, $F(t_k)$, an initial stochastic process model of $F(t)$ can always be constructed by maximizing the likelihood function

$$L_F = \frac{1}{(2\pi)^{n/2}(\sigma^2)|R_c|^{1/2}} \exp\left(-\frac{(F-\mu)' R_c^{-1}(Y-\mu)}{2\sigma^2}\right) \tag{6.8}$$

where $F = (F(t_1), F(t_2), \ldots, F(t_k))$ denote the sample responses of the limit state function, and R_c is the covariance matrix whose (i, j) is given by Eq. (6.7). After developing the initial one-dimensional stochastic process model, EGO updates this model iteratively by continuously searching for the most useful sample point that ensures a maximum improvement for the accuracy until the convergence criteria is satisfied. To update the stochastic process model, the EGO employs the expected improvement [23] metric to quantify the potential contribution of a new sample point to the existing response surface; the sample point that gives the largest expected improvement value is chosen at the next iteration. In what follows, the expected improvement metric will be introduced briefly and the procedure of employing the EGO technique for extreme time response identification will be summarized. A mathematic example is employed to demonstrate the extreme time response identification using the EGO technique.

Let us consider a continuous function $F(t)$ over time t that represents a limit state function over time for a given system design point in the design space. Here, we employ an expected improvement metric to determine the global minimum of $F(t)$. Due to limited samples of $F(t)$, the initial stochastic process model may introduce large model uncertainties, and consequently, the function approximation, denoted by $f(t)$, could be substantially biased compared with the real function $F(t)$. Due to the uncertainty involved in this model, in EGO, the function approximation of $f(t)$ at

time t is treated as a normal random variable whose mean and standard deviation are computed by an approximated response $F_r(t)$, and its standard error $e(t)$ is determined from the stochastic process model. With these notations, the improvement at time t can be defined as

$$I(t) = \max(F_{\min} - f(t), 0) \tag{6.9}$$

where $F_{\min}$ indicates the approximated global minimum value at the current EGO iteration. By integrating the expectation of the right part of Eq. (6.8), the expected improvement at any given time t can be presented as [17]

$$\begin{aligned} E[I(t)] &= E[\max(F_{\min} - f, 0)] \\ &= (F_{\min} - F_r(t))\Phi\left(\frac{F_{\min} - F_r(t)}{e(t)}\right) + e(t)\phi\left(\frac{F_{\min} - F_r(t)}{e(t)}\right) \end{aligned} \tag{6.10}$$

where $\Phi(\cdot)$ and $\phi(\cdot)$ are the cumulative distribution function and the probability density function for the standard Gaussian distribution, respectively. A larger expected improvement at time t means a greater probability of achieving a better global minimum approximation. Thus, a new sample should be evaluated at the particular time t_i, where the maximum expected improvement value is obtained to update the stochastic process model. With the updated model, a new global minimum approximation for $F(t)$ can be obtained. The same process can be repeated iteratively by evaluating a new sample at time t_i, which provides the maximum expected improvement value and updates the stochastic process model for the new global minimum approximation until the maximum expected improvement is small enough, and less than a critical value I_c (in this study, $I_c = |F_{min}|\%$, which is 1% of the absolute value of the current best global minimum approximation). The procedure for employing the EGO technique for extreme time response identification can be briefly summarized in five steps, as shown in Table 6.1.

A mathematical example is employed here to demonstrate the accuracy and efficacy of the EGO for extraction of the extreme responses of the limit state

Table 6.1 Procedure for EGO for extreme time response identification

Steps	Procedure
Step 1:	Identify a set of initial sample times t_i and evaluate the responses of the limited state function $F(t_i)$ for (i = 1, 2, …, k).
Step 2:	Develop a stochastic process model for $F(t)$ with existing sample points, and approximate the global minimum, F_{min}, and the time t_m.
Step 3:	Determine time t_i with maximum expected improvements, $E[I(t)]$.
Step 4:	Compare $max\{E[I(t)]\}$ with I_c: If $max\{E[I(t)]\}$ <= I_c, STOP and report t_m and F_{min}; Else, go to Step 5.
Step 5:	Evaluate the response at t_i, and repeat Steps 2–4.

function. Assume that a time-dependent limit state function for a particular point of the design space is provided by

$$F(t) = 1.042t^5 - 13.25t^4 + 59.78t^3 - 112.75t^2 + 75.17t \tag{6.11}$$

The objective here is to identify the extreme response of $F(t)$ (the global minimum) and pinpoint the corresponding time t within the time interval [1, 5]. Figure 6.8 shows the limit state function with respect to time, in which the global minimum occurs at time $t = 4.5021$ with $F(4.5021) = -3.5152$; whereas, the local minimum is located at $t = 1.9199$ with $F(1.9199) = 0.8472$.

Following the procedures outlined in Table 6.1, the performance function is first evaluated at initial samples $t_1 \sim t_5$: [1, 2, 3, 4, 5], and the obtained limit state function values are $F(t_1) \sim F(t_5)$: [10.992, 0.924, 7.776, 1.607, 9.600]. With these initial sample points, a one-dimensional stochastic process model can be built to approximate the limit state function $F(t)$, and the global optimum can be approximated as $F(2) = 0.924$. As indicated by Step 3 in Table 6.1, the expected improvement is calculated based on Eq. (6.10) throughout the time interval [1, 5], and the maximum expected improvement can be obtained as $max\{E[I(t_i)]\} = 5.2655$, where $t_i = 2.4572$. Because $max\{E[I(t_i)]\} > I_c$, where $I_c = 0.00001$ here, the limit state function will be evaluated at the new sample point $t_i = 2.4572$, which results in $F(2.4572) = 3.6114$. Figure 6.9a shows the above-discussed first EGO iteration for the extreme response identification, in which the top figure represents the current realization of the stochastic process model for $F(t)$, and the bottom one indicates the curve of the expected improvements over t. In the next iteration, by adding a new sample point to the existing ones at $t_1 \sim t_5$, a new stochastic process model with better accuracy can be built for the approximation of $F(t)$ over time and the identification of the extreme response of the limit state function (the global minimum).

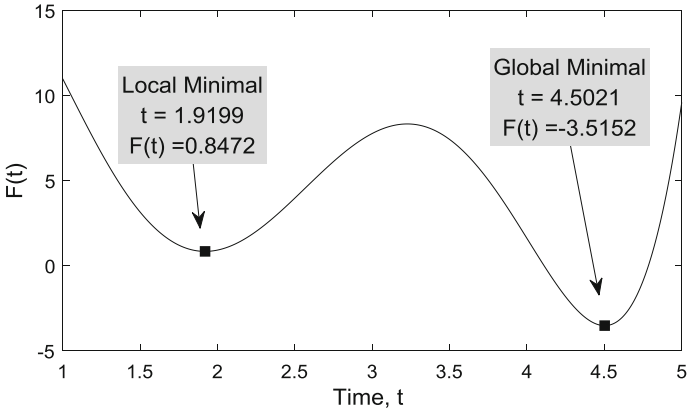


Fig. 6.8 Time-dependent limit state of the mathematical example

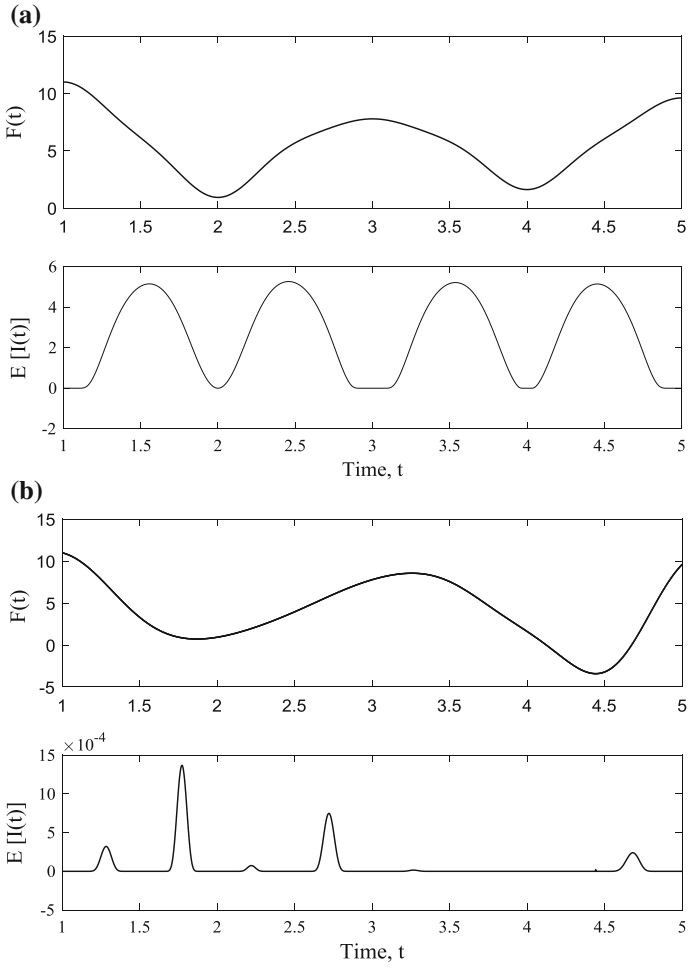


Fig. 6.9 **a** First EGO iteration for extreme response identification; **b** The eighth EGO iteration for extreme response identification

The procedures shown in Table 6.1 can be repeated until the convergence criterion is satisfied and the extreme response of the limit state function is identified with a desired accuracy level. Figure 6.9b shows the sixth EGO iteration for the extreme response identification. After a total of eight iterations, the expected improvement of including a new sample point for the response surface is clearly small enough and the convergence criterion is satisfied. Table 6.2 details all the EGO iterations for the extreme response identification of this mathematical example. As shown in this table, the accuracy of the estimated global minimum is improved after involving a new sample during the EGO process; thus, the minimum

Table 6.2 EGO iterative process for extreme response identification

Iteration	New sample point t			Approximated global minimum	
	Max{$E[I(t)]$}	t	$F(t)$	t_m	F_{min}
0	–	–	–	2.0002	0.924
1	5.2655E+00	2.4572	3.6114	2.0479	0.87574
2	2.0957E+00	3.5628	6.9557	2.0369	0.89421
3	2.0469E+00	4.4454	−3.4112	4.3888	−3.6272
4	4.8217E+00	1.5413	2.7243	4.3869	−3.6388
5	8.8917E−01	4.2659	−2.0063	4.4425	−3.4117
6	1.3699E−03	1.7734	1.1202	4.4524	−3.4138
7	4.1988E−05	4.4527	−3.4357	4.4922	−3.4991
8	1.3695E−05	4.4922	−3.5119	4.5021	−3.5152

F_{min}= −3.5152 in iteration 8 is more accurate than the result F_{min}= −3.4991 in iteration 7.

B. *Nested Time Prediction Model*

This section presents the procedure for developing a nested time prediction model using kriging. After repeating the EGO process outlined in subsection A for different system sample points in the design space, a set of data can be obtained, including initial sample points in the design space **X**, and the corresponding times T when the system responses approach their extreme values at these sample points. To build the NTPM using the NERS approach, design points are randomly generated in the design space based on the random properties of the design variables. To balance the accuracy and efficiency of the NTPM, initially 10 * (n − 1) samples are suggested to build the kriging-based NTPM for n-dimensional problems (n > 1). The accuracy and efficiency of NTPM is controlled by an adaptive response prediction and model maturation (ARPMM) mechanism, which will be detailed in the next subsection. The objective here is to develop a prediction model to estimate the time that leads to the extreme performances of the limit state function for any given system design in the design space. For this purpose, the kriging technique is employed, and a kriging model is constructed based on the sample dataset obtained during the EGO process. It is noteworthy that different response surface approaches, such as the simple linear regression model or artificial neural networks [24–26], could be applied here for development of the NTPM. In this study, kriging is used because it performs well for modeling nonlinear relationships between the extreme time responses with respect to system design variables.

Kriging is considered to be powerful and flexible for developing surrogate models among many widely used metamodeling techniques [20, 27]. One of the distinctive advantages of kriging is that it can provide not only prediction of extreme time responses at any design point but it can also define the uncertainties,

such as the mean square errors associated with the prediction. Considering a limit state function with *nd* random input variables, a kriging time prediction model can be developed with n sample points denoted by (x_i, t_i), in which $\mathbf{x}_i = (x_i^1 \ldots x_i^{nd})$ $(i = 1 \ldots n)$ are sample inputs, and t_i is the time when the limit state function approaches the extreme value for a given $\mathbf{x}_i$. In the kriging model, time responses are assumed to be generated from the model:

$$t(\mathbf{x}) = H(\mathbf{x}) + Z(\mathbf{x}) \tag{6.12}$$

where $H(\mathbf{x})$, as a polynomial function of $\mathbf{x}$, is the global model that represents the function mean, and $Z(\mathbf{x})$ is a Gaussian stochastic process with zero mean and variance σ^2. As indicated by the studies in references [20, 28], a constant global mean for $H(\mathbf{x})$ is usually sufficient in most engineering problems; it is also much less expensive and computationally more convenient. Thus, we use a constant global mean μ for the polynomial term $H(\mathbf{x})$, and accordingly, the kriging time prediction model in Eq. (6.16) can be expressed as

$$t(\mathbf{x}) = \mu + Z(\mathbf{x}) \tag{6.13}$$

In this kriging model, the covariance function of $Z(\mathrm{x})$ is given by

$$Cov[Z(\mathbf{x}_i), Z(\mathbf{x}_j)] = \sigma^2 R_c(\mathbf{x}_i, \mathbf{x}_j) \tag{6.14}$$

where $R_c(\mathbf{x}_i, \mathbf{x}_j)$ represents an $n \times n$ symmetric correlation matrix, and the (i, j) entry of this correlation matrix is a function of the spatial distance between two sample points $\mathbf{x}_i$ and $\mathbf{x}_j$, which is expressed as

$$R_c(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\sum_{p=1}^{nd} a_p |\mathbf{x}_i^p - \mathbf{x}_j^p|^{b_p} \right) \tag{6.15}$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ denote two sample points, $|\cdot|$ is the absolute value operator, and a_p and b_p are hyperparameters of the kriging model that need to be determined. In this equation, a_p is a positive weight factor related to each design variable, and b_p is a non-negative power factor with a value usually within the range [0, 2] [17, 29].

Note that other than the most commonly used Gaussian functions, other options are available to define the correlation matrix $R_c(\mathbf{x}_i, \mathbf{x}_j)$ and derive the covariance function $S(\mathbf{x})$ [16, 20, 30–32]. With n number of sample points $(\mathbf{x}_i, t_i)$ for $I = 1, \ldots, n$ for the kriging time prediction model, the likelihood function of the model hyperparameters can be given as

$$Likelihood = -\frac{1}{2}\left[n \ln(2\pi) + n \ln \sigma^2 + \ln|R_c| + \frac{1}{2\sigma^2}(t - A\mu)^T R_c^{-1}(t - A\mu) \right] \tag{6.16}$$

In this equation, we can solve for the values of μ and σ^2 by maximizing the likelihood function in closed form as

$$\mu = \left[A^T R_c^{-1} A\right]^{-1} A^T R_c^{-1} t \tag{6.17}$$

$$\sigma^2 = \frac{(t - A\mu)^T R_c^{-1} (t - A\mu)}{n} \tag{6.18}$$

where A is a matrix of basis functions for the global model. In this study, A is an $n \times 1$ unit vector since only the constant global mean μ is considered for the polynomial term $H(\mathbf{x})$. Substituting Eqs. (6.17) and (6.18) into Eq. (6.16), the likelihood function is transformed to a concentrated likelihood function, which depends only upon the hyperparameters a_p and b_p for any p within [1, nd]. Then, a_p and b_p can be obtained by maximizing the concentrated likelihood function, and thereafter the correlation matrix R_c can be computed. With the kriging time prediction model, the extreme time response for any given new point x′ can be estimated as

$$t(\mathbf{x}') = \mu + r^T(\mathbf{x}') R_c^{-1} (t - A\mu) \tag{6.19}$$

where $r(\mathbf{x}')$ is the correlation vector between $\mathbf{x}'$ and the sampled points $\mathbf{x}_1 \sim \mathbf{x}_n$, in which the ith element of r is given by $r_i(\mathbf{x}') = R_c(\mathbf{x}', \mathbf{x}_i)$.

C. *Adaptive Response Prediction and Model Maturation*

Model prediction accuracy is of vital importance when employing the nested time prediction model for design. Thus, during the design process, a mechanism is needed for model maturation that will automatically enroll new sample points to improve the accuracy of the nested time prediction model when the accuracy condition is not satisfied. This paper develops an adaptive response prediction and model maturation (ARPMM) mechanism based on the mean square error $e(x)$ of the current best prediction. Figure 6.10 shows the flowchart of the developed mechanism.

Before predicting the time response of a new design point x using the latest update of the NTPM, the ARPMM mechanism can be employed by first calculating the current mean square error $e(x)$ of the best prediction as

$$e(\mathbf{x}) = \sigma^2 \left[1 - r^T R^{-1} r + \frac{(1 - A^T R^{-1} r)^2}{A^T R^{-1} A}\right] \tag{6.20}$$

To reduce the numerical error, the relative MSE is suggested as a prediction performance measure for the NTPM, which is given by

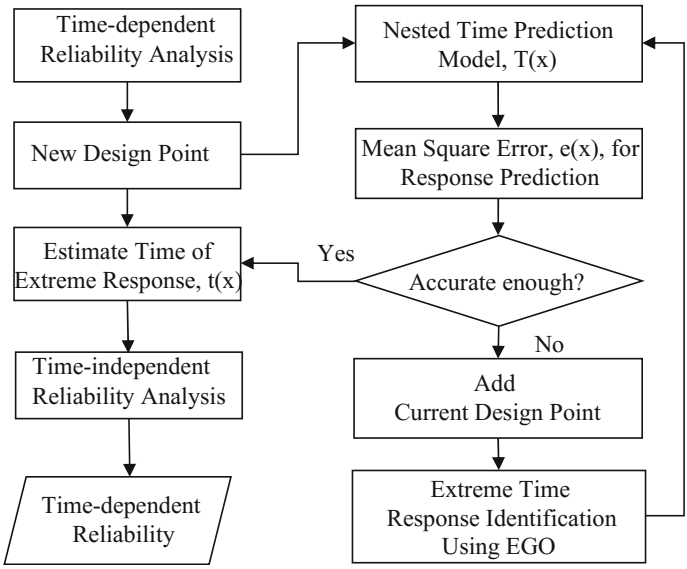


Fig. 6.10 Flowchart of the ARPMM mechanism

$$\xi(\mathbf{x}) = \frac{e(\mathbf{x})}{\mu} \tag{6.21}$$

Prediction of a time response t' using the NTPM for a new design point $\mathbf{x}$ is accepted only if the relative error $\xi(\mathbf{x})$ is less than a user-defined threshold ξ_t. To balance a smooth design process and a desired prediction accuracy, we suggest a value of ξ_t within the range $[10^{-3}, 10^{-2}]$. Once the prediction on this particular design point x is accepted, the time response t' of extreme performance is estimated using Eq. (6.23) and returned to the time-dependent reliability analysis process. If the relative error is larger than the threshold, then x will be enrolled as a new sample input, and the EGO process, as discussed in subsection A, will be employed to extract the true time response when the limit state function approaches its extreme performance for x. With the new design point $\mathbf{x}$ and the true time response for $\mathbf{x}$, the NTPM will be updated, as discussed in subsection B. The procedure for the ARPMM mechanism of NTPM is provided in Table 6.3. Through the developed ARPMM mechanism, the NTPM can be updated adaptively during the time-dependent reliability analysis process to guarantee accuracy and simultaneously maintain efficiency. Note that the ARPMM mechanism automates the improvement of the kriging model during the design process; under rare cases, the stability issues induced by singular matrices may occur when several design points located closely together are used to seed the kriging model. Thus, we suggest an extra step be included in the ARPMM process by checking the singularity after adding new sample points for the prediction accuracy improvement of the kriging model.

Table 6.3 The procedure of developing the nested time prediction model using ARPMM

Steps	Procedure
Step 1:	Identify an initial set of design points, **X**, and extract time responses **T** when the limited state function approaches extreme values at **X**, correspondingly.
Step 2:	Develop the nested time prediction model (NTPM) using existing data set $D = [\mathbf{X}, \mathbf{T}]$.
Step 3:	For a new design point **x**, calculate prediction error, $\xi(\mathbf{x})$, using the latest NTPM.
Step 4:	Compare the prediction error with the error threshold, ξ_t: • If $\xi(\mathbf{x}) < \xi_t$, estimate $t'(x)$ at the new point **x** and return to reliability analysis; • If $\xi(\mathbf{x}) \geq \xi_t$, determine the extreme response at t' of using EGO, add $(\mathbf{x}, t')$ to D, then go to Step 2.

6.3.3 *Example of Reliability Analysis Using NERS*

In this section, the same mathematical example as used in Sect. 6.2.2 is employed to demonstrate time-dependent reliability analysis using the NERS approach. In the example, the time-dependent limit state function $G(\mathbf{X}, t)$ is given by

$$G(\mathbf{X}, t) = 20 - X_1^2 X_2 + 3X_1 t - (X_2 + 0.5)t^2 \tag{6.22}$$

where t represents the time variable that varies within [0, 5] and x_1 and x_2 are normally distributed random variables: $X_1 \sim Normal(5.8, 0.5^2)$ and $X_2 \sim Normal(2.2, 0.5^2)$. Figure 6.11 shows the failure surface of the instantaneous time-dependent limit states as it changes with the time interval [0, 6], in which limit state functions at different time nodes equal zero. As a benchmark solution, we employed Monte Carlo simulation to calculate the reliability, with an obtained reliability of 0.9626. In the MCS, first, 100,000 samples were generated for each input variable, and then, the time variable was discretized evenly into 100 time nodes within the interval [0, 6]. The limit state function was evaluated for all sample points at each time node, and the reliability was then estimated by counting the number of safe samples accordingly. A sample point was considered a safe sample if the minimum of the limit state function values over the 100 time nodes was larger than zero. In what follows, the reliability analysis for this case study is carried out using the NERS approach.

The NERS approach, which converts the time-dependent reliability analysis to a time-independent analysis, begins with the development of the NTPM. As shown in Table 6.4, an initial set of eight design points can be sampled randomly in the design space, based on the randomness of the design variables X_1 and X_2. With the initial set of design points, the EGO technique can be employed for identification of the extreme responses (maximum) of the limit state and relative times when the limit state approaches its extreme values. The results are also shown in the last two columns of Table 6.4. For example, for the design point [3.4134, 3.2655], the limit state function approaches its extreme response −11.0843 when time $t = 1.3574$, as

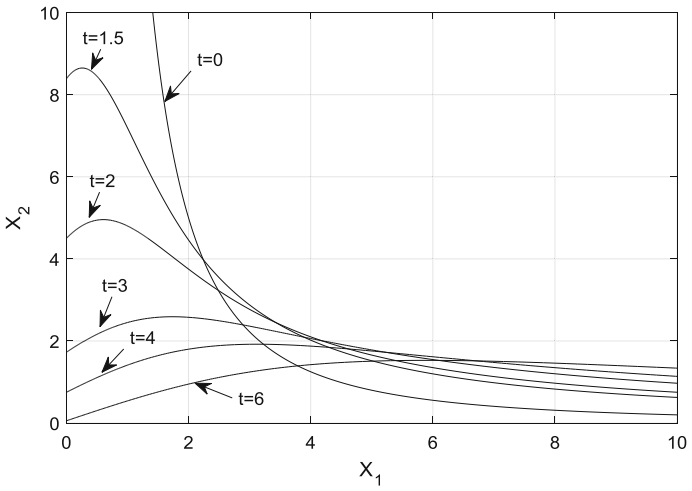


Fig. 6.11 Instantaneous limit states for time interval [0, 5]

Table 6.4 Randomly selected sample points

No.	X_1	X_2	Extreme response	Time
1	3.4134	3.2655	−11.0843	1.3574
2	4.5620	3.8518	−49.4041	1.5736
3	3.9249	3.9153	−32.4660	1.3333
4	4.5989	3.5194	−42.5950	1.7177
5	3.6763	3.2851	−16.3647	1.4595
6	2.4852	3.2229	3.8272	1.0030
7	3.2196	3.0563	−5.1216	1.3574
8	3.3746	3.9604	−19.3558	1.1351

shown in the first row of the table. Using these eight initial sample points, denoted by $(\mathbf{x}_i, t_i)$ for $i = 1, 2, \ldots, 8$, by following the procedure discussed in Sect. 6.3.2B, a kriging-based NTPM can be developed. The unknown parameters of the kriging model are estimated, and the NTPM is obtained by maximizing the concentrated likelihood function, as shown in Eq. (6.16). Figure 6.12 shows the developed NTPM for this case study.

During the reliability analysis process, the ARPMM mechanism is used to update the NTPM as necessary and predict the time responses for new design points. For the ARPMM mechanism, the mean square error $e(\mathbf{x})$ is employed as the decision-making metric. When the NTPM is used to predict the time when the limit state function approaches its extreme value for the new design point $\mathbf{x}$, if the relative error $\xi(\mathbf{x})$, based on $e(\mathbf{x})$, is greater than ξ_t, the ARPMM mechanism will trigger the

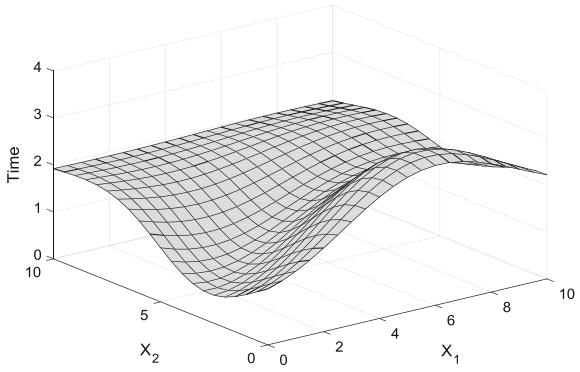


Fig. 6.12 Initial NTPM for the mathematical example

EGO process for identification of the true time response t' for $\mathbf{x}$. After the EGO process, the new design point x and the corresponding time response t' will be included as a new sample point to D, and the initial NTPM will be updated with this new point.

After developing the NTPM, the time-dependent reliability analysis can be converted to a time-independent one. Commonly used reliability analysis tools, such as the FORM, can be employed for this purpose. By applying the FORM, the time-independent limit state function is linearized at the most probable point (MPP), and the reliability index can be calculated in the standard normal space as the distance from the MPP to the origin, while the HL-RF algorithm [33] can be used for the MPP search in FORM. As the FORM linearizes the limit state function for the reliability calculation, error will be introduced due to this linearization. Thus, MCS is also employed with NTPM to study the reliability analysis errors introduced by FORM and by NTPM. Table 6.5 shows the results of reliability analysis with the NERS approach.

Table 6.5 Reliability analysis results using the NERS approach

Time interval	True reliability	NERS with FORM		NERS with MCS	
		Reliability	Error (%)	Reliability	Error (%)
[0, 6]	0.8563	0.8621	0.6816	0.8603	0.4695
[0, 5]	0.8563	0.8621	0.6816	0.8605	0.4998
[0, 4]	0.8584	0.8633	0.5763	0.8631	0.5452
[0, 3]	0.8788	0.8842	0.6181	0.8831	0.4870
[0, 2]	0.9327	0.9367	0.4261	0.9341	0.1469
[0, 1]	0.9832	0.9850	0.1810	0.9836	0.0437
0	0.9989	0.9991	0.0211	0.9989	0.0000

6.4 Composite Limit State Methods

The composite limit state (CLS) approach [34] discretizes the continuous time variable into a finite number of time intervals and consequently time is treated as a constant value within each time interval. The time-dependent reliability analysis can then be converted into reliability analysis of a serially connected system where the limit state function defined for each time interval is treated as a system component.

6.4.1 Introduction to the Composite Limit State Method

Let time t_l denote the designed life of interest, which is discretized into N time intervals using a fixed time step Δt. Let $E_n = \{\mathbf{x}|G(\mathbf{x}, t_n) \leq 0\}$ denote the instantaneous failure at the discrete time t_n, and let $\cup E_n$ for $n = 1, 2, \ldots$, be the composite limit state defined as the union event of all instantaneous failure events defined for each discretized time interval. The cumulative probability of failure can then be approximated by

$$P_f(0, t_l) = P\left(\bigcup_{n=0}^{N} G(\mathbf{X}, t_n) < 0,\ t_n \in [0,\ t_l]\right), \tag{6.23}$$

where failure occurs if $G(\mathbf{X}, t_n) < 0$ for any $n = 1, 2, \ldots, N$. Although the CLS approach converts the time-dependent reliability analysis to a time-independent one, determining the CLS itself is difficult and computationally expensive because it requires the comparison of all instantaneous failure events for each design point. The discretization of time will greatly simplify the time-dependent reliability analysis because it converts the time-dependent analysis problem into several time-independent reliability problems. However, despite the several-fold increase in computational cost, this method also raises the error when the time-dependent limit state is highly nonlinear within different time intervals.

6.4.2 Example of Time-Dependent Reliability Analysis Using the CLS Method

In this section, the same mathematical example as used in Sect. 6.2.2 is employed to demonstrate time-dependent reliability analysis using the CLS approach. By using the CLS method, the time within [0, 5] is discretized into 5 intervals, and within each time interval, the performance function is considered time-independent, as shown in Fig. 6.13.

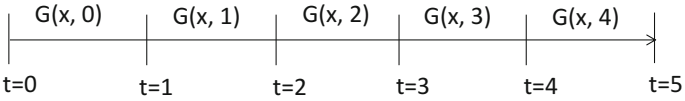


Fig. 6.13 Descretization of the time variables into time intervals

Table 6.6 Reliability values estimated in different time intervals

Time Interval	[0, 1]	[1, 2]	[2, 3]	[3, 4]	[4, 5]
Reliability	0.9996	0.9382	0.8308	0.9532	0.9982

After discretizing the time variable into time intervals, the time-dependent performance function can be accordingly converted into time-independent ones in different time intervals. With the time-independent performance function in each time interval, reliability analysis can be carried out using existing reliability analysis methods, such as MCS or FORM.

In this example, the MCS is used to find out the reliability at each time interval, in order to compute the time-dependent reliability estimates. Table 6.6 lists the reliability values estimated for these time intervals. The time-dependent reliability can be accordingly approximated based upon the reliability at each time interval as:

$$R = \prod_{i=1}^{t} R_i = 0.7413$$

6.4.3 Further Development of CLS-Based Methods

As compared to the time-dependent reliability estimates shown in the results obtained in Sects. 6.2.2 and 6.3.3, the CLS method produces errors for time-dependent reliability estimates. The error is dependent on the interval length determined while discretizing the time variable, as well as the reliability analysis method used for the time-independent reliability estimation. To improve the efficiency of the composite limit state method, a niching genetic algorithm with lazy learning metamodeling has been developed. Although a niching genetic algorithm with lazy learning metamodeling was performed to build the CLS, the computational cost was still a large burden, as demonstrated by reported case studies [34].

6.5 Outcrossing Rate Based Methods

Outcrossing rate based methods relate the probability of failure to the mean number of outcrossings of the time-dependent limit state function through the limit state surface. Outcrossing rate based methods generally perform time-dependent reliability analysis through estimation when the outcrossings may be considered independent and Poisson distributed, or by estimating an upper bound in general cases. In the following section, the outcrossing rate is introduced and time-dependent reliability analysis that employs the outcrossing rate is presented. A case study is used to demonstrate the method.

6.5.1 Introduction to the Outcrossing Rate

The approach based on the outcrossing rate calculates the probability of failure based on the expected mean number of outcrossings of the random process through the defined failure surface [35]. The instantaneous outcrossing rate at time τ is then defined as

$$v^+(\tau) = \lim_{\Delta\tau \to 0,\, \Delta\tau > 0} \frac{P[G(\mathbf{X}, \tau) > 0 \cap G(\mathbf{X}, \tau + \Delta\tau) < 0]}{\Delta\tau} \tag{6.24}$$

Let $N(0,\ t_l)$ denote the number of outcrossings of zero value from the safe domain to the failure domain within $[0,\ t_l]$. Basic probability theory shows that $N(0,\ t_l)$ follows a binomial distribution. When the probability of outcrossing is very small, it is equal to the mean number of outcrossings per unit time (the outcrossing rate). Because the binomial distribution converges to the Poisson distribution when the time period is sufficiently long or the dependence between crossings is negligible, the outcrossings are assumed to be statistically independent [36]. With this assumption, the outcrossing rate becomes the first-time crossing rate, or the failure rate. Then, the probability of failure can be estimated from the upcrossing rate. The probability of failure defined in Eq. (6.3) also reads as

$$P_f(0,\ t_l) = \Pr(\{G(\mathbf{X}, 0) < 0\} \cup \{N(0,\ t_l) > 0\}) \tag{6.25}$$

Equation (6.25) can be interpreted as that failure within time interval $[0,\ t_l]$ that corresponds either to failure at the initial instant $t = 0$ or to a later outcrossing of the limit state surface if the system is in the safe domain at $t = 0$. It is reported that the upper bound on $P_f(0,\ t_l)$ is available as [35]

$$P_f(0,\ t_l) \le P_f(0) + E[N(0, t_l)] \tag{6.26}$$

where the mean number of outcrossings is computed as

$$E[N(0,t_l)] = \int_0^{t_l} v^+(t)dt \tag{6.27}$$

6.5.2 Computing the Outcrossing Rate

Although the outcrossing rate approach tends to provide a rigorous mathematical derivation of the time-dependent reliability, it is difficult to evaluate the outcrossing rate for the general stochastic processes. An analytical outcrossing rate is available [37] for only special stochastic processes, such as stationary Gaussian processes. One commonly used method is Rice's formula [8], which has been used for scalar differentiable random processes. Rice's formula has been generalized by Belayev [38] for vector differentiable processes. Studies have also been conducted with specialization to differentiable Gaussian processes and for the case of scalar and vector rectangular wave renewal processes. A large number of methods have focused on the asymptotic integration approach to calculate the outcrossing rate [8, 9]. In the following section, one method for computing the outcrossing rate, namely the PHI2 method [39], will be briefly discussed and used in a case study.

For the PHI2 method, the outcrossing rate is computed by considering the two-component parallel system reliability analysis in Eq. (6.24). It consists of making two successive analyses using time-independent reliability analysis methods, such as MCS or FORM. Then, the binormal cumulative distribution function is evaluated, as first presented by Hagen and Tvedt [35] and then used by Li and Der Kiureghian [7].

Applying a finite difference-like operation of Eq. (6.24), the outcrossing rate can be replaced as

$$v^+_{\mathrm{PHI2}}(\tau) = \frac{\Pr[G(\mathbf{X},\tau) > 0 \cap G(\mathbf{X},\tau+\Delta\tau) < 0]}{\Delta\tau} \tag{6.28}$$

Thus, it is important that an appropriate time increment $\Delta\tau$ be selected. Assuming that FORM is used for time-independent reliability analysis, the following steps could be followed to carry out time-dependent reliability analysis using the PHI2 method.

- Conduct time-invariant reliability analysis for $\{G(\mathbf{X}, \tau) \leq 0\}$ using FORM. Thereby, the reliability index $\beta(\tau)$ is obtained by approximating the limit state surface by the hyper plane $\alpha(\tau)\cdot\mathbf{u} + \beta(\tau) = 0$ in the standard normal space.
- Similarly, conduct time-invariant reliability analysis for $\{G(\mathbf{X}, \tau+ \Delta\tau) \leq 0\}$ using FORM. Thereby, the reliability index $\beta(\tau + \Delta\tau)$ is obtained by

approximating the limit state surface by the hyper plane $\alpha(\tau + \Delta\tau)\cdot\mathbf{u} + \beta(\tau + \Delta\tau) = 0$ in the standard normal space.

- Denote the correlation between the two events $\{G(\mathbf{X}, \tau) > 0\}$ and $\{G(\mathbf{X}, \tau + \Delta\tau) \leq 0\}$ by

$$\rho_G(\tau, \tau + \Delta\tau) = -\alpha(\tau) \cdot \alpha(\tau + \Delta\tau) \tag{6.29}$$

- Compute the probability of a parallel system, leading to

$$v^{+}_{\mathrm{PHI2}}(\tau) = \frac{\Phi_2[\beta(\tau), -\beta(\tau + \Delta\tau); \rho_G(\tau, \tau + \Delta\tau)]}{\Delta\tau} \tag{6.30}$$

where F2 denotes the binormal cumulative distribution function.

- The time-dependent reliability can then be calculated by applying Eq. (6.29) to Eqs. (6.26) and (6.27).

6.5.3 *Example of Time-Dependent Reliability Analysis Using the Upcrossing Rate*

In this section, the same mathematical example as used in Sects. 6.2.2 and 6.3.3 is employed to demonstrate time-dependent reliability analysis using the PHI2 method. In the example, a time-dependent limit state function $G(\mathbf{X}, t)$ is given by

$$G(\mathbf{X}, t) = 20 - X_1^2 X_2 + 5X_1 t - (X_2 + 1)t^2$$

where t represents the time variable that varies within [0, 5]. The random variables X_1 and X_2 are normally distributed random variables: $X_1 \sim$ Normal(3.5, 0.32^2) and $X_2 \sim$ Normal(3.5, 0.32^2). While computing the time-dependent reliability of this example problem using the PHI2 methods, six different time increments have been used. The results, as compared with the MCS results in Sect. 6.2.2, are summarized in Table 6.7.

Table 6.7 Reliability analysis results using the PHI2 method

$\Delta\tau$	0.10	0.12	0.14	0.16	0.18	0.20
$R(0, 5)$	0.8311	0.8309	0.8334	0.8710	0.9209	0.9827
Error (%)	1.85	1.83	2.13	6.74	13.85	20.43

6.5.4 *Further Development of Outcrossing Rate Based Methods*

As one of the outcrossing rate based methods, the PHI2 method enables the use of classical time-independent reliability methods, such as FORM, for solving the time-dependent problem. The disadvantages of PHI2 were that only a bound of the system reliability could be obtained, and error could rise because of the nonlinearity of limit states due to the use of FORM. Sudret [40] developed a new formula based on the original PHI2 method to stabilize the step-size effect in calculating the time-dependent reliability, denoted as the PHI2+ method. Zhang and Du [41] and Du [42] proposed a mean value first-passage method to perform time-dependent reliability analysis for the function generator mechanism. In this approach, the analytical equations were derived for the up-crossing and down-crossing rates first, and a numerical procedure was then proposed to integrate these two rates in order to compute the time-dependent reliability. The assumption of this approach was that the motion error of the mechanism was a non-stationary Gaussian process. Son and Savage [43] tracked the time-variant limit state in standard normal space to compute the system reliability. For this approach, time was divided into several intervals, and the incremental failure probabilities were calculated for all of them. The probability of failure estimate was then obtained through the summation of all incremental failure probabilities. This approach simplified time-dependent reliability analysis by discretizing time into several time intervals and further assuming independency between these intervals.

6.6 Conclusion

This chapter introduced time-dependent reliability analysis problems and four different methods to carry out time-dependent reliability analysis in practical engineering analysis and design applications. Here, time-dependent reliability is defined as the probability that the time-dependent probabilistic constraints will be satisfied throughout the designed lifetime. Different from time-independent reliability analysis, the limit state function changes with time; thus, in time-dependent reliability analysis, a family of instantaneous limit states will be obtained over the system's designed life. Due to the difficulty in handling the time dependency of time-dependent limit states using existing time-independent probability analysis approaches, it is more challenging to calculate time-dependent reliability and carry out system design with time-dependent probabilistic constraints, as compared to time-independent cases.

While introducing methods for time-dependent reliability analysis, our focus has been on simulation-based methods and one extreme value based method, namely

the nested extreme response surface (NERS) approach. The principles of composite limit state based methods and outcrossing rate based methods have been briefly discussed, with examples to demonstrate the analysis processes.

Exercise Problems

1. Consider a time-dependent limit state function $G(\mathbf{X}, t)$, given by

$$G(\mathbf{X}, t) = 20 - X_1 X_2 + X_1^2 t - (X_2 + 1)t^2$$

where t represents the time variable that varies within [0, 5], and X_1 and X_2 are normally distributed random variables: $X_1 \sim Normal(3.5, 0.3^2)$ and $X_2 \sim Normal(3.5, 0.3^2)$. Calculate the time-dependent probability of failure, $P(G(\mathbf{X}, t) \leq 0)$. Following the procedure as shown in Fig. 6.1, the following steps can be used to estimate the time-dependent reliability of this example.

   a. Compute the time-dependent reliability using the MCS method.
   b. Compute the time-dependent reliability using the NERS method.
   c. Compute the time-dependent reliability using the CLS method with 10 time intervals evenly distributed.
   d. Compute the time-dependent reliability using the outcrossing rate method.

2. Consider a time-dependent limit state function $G(\mathbf{X}, t)$, given by

$$G(\mathbf{X}, t) = 1 - \frac{(X_1 + X_2 t - 10)^2}{30} - \frac{(0.2t^2 + X_1 t - X_2 - 16)^2}{120}$$

where t represents the time variable that varies within [0, 5], and X_1 and X_2 are normally distributed random variables: $X_1 \sim Normal(3.0, 0.5^2)$ and $X_2 \sim Normal(3.0, 0.5^2)$. Calculate the time-dependent probability of failure, $P(G(\mathbf{X}, t) \leq 0)$.

   a. Compute the time-dependent reliability using the MCS method.
   b. Compute the time-dependent reliability using the NERS method.
   c. Compute the time-dependent reliability using the CLS method with 10 time intervals evenly distributed.
   d. Compute the time-dependent reliability using the outcrossing rate method.

3. For the two-slider crank mechanism shown in Fig. 6.14, the time-dependent limit state function can be given by

$$G(R_1, R_2, R_3, R_4, \theta) = \Delta s_{desired} - \Delta s_{actual}$$

where

$$\Delta s_{actual} = R_1 \cos(\theta - \theta_0) + \sqrt{R_2^2 - R_1^2 \sin^2(\theta - \theta_0)} - R_3 \cos(\theta_1 + \theta_0 - \theta - \delta_0) - \sqrt{R_4^2 - R_3^2 \sin^2(\theta_1 + \theta_0 - \theta - \delta_0)}$$

$$\Delta s_{desired} = 108 \cos(\theta - \theta_0) + \sqrt{211^2 - 108^2 \sin^2(\theta - \theta_0)} - 100 \cos(\theta_1 + \theta_0 - \theta - \delta_0) - \sqrt{213^2 - 100^2 \sin^2(\theta_1 + \theta_0 - \theta - \delta_0)}$$

The random variables and parameters in the performance function are given in the table below.

Variable	Mean	Standard deviation	Distribution
R_1	108 mm	0.05 mm	Normal
R_2	211 mm	0.2 mm	Normal
R_3	100 mm	0.05 mm	Normal
R_4	213 mm	0.2 mm	Normal
θ_0	45°	0	Deterministic
θ_1	60°	0	Deterministic
δ_0	10°	0	Deterministic
δ_1	π rad/s	0	Deterministic

Considering the time variable θ within the range of [0, 360°],

a. Compute the time-dependent reliability using the MCS method.
b. Compute the time-dependent reliability using the NERS method.

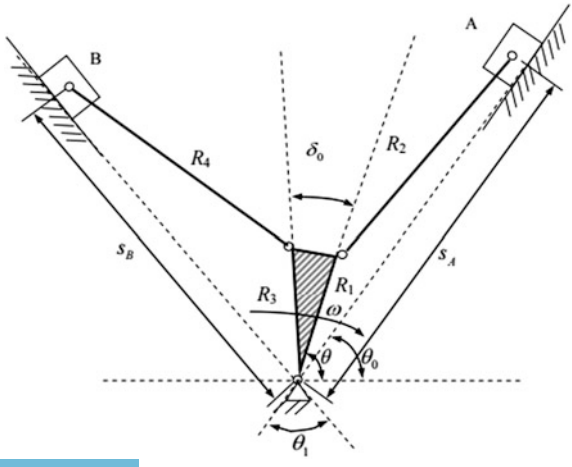


Fig. 6.14 Two-slider crank mechanism

c. Compute the time-dependent reliability using the CLS method with 10 time intervals evenly distributed.
d. Compute the time-dependent reliability using the outcrossing rate method.

References

1. Currin, C., Mitchell, T., Morris, M., & Ylvisaker, D. (1991). Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments. *Journal of the American Statistical Association, 86*(416), 953–963.
2. Li, J., Chen, J., & Fan, W. (2007). The equivalent extreme-value event and evaluation of the structural system reliability. *Structural Safety, 29*(2), 112–131.
3. Chen, J. B., & Li, J. (2007). The extreme value distribution and dynamic reliability analysis of nonlinear structures with uncertain parameters. *Structural Safety, 29*(2), 77–93.
4. Li, J., & Mourelatos, Z. P. (2009). Time-dependent reliability estimation for dynamic problems using a Niching genetic algorithm. *Journal of Mechanical Design, 131*(7), 071009.
5. Lutes, L. D., & Sarkani, S. (2009). *Reliability analysis of systems subject to first-passage failure* (NASA Technical Report No. NASA/CR-2009-215782).
6. Kuschel, N., & Rackwitz, R. (2000). Optimal design under time-variant reliability constraints. *Structural Safety, 22*(2), 113–127.
7. Li, C., & Der Kiureghian, A. (1995). Mean out-crossing rate of nonlinear response to stochastic input. In *Proceedings of ICASP-7*, Balkema, Rotterdam (pp. 295–302).
8. Schrupp, K., & Rackwitz, R. (1988). Out-crossing rates of marked Poisson cluster processes in structural reliability. *Applied Mathematical Modelling, 12*(5), 482–490.
9. Breitung, K. (1994). *Asymptotic approximations for the crossing rates of Poisson square waves* (pp. 75–75). NIST Special Publication SP.
10. Wang, Z., & Wang, P. (2013). A new approach for reliability analysis with time-variant performance characteristics. *Reliability Engineering and System Safety, 115,* 70–81.
11. Wang, Z., & Wang, P. (2012). A nested response surface approach for time-dependent reliability-based design optimization. *Journal of Mechanical Design, 134*(12), 121007(14).
12. Wang, P., Wang, Z., & Almaktoom, A. T. (2014). Dynamic reliability-based robust design optimization with time-variant probabilistic constraints. *Engineering Optimization, 46*(6), 784–809.
13. Hu, Z., & Mahadevan, S. (2016). A single-loop kriging surrogate modeling for time-dependent reliability analysis. *Journal of Mechanical Design, 138*(6), 061406(10).
14. Xu, H., & Rahman, S. (2005). Decomposition methods for structural reliability analysis. *Probabilistic Engineering Mechanics, 20*(3), 239–250.
15. Youn, B. D., & Xi, Z. (2009). Reliability-based robust design optimization using the Eigenvector Dimension Reduction (EDR) method. *Structural and Multidisciplinary Optimization, 37*(5), 475–492.
16. Xu, H., & Rahman, S. (2004). A generalized dimension-reduction method for multidimensional integration in stochastic mechanics. *International Journal for Numerical Methods in Engineering, 61*(12), 1992–2019.
17. Jones, D. R., Schonlau, M., & Welch, W. J. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization, 13*(4), 455–492.
18. Schonlau, M. (1997). *Computer experiments and global optimization* (Ph.D. Dissertation). University of Waterloo, Waterloo, Ontario, Canada.
19. Stuckman, B. E. (1988). A global search method for optimizing nonlinear systems. *IEEE Transactions on Systems, Man and Cybernetics, 18*(6), 965–977.
20. Žilinskas, A. (1992). A review of statistical models for global optimization. *Journal of Global Optimization, 2*(2), 145–153.

21. Koehler, J., & Owen, A. (1996). Computer experiments. In S. Ghosh & C. R. Rao (Eds.), *Handbook of statistics, 13: Design and analysis of experiments* (pp. 261–308). Amsterdam: Elsevier.
22. Sacks, J., Welch, W. J., Mitchell, T. J., & Wynn, H. P. (1989). Design and analysis of computer experiments. *Statistical Science, 4*(4), 409–423.
23. Mockus, J., Tiesis, V., & Zilinskas, A. (1978). The application of Bayesian methods for seeking the extreme. In L. C. W. Dixon & G. P. Szego (Eds.), *Towards global optimization* (Vol. 2, pp. 117–129). Amsterdam, The Netherlands: Elsevier.
24. Haftka, R. T., & Watsonft, L. T. (1999). Response surface models combining linear and Euler aerodynamics for supersonic transport design. *Journal of Aircraft, 36*(1), 75–86.
25. Madsen, J. I., Shyy, W., & Haftka, R. T. (2000). Response surface techniques for diffuser shape optimization. *AIAA Journal, 38*(9), 1512–1518.
26. Welch, W. J., Buck, R. J., Sacks, J., Wynn, H. P., Mitchell, T. J., & Morris, M. D. (1992). Screening, predicting, and computer experiments. *Technometrics, 34*(1), 15–25.
27. Wang, G. G., Dong, Z., & Aitchison, P. (2001). Adaptive response surface method—A global optimization scheme for approximation-based design problems. *Engineering Optimization, 33*(6), 707–733.
28. Keane, A. J., & Nair, P. B. (2005). *Computational approaches for aerospace design* (p. 582). Wiley: West Sussex.
29. Simpson, T. W., Mauery, T. M., Korte, J. J., & Mistree, F. (1998). Comparison of response surface and kriging models for multidisciplinary design optimization. *AIAA paper 98*, 4758(7).
30. Paciorek, C. J. (2003). *Nonstationary Gaussian processes for regression and spatial modelling* (Ph.D. dissertation). Carnegie Mellon University, Pittsburgh, PA.
31. Farhang-Mehr, A., & Azarm, S. (2005). Bayesian metamodeling of engineering design simulations: A sequential approach with adaptation to irregularities in the response behavior. *International Journal for Numerical Methods in Engineering, 62*(15), 2104–2126.
32. Qin, S., & Cui, W. (2003). Effect of corrosion models on the time-dependent reliability of steel plated elements. *Marine Structures, 16*(1), 15–34.
33. Madsen, H. O., Krenk, S., & Lind, N. C. (2006). *Methods of structural safety*. USA: Dover Publications.
34. Singh, A., Mourelatos, Z. P., & Li, J. (2010). Design for lifecycle cost using time-dependent reliability. *Journal of Mechanical Design, 132*(9), 091008.
35. Hagen, O., & Tvedt, L. (1991). Vector process out-crossing as parallel system sensitivity measure. *Journal of Engineering Mechanics, 117*(10), 2201–2220.
36. Rackwitz, R. (1998). Computational techniques in stationary and non-stationary load combination—A review and some extensions. *Journal of Structural Engineering, 25*(1), 1–20.
37. Breitung, K. (1988). Asymptotic crossing rates for stationary gaussian vector processes. *Stochastic Processes and Their Applications, 29*(2), 195–207.
38. Belyaev, Y. K. (1968). On the number of exits across the boundary of a region by a vector stochastic process. *Theory of Probability & Its Applications, 13,* 320–324.
39. Andrieu-Renaud, C., Sudret, B., & Lemaire, M. (2004). The PHI2 method: A way to compute time-variant reliability. *Reliability Engineering & System Safety, 84*(1), 75–86.
40. Sudret, B. (2008). Analytical derivation of the out-crossing rate in time-variant reliability problems. *Structure and Infrastructure Engineering, 4*(5), 353–362.
41. Zhang, J., & Du, X. (2011). Time-dependent reliability analysis for function generator mechanisms. *Journal of Mechanical Design, 133*(3), 031005(9).
42. Du, X. (2012). Toward time-dependent robustness metrics. *Journal of Mechanical Design, 134*(1), 011004(8).
43. Son, Y. K., & Savage, G. J. (2007). Set theoretic formulation of performance reliability of multiple response time-variant systems due to degradations in system components. *Quality and Reliability Engineering International, 23*(2), 171–188.

Chapter 7
Reliability-Based Design Optimization

As mentioned in earlier chapters, many system failures can be traced back to various difficulties in evaluating and designing complex systems under highly uncertain manufacturing and operational conditions. Our attempt to address this challenge continues with the discussion of reliability-based design optimization (RBDO). RBDO is a probabilistic approach to engineering system design that accounts for the stochastic nature of engineered systems. Our discussion of RBDO will cover the problem statement and formulation of RBDO as well as several probabilistic design approaches for RBDO.

7.1 Problem Statement and Formulation

It has been widely recognized that engineering design should account for the stochastic nature of design variables and parameters in engineered systems. Reliability-based design optimization (RBDO) integrates the techniques of reliability analysis and design optimization and offers probabilistic approaches to engineering design [1–11]. RBDO attempts to find the optimum design that minimizes a cost and satisfies an allocated reliability target with respect to system performance function(s), while accounting for various sources of uncertainty (e.g., material properties, geometric tolerances, and loading conditions). In general, a RBDO problem for an engineered system can be formulated as follows:

$$\begin{aligned} &\text{Minimize} && f(\mathbf{d}) \\ &\text{Subject to} && R = \Pr\left(G_j(\mathbf{X}, \boldsymbol{\Theta}; \mathbf{d}) \le 0\right) \ge \Phi(\beta^t) = R^t, \quad j = 1, \ldots, N_c \\ & && d_i^L \le d_i \le d_i^U, \quad i = 1, \ldots, N_d \end{aligned} \tag{7.1}$$

where the objective function is often the cost (e.g. price, volume, and mass) of the system, the random design vector $\mathbf{X} = (X_1, \ldots, X_{Nd})^{\mathrm{T}}$ and the random parameter vector $\boldsymbol{\Theta} = (\theta_1, \ldots, \theta_{Nr})^{\mathrm{T}}$, N_d and N_r are the number of design and parameter

C. Hu et al., *Engineering Design under Uncertainty and Health Prognostics*,
Springer Series in Reliability Engineering,
https://doi.org/10.1007/978-3-319-92574-5_7

variables, respectively, $\mathbf{d} = (d_1, \ldots, d_{Nd})^T = \boldsymbol{\mu}(\mathbf{X})$ is the mean design vector (or the design variable set), R is the reliability level of a given system design $\mathbf{d}$, G_j is the performance function of the jth design constraint, for $j = 1, \ldots, N_c$, N_c is the number of constraints, Φ is the cumulative distribution function of the standard normal distribution, R^t is the target reliability level, which corresponds to a target reliability index β^t, and d_i^L and d_i^U are, respectively, the lower and upper bounds on d_i, for $i = 1, \ldots, N_d$. The concept of RBDO and its comparison to deterministic design optimization (DDO) is illustrated with a two-dimensional design example in Fig. 7.1. In this example, the engineered system under design is the lower control arm of an army ground vehicle. The control arm serves as a connection between the wheels and the main body of the vehicle. The majority of dynamic loading in the suspension system is transmitted through the control arm, making it susceptible to fatigue failure. Here, a computer simulation model is built to predict the fatigue lives of the nodes on the surface of the control arm, and simulation-based design is used to optimize the thicknesses of two components, X_1 and X_2, with an aim to minimize the design cost and satisfy two fatigue design constraints.

The design constraints are defined in terms of the fatigue lives at two hotspots, and are graphically represented as two failure surfaces, $G_1(X_1, X_2) = 0$ and $G_2(X_1, X_2) = 0$, that separate the safe regions from the failure regions. For example, at any design point below the failure surface $G_1 = 0$, the fatigue life at the 1st hotspot is longer than the specification limit (i.e., $G_1 < 0$) and at any point above the surface, the fatigue life is shorter than the limit (i.e., $G_1 > 0$). Thus, on or below the surface is the safe region for the 1st constraint (i.e., $G_1 \leq 0$) and above the surface is the failure region (i.e., $G_1 > 0$). Similarly, on or to the left of the failure surface $G_2 = 0$ is the safe region for the 2nd design constraint (i.e., $G_2 \leq 0$) and to the right is the failure region (i.e., $G_2 > 0$). Therefore, any design point in the joint safe region is safe with respect to both constraints (i.e., $G_1 \leq 0$ and $G_2 \leq 0$). This joint region is treated as the system safe (or feasible) region in this design example. The concentric

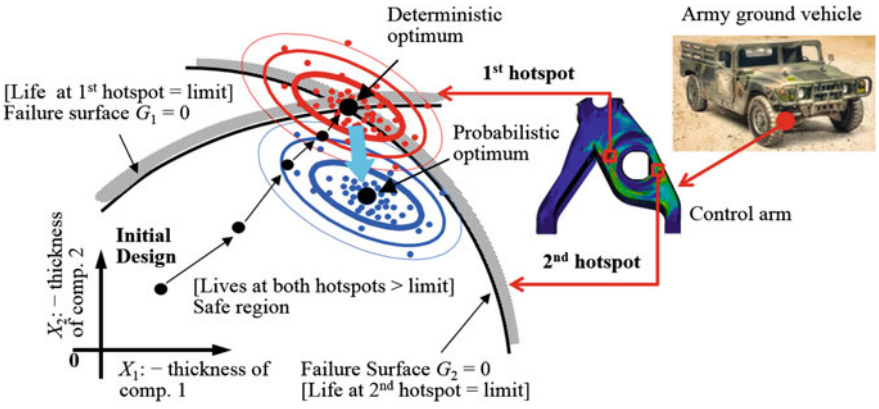


Fig. 7.1 Schematic of RBDO and its comparison with DDO in a two-dimensional design space

ellipses centering at either optimum design point $\boldsymbol{\mu}(\mathbf{X})$ represent the contours of the joint PDF of the two random variables X_1 and X_2. The small dots that scatter around the optimum design point are the design realizations that are randomly generated based on the joint PDF. Two important observations can be made from the figure:

- Starting with an initial design in the safe region, DDO and RBDO both attempt to find an optimum design solution that minimizes the objective function and satisfies the design constraints. In this case, the objective function decreases along the upper-right direction in the design space, i.e., the farther towards the upper right that the design point is located, the smaller the corresponding objective. Thus, both design approaches move the design point as far toward the upper-right as possible while keeping the design point in the safe region.
- DDO and RBDO differ intrinsically in the way they deal with the design constraints, and this difference leads to two different optimal design solutions (see the deterministic optimum and probabilistic optimum in Fig. 7.1). DDO considers two deterministic constraints, $G_1(d_1, d_2) \leq 0$ and $G_2(d_1, d_2) \leq 0$, that do not account for the uncertainty in the thicknesses of the components, while RBDO addresses two probabilistic constraints (or fatigue reliabilities), $\Pr(G_1(X_1, X_2) \leq 0) \geq R^t$ and $\Pr(G_2(X_1, X_2) \leq 0) \geq R^t$, that explicitly account for the uncertainty. Consequently, DDO pushes the optimum design point onto the boundary of the safe region, leaving little room for accommodating uncertainty in the design variables (depicted by red dots around the deterministic optimum design). By doing so, DDO finds the optimum design point that minimizes the objective function while satisfying the deterministic constraints. However, many random design realizations (or actual control arm units) around the deterministic optimum fall out of the safe region and are deemed unreliable. RBDO pushes the deterministic optimum design back to the safe region in order to create a safety margin that accommodates the uncertainty in the design variables. Due to the safety margin around the probabilistic optimum, this design solution is more reliable in that most of the random design realizations (blue dots) are located within the safe region.

In its search for the probabilistic optimum, RBDO must evaluate the feasibility of probabilistic constraints at a candidate design point through reliability analysis under uncertainty. Intuitively, RBDO needs to determine whether the safety margin shown in Fig. 7.1 is large enough to satisfy both probabilistic constraints. During the past two decades, many attempts have been made to develop efficient strategies to perform feasibility identification of the probabilistic constraints. These strategies can be broadly categorized as: double-loop RBDO [1–5], decoupled, sequential single-loop RBDO (or decoupled RBDO) [6], integrated single-loop RBDO (or single-loop RBDO) [7–11], and metamodel-based RBDO [12]. The rest of this chapter introduces these strategies as well as discusses an emerging design topic, RBDO under time-dependent uncertainty.

7.2 Double-Loop RBDO

The double-loop RBDO strategies often consist of a nested structure of outer and inner loops, where the outer loop performs design optimization in the original design variable space (**X**-space), and the inner loop performs reliability analysis at a given design point in the transformed standard normal space (**U**-space). The two steps (design optimization and reliability analysis) are iteratively repeated until the optimum design of an engineered system is found that minimizes the cost and meets the probabilistic constraints.

The *j*th probabilistic design constraint in the RBDO formulation shown in Eq. (7.1) can be expressed in terms of the CDF of the *j*th performance function as

$$\Pr\left(G_j(\mathbf{X}) \le 0\right) = F_{G_j}(0) \ge \Phi(\beta^t) \tag{7.2}$$

where $F_{G_j}(\cdot)$ is the CDF of G_j, and its value at $G_j = 0$ is the reliability of the *j*th design constraint, expressed as

$$F_{G_j}(0) = \int_{-\infty}^{0} f_{G_j}(g_j)\, dg_j = \int_{\Omega_j^S} f_{\mathbf{X}}(\mathbf{x})\, d\mathbf{x} \tag{7.3}$$

Here $f_{\mathbf{X}}(\mathbf{x})$ is the joint PDF of all the random variables, and Ω_j^S is the safety region defined as $\Omega_j^S = \{\mathbf{x} : G_j(\mathbf{x}) < 0\}$. For notational convenience, we assume the new random vector consists of both design and parameter vectors, i.e., $\mathbf{X} = [\mathbf{X}^{\mathrm{T}}, \boldsymbol{\Theta}^{\mathrm{T}}]^{\mathrm{T}}$. The evaluation of Eq. (7.3) is essentially reliability analysis under time-independent uncertainty, which has been extensively discussed in Chap. 5. Among the various methods for reliability analysis, an approximate probability integration method is known to provide efficient solutions; this is the first-order reliability method (FORM). Recall from Sect. 5.3 that reliability analysis in FORM requires a transformation **T** of the original random variables **X** to the standard normal random variables **U**. Correspondingly, the performance function $G(\mathbf{X})$ can be mapped from the original **X**-space onto the transformed **U**-space, i.e., $G[\mathbf{T}(\mathbf{X})] \equiv G(\mathbf{U})$.

There are in general three double-loop approaches to RBDO, and these approaches further express the probabilistic constraint in Eq. (7.2) through inverse transformations or using approximate statistical moments:

$$\text{RIA: } G_j^r = \beta^t - \Phi^{-1}\left(F_{G_j}(0)\right) = \beta^t - \beta_j^s \le 0 \tag{7.4}$$

$$\text{PMA: } G_j^p = F_{G_j}^{-1}(\Phi(\beta^t)) \le 0 \tag{7.5}$$

$$\text{AMA: } G_j^m = G_j(\boldsymbol{\mu}) + \upsilon\sigma_{G_j} \leq 0 \tag{7.6}$$

where β_j^s, G_j^p, and G_j^m are respectively the safety reliability index, probabilistic performance measure; and probabilistic constraint in AMA for the jth design constraint, υ is a target reliability coefficient, and σ_{G_j} is the second statistical moment of the performance function G_j. Equation (7.4) uses the reliability index to describe the probabilistic constraint in Eq. (7.1); this RBDO approach is called reliability index approach (RIA) [13]. Similarly, Eq. (7.5) replaces the probabilistic constraint in Eq. (7.1) with the probabilistic performance measure; this approach is known as performance measure approach (PMA). Because the (often) approximate first and second statistical moments in Eq. (7.6) are used to describe the probabilistic constraint in the RBDO formulation, this approach is termed the approximate moment approach (AMA), which originated from the robust design optimization concept [14, 15]. In what follows, these RBDO approaches will be introduced, and a focus will be put on RIA and PMA, which are more widely used in system design than AMA.

7.2.1 RIA in RBDO

In RIA, the probabilistic constraint in Eq. (7.2) is represented by using a reliability index, and an inverse transformation $\Phi^{-1}(\cdot)$ is used to convert an approximate reliability to a reliability index. Replacing the probabilistic constraints in Eq. (7.1) with Eq. (7.4) yields the following reformulation of the RBDO problem

$$\begin{aligned} &\text{Minimize} && f(\mathbf{d}) \\ &\text{Subject to} && G_j^r = \beta^t - \beta_j^s \leq 0, \quad j = 1, \ldots, N_c \\ & && d_i^L \leq d_i \leq d_i^U, \quad i = 1, \ldots, N_d \end{aligned} \tag{7.7}$$

The reliability index in the probabilistic constraint in RIA can be evaluated using FORM, which solves an optimization problem in the transformed **U**-space (see Sect. 5.3.1). In FORM, the equality constraint is the failure surface $G_j(\mathbf{U}) = 0$. The point on the failure surface that has the shortest distance to the origin is called the most probable point (MPP) $\mathbf{u}_j^*$, and the first-order estimate of the reliability index is defined as the distance between the MPP and origin, $\beta_i^{s,\text{FORM}} = \|\mathbf{u}_j^*\|$. Because of its simplicity and efficiency, the HL–RF method described in Sect. 5.3.2 is often employed to search for the MPP for reliability analysis in RIA.

The MPP search space in RIA is illustrated over a two-dimensional design space in Fig. 7.2, where the first-order reliability indices in Eq. (7.4) are $\beta_j^{s,\text{FORM}} = \beta_j(\mathbf{x}_j^*; \mathbf{d}_k) = \|\mathbf{T}(\mathbf{x}_j^*)\|$, $j = 1, 2$, at the kth design iteration. Reliability analysis in RIA is carried out by determining the minimum distance between the mean design point $\mathbf{d}_k$ and the MPP $\mathbf{x}_j^*$ on the failure surface $G_j(\mathbf{X}) = 0$, $j = 1, 2$. A comparison of the two probabilistic constraints in Fig. 7.2 suggests that the first constraint is slightly

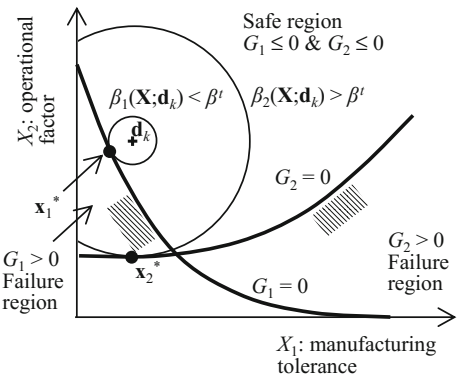


Fig. 7.2 Random search space of RIA in a two-dimensional design space

violated, i.e., $\beta_1(\mathbf{x}_1^*; \mathbf{d}_k) < \beta^t$, and the second is largely inactive, i.e., $\beta_2(\mathbf{x}_2^*; \mathbf{d}_k) > \beta^t$. Consequently, the MPP search space (see the smaller circle) for the first constraint is smaller than the search space (see the larger circle) for the second constraint. It has been reported in [1, 3] that the size of the MPP search space in reliability analysis could affect the efficiency of the MPP search but may not be a crucial factor. Rather, it was found that PMA with the spherical equality constraint (see Eq. (7.9) in Sect. 7.2.2) is often easier to solve than RIA with an often complicated constraint (i.e., $G_j(\mathbf{X}) = 0$). In other words, it is easier to minimize a complex cost function subject to a simple constraint function (PMA) than to minimize a simple cost function subject to a complicated constraint function (RIA).

7.2.2 PMA in RBDO

In PMA, the probability constraint in Eq. (7.2) is expressed as a probabilistic performance measure through an inverse transformation $F_{G_j}^{-1}(\cdot)$. The transformation converts the probability measure $\Phi(\beta^t)$ for the jth design constraint to a performance measure G_j^p. Replacing the probabilistic constraints with the corresponding performance measures, the RBDO problem in Eq. (7.1) can be reformulated as

$$\begin{aligned} &\text{Minimize} \quad f(\mathbf{d}) \\ &\text{Subject to} \quad G_j^p = F_{G_j}^{-1}(\Phi(\beta^t)) \le 0, \quad j = 1, \ldots, N_c \\ &\qquad\qquad d_i^L \le d_i \le d_i^U, \quad i = 1, \ldots, N_d \end{aligned} \tag{7.8}$$

The evaluation of the probabilistic constraint in PMA requires inverse reliability analysis, i.e., an inverse problem of reliability analysis. The first-order estimate of the probabilistic constraint G_j^p can be obtained by solving the following optimization problem with one equality constraint in the **U**-space, expressed as

$$\begin{array}{ll} \text{Maximize} & G_j(\mathbf{U}) \\ \text{Subject to} & \|\mathbf{U}\| = \beta^t \end{array} \tag{7.9}$$

The point on the target reliability surface $\|\mathbf{U}\| = \beta^t$ with the maximum value of the performance function is called the MPP $\mathbf{u}_j^*$ with the prescribed reliability $\|\mathbf{u}_j^*\| = \beta^t$. Then, the probabilistic performance measure is defined as $G_j^{p,\text{FORM}} = G(\mathbf{u}_j^*)$. Unlike RIA, the MPP search in PMA requires only the direction vector $\mathbf{u}_j^*/\|\mathbf{u}_j^*\|$ given the spherical equality constraint $\|\mathbf{U}\| = \beta^t$. Three numerical methods for PMA can be used to perform the MPP search in Eq. (7.9): the advanced mean value (AMV) method [15], the conjugate mean value (CMV) method [3, 17], and the hybrid mean value (HMV) method [3].

The MPP search space in PMA is illustrated over a two-dimensional design space in Fig. 7.3, where the first-order performance measures in Eq. (7.5) are $G_j^{p,\text{FORM}} = g_j(\mathbf{x}_j^*)$, $j = 1, 2$. Reliability analysis in PMA is carried out by determining the maximum performance value $G_i(\mathbf{x}_j^*)$ on the explicit sphere of the target reliability $\beta_j(\mathbf{X}; \mathbf{d}_k) = \beta^t$, $j = 1, 2$. Although the two probabilistic constraints in Fig. 7.3 significantly differ in terms of feasibility, they share the same MPP search space (see the circle in Fig. 7.3). As mentioned, it is often more efficient to perform the MPP search in PMA with the spherical equality constraint, than to perform RIA with a complicated constraint (i.e., $G_j(\mathbf{X}) = 0$).

AMV for MPP Search in PMA

The advanced mean value (AMV) method is, in general, well-suited for solving the optimization problem in Eq. (7.9) due to its simplicity and efficiency [16]. The first-order AMV method starts the MPP search with the initial MPP estimate expressed as follows [16]:

$$\mathbf{u}^{(0)} = \beta^t \mathbf{n}^{(0)} = \beta^t \frac{\nabla_{\mathbf{U}} G(\mathbf{u} = \mathbf{0})}{\|\nabla_{\mathbf{U}} G(\mathbf{u} = \mathbf{0})\|} \tag{7.10}$$

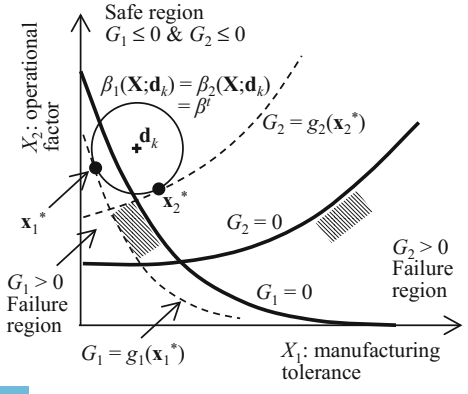


Fig. 7.3 Random search space of PMA in a two-dimensional design space

To maximize the objective function $G(\mathbf{U})$ in Eq. (7.9), the AMV method first uses the normalized steepest descent direction $\mathbf{n}^{(0)}$ obtained at the origin $\mathbf{u} = \mathbf{0}$ in the $\mathbf{U}$-space. Note that $\mathbf{u} = \mathbf{0}$ corresponds to the mean values of $\mathbf{X}$. In subsequent steps, the method iteratively updates the search direction at the current iteration $(k + 1)$ as the steepest descent direction $\mathbf{n}^{(k)}$ at the MPP $\mathbf{u}^{(k)}$ obtained at the previous iteration k. The iterative algorithm works according to the following:

$$\mathbf{u}^{(1)} = \beta^t \mathbf{n}^{(0)} = \beta^t \frac{\nabla_{\mathbf{U}} G\left(\mathbf{u}^{(0)}\right)}{\left\|\nabla_{\mathbf{U}} G(\mathbf{u}^{(0)})\right\|}, \quad \mathbf{u}^{(k+1)} = \beta^t \mathbf{n}^{(k)} = \beta^t \frac{\nabla_{\mathbf{U}} G\left(\mathbf{u}^{(k)}\right)}{\left\|\nabla_{\mathbf{U}} G(\mathbf{u}^{(k)})\right\|} \tag{7.11}$$

It was reported in [3] that the AMV method often behaves well for a convex performance function, but may exhibit instability and inefficiency for a concave performance function due to the sole use of the gradient information at the previous MPP.

CMV for MPP Search in PMA

When dealing with a concave performance function, the AMV method tends to show slow convergence or even divergence. This numerical difficulty can be addressed by employing an alternative MPP search method, the conjugate mean value (CMV) method, which updates the search direction through a combined use of the steepest descent directions at the three previous iterations [3, 17]. The update of the search direction at the current iterative $(k + 1)$, $k \geq 2$, is performed according to the following:

$$\mathbf{u}^{(k+1)} = \beta^t \frac{\mathbf{n}^{(k)} + \mathbf{n}^{(k-1)} + \mathbf{n}^{(k-2)}}{\left\|\mathbf{n}^{(k)} + \mathbf{n}^{(k-1)} + \mathbf{n}^{(k-2)}\right\|} \tag{7.12}$$

where

$$\mathbf{n}^{(k)} = \frac{\nabla_{\mathbf{U}} G\left(\mathbf{u}^{(k)}\right)}{\left\|\nabla_{\mathbf{U}} G(\mathbf{u}^{(k)})\right\|} \tag{7.13}$$

It can be observed from the above equations that the conjugate steepest descent direction is a weighted sum of the previous three consecutive steepest descent directions. This way of updating the search direction improves the rate of convergence and the stability over the AMV method for concave performance functions.

HMV for MPP Search in PMA

Although the CMV method works well on concave functions, the method is often less efficient than the AMV method for convex functions. To combine the strengths of the two methods, the hybrid mean value (HMV) method was developed and shown to attain both stability and efficiency in the MPP search in PMA [3]. The HMV method first determines the type (i.e., convex or concave) of a

performance function based on the steepest descent directions at three most recent iterations, and then adaptively selects one of the two algorithms, AMV or CMV, for the MPP search. A more detailed description of the method can be found in [3, 18].

7.2.3 AMA in RBDO

The RBDO problem in Eq. (7.1) can be redefined using AMA in Eq. (7.6) as

$$\begin{array}{ll} \text{Minimize} & f(\mathbf{d}) \\ \text{Subject to} & G_j^m = G_j(\boldsymbol{\mu}) + k\sigma_{G_j} \le 0, \quad j = 1, \ldots, N_c \\ & d_i^L \le d_i \le d_i^U, \quad i = 1, \ldots, N_d \end{array} \tag{7.14}$$

As described in Sect. 5.2, the second statistical moment (standard deviation) of the performance function can be approximated using the first-order Taylor series expansion at the mean values $\boldsymbol{\mu}_{\mathbf{X}}$ as

$$G_j(\mathbf{X}) \approx G_j(\boldsymbol{\mu}_{\mathbf{X}}) + \sum_{i=1}^{N} \frac{\partial G_j(\boldsymbol{\mu}_{\mathbf{X}})}{\partial X_i} \left(X_i - \mu_{X_i}\right) \tag{7.15}$$

This approximation may yield inaccurate results if a large number of random realizations of $\mathbf{X}$ are not close to the mean values $\boldsymbol{\mu}_{\mathbf{X}}$, which occurs if the standard deviations $\boldsymbol{\sigma}_{\mathbf{X}}$ of the random variables $\mathbf{X}$ are large. Similarly, a second-order Taylor series expansion can be used to approximate the standard deviation of the performance function [15], but this higher-order approximation may suffer from the same issue. Alternatively, the stochastic response surface methods (SRSMs) described in Sect. 5.4 can be used to approximate the second statistical moment, and often yields better accuracy with comparable efficiency. Unlike RIA and PMA, AMA does not require reliability analysis, but requires second- or higher-order sensitivity analysis that may demand a large amount of computational effort.

7.2.4 Comparisons of Different Double-Loop RBDO Approaches

The numerical behavior of different probabilistic approaches in RBDO has been studied in earlier works [1, 3]. It has been shown that PMA is better than RIA in terms of numerical efficiency and stability [1]. More specifically, there are several numerical advantages of PMA as compared to RIA. First, the convergence of the reliability analysis using PMA is inherently more robust and efficient than that using RIA because it is easier to minimize a complicated function subject to a simple constraint function using PMA than to minimize a simple function subject to

a complicated constraint function using RIA [1, 3, 17]. Second, the nonlinearity of the PMA reliability constraint is less dependent on probabilistic model types, such as normal, lognormal, Weibull, Gumbel, and uniform distributions, than the RIA reliability constraint [17]. Thus, RIA tends to diverge for distributions other than the normal distribution, whereas PMA converges well for all types of distributions.

Here, we also list some observations on AMA from an earlier study [18]. First, without knowing the output probabilistic distribution type, a reliability requirement is directly assigned by the first two moments of the performance function. Therefore, a non-normal and skewed output distribution with even a small variation produces a large error when estimating the reliability. Second, another numerical error can be generated when estimating the first two moments based on a Taylor series expansion at the mean values of the random variables. Third, AMA involves intensive computations because it requires the second- or higher-order sensitivity of the performance function to evaluate the sensitivity of the probabilistic constraint, whereas PMA and RIA require only the first-order sensitivity.

Table 7.1 summarizes the comparisons of these double-loop approaches in terms of several numerical attributes. As shown in Table 7.1, PMA is in general more desirable than RIA and AMA for RBDO from several numerical perspectives.

Table 7.1 Comparisons of RIA, PMA and AMA for RBDO

Attribute	RIA	PMA	AMA
Application domain	Reliability analysis and design optimization	Design optimization	Reliability analysis and design optimization
Required sensitivity information	First-order design sensitivity	First-order design sensitivity	Second- or higher-order design sensitivity
Ease in setting a target reliability	Easy to describe a target reliability using β_t	Easy to describe a target reliability using β_t	Difficult to describe precisely a target reliability using v
Capability in handling non-normal distributions of random variables	Capable of dealing with all distribution types	Capable of dealing with all distribution types	Not capable of handling different distribution types
Accuracy for non-normal and skewed output distribution	Accurate even with large variation	Accurate even with large variation	Inaccurate even with small variation
Efficiency and stability for high reliability and/or nonlinear performance function	Maybe inefficient and unstable	Efficient and stable	Efficient and stable

7.2.5 Reliability-Based Sensitivity Analysis in RBDO

In the design optimization process, an important component of RBDO is the sensitivity analysis of a reliability (or failure probability) estimate with respect to both random and deterministic design variables; this is known as reliability-based design sensitivity analysis. The sensitivity information is useful since it quantifies the effect of variations in design variables on the reliability of a system design. In what follows, the reliability-based design sensitivity analysis of the probabilistic constraint in RIA and PMA are presented.

Reliability-Based Design Sensitivity Analysis in RIA

In RIA, probabilistic constraints are expressed in terms of the safety reliability index. The derivative of safety reliability with respect to design parameter d_i, $i = 1, \ldots, n$ can be obtained by using a chain rule as

$$\begin{aligned}\frac{\partial \beta_j^{s,\mathrm{FORM}}}{\partial d_i} &= \left.\frac{\partial \left(\mathbf{U}^{\mathrm{T}}\mathbf{U}\right)^{1/2}}{\partial d_i}\right|_{\mathbf{U}=\mathbf{u}^*_{G_j(\mathbf{U})=0}} \\ &= \left.\frac{\partial \left(\mathbf{U}^{\mathrm{T}}\mathbf{U}\right)^{1/2}}{\partial \mathbf{U}} \cdot \frac{\partial \mathbf{U}}{\partial d_i}\right|_{\mathbf{U}=\mathbf{u}^*_{G_j(\mathbf{U})=0}} \\ &= \left.\frac{1}{2}\left(\mathbf{U}^{\mathrm{T}}\mathbf{U}\right)^{-1/2} \cdot \left(2\mathbf{U}^{\mathrm{T}}\right) \cdot \frac{\partial \mathbf{U}}{\partial d_i}\right|_{\mathbf{U}=\mathbf{u}^*_{G_j(\mathbf{U})=0}} \\ &= \left.\frac{\mathbf{U}^{\mathrm{T}}}{\beta_j^{s,\mathrm{FORM}}} \cdot \frac{\partial \mathbf{U}}{\partial d_i}\right|_{\mathbf{U}=\mathbf{u}^*_{G_j(\mathbf{U})=0}}\end{aligned} \tag{7.16}$$

where $\mathbf{u}^*_{G_j(\mathbf{U})=0}$ is the MPP in the $\mathbf{U}$-space that may be estimated by the HL-RF method (see Sect. 7.2.1). Using the transformation $\mathbf{U} = \mathbf{T}(\mathbf{X}; \mathbf{d})$, Eq. (7.16) can be rewritten as

$$\frac{\partial \beta_j^{s,\mathrm{FORM}}}{\partial d_i} = \left.\frac{\mathbf{T}(\mathbf{X};\mathbf{d})^{\mathrm{T}}}{\beta_{s,\mathrm{FORM}}} \frac{\partial \mathbf{T}(\mathbf{X};\mathbf{d})}{\partial d_i}\right|_{\mathbf{X}=\mathbf{x}^*_{G_j(\mathbf{X})=0}} \tag{7.17}$$

where $\mathbf{x}^*_{G_j(\mathbf{X})=0}$ is the corresponding MPP in the $\mathbf{X}$-space.

Reliability-Based Design Sensitivity Analysis in PMA

In PMA, probabilistic constraints are described in terms of the probabilistic performance measure, i.e., the performance measure evaluated at the MPP. The derivative of the estimate of the probabilistic performance measure with respect to design parameter d_i, $i = 1, \ldots, N_d$ can be obtained as

$$\frac{\partial G_j^{p,\mathrm{FORM}}}{\partial d_i} = \frac{\partial G_j(\mathbf{U})}{\partial d_i}\bigg|_{\mathbf{U}=\mathbf{u}^*_{\beta_j=\beta^t}} \tag{7.18}$$

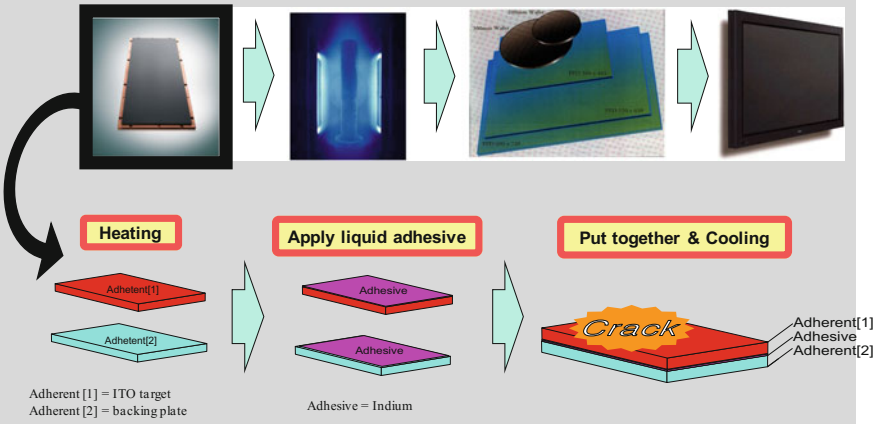
where $\mathbf{u}^*_{\beta_j(\mathbf{U})=\beta^t}$ is the MPP in the **U**-space that may be estimated by the AMV, CMV, or HMV method (see Sect. 7.2.2). Using the transformation $\mathbf{U} = \mathbf{T}(\mathbf{X}; \mathbf{d})$, Eq. (7.18) can be rewritten as

$$\frac{\partial G_j^{p,\mathrm{FORM}}}{\partial d_i} = \frac{\partial G_j(\mathbf{T}(\mathbf{X};\mathbf{d}))}{\partial d_i}\bigg|_{\mathbf{X}=\mathbf{x}^*_{\beta_j=\beta_t}} \tag{7.19}$$

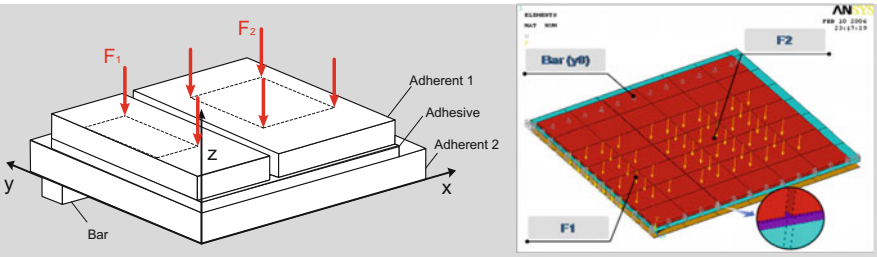
where $\mathbf{x}^*_{\beta_j(\mathbf{U})=\beta^t}$ is the corresponding MPP in the **X**-space.

Example 7.1 LCD Manufacturing Process
The bonding process of layered plates (called an Indium-Tin-Oxide (ITO) sputtering target process) is very popular in the manufacturing of semi-conductor or electronic display components. During this process, two plates (glass and copper) are bonded together by a suitable adhesive to form laminated stacks, which can be further processed in the following 4 steps:

(1) heating the two plates above the melting temperature of the adhesive;
(2) applying the adhesive at each surface of the plate;
(3) putting them in contact with each other;
(4) cooling them down to room temperature.



In this process, residual stress due to the mismatch of the thermal expansion coefficients of two layered plates could result in failures of the component,



(a) Isometric view of the quarter model (b) FE model

Fig. 7.4 Target bonding process and FE model for Problem 7.2

such as crack, distortion, and interfacial delamination. Therefore, it is important to accurately estimate the stress in order to improve the product quality. Here, a transient thermal finite element (FE) analysis was used to predict the stress and deformation of plates. The model for the layered bonding plates is shown in Fig. 7.4. Considering the symmetry of the problem, a quarter of the model is used, as shown in Fig. 7.4a. Due to the brittle property and high stress at the adherent 1, cracks and distortion could occur. To reduce such defects, weights are applied on top of the adherent 1, as shown in Fig. 7.4a from the beginning of the process, and are removed at the end of the cooling process. The bonding assembly is placed on a pair of supporting bars, as shown in Fig. 7.4a. Three design variables, weight at the edge (X_1 or F_2), weight at the center (X_2 or F_1), and height of the bar (X_3 or y_0), are considered in this problem. Their statistical information is shown in Table 7.2. The objective is to minimize the sum of the mean and standard deviation of residual stress. Two constraints are maximum stress during the process (<130 MPa) and center displacement (<3 mm).

A reliability-based robust design optimization (RBRDO) problem is formulated as

$$\begin{aligned} &\text{Minimize} && Q = \mu_r + \sigma_r \\ &\text{Subject to} && R_j = P(G_j(\mathbf{X}) \le 0) \ge \Phi(\beta^t), \quad j = 1, 2 \\ & && 2000 \le X_1 \le 10000; 1000 \le X_2 \le 5000; 1 \le X_3 \le 5; \end{aligned}$$

Table 7.2 Statistical properties of design variables in layered plate bonding model for Problem 7.2

Design variable	Distribution type	Mean	Std. dev.
X_1	Normal	4000	400
X_2	Normal	2000	200
X_3	Normal	1	0.1

where μ_r and σ_r are the mean and standard deviation of residual stress, $G_1(\mathbf{X})$ is the instantaneous stress, $G_2(\mathbf{X})$ is the edge displacement, and the target reliability index $\beta^t = 3$.
The eigenvector dimension reduction (EDR) method [19], as a variant of univariate dimension reduction (UDR) introduced in Sect. 5.5.1, is carried out to evaluate the quality (= mean + standard deviation) of residual stress and the reliabilities of two constraints. The sampling scheme for the EDR method is adaptively chosen in the RBRDO process to tackle the high nonlinearity in system responses. First, RBRDO starts with a $2N_d + 1$ sampling scheme for the EDR method. Then when satisfying a relaxed convergence criteria ($\varepsilon \leq 0.1$), the RBRDO process turns the $4N + 1$ sampling on. In this example, the standard deviation at the fourth design iteration is quite small but this estimation is not accurate enough because of highly nonlinear responses. Therefore, after the fourth design iteration, RBRDO is performed with the $4N + 1$ sampling scheme to enhance accuracy of the quality and reliability estimates. Sequential quadratic programming (SQP) is used as a design optimizer to solve the RBRDO problem. Table 7.3 shows the design history of this problem. After eight design iterations, an optimum design is found where X_2 is close to the upper bound. The EDR method requires in total 87 function evaluations for RBRDO. MCS with 1000 random samples is used to confirm the EDR results at the optimum design. It is found that the EDR estimates for the mean (μ_r) and standard deviation (σ_r) of the residual stress at the optimum design are very close to those using MCS. The overall quality is drastically improved by 38%.

7.2.6 *Limitations of Double-Loop RBDO*

The double-loop RBDO approaches require two nested optimization loops—an outer loop for design optimization and an inner loop for reliability analysis (see Fig. 7.5). The latter is needed to evaluate probabilistic constraints at each design iteration. The probability constraint in Eq. (7.1) is evaluated by the reliability index or the probabilistic performance measure, which requires the use of a reliability analysis method, such as the FORM. As the design iteration proceeds, the double-loop method tends to converge to a reliability-based optimum design while satisfying the feasibility of constraints. However, a prohibitive computational cost could be required due to the inherent nature of the nested structure. To overcome this inefficiency of the nested loop, (1) decoupled approaches and (2) single-loop approaches have been proposed. Figure 7.6 shows the categorization of RBDO methodologies and relevant literature.

Table 7.3 Design history of layered bonding plates model

Iteration	Obj	Mean	Std. dev.	X_1	X_2	X_3	G_1	G_2	# of FE evaluations
0	23.322	23.020	0.302	4000.000	2000.000	1.000	−94.876	1.051	7
1	21.437	21.350	0.087	4579.841	3633.035	2.317	−85.742	0.108	7
2	21.358	21.215	0.143	4659.514	4704.467	3.356	−79.354	−0.467	7
3	21.177	21.040	0.137	4316.124	5000.000	3.734	−77.240	−0.631	7
4	20.884	20.808	0.075	3121.245	5000.000	3.772	−77.371	−0.567	7
5	20.976	20.862	0.115	3121.245	5000.000	3.772	−77.342	−0.563	13
6	20.909	20.802	0.110	2752.275	4996.178	3.024	−80.775	−0.207	13
7	20.900	20.798	0.102	2554.780	4998.089	2.862	−81.861	−0.122	13
8	20.898	20.795	0.103	2520.106	4998.208	2.849	−82.046	−0.114	13
Optimum	20.898	20.795	0.103	2520.106	4998.208	2.849	Inactive	Inactive	87
MCS	20.891	20.786	0.105	2520.106	4998.208	2.849	Inactive	Inactive	1000

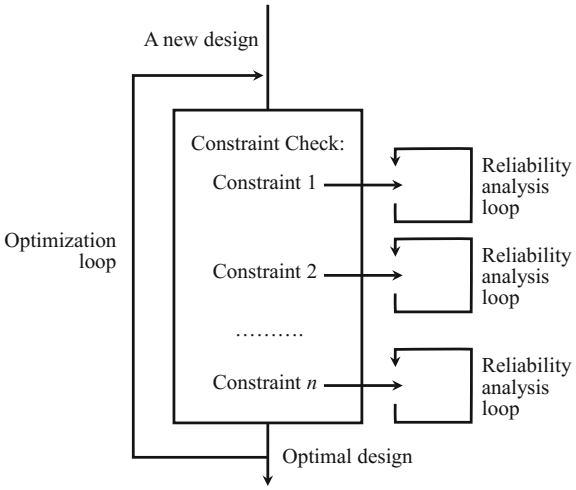


Fig. 7.5 Nested double-loop of RBDO. Reprinted (adapted) with permission from Ref. [20]

RBDO Methodologies

Double-Loop Approaches	Decoupled (or Sequential) Approaches	Single-Loop Approaches
- Sensitivity-based approximation [64] - RIA-based [1,12] - PMA-based [1-3, 5, 15, 16]	- Safety factor-based [66] - Sequential optimization and reliability assessment (SORA) [6] - Direct decoupling approach [20]	- Single-loop single vector (SLSV) [10] - SLSV with most-probable point [3] - SLSV with Karush-Kuhn-Tucker condition [7] - Complete single-loop [67] - Semi-single loop [68]

Fig. 7.6 Categorization of RBDO methodologies

7.3 Decoupled RBDO

One of strategies to decouple the nested loop is to decouple two loops: the outer loop for deterministic design optimization and the inner loop for reliability analysis. The separated two loops are performed sequentially until a design optimization converges (see Fig. 7.7). Compared to the double-loop RBDO, which conducts the reliability analysis for all design changes in the outer loop, the decoupled RBDO conducts the reliability analysis only once after the deterministic optimum design from the outer loop is achieved. That is, the outer loop may have several iterations but it does not call the inner loop each time. This reduces the number of the reliability analyses, which accounts for a majority of the computational cost. In the

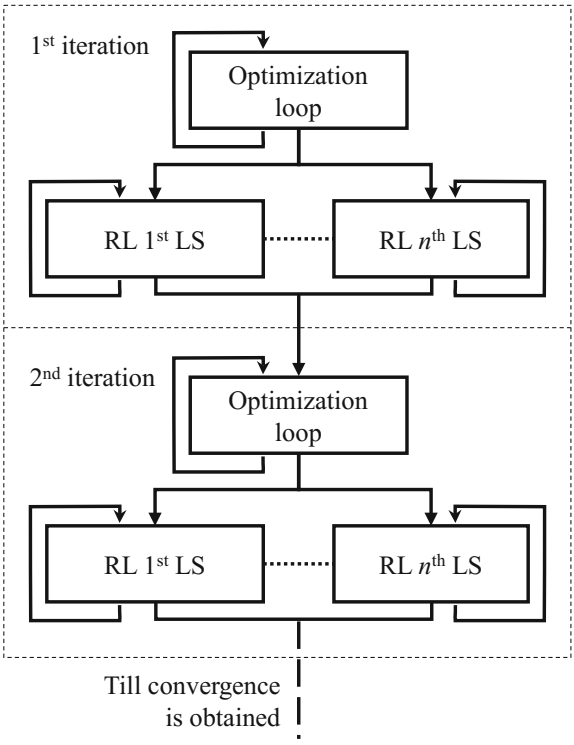


Fig. 7.7 Flowchart of decoupled RBDO [6, 21]; RL: Reliability analysis loop, LS: Limit state

outer loop, the deterministic design optimization is conducted with deterministic constraints approximating probabilistic constraints. Equation (7.20) shows the decoupled RBDO problem proposed in [6]. Its key idea is the use of shifting vector $\boldsymbol{s}_i^{(k+1)}$, which transforms deterministic constraints into shifted constraints comparable to the probabilistic constraints (see Fig. 5 in Ref. [6]). The shifting vector is updated after every reliability analysis, $\mathbf{s}_i^{(k+1)} = \boldsymbol{\mu}_{\mathbf{X}}^{(k)} - \mathbf{X}_{i\text{MPP}}^{(k)}$, and as the design optimization proceeds, the difference between the shifted and probabilistic constraints diminishes. This approach is based upon the assumption that the reliability levels of different designs are comparable, and can be estimated with the approximated deterministic constraints. The main concern of the decoupled RBDO is that the reliability levels of different designs are not always comparable and, therefore, the accuracy of reliability analysis can be low.

$$\begin{aligned} &\underset{\mathbf{d}}{\text{Minimize}} \quad f(\mathbf{d}) \\ &\text{Subject to } G_j \quad \left(\mathbf{d}, \boldsymbol{\mu}_{\mathbf{X}} - \mathbf{s}_j^{(k+1)}\right) \le 0, \quad j = 1, 2, \ldots, N_c \end{aligned} \tag{7.20}$$

where $\mathbf{s}_i^{(k+1)}$ is the shifting vector for the jth design constraint at the $(k+1)$th design iteration.

7.4 Single-Loop RBDO

Another strategy to decouple the nested loop is to eliminate the inner loop for reliability analysis by approximating the probabilistic constraints as deterministic ones. Once the probabilistic constraints are approximated into deterministic ones, a simple deterministic design optimization can be conducted without additional reliability analysis (see Fig. 7.8).

There are two major approaches to approximating the probabilistic constraints: single-loop single vector (SLSV) and Karush-Kuhn-Tucker (KKT) optimality condition. The SLSV approach leverages the sensitivity of the design variables to remove reliability analysis by finding the MPP iteratively. The main drawback of SLSV is that the active constraints should be identified in advance. The second approach with the KKT condition treats the inner loop of reliability analysis as the equality constraints in the outer design optimization loop. The single-loop formulation using the KKT optimality condition is given as follows

$$\begin{aligned} &\underset{\mathbf{d}}{\text{Minimize}}\, f(\mathbf{d}) \\ &\text{Subject to } G^{P^t} \cong G(\mathbf{d}, \mathbf{X}(\mathbf{u}^t)) \le 0 \\ &\text{where } \mathbf{u}^t \cong \beta^t \cdot \hat{\boldsymbol{\alpha}}^t \\ &\hat{\boldsymbol{\alpha}}^t \cong \left(-\frac{\nabla_{\mathbf{X}} G(\mathbf{d}, \mathbf{X}(\mathbf{u})) \mathbf{J}_{\mathbf{X},\mathbf{u}}}{\left\| \nabla_{\mathbf{X}} G(\mathbf{d}, \mathbf{X}(\mathbf{u})) \mathbf{J}_{\mathbf{X},\mathbf{u}} \right\|} \right)_{\mathbf{u}=\tilde{\mathbf{u}}} \end{aligned} \tag{7.21}$$

where G^{P^t} is P^t-percentile of the performance function $G(\cdot)$, $\beta^t = -\Phi^{-1}(P^t)$ is a target reliability index, $\mathbf{J}_{\mathbf{X},\mathbf{u}}$ is the Jacobian matrix of the transformation, and $\hat{\boldsymbol{\alpha}}^t$ is the negative normalized gradient vector of the performance function $G(\cdot)$ evaluated at the approximate location for the performance function value $\tilde{\mathbf{u}}$, which is the solution of KKT optimality condition.

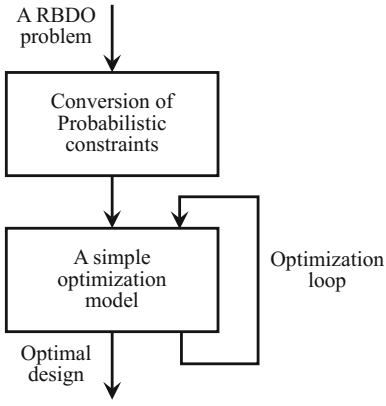


Fig. 7.8 Flowchart of single-loop RBDO. Reprinted (adapted) with permission from Ref. [20]

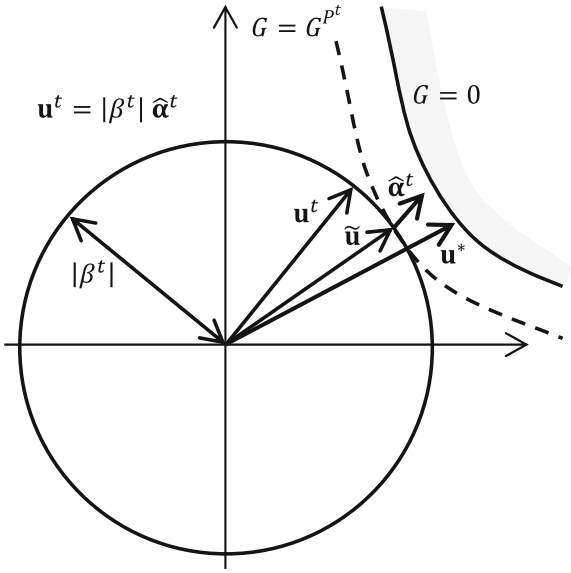


Fig. 7.9 Actual MPP ($\mathbf{u}^*$) and approximate location for the performance function value ($\mathbf{u}^t$). Reprinted (adapted) with permission from Ref. [22]

Instead of searching for the exact MPP at each design iteration, the single-loop approach obtains an approximate location for the performance function value $\tilde{\mathbf{u}}_j$ by solving the system equation given by the KKT condition. In Fig. 7.9, the dashed line $G = G^{P^t}$ is the approximate limit-state function from the conversion of the probabilistic constraint. This approach may require high computational resources in order to handle a large number of design variables and calculate second-order derivatives.

As the single-loop approaches do not require reliability analysis throughout the optimization process (see Fig. 7.8), they can reduce computational costs significantly. However, these approaches can also produce an infeasible design for highly nonlinear design problems in which the accuracy of the approximation is not guaranteed.

7.5 Metamodel-Based RBDO

This section introduces RBDO using metamodels, where Kriging-based surrogate models with adaptive sequential sampling are employed to perform reliability analysis and design sensitivity analysis in the RBDO framework. Section 7.5.1 first introduces the reliability analysis in the iterative RBDO process. Section 7.5.2 then presents the adaptive surrogate modeling with adaptive sequential sampling, and two case studies are presented in Sect. 7.5.3.

7.5.1 Review of Reliability Analysis with Surrogate Models

This section briefly reviews the use of the direct Monte Carlo simulation (MCS) based on surrogate models for reliability analysis. As discussed in Sect. 4.1, the probability of failure given a performance function $G(\mathbf{X})$ can be expressed as

$$P_f = P(G(\mathbf{X}) > 0) = \int \cdots \int_{G(\mathbf{x}) > 0} f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} \tag{7.22}$$

where $f_{\mathbf{X}}(\mathbf{x})$ is the joint PDF of the system random inputs $\mathbf{X}$. The direct MCS first draws from the distribution $f_{\mathbf{X}}(\mathbf{x})$ a large number of random samples, and then evaluates the performance function $G(\mathbf{X})$ at these random samples to estimate the probability of failure:

$$P_f = P(G(\mathbf{X}) > 0) = \int \cdots \int I_f(\mathbf{x}) f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} = E[I_f(\mathbf{x})] \tag{7.23}$$

where $E[.]$ is the expectation operator, and $I_f(\mathbf{x})$ represents an indicator function, defined as

$$I_f(\mathbf{x}) = \begin{cases} 1, & \text{if } G(\mathbf{x}) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{7.24}$$

7.5.2 Surrogate Modeling with Adaptive Sequential Sampling

This section introduces the surrogate modeling with adaptive sequential sampling (SMASS) approach. Section 7.5.2.1 first presents the Kriging-based surrogate modeling, whereas Sect. 7.5.2.2 introduces the sampling scheme for initial surrogate model development. Section 7.5.2.3 then presents a new classification confidence value (CCV)-based adaptive sequential sampling technique for the updating of Kriging surrogate models; Sect. 7.5.2.4 summarizes the procedure of the SMASS approach.

7.5.2.1 Kriging-Based Surrogate Modeling

Kriging is a nonparametric interpolation model that requires training samples for model construction and predicting the performance function values (or responses) at new sample points [23]. An ordinary Kriging model can be generally expressed as

$$G_K(\mathbf{X}) = \mu + Z(\mathbf{X}) \tag{7.25}$$

where μ is the mean response, and $Z(\mathbf{x})$ is Gaussian stochastic process with mean equal to zero and variance equal to σ^2. $G_K(\mathbf{X})$ is the Kriging predicted response as a function of $\mathbf{X}$. The covariance function between two input points, $\mathbf{x}_i$ and $\mathbf{x}_j$, is expressed as

$$\mathbf{Cov}_{(i,j)} = \sigma^2 \mathbf{R}_{(i,j)} \tag{7.26}$$

where $\mathbf{R}$ is the correlation matrix. The (i, j) entry of matrix $\mathbf{R}$ is defined as

$$\mathbf{R}_{(i,j)} = Corr(\mathbf{x}_i, \mathbf{x}_j) = \exp\left[-\sum_{p=1}^{N} a_p |x_i^p - x_j^p|^{b_p}\right] \tag{7.27}$$

where $Corr()$ is the correlation function; and a_p and b_p are parameters of the Kriging model. With n number of observations, $\mathbf{G}_{tr} = [G(\mathbf{x}_1), \ldots, G(\mathbf{x}_n)]$, at training samples $\mathbf{X}_{tr} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$, the log likelihood function of the Kriging model can be expressed as

$$Likelihood = -\frac{1}{2}[n \ln(2\pi) + n \ln \sigma^2 + \ln|\mathbf{R}| + \frac{1}{2\sigma^2}(\mathbf{G}_{tr} - \mathbf{A}\mu)^T \mathbf{R}^{-1}(\mathbf{G}_{tr} - \mathbf{A}\mu)] \tag{7.28}$$

where $\mathbf{A}$ is an $n \times 1$ unit vector. Then μ and σ^2 can be obtained by maximizing the likelihood function as

$$\mu = \left[\mathbf{A}^T \mathbf{R}^{-1} \mathbf{A}\right]^{-1} \mathbf{A}^T \mathbf{R}^{-1} \mathbf{G}_{tr} \tag{7.29}$$

$$\sigma^2 = \frac{(\mathbf{G}_{tr} - \mathbf{A}\mu)^T \mathbf{R}^{-1}(\mathbf{G}_{tr} - \mathbf{A}\mu)}{n} \tag{7.30}$$

With the Kriging model, the response for any given point $\mathbf{x}'$ can be estimated as

$$G_K(\mathbf{x}') = \mu + \mathbf{r}^T \mathbf{R}^{-1}(\mathbf{G}_{tr} - \mathbf{A}\mu) \tag{7.31}$$

where $\mathbf{r}$ is the correlation vector between $\mathbf{x}'$ and the training points $\mathbf{X}_{tr} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$, and the ith element of $\mathbf{r}$ is given by $\mathbf{r}(i) = Corr(\mathbf{x}', \mathbf{x}_i)$. The mean square error $e(\mathbf{x}')$ can be estimated by

$$e(\mathbf{x}') = \sigma^2 \left[1 - \mathbf{r}^T \mathbf{R}^{-1} \mathbf{r} + \frac{(1 - \mathbf{A}^T \mathbf{R}^{-1} \mathbf{r})^2}{\mathbf{A}^T \mathbf{R}^{-1} \mathbf{A}}\right] \tag{7.32}$$

Therefore, the prediction of response at point $\mathbf{x}'$ from a Kriging model can be considered as a random variable that follows a normal distribution with mean $G_K(\mathbf{x}')$ and variance $e(\mathbf{x}')$.

7.5.2.2 Sampling for Initial Surrogate Model Development

A crucial issue in sequential surrogate modeling is the sampling strategy used to generate random sample points in order to construct the initial low fidelity surrogate model. The one-step sampling method, i.e., Latin hypercube sampling (LHS) method [24, 25] has been widely used for this purpose. By using the LHS, the random input domain will be occupied most evenly by sample points, thereby enabling much information about the true model to be obtained. However, it usually provides similar sample point profiles to occupy the random input domain evenly regardless of the distribution of true system responses. On the contrary, the importance sampling technique [26, 27] generates sample points around the limit state area and predicts the response accurately around the limit state. However, it usually only provides good local surrogate models and does not represent the true model accurately enough in other areas of the random input domain. Besides the LHS method and the importance sampling method, there are other advanced sampling techniques, such as the Improved Distributed Hypercube Sampling (HIS) [27], orthogonal-array-based LHS designs [29], and maximin LHS designs [30], which can all be used to generate sample points to construct the initial surrogate model. A comparison of these different sampling methods can be found in Ref. [31]. Without losing the generosity, the LHS method is employed in this study to generate the initial Kriging model for the SMASS approach.

7.5.2.3 Confidence-Based Adaptive Sequential Sampling

With an initial set of sample points $\mathbf{X}_{\mathrm{E}}$ and system responses $\mathbf{Y}_{\mathrm{E}}$, a Kriging model $\boldsymbol{M}$ ($\equiv G_K$) can be constructed accordingly. However, this Kriging model usually has a low fidelity, and thus needs to be updated. This subsection introduces a new confidence-based adaptive sampling scheme for sequential updating of the Kriging models.

The prediction of the response at point $\mathbf{x}_i$ from a Kriging model can be considered as a random variable that follows normal distribution. For any given sample point $\mathbf{x}_i$, based on the Kriging prediction of its response, $G_K(\mathbf{x}_i)$, it can be accordingly classified as a sampling point in the failure region or safe region. With this classification, all Monte Carlo sample points can be accordingly categorized into two classes, as shown in Fig. 7.10, the failure class and the safe class, where the failure class includes all sample points at which the predicted responses $G_K(\mathbf{x}_i) > 0$, and the safe class at which $G_K(\mathbf{x}_i) \leq 0$. Knowing that the Kriging prediction can be considered as a random variable, thus the classification of sample point becomes probabilistic. Here we define the probability of having a correct

classification of a sample point as the classification confidence value (CCV). To compute the CCV, the sample points at two difference classes must be treated differently. For sample points in the failure class, since failure is defined as $G(\mathbf{x}_i) > 0$, the CCV value indicates the probability that the sample point is at the failure region, which can be accordingly calculated as the area of the normal cumulative distribution function in the interval of $(0, \infty)$ as

$$\mathrm{CCV}(\mathbf{x}_i) = P(G(\mathbf{x}_i) > 0) = \frac{1}{\sqrt{2\pi \cdot e(\mathbf{x}_i)}} \int_0^{\infty} e^{-\frac{1}{2}\frac{[y - G_K(\mathbf{x}_i)]^2}{e(\mathbf{x}_i)}} dy, \quad \text{for all } i \text{ where } G_K(\mathbf{x}_i) > 0 \tag{7.33}$$

where $G_K(\mathbf{x}_i)$ and $e(\mathbf{x}_i)$ are the predicted response at point $\mathbf{x}_i$ and the standard deviation of the prediction, respectively. Similarly, for sample points in the safe class, the CCV value indicates the probability that the sample point is at the safe region, which can be accordingly calculated as the area of the normal cumulative distribution function in the interval of $(-\infty, 0)$ as

$$\mathrm{CCV}(\mathbf{x}_i) = P(G(\mathbf{x}_i) \le 0) = \frac{1}{\sqrt{2\pi \cdot e(\mathbf{x}_i)}} \int_{-\infty}^{0} e^{-\frac{1}{2}\frac{[y - G_K(\mathbf{x}_i)]^2}{e(\mathbf{x}_i)}} dy, \quad \text{for all } i \text{ where } G_K(\mathbf{x}_i) \le 0 \tag{7.34}$$

Based upon the definition, it is clear from Fig. 7.10 that the CCV should be a positive value within (0.5, 1), where a higher value indicates higher classification confidence. Combining Eqs. (7.33) and (7.34), the CCV of the sample point $\mathbf{x}_i$ can be generally calculated as

$$\mathrm{CCV}(\mathbf{x}_i) = \Phi\left(\frac{|G_k(\mathbf{x}_i)|}{\sqrt{e(\mathbf{x}_i)}}\right), \quad i = 1, 2, \ldots, n \tag{7.35}$$

where Φ is standard normal cumulative distribution function, $|.|$ is the absolute operator, n is the total number of Monte Carlo samples, and $G_K(\mathbf{x}_i)$ and $e(\mathbf{x}_i)$ is the predicted response at the sample point $\mathbf{x}_i$ and the standard deviation of this

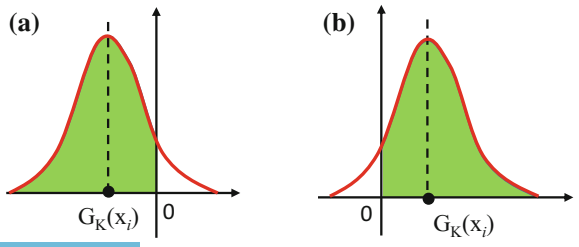


Fig. 7.10 Sample classification: **a** $G_K(\mathbf{x}_i) < 0$, $\mathbf{x}_i$ is classified as a safe sample point, and **b** $G_K(\mathbf{x}_i) > 0$, $\mathbf{x}_i$ is classified as the failure sample point

prediction, respectively. These values can be obtained directly from the constructed Kriging model. By using Eq. (7.35), failure potentials of the Monte Carlo samples can be calculated based on their Kriging-predicted means and standard deviations of the responses.

In order to improve the fidelity of the Kriging model, especially for the problems with multiple disjointed failure regions, sample points must be accordingly chosen from different disjointed failure regions during the sequential Kriging model updating process, as those sample points could bring much information about the system performance function at particular failure regions and thus are more valuable. Therefore, in the developed SMASS approach, the sample point will be selected based upon a sampling rule of improving the classification confidence values using the Kriging model, thus the sample point with the minimum CCV, $\mathbf{x}^*$, will be selected in each Kriging model updating iteration, and the corresponding performance value $\mathbf{y}^*$ will be evaluated. This selected sample $\mathbf{x}^*$ with its actual response value $\mathbf{y}^*$ is then added into $\mathbf{X}_E$ and $\mathbf{Y}_E$, respectively, and the Kriging model will be accordingly updated with new sample points added. To prevent same sample points being used repeatedly in different updating iterations, the selected sample point $\mathbf{x}^*$ will be excluded from the original Monte Carlo samples in each updating iteration. The updated Kriging model is then used to predict the responses of Monte Carlo samples again. This search and update process works iteratively and it is terminated when the minimum CCV value reaches a predefined threshold, CCV_t. This stopping rule is defined as

$$\min \quad CCV_i \geq CCV_t, \quad i = 1, 2, \ldots, n \tag{7.36}$$

where CCV_i is the CCV value for the sample point $\mathbf{x}_i$, and CCV_t is the predefined CCV threshold. To ensure a good balance of accuracy and efficiency, it is suggested that a value of CCV_t defined between [0.95, 1) is desirable; in this study, 0.95 has been used for the CCV_t. In implementation of the SMASS approach, it is suggested that the minimum CCV_i in Eq. (7.36) can often be replaced by the average minimum CCV value obtained at the last few updating iterations (e.g., the last five iterations), in order to ensure a more robust convergence.

7.5.2.4 Procedure of Reliability Analysis Using SMASS

The detailed procedure of the approach is presented in this section. A flowchart of the SMASS approach in Fig. 7.11 reveals three consecutive steps: (1) produce an initial set of sample points $\mathbf{X}_E$ and the corresponding system responses $\mathbf{Y}_E$ in order to develop the initial Kriging surrogate model. In this study, the Latin Hypercube Sampling (LHS) method is used for this purpose, and the system responses $\mathbf{Y}_E$ of the generated samples, $\mathbf{X}_E$, are then evaluated through the design of experiments. (2) Use $\mathbf{X}_E$ and $\mathbf{Y}_E$ to construct a Kriging model, $\boldsymbol{M}$, and evaluate the performance of the model. In this step, the Kriging model is used to predict the responses at a set of Monte Carlo samples, and the minimum CCV of the predictions, as explained in

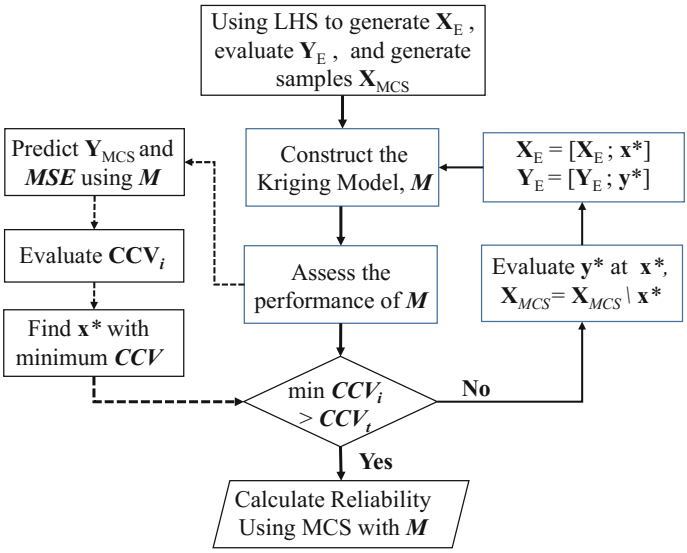


Fig. 7.11 Flowchart of the SMASS approach

Sect. 7.5.2.3, is used as the performance metric of the Kriging model $\boldsymbol{M}$. (3) Update the Kriging model iteratively by adding sample points into $\mathbf{X}_E$ and $\mathbf{Y}_E$ using the CCV-based sampling scheme. The updated Kriging model is used to predict the responses for a new set of Monte Carlo samples for the probability of failure estimation.

7.5.3 *Stochastic Sensitivity Analysis with Surrogate Models for RBDO*

Design sensitivity information of reliability with respect to random design variables is essential in the iterative design process, as it not only affects the efficiency but also determines the convergence of the design process to an optimum design. In RBDO, while finite different method (FDM) is adopted for design sensitivity analysis, reliability analysis needs to be performed *nd* times for the perturbed design points where *nd* is number of design variables.

For design sensitivity analysis using sampling-based methods, taking the partial derivative of probability of failure with respect to the *i*th design variable d_i yields

$$\frac{\partial P_f}{\partial d_i} = \frac{\partial}{\partial d_i} \int_{R^{nr}} I_f(\mathbf{x}) f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} \tag{7.37}$$

According to the Leibniz's rule of differentiation [12], the differential and integral operators in the Eq. (7.37) can be interchanged, thus yielding the following

$$\begin{aligned}\frac{\partial P_f}{\partial d_i} &= \int_{R^{nr}} I_f(\mathbf{x}) \frac{\partial f_{\mathbf{X}}(\mathbf{x})}{\partial d_i} d\mathbf{x} \\ &= \int_{R^{nr}} I_f(\mathbf{x}) \frac{\partial \ln f_{\mathbf{X}}(\mathbf{x})}{\partial d_i} f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} = E[I_f(\mathbf{x}) \frac{\partial \ln f_{\mathbf{X}}(\mathbf{x})}{\partial d_i}]\end{aligned} \tag{7.38}$$

Although the analytical form for the sensitivity of reliability can be derived, it cannot be used to compute the sensitivity when all the samples in MCS are identified as safe. If I_{RF} equals 0 for all the N-samples, Eq. (7.38) becomes

$$\frac{\partial P_f}{\partial d_i} = \frac{\partial}{\partial d_i} \int_{R^{nr}} I_f(\mathbf{x}) f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} = I_f(\mathbf{x}) \times \frac{\partial}{\partial d_i} \int_{R^{nr}} f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} = 0 \tag{7.39}$$

Although zero estimation of the sensitivity based on MCS samples may not lead to a divergence of the RBDO process, it could result in substantially more design iterations since the new design point in the subsequent design iteration will be affected by this sensitivity estimation. This is especially true for a high reliability target scenario in RBDO, as the sensitivity estimated using the MCS samples based on Eq. (7.38) will frequently be zero. In same extreme cases, the non-smooth sensitivity estimation will substantially increase the total number of design iterations, and could also make the RBDO process fail to converge to the optimum design. To alleviate such a difficulty, a new way to calculate smooth sensitivity of reliability without extra computational cost is presented here [32]. Defined as the integration of the probability density function of system input variables over the safe region ($G(\mathbf{X}) \leq 0$), the reliability has a monotonically one-to-one mapping relationship to the ratio of the mean and the variance of the performance function, which can be expressed as

$$\begin{aligned}R = P(G(\mathbf{X}) \leq 0) &= \int \cdots \int_{G(\mathbf{x})<0} f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} \approx \Phi\left(\frac{\mu_{G(\mathbf{x})}}{\sigma_{G(\mathbf{x})}}\right) \propto \frac{\mu_{G(\mathbf{x})}}{\sigma_{G(\mathbf{x})}} \\ &= \frac{\int_{R^{nr}} G(\mathbf{x}) f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}}{\sqrt{\int_{R^{nr}} (G(\mathbf{x}) - \mu_{G(\mathbf{x})})^2 f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}}}\end{aligned} \tag{7.40}$$

where $\mu_{G(\mathbf{x})}$ and $\sigma_{G(\mathbf{x})}$ are the mean and variance of performance function $G(\mathbf{X})$ given the random input $\mathbf{x}$. It is should be noticed that the failure probability is computed by the integration of the probability density function over all of the failure region, and can be determined by the randomness properties of input $\mathbf{x}$ and performance function $G(\mathbf{x})$. Thus, the probability of failure is a function of the design variable $\mathbf{d}$. In this equation, the approximate equality should be equality if

the random response $G(\mathbf{x})$ follows the Gaussian distribution given the randomness of input $\mathbf{x}$. The sensitivity of reliability with respect to the design variable $\mathbf{d}$ can then be approximated as

$$\frac{\partial R}{\partial \mathbf{d}} \propto \nabla \frac{\mu_{G(\mathbf{x})}}{\sigma_{G(\mathbf{x})}} \tag{7.41}$$

Note that the right part of Eq. (7.41) only provides an estimated sensitivity vector that is proportional to the true design sensitivity. Thus, the sensitivity information in Eq. (7.41) can be normalized and derived as

$$\frac{\partial R}{\partial \mathbf{d}} \approx a \times \nabla \left[\frac{\mu_{G(\mathbf{x})}}{\sigma_{G(\mathbf{x})}}\right] \Big/ \left\| \nabla \left[\frac{\mu_{G(\mathbf{x})}}{\sigma_{G(\mathbf{x})}}\right] \right\| \tag{7.42}$$

where a is a proportional coefficient; $\|.\|$ is norm operation. To calculate the sensitivity of reliability, the derivative term on the right-hand side of Eq. (7.42) can be derived as

$$\frac{\partial}{\partial \mathbf{d}} \left[\frac{\mu_{G(\mathbf{x})}}{\sigma_{G(\mathbf{x})}}\right] = \frac{\partial \mu_{G(\mathbf{x})}}{\partial \mathbf{d}} \times \frac{1}{\sigma_{G(\mathbf{x})}} - \mu_{G(\mathbf{x})} \times \frac{1}{{\sigma_{G(\mathbf{x})}}^2} \times \frac{\partial \left[\sigma_{G(\mathbf{x})}\right]}{\partial \mathbf{d}} \tag{7.43}$$

The derivative of $\mu_{G(\mathbf{x})}$ and $\sigma_{G(\mathbf{x})}$ with respect to $\mathbf{d}$ can be estimated as

$$\begin{aligned}
\frac{\partial \left[\mu_{G(\mathbf{x})}\right]}{\partial \mathbf{d}} &= \frac{\partial \left[\int_{R^{nr}} G(\mathbf{x}) f_{\mathbf{X}}(\mathbf{x}) dx\right]}{\partial \mathbf{d}} \\
&\approx \frac{1}{N} \sum_{i=1}^{N} \frac{\partial \left[G_K(\mathbf{x}_{m,i})\right]}{\partial \mathbf{d}} \\
\frac{\partial \left[\sigma_{G(\mathbf{x})}\right]}{\partial \mathbf{d}} &= \frac{\partial \left[\int_{R^{nr}} \left(G(\mathbf{x}) - \mu_{G(\mathbf{x})}\right)^2 f_{\mathbf{X}}(\mathbf{x}) dx\right]}{\partial \mathbf{d}} \\
&\approx \frac{1}{N} \sum_{i=1}^{N} [2 \times G_K(\mathbf{x}_{m,i}) \times \frac{\partial \left[G_K(\mathbf{x}_{m,i})\right]}{\partial \mathbf{d}} - 2 \times \mu_{G(\mathbf{x})} \times \frac{\partial \left[G_K(\mathbf{x}_{m,i})\right]}{\partial \mathbf{d}} \\
&\quad - 2 \times G_K(\mathbf{x}_{m,i}) \times \frac{\partial \left[\mu_{G(\mathbf{x})}\right]}{\partial \mathbf{d}} + 2 \times \mu_{G(\mathbf{x})} \times \frac{\partial \left[\mu_{G(\mathbf{x})}\right]}{\partial \mathbf{d}}]
\end{aligned} \tag{7.44}$$

To calculate the sensitivity of reliability appropriately, the proportional coefficient a should be determined. In this paper, a is set to one initially at the first iteration and will be updated based on the reliabilities of current and previous designs. Let R_k and R_{k+1}, respectively, represent the reliabilities of the designs at the kth and $(k + 1)$th iterations, $\mathbf{d}_k$ and $\mathbf{d}_{k+1}$ be the designs at the kth and $(k + 1)$th iterations, and SR_i denote the sensitivity of reliability calculated using Eq. (7.42) at

the ith iteration. Also let a_k and a_{k+1} be the proportional coefficients for kth and $(k + 1)$th iterations. With these notations, the proportional coefficient a_{k+1} can be updated by

$$a_{k+1} = \begin{cases} -\frac{R_{k+1} - R_k}{(d_{k+1} - d_k) \cdot (SR_k)}, & \text{if} |R_{k+1} - R_k| > C_a \\ a_k, & \text{otherwise} \end{cases} \tag{7.45}$$

where C_a is a predefined critical threshold within 10^{-1} to 10^{-4}.

7.5.4 Case Studies

Two case studies are employed in this section to demonstrate the proposed approach to reliability analysis of problems with disjointed active failure regions.

7.5.4.1 A Mathematical Problem

In this example, a two-dimensional nonlinear limit state function $G(\mathbf{x})$ is provided as shown in Eq. (7.46), in which the two random variables follow normal distributions with $X_1 \sim N(0, 0.8^2)$ and $X_2 \sim N(2, 0.8^2)$, respectively.

$$G(\mathbf{X}) = \frac{1}{20}(X_1^2 + 4) \times (X_2 - 1) - \sin(2.5X_1) - 2 \tag{7.46}$$

The contour of the limit state function $G(\mathbf{X}) = 0$ in Fig. 7.12 shows three disjointed active failure regions, denoted in the figure by failure region 1, failure region 2, and failure region 3, respectively. The probability of failure analysis of this mathematical example is conducted with the direct MCS with a very large number of sample points in order to come up with a benchmark probability of failure value so that the accuracy of other reliability analysis methods, including the develop SMASS approach, can be compared. With a sample size of 10^8, direct MCS produces an estimated probability of failure of 2.56×10^{-4}.

The FORM is also implemented for the case study. As shown in Fig. 7.12, after a total of 84 iterations, the MPP search converges to the MPP, [−2.895363, 2.911457], which gives an estimated reliability index of 3.7943. Accordingly, the probability of failure can be estimated by the FORM as 7.4×10^{-5}. For each iteration of MPP search, there are three function evaluations for calculating the performance function value and its first-order gradient information with respect to the input variables $\mathbf{X}$. Therefore, the FORM requires a total of 252 function evaluations. It can be seen from Fig. 7.12 that the FORM method can only find one MPP in one failure region while ignoring all other potential failure regions. This results in substantial errors in the probability of failure estimation, as compared with the MCS estimate.

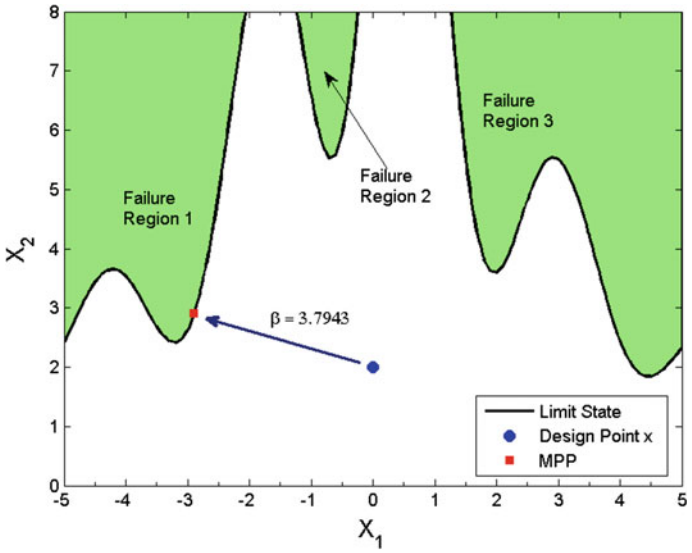


Fig. 7.12 Contour of the limit state function $G(\mathbf{X}) = 0$ and the MPP found by the FORM

A dimension reduction (DR) method, univariate dimension reduction (UDR) as introduced in Sect. 5.5.1, is also employed for this case study. The UDR method estimates the probability density function of the performance function through decomposing a multivariate joint probability function to multiple univariate probability functions using the additive decomposition method. As shown in Fig. 7.13, the approximated probability density function of $G(\mathbf{X})$ using UDR is generally with a single mode. However, when there are multiple disjointed active failure regions, the PDF of the performance function $G(\mathbf{X})$ is usually multimode. With two random input variables, the DR method only requires 5 function evaluations. However, due

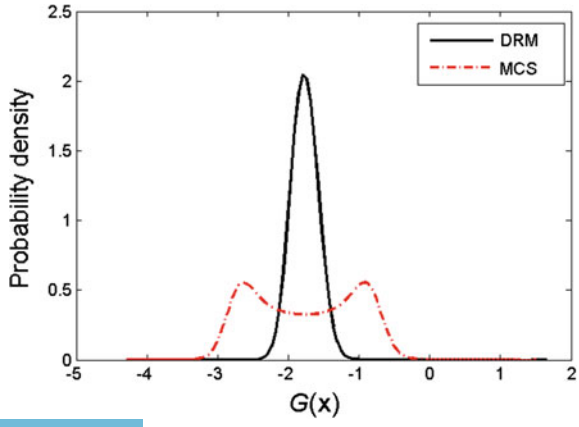


Fig. 7.13 Approximated PDFs of $G(\mathbf{X})$ by the MCS and DRM

to the incapability of estimating multimode probability density functions, the failure probability obtained by DRM, which is the 2.78×10^{-5} in this case study, shows a large error, as compared to the MCS result.

In order to further demonstrate the efficacy of the developed SMASS method, the Kriging surrogate modeling approach without adaptive sequential sampling (referred to as "simple Kriging") is employed for the case study. In order to construct a relatively accurate Kriging surrogate model, a total number of 150 training sample points have been used. In order to avoid the randomness effect of the training sample points generated by the LHS, the Kriging surrogate model has been repeatedly developed 100 times, and the mean estimated probability of failure based upon these 100 Kriging models is 2.17×10^{-4}.

The SMASS approach is applied to the case study. As the procedure of SMASS introduced in Sect. 7.5.2.4, in this case study it is implemented first with 15 initial LHS sample points, as denoted by the blue circle points in Fig. 7.14. Meanwhile, a total of 10^5 Monte Carlo sample points, $\mathbf{X}_{MCS}$, are also generated in order to identify the best sample points for the sequential updating of the Kriging model. After 23 iterations of updating of the Kriging model, the minimum CCV value for all the $\mathbf{X}_{MCS}$ sample points satisfies the target classification confidence value, CCV_t, which has been set as 0.95 in this case study. The 23 sequentially sampled points for the updating of the Kriging model are shown in Fig. 7.14 with stars. It is clear that the developed SMASS approach is able to locate most of the sample points to the disjointed active failure regions, in order to enhance the fidelity of the Kriging model. With a total of 38 evaluations of the performance function, $G(\mathbf{X})$, the approximated limit state function generated by the developed Kriging model, as shown by the red dash line in the figure, has a very good match with the true limit

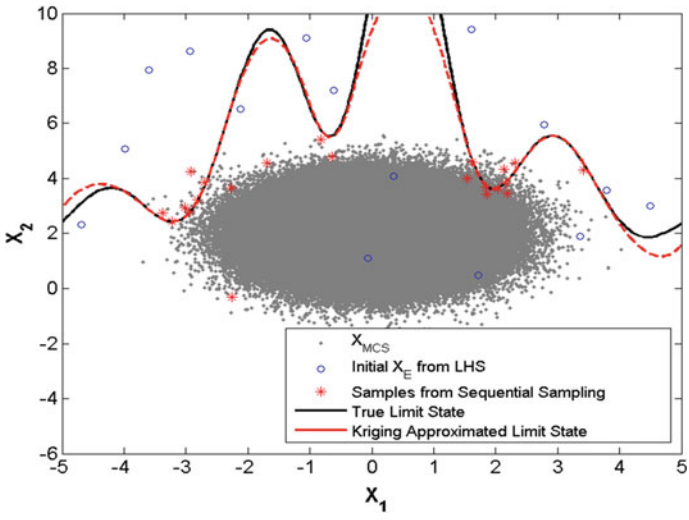


Fig. 7.14 Kriging-predicted limit state function after 27 iterations of updating

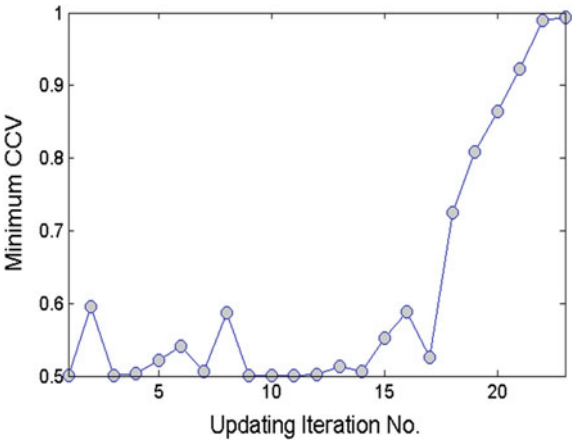


Fig. 7.15 History of the minimum CCV during the iterative Kriging model updating process

state function, as shown by the black solid line in the figure, respectively. The approximate limit state function is very accurate, especially at those critical failure surfaces from multiple disjointed active failure regions. Figure 7.15 shows the convergence history of the minimum CCV during the Kriging model updating process. With a total number of 38 function evaluations, the developed SMASS approach provides a probability of failure estimate of 2.55×10^{-4}. From the relative error comparison, the developed SMASS approach is able to provide the most accurate probability of failure estimation, compared with the FORM, DRM, and the simple Kriging method. In addition, the simple Kriging method and developed SMASS approach are significantly more accurate than FORM and the DRM. Furthermore, due to the novel adaptive sequential sampling mechanism, the

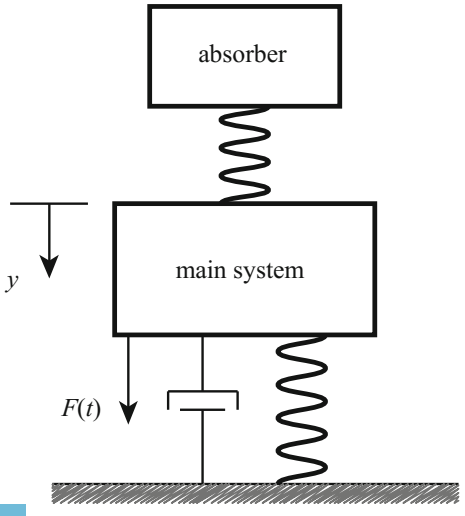


Fig. 7.16 A tuned vibration absorber

Table 7.4 Reliability analysis results by various approaches for the mathematical example

Approach	Probability of failure	Error (%)	Number of function evaluations
MCS	2.56×10^{-4}	N/A	10^8
FORM	7.4×10^{-5}	71.09	252
DRM	2.78×10^{-5}	89.24	5
Kriging	2.37×10^{-4}	7.42	150
SMASS	2.55×10^{-4}	0.39	38

developed SMASS approach is more accurate than simple Kriging, with a much smaller number of function evaluations. The quantitative results for the comparison of different probability of failure analysis methods employed in this mathematical example are summarized in Table 7.4.

7.5.4.2 A Vibration Absorber Problem

In this case study, a vibration absorber problem [33] is employed and the probability of failure analysis is carried out using the developed SMASS approach in comparison with FORM, DRM, and the simple Kriging method. A tuned damper system that includes a main system and a vibrational absorber is shown in Fig. 7.16. The main system is attached to the ground or a surface by a spring and a damper, and the absorber is attached to the main system by a spring only. The main system is subject to a harmonic force $F(t) = \cos(\omega \cdot t)$. The purpose of the absorber is to reduce or eliminate the vibrational amplitude of the main system. This type of problem often occurs when there is a need to reduce or eliminate the seismic effect on civil structures.

For the given vibration absorber system, as shown in Fig. 7.16, the normalized amplitude y of the main system can be calculated as

$$y = \frac{\left|1 - \left(\frac{1}{\beta_2}\right)^2\right|}{\sqrt{\left[1 - R\left(\frac{1}{\beta_1}\right)^2 - \left(\frac{1}{\beta_1}\right)^2 - \left(\frac{1}{\beta_2}\right)^2 + \frac{1}{\beta_1^2\beta_2^2}\right]^2 + 4\varsigma^2\left[\left(\frac{1}{\beta_1}\right) - \frac{1}{\beta_1\beta_2^2}\right]^2}} \tag{7.47}$$

where R is the ratio of the absorber's mass to the main system's mass, ζ is the damping ratio of the main system, β_1 is the ratio of the nature frequency of the main system to the harmonic force frequency, and β_2 is the ratio of the nature frequency of the absorber to the harmonic force frequency. In this case study, R and ζ are set as constants with $R = 0.01$ and $\zeta = 0.01$, whereas β_1 and β_2 are considered to be random variables that follow normal distributions, with $\beta_1 \sim N(1, 0.025^2)$ and $\beta_2 \sim N(1, 0.025^2)$, respectively.

For this case study, it is considered a system failure when the normalized amplitude y reaches beyond a critical value of 28, thereby the limit state equation

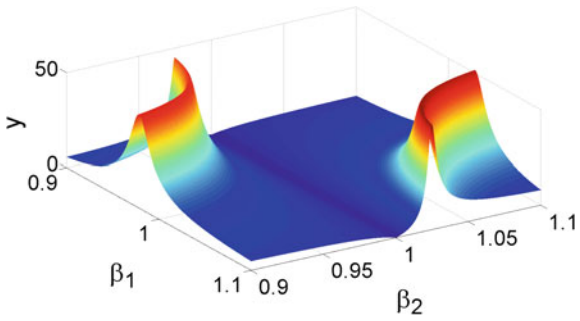


Fig. 7.17 Surface plot of the normalized amplitude y versus β_1 and β_2

can be accordingly defined as $G(\beta_1, \beta_2) = 28 - y(\beta_1, \beta_2)$. Figure 7.17 shows the function surface plot of the normalized amplitude y with respect to input variables β_1 and β_2. It is clearly shown in Fig. 7.17 that this vibration absorber problem has two distinct areas located in the random input space symmetrically, leading to two disjointed active failure regions. As the contour of the limit state function $G(\mathbf{X})$ shown in Fig. 7.18, there are two disjointed active failure regions, denoted in the figure by failure region 1 and failure region 2, respectively. The probability of failure analysis of this vibration absorber case study has been conducted with the direct MCS with a total number of 10^8 sample points. A probability of failure of 1.024×10^{-2} has been obtained, which is regarded as the accurate one, while comparing other probability of failure analysis methods.

Similar to the first case study, the FORM is also implemented for the case study. Due to the symmetric limit state functions on two disjointed active failure regions, the convergence of the FORM algorithm becomes starting point dependent, and particularly it encounters difficulty to converge when the initial starting point of the algorithm is set to the mean value point as [1, 1] in this case study. To avoid this divergence issue, a starting point close to failure region 1 has been intentionally provided. As shown in Fig. 7.18, after a total of 33 iterations of MPP searches, the FORM is converged to the MPP point [1.041170, 1.045821], providing a reliability index of 2.46392. Thus, the probability of failure by the FORM can be obtained as 6.871×10^{-3}. There are a total of 99 function evaluations used by FORM. From Fig. 7.18, it is clear that the FORM method can only estimate the probability of failure based upon one failure region while ignoring all others. Thus, it produces substantial errors in probability estimation, as compared with the MCS result.

The dimension reduction method (DRM), and more specifically, univariate dimension reduction (UDR), is employed for this case study. As shown in Fig. 7.19, the approximated PDF of $G(\mathbf{X})$ using the DRM is compared with the one obtained through MCS with a large number of simulation samples. With two random input variables, the DRM method only requires 5 function evaluations. However, due to the lack of the capability of accurately estimating the tail region of the probability density function of $G(\mathbf{X})$, as shown in the figure, the failure probability obtained by DRM, which is zero in this case study, shows a large error compared with the MCS result.

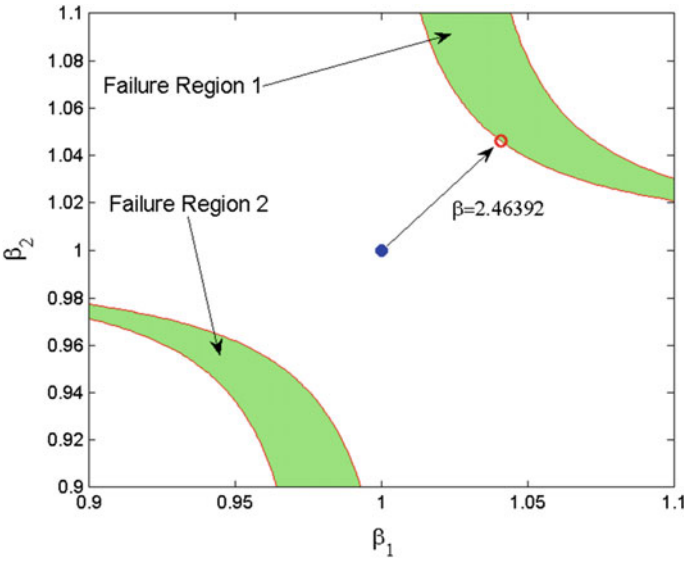


Fig. 7.18 Disjointed active failure regions by the limit state function and the MPP

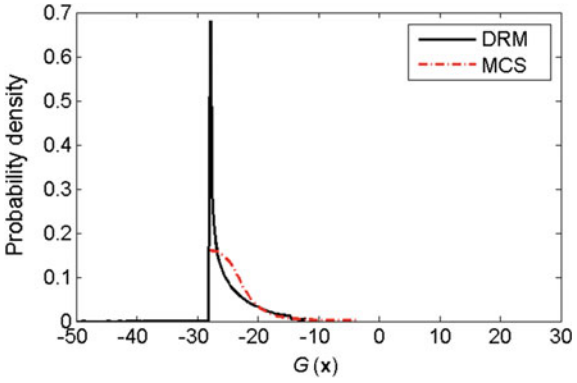


Fig. 7.19 Approximated PDFs of $G(\mathbf{X})$ by the MCS and DRM

Similar to the first case study, the Kriging surrogate modeling approach without adaptive sequential sampling, referred to as simple Kriging, has also been employed for the case study. In order to construct a relatively accurate Kriging surrogate model, a total number of 200 training sample points have been used. Figure 7.20 shows an approximated limit state function using a Kriging model constructed with 200 training sample points, as compared with the true limit state function. In order to avoid the randomness effect of the training sample points generated by the Latin Hypercube Sampling (LHS), the Kriging surrogate model has been repeatedly developed 100 times, and the estimated probability of failure on average is 1.163×10^{-2} based upon these 100 Kriging surrogate models.

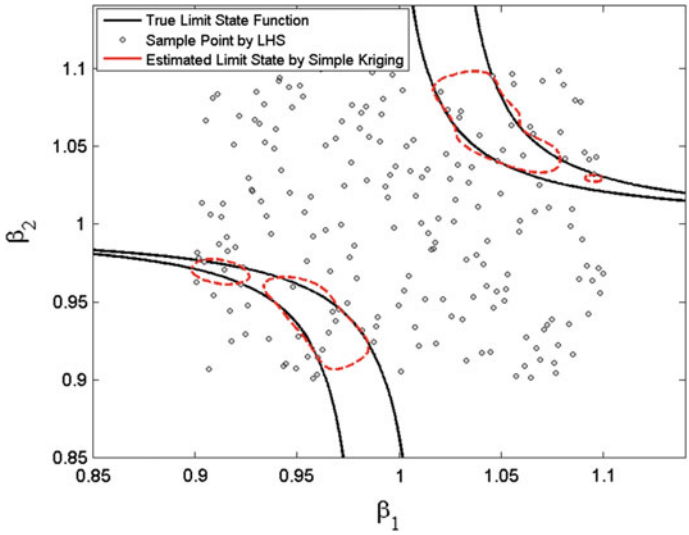


Fig. 7.20 Predicted limit state function by simple Kriging constructed with 200 sample points

The developed SMASS approach is applied to the case study. In this case study, it is implemented first with 15 initial LHS sample points, as denoted by the blue circle points in Fig. 7.14. Meanwhile, a total of 10^5 Monte Carlo sample points, $\mathbf{X}_{\mathrm{MCS}}$, are also generated in order to identify best sample points for the sequential updating of the Kriging model. After 53 iterations of updating of the Kriging model, the minimum CCV value for all the $\mathbf{X}_{\mathrm{MCS}}$ sample points satisfies the target classification confidence value, CCV_t, which has been set as 0.95 in this case study. The 53 sequentially sampled points for the updating of the Kriging model are shown in Fig. 7.21 with stars. From the identified sequential sampling points, clearly the developed SMASS approach is able to locate most of the sample points to the disjointed active failure regions, in order to enhance the fidelity of the Kriging model. With a total of 103 evaluations of the limit state function, $G(\mathbf{X})$, the approximated limit state function generated by the developed Kriging model, as shown by the red dash line in the figure, has a very good match with the true limit state function, as shown by the black solid line in the figure, respectively. The approximate limit state function is very accurate, especially at those critical failure surfaces from multiple disjointed active failure regions. Figure 7.22 shows the convergence history of the minimum CCV during the Kriging model updating process. With a total number of 103 function evaluations, the developed SMASS approach provides a probability of failure estimate of 1.033×10^{-2}. From the relative error comparison, the developed SMASS approach is able to provide the most accurate probability of failure estimation, compared with the FORM, DRM, and the simple Kriging method. Due to the novel adaptive sequential sampling

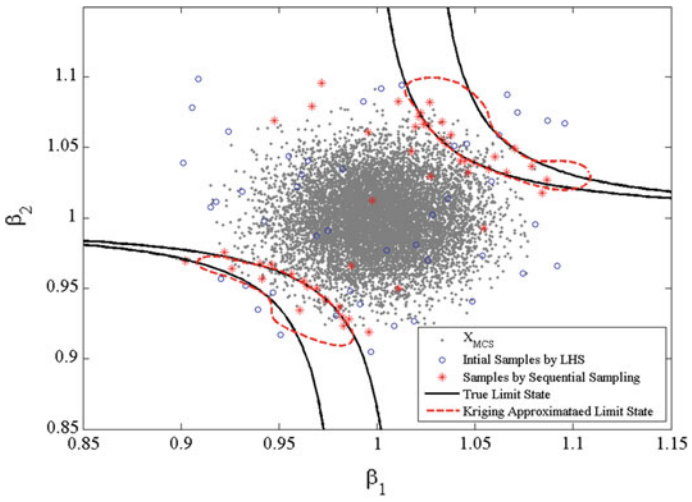


Fig. 7.21 Kriging-predicted limit state function after 53 iterations of updating

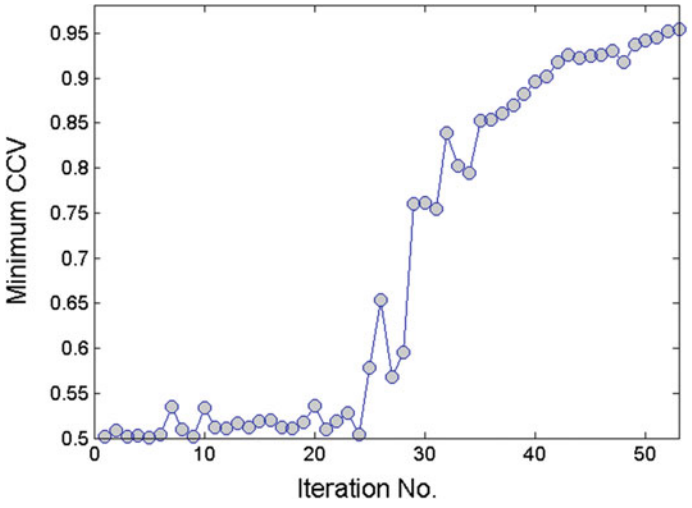


Fig. 7.22 History of the minimum CCV during the iterative Kriging model updating process

mechanism, the developed SMASS approach is more accurate than the simple Kriging with a smaller number of function evaluations. The quantitative results for the comparison of different probability of failure analysis methods employed in this mathematical example are summarized in Table 7.5.

Table 7.5 Reliability analysis results by various approaches for the vibration absorber example

Approach	Probability of failure	Error (%)	Number of function evaluations
MCS	1.024×10^{-2}	N/A	1×10^{8}
FORM	6.87×10^{-3}	32.92	99
DRM	0	100.00	5
Simple Kriging	1.163×10^{-2}	13.57	200
SMASS	1.033×10^{-2}	0.88	103

7.6 Exercises

7.1 Consider a mathematical problem for design optimization. The problem has two design variables $\mathbf{X} = [X_1, X_2]^{\mathrm{T}}$, and involves three performance functions that are defined as follows:

$$\begin{aligned} G_1(\mathbf{X}) &= X_1^2 X_2/20 - 1 \\ G_2(\mathbf{X}) &= (X_1 + X_2 - 5)^2/30 + (X_1 - X_2 - 12)^2/120 - 1 \\ G_3(\mathbf{X}) &= 80/\left(X_1^2 + 8X_2 + 5\right) - 1 \end{aligned} \tag{7.48}$$

The two variables follow independent normal distributions with means $\mathbf{d} = [d_1, d_2]^{\mathrm{T}}$ and standard deviations 0.3 and 0.3. The deterministic design optimization problem is formulated as follows

$$\begin{aligned} &\text{Minimize} && f(\mathbf{d}) = d_1 + d_2 \\ &\text{Subject to} && G_j(\mathbf{d}) \le 0, \quad j = 1, 2, 3 \\ & && 1 \le d_1 \le 10, \quad 1 \le d_2 \le 10 \end{aligned} \tag{7.49}$$

The RBDO problem with a target reliability of 99.87% (i.e., a target reliability index of $\beta^t = 3.0$) is formulated as follows

$$\begin{aligned} &\text{Minimize} && f(\mathbf{d}) = d_1 + d_2 \\ &\text{Subject to} && \Pr\left(G_j(\mathbf{X}; \mathbf{d}) \le 0\right) \ge \Phi(\beta^t), \quad j = 1, 2, 3 \\ & && 0 \le d_1 \le 10, \quad 0 \le d_2 \le 10 \end{aligned} \tag{7.50}$$

The initial design is $\mathbf{d}^{(0)} = [5.0, 5.0]^{\mathrm{T}}$.

(1) Solve the deterministic design optimization problem in Eq. (7.49) using the 'fmincon' function in MATLAB. Discuss and explain your conclusions.

(2) Solve the RBDO problem in Eq. (7.50) using PMA with the AMV method by modifying the 99-line RBDO code in Appendix. Start the design optimization from both the initial design $\mathbf{d}^{(0)} = [5.0, 5.0]^{\mathrm{T}}$ and deterministic optimum design obtained from (1). Discuss and explain your conclusions.

7.2 Solve the RBDO problem in Problem 7.1 using RIA with the HL-RF method (i.e., nm = 2 in the 99-line RBDO code). In this exercise, assume that X_2 is a non-normally distributed random variable and follows a lognormal distribution with the same mean and standard deviation as defined in the code. (Hint: Refer to the transformation between the X-space and the U-space for a lognormal distribution specified in Table 5.1, and consider adding the following lines of code for the 1st constraint).

①:
```
DXDU(1) = stdx(1);
sigmaL = sqrt(log(1+(stdx(2)/x(2))^2));
muL = log(x(2))-0.5*sigmaL^2;
DXDU(2) = exp(muL + sigmaL*u(2))*sigmaL;

dbeta = u./(beta*DXDU);
```

②:
```
x(1) = u(1).*stdx(1)+d(1);
sigmaL = sqrt(log(1+(stdx(2)/d(2))^2));
muL = log(d(2))-0.5*sigmaL^2;
x(2) = exp(muL + sigmaL*u(2));
```

③:
```
DXDU = x(2)*sigmaL;
GCeq(2) = -x(1)^2/20*DXDU;
```

7.3 Consider a vehicle side-impact problem for design optimization. The optimization task is to minimize the vehicle weight while meeting the side impact top safety-rating criteria shown in Table 7.6 [34]. There are nine design parameters used in the design optimization of vehicle side impact. The design variables are the thickness (X_1–X_7) and material properties (X_8, X_9) of critical parts, as shown in Table 7.7. The two (non-design) random parameters are barrier height and hitting position (X_{10}, X_{11}), which can vary from −30 to 30 mm according to the physical test.

Table 7.6 Regulations and requirements for vehicle side impact

Constraints		Safety criteria
G_1: Abdomen load (kN)		≤ 1
G_2–G_4: Rib deflection (mm)	Upper	≤ 32
	Middle	
	Lower	
G_5–G_7: Viscous criteria (m/s)	Upper	≤ 0.32
	Middle	
	Lower	
G_8: Pubic symphysis force (kN)		≤ 4
G_9: Velocity of B-pillar		≤ 9.9
G_{10}: Velocity of front door at B-pillar		≤ 15.7

Table 7.7 Statistical properties of random design and parameter variables (X_{10} and X_{11} both have 0 means) for Problem 7.1

Random variable	Distribution type	Std. dev.	d^L	d	d^U
X_1 (mm)	Normal	0.050	0.500	1.000	1.500
X_2 (mm)	Normal	0.050	0.500	1.000	1.500
X_3 (mm)	Normal	0.050	0.500	1.000	1.500
X_4 (mm)	Normal	0.050	0.500	1.000	1.500
X_5 (mm)	Normal	0.050	0.500	1.000	1.500
X_6 (mm)	Normal	0.050	0.500	1.000	1.500
X_7 (mm)	Normal	0.050	0.500	1.000	1.500
X_8 (GPa)	Lognormal	0.006	0.192	0.300	0.345
X_9 (GPa)	Lognormal	0.006	0.192	0.300	0.345
X_{10} (mm)	Normal	10.0	X_{10} and X_{11} are not design variables		
X_{11} (mm)	Normal	10.0			

The response surfaces for ten performance measures are constructed from a vehicle side-impact model as $\{G_j \leq g_j, j = 1, 2, \ldots, 10\}$, where the performance limits g_j form a vector $\mathbf{g} = [1, 32, 32, 32, 0.32, 0.32, 0.32, 4, 9.9, 15.7]^{\mathrm{T}}$. The response surfaces of the vehicle weight f and the performance measures G_i are defined as follows:

$$
\begin{aligned}
f(weight) &= 1.98 + 4.90d_1 + 6.67d_2 + 6.98d_3 + 4.01d_4 + 1.78d_5 + 2.73d_7;\\
G_1 &= 1.16 - 0.3717X_2X_4 - 0.00931X_2X_{10} - 0.484X_3X_9 + 0.01343X_6X_{10};\\
G_2 &= 28.98 + 3.818X_3 - 4.2X_1X_2 + 0.0207X_5X_{10} + 6.63X_6X_9 - 7.7X_7X_8 + 0.32X_9X_{10};\\
G_3 &= 33.86 + 2.95X_3 + 0.1792X_{10} - 5.057X_1X_2 - 11X_2X_8 - 0.0215X_5X_{10} - 9.98X_7X_8\\
&\quad + 22X_8X_9;\\
G_4 &= 46.36 - 9.9X_2 - 12.9X_1X_8 + 0.1107X_3X_{10};\\
G_5 &= 0.261 - 0.0159X_1X_2 - 0.188X_1X_8 - 0.019X_2X_7 + 0.0144X_3X_5 + 0.0008757X_5X_{10}\\
&\quad + 0.08045X_6X_9 + 0.00139X_8X_{11} + 0.00001575X_{10}X_{11};\\
G_6 &= 0.214 + 0.00817X_5 - 0.131X_1X_8 - 0.0704X_1X_9 + 0.03099X_2X_6 - 0.018X_2X_7\\
&\quad + 0.0208X_3X_8 + 0.121X_3X_9 - 0.00364X_5X_6 + 0.0007715X_5X_{10} - 0.0005354X_6X_{10}\\
&\quad + 0.00121X_8X_{11} + 0.00184X_9X_{10} - 0.018X_2^2;\\
G_7 &= 0.74 - 0.61X_2 - 0.163X_3X_8 + 0.001232X_3X_{10} - 0.166X_7X_9 + 0.227X_2^2;\\
G_8 &= 4.72 - 0.5X_4 - 0.19X_2X_3 - 0.0122X_4X_{10} + 0.009325X_6X_{10} + 0.000191X_{11}^2;\\
G_9 &= 10.58 - 0.674X_1X_2 - 1.95X_2X_8 + 0.02054X_3X_{10} - 0.0198X_4X_{10} + 0.028X_6X_{10};\\
G_{10} &= 16.45 - 0.489X_3X_7 - 0.843X_5X_6 + 0.0432X_9X_{10} - 0.0556X_9X_{11} - 0.000786X_{11}^2.
\end{aligned}
$$

The deterministic design optimization problem for vehicle crashworthiness is formulated as follows

$$\begin{array}{ll} \text{Minimize} & f(\mathbf{d}) \\ \text{Subject to} & G_j(\mathbf{d}) \le g_j, \quad j = 1, \ldots, 10 \\ & d_i^L \le d_i \le d_i^U, \quad i = 1, \ldots, 9 \end{array} \tag{7.51}$$

The RBDO problem for vehicle crashworthiness with a target reliability of 99.87% is formulated as follows

$$\begin{array}{ll} \text{Minimize} & f(\mathbf{d}) \\ \text{Subject to} & \Pr\left(G_j(\mathbf{X}; \mathbf{d}) \le g_j\right) \ge 99.87\%, \quad j = 1, \ldots, 10 \\ & d_i^L \le d_i \le d_i^U, \quad i = 1, \ldots, 9 \end{array} \tag{7.52}$$

(1) Solve the deterministic design optimization problem in Eq. (7.51) using the 'fmincon' function in MATLAB. Start the design optimization from the initial design (d_1–d_7 = 1.000, $d_8 = d_9 = 0.300$).
(2) Solve the RBDO problem in Eq. (7.52) using PMA with the AMV method by modifying the MATLAB code in the Appendix. Start the design optimization from both the initial design (d_1–d_7 = 1.000, $d_8 = d_9 = 0.300$) and deterministic optimum design obtained from (1).

Appendix: A 99-Line MATLAB Code for RBDO

```
%%%%%%%%%%  A 99 LINE RBDO CODE WRITTEN BY WANG P.F. & YOUN B.D.  %%%%%%%%
function RBDO()
clear all; close all; clc;
global nc nd nm bt stdx Iters Cost
nm=2; nc=3; nd=2; bt=norminv(0.99,0,1);
x0=[5,5]; stdx=[0.3,0.3]; lb=[0,0]; ub=[10,10];
xp=x0; Iters=0;
options = optimset('GradConstr','on','GradObj','on','LargeScale','off');
[x,fval]=fmincon(@Costfun,x0,[],[],[],[],lb,ub,@frelcon,options)
%====================    Obj. Function  ==============================%
function [f,g]= Costfun(x)
     f=x(1)+x(2);
     g=[1 1];
     Cost=f;
end
```

```
%=================== Define Constraints and Gradiants ================%
function [c,ceq,GC,GCeq] = frelcon(x)
     ceq=[]; GCeq=[];
     for j = 1:nc
        if nm==1
             [G,DG]=AMV(x,j);
             beta(j)=G;
             dbeta(:,j)=DG./stdx;
        elseif nm==2
             [G,DG]=HL_RF(x,j);
             beta(j)=bt-G;
             dbeta(:,j)=-DG;
        end
     end
     c=beta; GC=dbeta;
     dx=norm(x-xp);
     if  dx>1d-5  || Iters == 0
        Iters=Iters+1;
        SHOW(Iters,x,c,GC);
     end
     xp = x;
end
%================  PMA Approach with AMV Algorithm  ====================%
function [G,DG]=AMV(x,kc)
     u=zeros(1,nd); iter = 0; Dif=1;
     while Dif>1d-5 & iter<20
          iter=iter+1;
          if iter>1
             u=DG*bt/norm(DG);
          end
          [G,DG]=cons(u,x,kc);
          U(iter,:)=u/bt;
          if iter>1
             Dif=abs(U(iter,:)*U(iter-1,:)'-1);
          end
     end
end
%================= RIA Approach with HL_RF Algorithm ==================%
function [beta,dbeta]=HL_RF(x,kc)
     u=zeros(1,nd); iter=0;  Dif=1; sign = 1;
     while Dif >= 1d-5 && iter < 20
          iter=iter + 1;
          [ceq,GCeq]=cons(u,x,kc);
          u=(GCeq*u'-ceq)/norm(GCeq)^2*GCeq;
          U(iter,:)=u/norm(u);
```

```
        if iter ==1
           sign = -ceq/abs(ceq);
        elseif iter>1
           Dif=abs(U(iter-1,:)*U(iter,:)' - 1);
        end
    end
    beta = sign*norm(u);
    dbeta = -u./(beta*stdx);
end
%========================== Constraint Fun. ==========================%
function [ceq,GCeq]=cons(u,d,kc)
    x = u.*stdx+d;
    if kc == 1
      ceq=1-x(1)^2*x(2)/20;
      GCeq(1)=-x(1)*x(2)/10*stdx(1);
      GCeq(2)=-x(1)^2/20*stdx(2);
    elseif kc == 2
      ceq=1-(x(1)+x(2)-5)^2/30-(x(1)-x(2)-12)^2/120;
      GCeq(1)=(-(x(1)+x(2)-5)/15-(x(1)-x(2)-12)/60)*stdx(1);
      GCeq(2)=(-(x(1)+x(2)-5)/15+(x(1)-x(2)-12)/60)*stdx(2);
    elseif kc == 3
      ceq=1-80/(x(1)^2+8*x(2)+5);
      GCeq(1)=x(1)*160*stdx(1)/((x(1)^2+8*x(2)+5))^2;
      GCeq(2)=80*8*stdx(2)/((x(1)^2+8*x(2)+5))^2;
    end
end
function SHOW(Iters,x,c,GC)%==== Display the Iteration Information====%
    fprintf(1,'\n********** Iter.%d ***********\n',Iters);
    disp(['Des.:' sprintf('%6.4f ',x)]);
    disp(['Obj.:' sprintf('%6.4f',Cost)]);
    if nm==1
       disp(['Cons.:' sprintf('%6.4f ',c)]);
    elseif nm==2
       disp(['Index.:' sprintf('%6.4f',bt-c)]);
    end
    disp(['Sens.:' sprintf('%6.4f ',GC)]);
    fprintf('\n\n')
end
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%% Notes to the Code %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                               %
% This is a compact code for reliability based design optimization (RBDO)      %
% which includes PMA with AMV algorithm and RIA with HL_RF algorithm.           %
% One mathematical example is solved for demonstration purpose.                 %
%                                                                               %
% VARIABLE DEFINITION:                                                          %
% nc: number of constraints; nd: number of design variables;                    %
% bt: target rel. index; nm: method choice, nm=1 means PMA with AMV; nm=2       %
% means RIA with HL_RF;                                                         %
% x0: initial design; stdx: standard deviation for design variable;             %
% lb & ub: lower bound and upper bound for design variables;                    %
% iter: variable used to record iteration number of optimization process       %
%                                                                               %
% FUNCTION DEFINITION                                                           %
% RBDO(): main function                                                         %
% Costfun(): objective function & gradients w.r.t. design variables             %
% frelcon(): Define the probability constraints and gradients                   %
% AMV() & HL_RF(): AMV algorithm for PMA & HL_RF algorithm for RIA              %
% cons(): deterministic constraints and gradients evaluation                    %
% SHOW(): show the optimization information for every design iteration          %
%                                                                               %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

References

1. Tu, J., Choi, K. K., & Park, Y. H. (1999). A new study on reliability-based design optimization. *Journal of Mechanical Design, Transactions of the ASME, 121*(4), 557–564.
2. Youn, B. D., Choi, K. K., & Du, L. (2004). Enriched Performance Measure Approach (PMA+) and its numerical method for reliability-based design optimization. *AIAA Journal, 43,* 874–884.
3. Youn, B. D., Choi, K. K., & Park, Y. H. (2003). Hybrid analysis method for reliability-based design optimization. *Journal of Mechanical Design, 125*(2), 221–232.
4. Chiralaksanakul, A., & Mahadevan, S. (2005). First-order approximation methods in reliability-based design optimization. *Journal of Mechanical Design, 127*(5), 851–857.
5. Noh, Y., Choi, K., & Du, L. (2008). Reliability-based design optimization of problems with correlated input variables using a Gaussian Copula. *Structural and Multidisciplinary Optimization, 38*(1), 1–16.
6. Du, X. P., & Chen, W. (2004). Sequential optimization and reliability assessment method for efficient probabilistic design. *Journal of Mechanical Design, 126*(2), 225–233.
7. Liang, J. H., Mourelatos, Z. P., & Nikolaidis, E. (2007). A single-loop approach for system reliability-based design optimization. *Journal of Mechanical Design, 129*(12), 1215–1224.
8. Nguyen, T. H., Song, J., & Paulino, G. H. (2010). Single-loop system reliability-based design optimization using matrix-based system reliability method: Theory and applications. *ASME Journal of Mechanical Design, 132*, 011005-1–11.
9. Thanedar, P. B., & Kodiyalam, S. (1992). Structural optimization using probabilistic constraints. *Journal of Structural Optimization, 4,* 236–240.
10. Chen, X., Hasselman, T. K., & Neill, D. J. (1997, April). Reliability-based structural design optimization for practical applications. *AIAA Paper*, 97-1403.

11. Wang, L., & Kodiyalam, S. (2002). An efficient method for probabilistic and robust design with non-normal distributions. In *43rd AIAA Structures, Structural Dynamics, and Materials Conference*, April 2002.
12. Rahman, S. (2009). Stochastic sensitivity analysis by dimensional decomposition and score functions. *Probabilistic Engineering Mechanics*, *24*(3), 278–287.
13. Yu, X., Chang, K. H., & Choi, K. K. (1998). Probabilistic structural durability prediction. *AIAA Journal, 36*(4), 628–637.
14. Putko, M. M., Newman, P. A., Taylor, A. C., III, & Green, L. L. (2002). Approach for uncertainty propagation and robust design in CFD using sensitivity derivatives. *Journal of Fluids Engineering, 124*(1), 60–69.
15. Koch, P. N., Yang, R.-J., & Gu, L. (2004). Design for six sigma through robust optimization. *Structural and Multidisciplinary Optimization, 26*(3–4), 235–248.
16. Wu, Y. T., Millwater, H. R., & Cruse, T. A. (1990). Advanced probabilistic structural analysis method for implicit performance functions. *AIAA Journal, 28*(9), 1663–1669.
17. Youn, B. D., & Choi, K. K. (2004). An investigation of nonlinearity of reliability-based design optimization approaches. *Journal of Mechanical Design, 126*(3), 403–411.
18. Youn, B. D., & Choi, K. K. (2004). Selecting probabilistic approaches for reliability-based design optimization. *AIAA Journal, 42*(1), 124–131.
19. Youn, B. D., Xi, Z., & Wang, P. (2008). Eigenvector Dimension-Reduction (EDR) method for sensitivity-free uncertainty quantification. *Structural and Multidisciplinary Optimization, 37*(1), 13–28.
20. Shan, S., & Wang, G. G. (2008). Reliable design space and complete single-loop reliability-based design optimization. *Reliability Engineering & System Safety, 93*(8), 1218–1230.
21. Zou, T., & Mahadevan, S. (2006). A direct decoupling approach for efficient reliability-based design optimization. *Structural and Multidisciplinary Optimization, 31*(3), 190–200.
22. Nguyen, T. H., Song, J., & Paulino, G. H. (2011). Single-loop system reliability-based topology optimization considering statistical dependence between limit-states. *Structural and Multidisciplinary Optimization, 44*(5), 593–611.
23. Rubinstein, R. Y., & Kroese, D. P. (2011). *Simulation and the Monte Carlo method* (Vol. 707). New York: Wiley.
24. Stein, M. (1987). Large sample properties of simulations using latin hypercube sampling. *Technometrics, 29*(2), 143–151.
25. Goel, T., Haftka, R. T., & Shyy, W. (2008). Error measures for noise-free surrogate approximations. *AIAA Paper*, 2008-901.
26. Dey, A., & Mahadevan, S. (1998). Ductile structural system reliability analysis using importance sampling. *Structure Safety, 20*(2), 137–154.
27. Martino, L., Elvira, V., Luengo, D., & Corander, J. (2015). An adaptive population importance sampler: Learning from uncertainty. *IEEE Transactions on Signal Processing, 63* (16), 4422–4437.
28. Beachkofski, B., & Grandhi, R. (2002). Improved distributed hypercube sampling. In *43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. April, 2002.
29. Leary, S., Bhaskar, A., & Keane, A. (2003). Optimal Orthogonal-array-based latin hypercubes. *Journal of Applied Statistics, 30*(5), 585–598.
30. Joseph, V. R., & Hung, Y. (2008). Orthogonal-maximin latin hypercube designs. *Statistica Sinica, 18,* 171–186.
31. Deutsch, J. L., & Deutsch, C. V. (2012). Latin hypercube sampling with multidimensional uniformity. *Journal of Statistical Planning and Inference, 142,* 763–772.
32. Wang, Z., & Wang, P. (2014). A maximum confidence enhancement based sequential sampling scheme for simulation-based design. *Journal of Mechanical Design, 136*(2), 021006.

33. Kuczera, R. C., & Mourelatos, Z. P. (2009). On estimating the reliability of multiple failure region problems using approximate metamodels. *Journal of Mechanical Design, 131*(12), 121003–121013.
34. Youn, B. D., Choi, K. K., Gu, L., & Yang, R.-J. (2004). Reliability-based design optimization for crashworthiness of side impact. *Journal of Structural and Multidisciplinary Optimization, 26*(3–4), 272–283.

Chapter 8
Time-Dependent Reliability Analysis in Operation: Prognostics and Health Management

Over the past few decades, rapid adoption of sensing, computing, and communications technologies has created one of the key capabilities of modern engineered systems: the ability—at a low cost—to gather, store, and process large volumes of sensor data from an engineered system during operation. These sensor data may contain rich information about a system's behavior under both healthy and degraded conditions. A critical question now is how to leverage the new sensor information, which may be continuously or periodically collected, to assess the current health condition (health reasoning) and predict imminent failures (health prognostics) of the operating system over its life cycle. This health information can provide a timely warning about potential failures and potentially open a window of opportunity for implementing measures to avert these failures. This chapter presents techniques and approaches that enable (i) design of sensor networks (SNs) for health reasoning, (ii) extraction of health-relevant information from sensor signals and assessment of a system's health condition, and (iii) prediction of a system's remaining useful life (RUL).

8.1 Overview of Prognostics and Health Management

Accelerated life testing (ALT) is capable of providing a time-dependent reliability estimate for a population of system units based on the degradation characteristics of historical units. We refer to this approach as classical statistics-based, time-dependent reliability analysis. This type of analysis incorporates population characteristics into time-dependent reliability estimation by modeling a life distribution. This classical, statistics-based approach provides an overall (or population-wise) reliability estimate that determines the same life distribution for an entire population of units. In engineering practice, we may be more interested in investigating the specific reliability information of a particular unit under its actual operating conditions to predict the advent of an imminent failure and mitigate

C. Hu et al., *Engineering Design under Uncertainty and Health Prognostics*,
Springer Series in Reliability Engineering,
https://doi.org/10.1007/978-3-319-92574-5_8

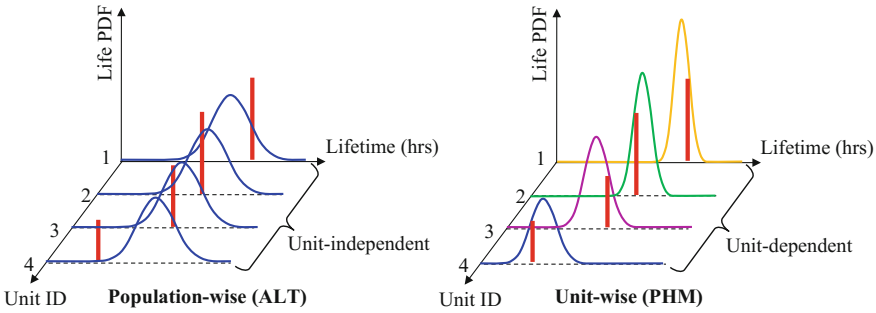


Fig. 8.1 Population- and unit-wise time-dependent reliability analyses

potential risk. In other words, we may need to perform unit-wise reliability analysis that estimates the time-dependent reliability of a particular unit. Figure 8.1 provides a graphical illustration of the difference between population- and unit-wise reliability analyses.

Recent decades have seen a growing interest in moving from traditional nondestructive testing (NDT) to nondestructive evaluation (NDE) and structural health monitoring (SHM), and towards automated data analytics for prognostics and health management (PHM) [1], as shown in Fig. 8.2. Among these major enabling technologies for unit-wise, time-dependent reliability analysis, PHM has recently emerged as a key technology that uses data analytics to assess the current health condition of an engineered system (health reasoning) and predict when and how the system is likely to fail (health prognostics) throughout the system's lifetime. The need for PHM is also being driven by an increased demand for condition-based maintenance and life extension of high-value engineered systems like bridges and energy infrastructure (e.g., nuclear power plants, wind turbines, and pipelines).

In general, PHM consists of four basic functions: health sensing, health reasoning, health prognostics, and health management (see Fig. 8.3). A brief description of each function is given here:

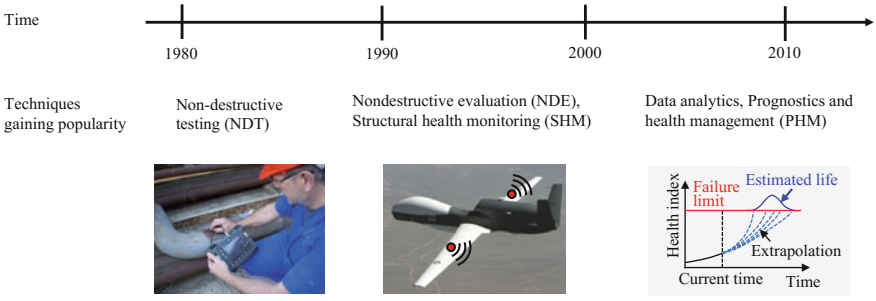


Fig. 8.2 Evolution of key enabling technologies for unit-wise, time-dependent reliability analysis

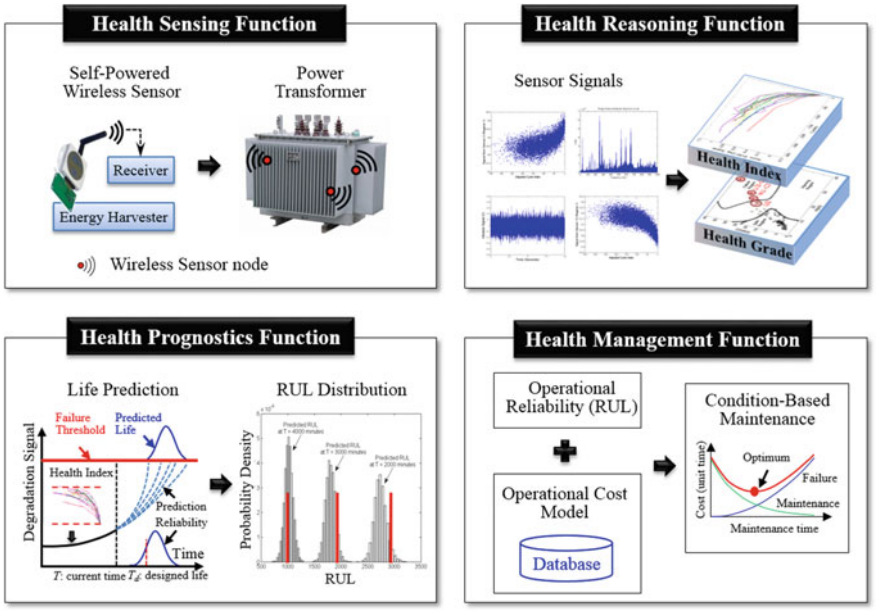


Fig. 8.3 The four basic PHM functions

- Health Sensing: To acquire sensor signals from an engineered system through in-situ monitoring techniques and to ensure a high likelihood of damage detection by designing an optimal SN.
- Health Reasoning: To extract health-relevant system information from the acquired sensor signals using feature extraction techniques and to classify system health states through the use of health classification techniques.
- Health Prognostics: To define a threshold for each anomaly state and predict the RUL, i.e., the amount of time remaining before the system can no longer perform its required function(s).
- Health Management: To enable optimal decision making on maintenance of the system, based on the RUL prediction, to minimize the life-cycle cost (LCC).

In today's engineered systems, most system maintenance is either corrective (i.e., repairing or replacing a system after it fails) or preventive (i.e., inspecting a system on a routine schedule regardless of whether inspection is actually needed). The former strategy is called corrective maintenance (CM) and the latter is called preventive maintenance (PM). Both approaches are expensive and incur high LCCs. PHM, meanwhile, uses sensor signals from a system in operation to assess the system's current health and predict when and how the system is likely to fail. The health and life information provided by PHM enables field engineers to take a maintenance action only when needed. This is referred to as a condition-based maintenance (CBM) strategy. CBM often results in a lower LCC than strategies that use CM and/or PM (see Fig. 8.4). Consider nuclear power plants as an example.

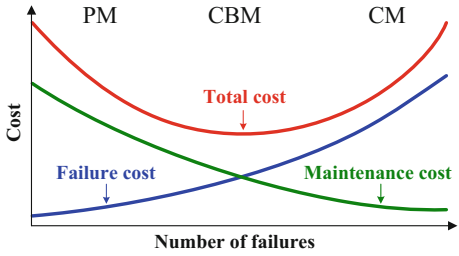


Fig. 8.4 Costs associated with different maintenance strategies

A Nuclear Energy Research Initiative study suggested that, based on a 6% discount rate and a life of 40 years, the deployment of PHM in all key equipment in the 104 U.S. legacy power plants could potentially save over 1 billion U.S. dollars per year [2]. When the economic analysis is extended from the nuclear power industry to consider other sectors in U.S. manufacturing industries, PHM could result in potential savings in the multi-trillions of dollars.

Another example that illustrates the benefits of PHM is a recent report published by General Electric (GE) [3]. As mentioned above, PHM enables the marriage of data analytics and engineered systems for improving the reliability, safety, and operational efficiency of these systems. The GE report takes a conservative look at the monetary benefits of this marriage in several industries (see a summary of these benefits in Table 8.1). An important takeaway from the table is that, even if data analytics achieves just a one percent efficiency improvement, the results will still be substantial. For example, in the commercial aviation industry alone, a one percent improvement in fuel savings would yield an annual savings of $2–3 billion. Likewise, a one percent uptime improvement in the gas-fired power plant fleet could yield a $5–7 billion annual savings. Healthcare and rail transportation would also benefit by multiple billions of dollars, through improvements in process efficiency. As PHM employs data analytics to determine the health and remaining life of a system and enable optimal decision making on maintenance, the resulting improvement in reliability, safety, and uptime will have a similarly big impact on profitability.

It is also important to note that PHM technology can scale well for "Internet of Things" applications. Specifically, data analytics can be embedded in a cloud computing environment to allow real-time analysis of large volumes of sensor data collected from individual system units in the field, and assess and predict performance degradation of these units. In recent years, PHM has been successfully

Table 8.1 Illustrative examples based on a potential one percent savings applied across specific global industry sectors [3]

Industry	Power of 1%	Annual impact
Airlines	1% fuel saving	$2–3B
Oil and gas	1% uptime	$5–7B
Healthcare	1% productivity	$4–5B
Transportation	1% productivity	$1.5–2B

applied to assess the health conditions of a number of engineered systems and to predict their RULs for failure prevention and reliability improvement. These engineered systems can capitalize on PHM to enable early anticipation of failures, to develop cost-effective maintenance strategies, and to seek opportunities for life extension. Example of these systems include wind turbines, nuclear power plants, aircraft engines, power transformers, lithium-ion (Li-ion) batteries, oil pipelines, and micro-machining tools.

8.2 The Health Sensing Function

As mentioned in Sect. 8.1, the health sensing function of PHM aims at acquiring sensor signals from an engineered system through in-situ monitoring techniques and ensuring a high likelihood of damage detection by designing an optimal SN. The effectiveness of PHM in failure prevention and reliability improvement relies greatly on the usefulness and completeness of health-relevant information conveyed by the sensor signals. These measurable physical quantities can be classified into two major categories: environmental signals (e.g., temperature, pressure, and humidity) and operating signals (e.g., voltage, current, vibration, and power). In order to identify an approximate set of sensing quantities, we can first conduct failure modes and effects analysis (FMEA) to determine critical failure modes and their effects, and then identify measureable quantities that may be affected by these modes and/or effects. Potential failure modes and the corresponding sensing quantities of several engineered systems are shown in Fig. 8.5.

Identifying appropriate sensing quantities (or selecting approximate sensor types) is one important aspect of the health sensing function. In a broader sense, there may be interest in designing an optimal SN with high detectability, while accounting for various sources of uncertainty (e.g., material properties, geometric tolerances and loading conditions). Sections 8.2.1–8.2.3 present a

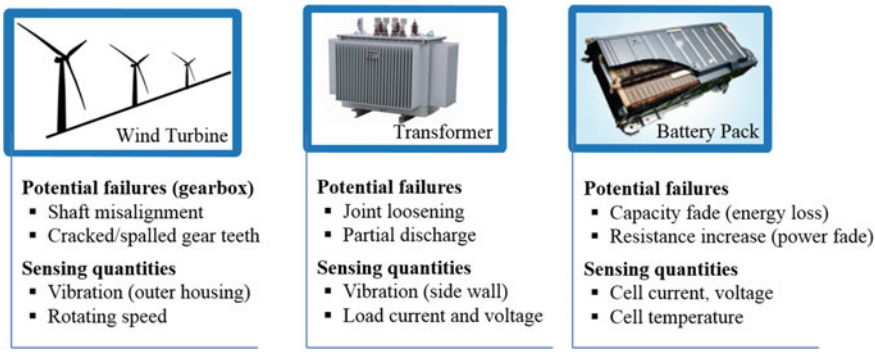


Fig. 8.5 Potential failures and sensing quantities of selected engineered systems

detectability-based SN design framework for system health monitoring and prognostics. Section 8.2.4 discusses self-powered wireless sensors that are increasingly being used in the health sensing function. Other issues related to design of the health sensing function will be discussed in Sect. 8.2.5.

8.2.1 Detectability Analysis

The detectability of a given SN design can be derived based on a probability-of-detection (PoD) matrix [4, 5]. A general form of the PoD matrix for an engineered system with a number of health states (i.e., HS_i, $i = 1, 2, \ldots, N_{HS}$) is shown in Table 8.2, where one element P_{ij} is defined as the conditional probability that the system is detected to be operating at HS_j by the SN, given that the system is actually operating at HS_i. Clearly, P_{ij} represents the probabilistic relationship between the true system health state and the health state detected by the SN. Mathematically, P_{ij} can be expressed as

$$P_{ij} = \Pr\left(\text{Detected as } HS_j | \text{System is at } HS_i\right) \tag{8.1}$$

By definition, the ith diagonal term in the PoD matrix represents a conditional probability of correct detection for the ith health state; this can be defined as the detectability of the ith system health state HS_i as

$$D_i = P_{ii} = \Pr(\text{Detected as } HS_i | \text{System is at } HS_i) \tag{8.2}$$

The definition above provides a probabilistic measure for the diagnostic/prognostic performance of an SN while considering uncertainty in manufacturing and system operation processes. The diagonal terms in the PoD matrix, which represent the probabilities of correct detection for predefined health states, will determine the overall detection performance of the SN. Using the predefined detectability requirements, these diagonal terms in the PoD matrix will then constitute N_{HS} number of detectability constraints in the SN design process. Since these detectability constraints involve the computation of multiple conditional probabilities, an efficient and accurate methodology for detectability analysis must be developed.

Table 8.2 Probability of detection (PoD) matrix

Probability		Detected health state			
		1	2	…	N_{HS}
True health state	1	P_{11}	P_{12}	…	$P_{1N_{HS}}$
	2	P_{21}	P_{22}	…	$P_{2N_{HS}}$
	…	…	…	…	…
	N_{HS}	$P_{N_{HS}1}$	$P_{N_{HS}2}$	…	$P_{N_{HS}N_{HS}}$

8.2.2 SN Design Optimization

Appropriate selection of sensing devices—such as fiber optic sensors, piezoelectric sensors, MEMS sensors, accelerometers, or acoustic sensors—is determined by each sensor's characteristic attributes, such as full-scale dynamic range, sensitivity, noise floor, and analog-to-digital converter resolution. Thus, the design variables involved in the proposed SN design framework are the decision variables for (i) the selection of sensing devices, (ii) the number of sensing devices, (iii) the location of sensing devices, and (iv) the parameters for controlling the sensing process, such as sampling frequency, sampling period, and power configuration. The design constraints are detectability requirements that consider the uncertainty present in the manufacturing and system operation processes. Considering all factors outlined above, the SN design optimization problem can be formulated as [4]:

$$\begin{array}{ll} \text{Minimize} & C \\ \text{subject to} & D_i(\mathbf{X_T}, \mathbf{X_N}, \mathbf{X_{Loc}}, \mathbf{X_s}) \geq D_i^t \\ & (i = 1, 2, \ldots, N_{HS}) \end{array} \tag{8.3}$$

where C is the cost involved (calculated as the product of the number of sensors and the sum of the sensor material and installation costs), $\mathrm{X_T}$ is a vector of the binary decision variables for selection of the types of sensing devices, $\mathrm{X_N}$ is a vector consisting of the number of each selected type of sensing devices, $\mathrm{X_{Loc}}$ is a 3-D vector of the location of each sensing device, $\mathbf{X_s}$ is a vector of the sensing control parameters, N_{HS} is the total number of predefined health states for the engineered system, D_i is the detectability of the SN for the ith predefined health state, which is a function of the design variables $\mathrm{X_T}$, $\mathrm{X_N}$, $\mathrm{X_{Loc}}$, and $\mathbf{X_s}$, and D_i^t is the target SN detectability for the ith predefined health state. Note that the formulation of the SN design optimization problem bears a resemblance to that of the reliability-based design optimization problem [6, 7] with the exception that the former uses the detectability as the constraint and the latter uses the reliability as the constraint.

The SN design optimization problem in Eq. (8.3) contains discrete decision variables for the selection of sensing devices, integer variables for the number of selected sensing devices, as well as continuous variables for the sensor locations. Thus, the optimization problem is formulated as a mixed-integer nonlinear programming (MINLP) problem [8], and heuristic algorithms, such as genetic algorithms (GAs), can be used as the optimizer for the optimization process. In this textbook, the GA is employed for the example problem that will be detailed in the subsequent section. More alternative algorithms for solving the MINLP problem can be found in references [8, 9].

8.2.3 Overall Procedure for SN Design

The flowchart of the SN design optimization process is shown in Fig. 8.6 [4, 5]. As shown in the figure, the process starts from an initial SN design and goes into the design optimization subroutine (the grey box on the right-hand side), which will carry out the SN cost analysis, call the performance analysis subroutine (the grey box on the left-hand side) to evaluate the performance of the SN in its current design, and execute the optimizer to generate a new SN design if the optimality condition is not met. In the performance analysis subroutine, the detectability analysis (as discussed in the previous section) will be carried out. Before solving the optimization problem, valid system simulation models have to be built and computer simulations have to be accomplished so that the training and testing data sets for each predefined health state are available.

It is interesting to note that this design optimization procedure bears a striking resemblance to the RBDO procedure. The two key elements in the SN design, the detectability analysis and the design optimization, can be equivalently mapped to the two key elements in the RBDO, the reliability analysis and design optimization. This finding can be generalized to other optimization problems, and we can conclude that all optimization problems share the same basic structure: the optimization routine performs design change based on the cost (e.g., product volume, number of sensors) and constraint (e.g., reliability, detectability) analyses. The two key elements in SN design, the detectability analysis and the design optimization, will be discussed in detail in subsequent sections.

Example 8.1 SN Design to Detect Power Transformer Mechanical Joint Failure

Power transformers are among the most expensive elements in high-voltage power systems. Effective monitoring of power transformers can enable a transition from traditional PM to CBM, resulting in significant reductions in

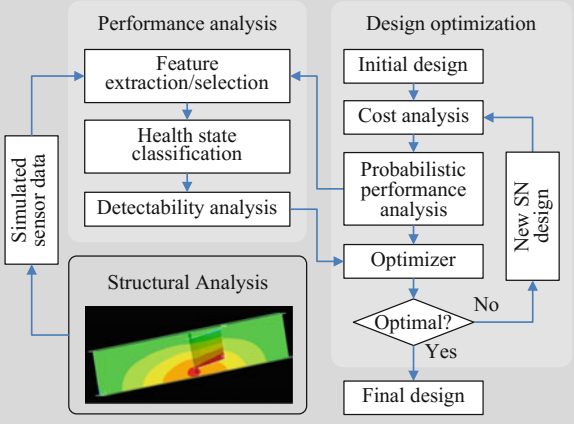


Fig. 8.6 Flowchart of detectability-based SN design [4]

LCCs. Due to the difficulties associated with direct sensing inside transformers, the data that are most often used for both diagnosis and prognosis of transformers are obtained through indirect measurements. This example aims to design an optimum SN on the front wall surface of a power transformer. The measurements of the transformer's vibration responses induced by the magnetic field loading enable the detection of mechanical failures of the winding support joints inside the transformer.

Description of the Example

In this example, a loosening of the winding support joint is considered to be the failure mode. Detection of the failure will be realized by collecting the vibration signal, induced by magnetic field loading with a fixed frequency on the power transformer core, using an optimally designed SN on the external surface of the transformer. The validated finite element (FE) model of a power transformer was created in ANSYS 10, as shown in Fig. 8.7, where one exterior wall is removed to make the interior structure visible. Figure 8.8 shows 12 simplified winding support joints, 4 for each winding. The transformer is fixed at the bottom surface and a vibration load with a frequency of 120 Hz is applied to the transformer core. The joint loosening was realized by

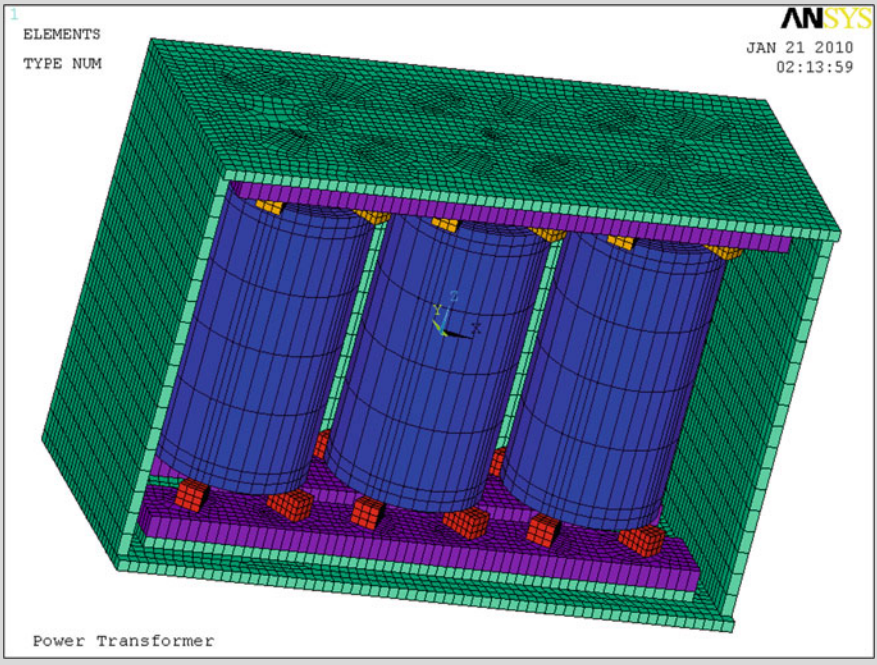


Fig. 8.7 An FE model of a power transformer (without the covering wall). Reprinted (adapted) with permission from Ref. [4]

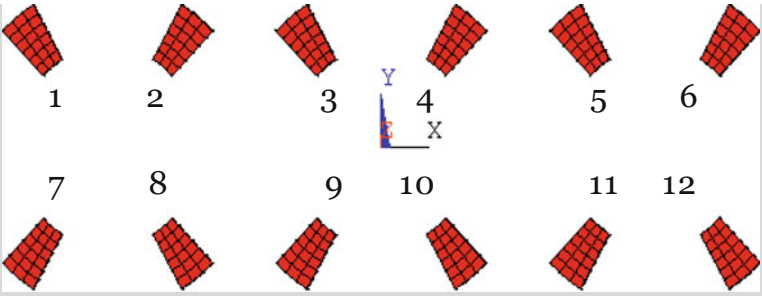


Fig. 8.8 Winding support joints and their numbering. Reprinted (adapted) with permission from Ref. [4]

reducing the stiffness of the joint itself. Different combinations of the loosening joints will be treated as different health states of the power transformer; these will be detailed in the next subsection.

The uncertainties in this example are modeled as random parameters with corresponding statistical distributions, as listed in Table 8.3. These uncertainties include the material properties (e.g., Young's modulus, densities, and Poisson ratios) for support joints and windings, as well other parts in the power transformer system. In addition, geometric parameters are also considered as random variables. These uncertainties will be propagated into the structural vibration responses and will be accounted for when designing the optimum SN.

Health States

For the purpose of demonstrating the proposed SN design methodology, 9 representative health states (see Table 8.4) were selected from all possible

Table 8.3 Random properties of the power transformer. Reprinted (adapted) with permission from Ref. [4]

Random variable	Physical meaning	Randomness (cm, g, degree)
X_1	Wall thickness	$N(3, 0.06^2)$
X_2	Angular width of support joints	$N(15, 0.3^2)$
X_3	Height of support joints	$N(6, 0.12^2)$
X_4	Young's modulus of support joint	$N(2e12, 4e10^2)$
X_5	Young's modulus of loosening joints	$N(2e10, 4e8^2)$
X_6	Young's modulus of winding	$N(1.28e12, 3e10^2)$
X_7	Poisson ratio of joints	$N(0.27, 0.0054^2)$
X_8	Poisson ratio of winding	$N(0.34, 0.0068^2)$
X_9	Density of joints	$N(7.85, 0.157^2)$
X_{10}	Density of windings	$N(8.96, 0.179^2)$

Table 8.4 Definition of system health states

Health state	1	2	3	4	5	6	7	8	9
Loosening joints	–	1	2	3	1, 2	1, 3	1, 5	1, 9	1, 11

combinations of 12 winding support joint failures. Among these 9 selected health states, HS_1 denotes a healthy condition without the presence of any loosening joints. HS_2 to HS_9 are health states with either one or two loosening joints. Figure 8.9 shows the stress contour of a power transformer in a healthy state at the nominal values of the random parameters used for the structural simulation. The first 100 sets of simulation results were used as the training data set; the remaining simulations were used as the test data set. These simulation results were later used to evaluate the SN's detection capabilities. The problem is formulated with the goal of designing an SN on the surface of the covering wall of the power transformer to minimize the cost of the SN (or the number of sensors), while satisfying the detectability constraints for each health state (i.e., the detectability should be greater than the chosen target detectability of 0.95).

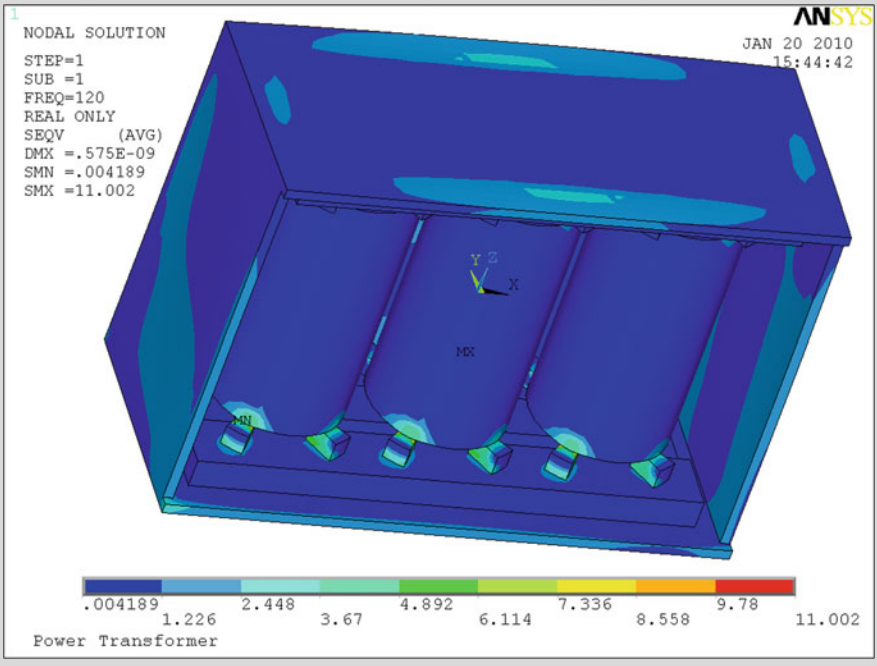


Fig. 8.9 Stress contour of the winding supports for a healthy-state power transformer. Reprinted (adapted) with permission from Ref. [4]

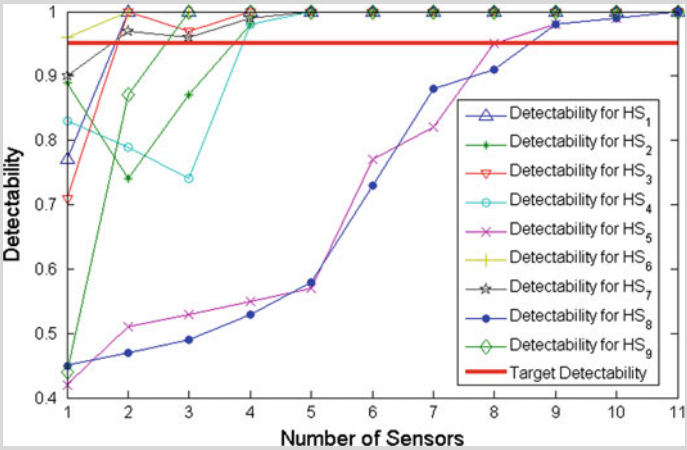


Fig. 8.10 Detectability with an optimum design and detectability with different numbers of sensors. Reprinted (adapted) with permission from Ref. [4]

The vibration amplitude of each node on the surface of the covering wall was used as the simulated sensor (accelerometer) output. Thus, the design variables in this example include: (i) the total number of accelerometers, (ii) the location of each accelerometer, and (iii) the direction (X or Z) of each accelerometer.

Results and Discussion

The SN design problem in this example was solved using the genetic algorithm. Figure 8.10 shows the detectability for each of the 9 health states at the optimum SN design, and the detectability for each different number of total sensors. Using a target detectability of 0.95, we obtained the optimum SN design on the outer wall surface (140 cm × 90 cm) with a total of 9 sensors. The results of this example suggest that the proposed SN design framework is capable of solving SN design problems for complicated engineered systems with multiple system health states and a wide variety of system input uncertainties.

8.2.4 Self-powered Wireless Sensors for Health Sensing

The effectiveness of PHM for failure prevention and reliability improvement relies significantly on the usefulness and completeness of health-relevant information conveyed by the sensor signals. Advances in wireless communications and low-power electronics have allowed the deployment of wireless sensor networks

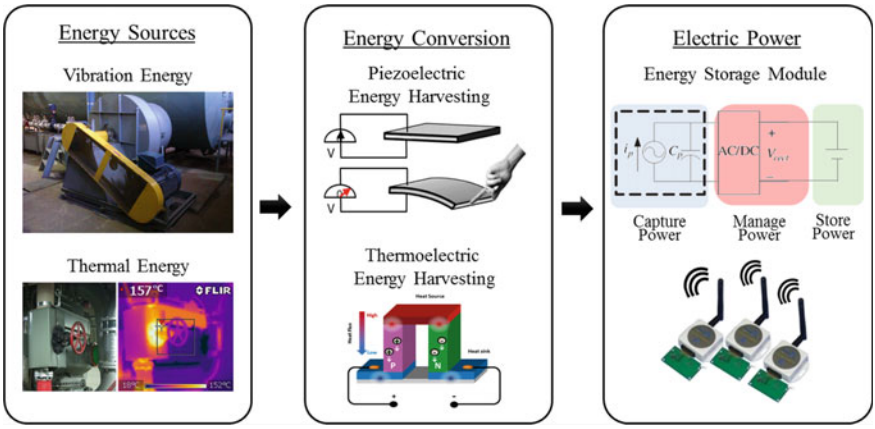


Fig. 8.11 Concept of energy harvesting for self-powered wireless sensors

(WSNs) for PHM. However, because the powering of wireless sensors still relies on chemical batteries, the limited lifespan of chemical batteries makes it difficult to use wireless sensors, especially when replacement is needed in inaccessible or remote locations. Furthermore, according to the U.S. Department of Energy [10], estimated battery replacement costs of $80–$500 (including labor) exceed the price of the sensor. This battery issue that affects wireless sensors used in health sensing is prompting research interest in developing a self-powered solution.

Energy harvesting has received much attention as an alternative solution to possibly eliminate the replacement cost of the chemical batteries in wireless sensors. Energy harvesting technology converts ambient, otherwise wasted, energy sources into electric power that can be used for operating wireless sensors. Figure 8.11 shows the concept of energy harvesting for self-powered wireless sensors.

Energy Conversion (Transduction) Mechanisms

Examples of ambient, otherwise wasted, energy sources include light, fluid flow, temperature difference, and vibration. A solar cell can convert the energy from light directly into electric power. This transduction mechanism is called the photovoltaic effect. While solar cells can produce relatively high power density; their use is limited in a dim light conditions and they are unsuitable where light is not accessible.

As shown in Fig. 8.11, thermal energy, such as temperature difference, can be converted into electric power using the thermoelectric transduction mechanism. The thermoelectric effect was discovered by the Baltic German physicist, Thomas Johann Seebeck. Thermoelectricity refers to the direct conversion of temperature differences to electric voltage, and vice versa. The output voltage generated by the thermoelectric effect is proportional to the temperature difference between the junctions of dissimilar conductors.

Likewise, vibration energy, one widely available ambient energy source, can be converted into electric power using piezoelectric, electromagnetic, electrostatic, and/or magnetostrictive transduction mechanisms. Among vibration-based energy harvesting technologies, piezoelectric energy harvesting has been preferred due to its high energy density and easy installation. In 1880, French physicists, Jacques and Pierre Curie, first discovered piezoelectricity. The prefix 'piezo' comes from the Greek 'piezein' which means to pressure or squeeze. In this method, as shown in the second column in Fig. 8.11, a piezoelectric material produces electric polarization in response to mechanical strain. This transduction mechanism is called the direct piezoelectric effect. The amount of output voltage generated by the piezoelectric material is proportional to the mechanical strain.

Among the aforementioned energy conversion mechanisms, piezoelectric energy harvesting could be a very attractive solution for powering wireless sensors, because engineered systems usually induce vibrations in operation. To help the readers better understand how to realize self-powered wireless sensors for PHM, the following subsection provides a brief overview of the key issues in piezoelectric energy harvesting.

Key Issues in Piezoelectric Energy Harvesting

Research in piezoelectric energy harvesting can be broken into four key issues, specifically: (i) development of materials, (ii) modeling and analysis, (iii) mechanics-based design, and (iv) circuit design. To successfully realize self-powered wireless sensors using piezoelectric energy harvesting, it is necessary to thoroughly understand the key issues and make connections between them.

- Development of materials: Piezoelectric materials include lead zirconate titanate (PZT), zinc oxide (ZnO), polyvinylidene difluoride (PVDF), lead magnesium niobate-lead titanate (PMN-PT), and polypropylene (PP) polymer. It is never enough to emphasize only the material issues in order to improve the mechanical and electrical properties of the piezoelectric material. For example, piezoelectric ceramics, such as PZT, have a high piezoelectric and dielectric constant, but are inherently brittle and less durable. Meanwhile, piezoelectric polymers, such as PVDF, have high flexibility but low electromechanical coupling. For this reason, many material scientists have been devoted to developing flexible as well as electromechanically efficient piezoelectric materials based on nanotechnology.
- Modeling and analysis: Prior to designing the piezoelectric energy harvester and selecting the best sites for installation, it is essential to make a preliminary estimate of the output power based on the vibration data acquired from the engineered system. This issue drives current research interest in developing an electromechanically-coupled model with high predictive capability, based on rigorous theories and mechanics. Many research efforts have been made to advance an analytical model (e.g., lumped-parameter, Rayleigh-Ritz method, and distributed-parameter) that can describe the physics of the electromechanical behavior of the piezoelectric energy harvester. Since the commercialized

piezoelectric energy harvester is generally manufactured as a thin cantilevered structure, the partial differential equation governing piezoelectric coupling under base excitation is derived based on the Euler-Bernoulli beam (large aspect ratio) or the Kirchhoff plate (small aspect ratio) theories. To obtain the output current generated by the piezoelectric energy harvester in response to the mechanical strain, the coupled electrical circuit equation is derived by substituting the electric displacement component of the piezoelectric constitutive equation into the integral form of Gauss's law. Most electromechanically coupled models have been developed under the assumption that the input vibration signal is a harmonic sinusoidal function. Since the inherent random nature (e.g., the variation of an amplitude and driving frequency) in realistic vibrations induced by engineered systems significantly affects the output power generated by the piezoelectric energy harvester, one remaining challenge in the modeling and analysis involves how to stochastically quantify the output power under non-stationary random vibrations.

- Mechanics-based design: This research avenue aims to provide an optimal design of the piezoelectric energy harvester to enhance the conversion efficiency. The most commonly used design of the piezoelectric energy harvester is a cantilever beam, which can achieve higher mechanical strain and lower resonance frequency, thereby producing relatively more electric power. The cantilever piezoelectric energy harvester is designed as a linear resonator. This fact implies that the maximum output power can be obtained when the fundamental resonance frequency of the piezoelectric energy harvester matches the dominant driving frequency of the ambient vibrations. As a shape design issue, a trapezoidal cantilever beam exhibits better electromechanical performances than a rectangular cantilever beam due to the uniformly large strain at every point on the beam. In recent years, the best performance of resonance-based piezoelectric energy harvesters has been limited to a very narrow bandwidth adjacent to the resonance frequency; therefore, exploitation of nonlinearity has been proposed to broaden the bandwidth (e.g., designs such as monostable Duffing, impact, and bistable oscillator). Another design issue is how to attach the piezoelectric patch onto the surface of vibrating engineering systems. Since output voltage is proportional to the strain, tensile strain yields positive output voltage, and vice-versa. This implies that the output voltage could decrease significantly when the piezoelectric patch possesses a strain nodal line; this is called voltage cancellation. Therefore, it is worth noting that the optimal placement of the piezoelectric patches should be determined to generate the greatest amount of electric power.
- Circuit design: It is of great importance to develop a power conditioning circuit to improve energy transfer from the piezoelectric energy harvester to an electrical load (e.g., external electrical resistance) and/or a storage. For instance, electrical regulation should be optimized to maximize the output power, which is generally composed of three stages: (i) energy capture, (ii) energy rectification, and (iii) energy storage. One of the most important aspects in this electrical circuit configuration is the impedance matching between a piezoelectric energy

harvester and electrical regulation. Moreover, while the piezoelectric energy harvester may produce alternating current (AC) in accordance with the sign change of the curvatures of the dynamic strains, charging a capacitor requires stable direct current (DC). Thus, in this case, an AC/DC converter is needed to supply the electric power needed to operate electronic devices.

Example 8.2 Self-Powered Wireless Sensors Using a Piezoelectric Energy Harvesting Skin

Figure 8.12 shows an advanced design concept for piezoelectric energy harvesting, referred to as a multimodal EH skin, which is composed of piezoelectric patches (e.g., lead zirconate titanate, polyvinylidene fluoride, and macro fiber composites) directly attached to the structural layer (the surface of a vibrating engineered system) as one embodiment. Conventional piezoelectric energy harvesters that are manufactured as a cantilever beam have some drawbacks from a practical point of view: (i) additional space is required for a proof mass and clamping fixture, (ii) significant vibration energy could be lost when a clamping condition becomes loosened after long periods of use, and (iii) fatigue failure is expected due to the excessive strain on the clamping part. In contrast, an EH skin does not require clamping fixtures and proof mass. This example aims to demonstrate how EH skin can be used to scavenge electric power from realistic vibrations induced by the electromagnetic motor of a cooling fan (1170–1180 RPM) to enable self-powered wireless sensors in real-time.

Design Methodology of the EH Skin

In the design of the EH skin, topology optimization is implemented to determine a highly efficient piezoelectric material distribution based on the finite element (FE), as shown in Fig. 8.13. Initially, the design space is fully covered with the piezoelectric material. Since the larger absolute value of the in-plane normal strains ensures a higher output voltage, topology optimization tries to eliminate some piezoelectric materials where the sum of the

Fig. 8.12 Piezoelectric energy harvesting skin for powering wireless sensors

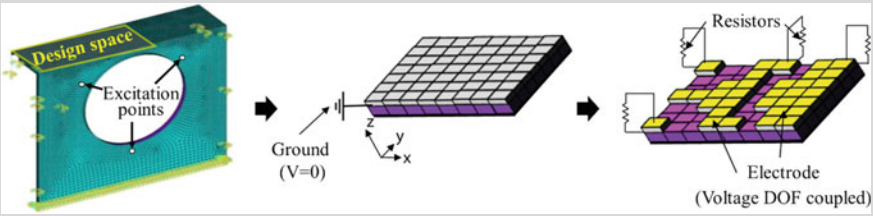


Fig. 8.13 Topology optimization of a piezoelectric energy harvesting skin

in-plane normal strains is locally small. Based on this design rationale, it is important to note that topology optimization can minimize voltage cancellation by segmenting the piezoelectric materials that possess inflection lines. After topology optimization, the optimal resistors are obtained to maximize the output power.

Demonstration of In-Situ, Self-powered Wireless Sensors

Figure 8.14 shows the experimental setup devised to demonstrate the capability of an EH skin for powering wireless sensors in real time. As shown in Fig. 8.14, four three-dimensional acceleration sensors and one analog temperature sensor were connected to wireless platforms (AmbioMote24 from AmbioSystems, LLC); these platforms were used to transmit sensor signals to a laptop computer. Furthermore, the real-time signal acquired from two acceleration sensors (ID: 19 and 41), as carried by a wireless platform (AmbioMote24-A), and two acceleration sensors and one temperature sensor

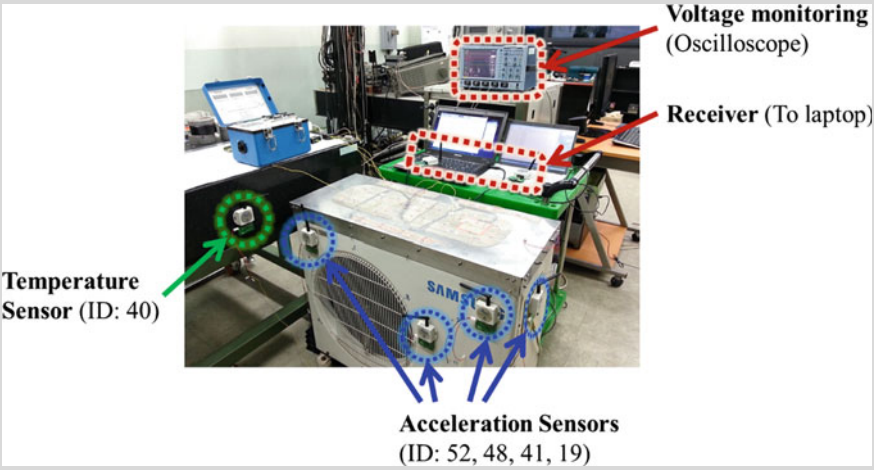


Fig. 8.14 Experimental setup for in-situ, self-powered wireless sensors

(ID: 48, 52, and 40) were connected to the other platform (AmbioMote24-B). An oscilloscope (LT354 M from LeCroy) was used to measure the output voltage in real time. While the outdoor condensing unit is in operation, an output voltage of 4–5 V (peak-to-peak) was measured and all five wireless sensor signals were successfully transmitted to the laptop computers in real time.

8.2.5 *Other Issues in Designing the Health Sensing Function*

SN optimization can determine the types, numbers, and locations of sensors by maximizing the detectability of potential failures. Remaining issues include how to design data acquisition, data communication, and data management strategies for robust health sensing. First, the data acquisition strategy concerns determination of the data sampling rate, the data acquisition period, data acquisition duration, etc. In the case of a steam turbine, gap displacement signals are measured at a sampling rate of 1024–4096 Hz every ten to sixty minutes and when an anomaly happens. A lower sampling rate or period can make it difficult to extract useful health information out of gap signals. In contrast, a higher sampling rate or period burdens data acquisition, communication, and management solutions. Therefore, decision making about data acquisition is extremely important for the rest of the health sensing, reasoning, and prognostics functions. Second, data communication and networking deal with data logging, wired/wireless data communication, and data networking protocols (Ethernet, TCP/IP, etc.) and technologies (RFID, WiFi, BT, LTE, etc.). In the case of a smart factory, wireless data communication and networking is preferred; however, some technical challenges remain in terms of data loss, power supply to wireless sensors, cyber-security, etc. This decision must be made to maximize the reliability of data transmission, which is highly influenced by data size and structure (determined by the data acquisition strategy), engineering site (i.e., outdoor/indoor), facility types, budget, and so on. Third, the data management strategy is concerned about the development and execution of architectures, procedures, and practices that adequately manage the full data lifecycle needs of the engineering assets. The data management strategy should be designed to ensure a high level of data quality and accessibility for big data analytics and to lay the foundation for the rest of the PHM process.

8.3 The Health Reasoning Function

The primary tasks of the health reasoning function are extraction of health-relevant system information (or features) from raw sensor signals and early detection of faults based on the extracted health-relevant information. These tasks can be accomplished by (i) continuously or periodically monitoring the operation of an engineered system, (ii) detecting and diagnosing abnormal conditions or faults of the system using feature extraction and health classification techniques, and (iii) assessing the significance of the detected faults. The procedure for executing the health reasoning function is shown in Fig. 8.15. The process involves the following key steps:

- Signal preprocessing: The aims of signal preprocessing are to isolate specific signals of interest, filter out noise and outliers, and/or achieve a normalized scale. Signals that are preprocessed are expected to achieve better accuracy in fault detection and classification than those that are not preprocessed. The applicability of one specific signal preprocessing technique depends on the kinds of signals that need to be preprocessed, how noisy they are, how many outliers they contain, and what techniques will be used in the subsequent processing steps. Some of the most important techniques include time synchronous averaging resampling, signal filtering, and various averaging techniques.
- Feature extraction: This step extracts health-relevant features from raw or preprocessed sensor measurements acquired from continuous or periodic sensing of an engineered system. Inherent in this feature extraction step is the condensing of raw sensor data. Commonly used feature extraction techniques include time domain analysis (e.g., statistical moment calculation), frequency domain analysis (e.g., FFT), and time-frequency domain analysis (e.g., wavelet transform).
- Feature selection: Feature selection aims at selecting an optimum subset of features that minimize redundancy and focus on features of maximum relevance to the health states of the system. Both non-adaptive and adaptive approaches can be used to select features that are capable of discriminating measurements that belong to different health states.
- Fault detection and classification (health diagnostics): This step involves (i) fault detection that determines whether some type of fault has occurred, and (ii) fault classification that identifies to which of a set of health states (defined based on fault type and location) a new measurement belongs. An additional process is often needed to quantify the severity of a detected fault (e.g., the size of a crack on a plate, the loss of power from a battery) in the form of a normalized health measure, or health index. This additional process yields a quantitative measure of the fault and is particularly useful for health prognostics.

Together, fault detection and classification are commonly called health diagnostics. In most engineered systems equipped with the capability of health diagnostics, fault detection runs continuously, while fault classification is triggered only upon the

detection of a fault. In other systems, fault detection and classification may run in parallel and be performed simultaneously.

8.3.1 Signal Preprocessing

Prior to feature extraction, raw sensor signals are often preprocessed to isolate specific signals of interest, to remove noise and outliers, and/or to achieve a normalized scale. Signal preprocessing is essential to ensuring good accuracy in fault detection and classification. Several important techniques for signal preprocessing are discussed in the following sections.

8.3.1.1 Resampling

Resampling is a signal preprocessing technique that changes the sampling rate of the raw sensor signals. Since most of the signals are acquired using a pre-determined sampling rate, the signals may require resampling to consider the signal characteristics. Reducing the sampling rate by an integer factor is called downsampling;

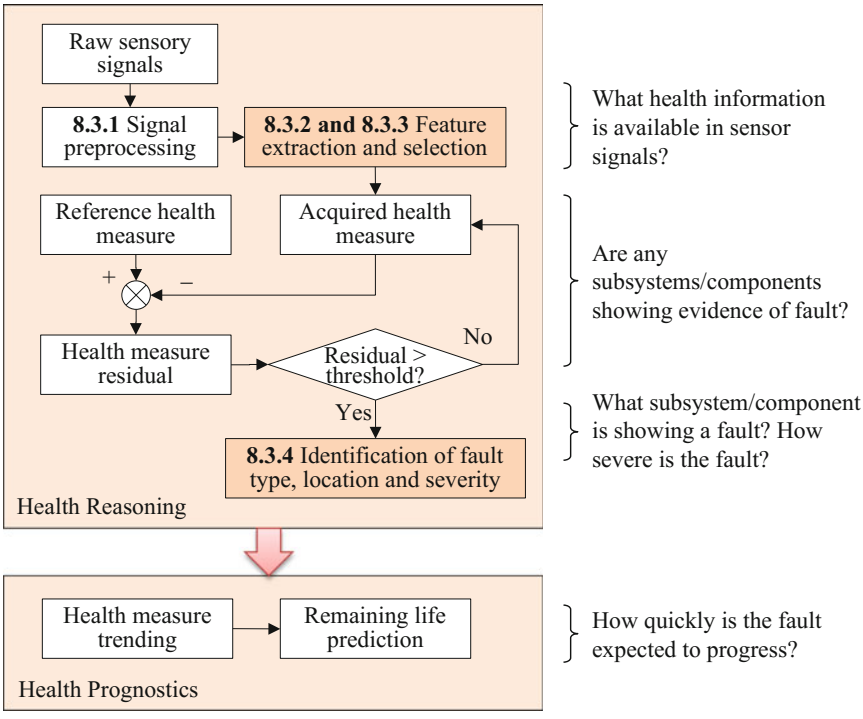


Fig. 8.15 Flowchart for health reasoning and prognostics

whereas, increasing the sampling rate by an integer factor is called upsampling. Note that downsampling is used with an anti-aliasing filter because the sampling rate reduction causes distortion of high-frequency components. Combining downsampling and upsampling, sampling rate conversion by a noninteger factor can be achieved. For example, the rotating speed of a rotor system fluctuates, while the sampling rate is fixed. Thus, the raw vibration signals have a different number of points per a rotation of the rotor, which will increase uncertainties in the analysis procedure. Uncertainty can be reduced by resampling the signals into a fixed number of points per cycle with respect to the tachometer signals.

8.3.1.2 Time Synchronous Averaging

Time synchronous averaging (TSA) is a signal preprocessing technique that extracts periodic waveforms from noisy data. TSA has been widely used for gearbox diagnostics [11–14], where it can allow the vibration signal of a gear of interest to be isolated from noisy vibration signals. The isolated signal of the gear ideally does not contain the signals of other gears or noise sources that are not synchronous with the gear. The two-step procedure for conventional TSA is shown in Fig. 8.16. In Step 1, the raw sensor signal is divided into N segments based on the rotational frequency of the gear of interest. In Step 2, an ensemble average over the divided segments is taken to form a preprocessed signal [14]. Since vibration signals are collected with a pre-determined sampling rate, while the system operates with a varying rotational speed, the number of the divided segments in Fig. 8.16 may vary from one revolution to the next. Thus, the vibration signals should be resampled so that the number of samples assigned during a single revolution of the gear remains constant [15]. This can be achieved by interpolating the vibration signal with a constant angle interval of consecutive samples. Using TSA, the vibration signal produced by the gear of interest remains in its own shape because every divided segment contains similar vibration patterns created by meshing of the gear of

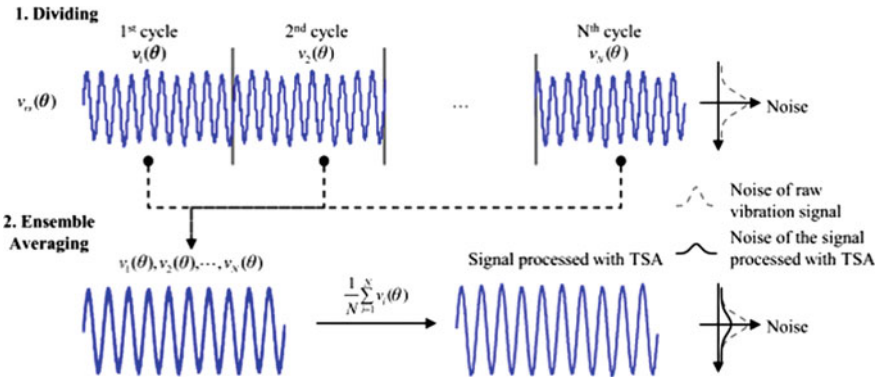


Fig. 8.16 Procedure for conventional TSA. Reprinted (adapted) with permission from Ref. [14]

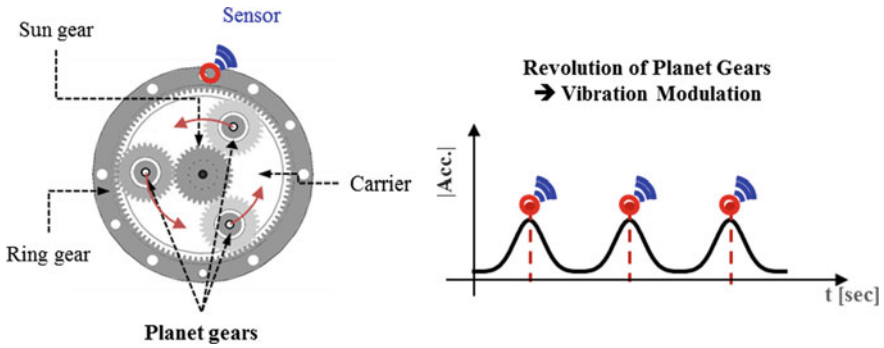


Fig. 8.17 Vibration modulation due to the revolution of the planet gears of a planetary gearbox

interest. On the other hand, the noise term converges to zero as a considerable number of segments accumulates, as shown in the lower right of Fig. 8.16.

TSA needs to be revised for use in more complex gearboxes, such as planetary gearboxes in which planet gears revolve around a sun gear (see Fig. 8.17). Because a typical sensor is fixed at a gearbox housing, the measured vibration signal is modulated as a function of the distance from the fixed sensor to the revolving planet gears.

For such systems, TSA has been developed with a window function [14]. The window function extracts the vibration signals as the planet gears approach the sensor so that the extracted signals can be used for ensemble averaging, as shown in Fig. 8.16. This process enables the vibration signals that are out of interest to be ignored, while increasing the signal-to-noise (S/N) ratio. Typical mathematically defined window functions, such as Tukey window and Hann window, have a bell-shape to highlight the instances in which the planet gears pass the sensor and reduce the signal amplitude when the planet gears are located far from the sensor. Extracted vibration signals from the bell-shaped window function would have high similarity and could serve as good sources for the ensemble averaging process shown in Fig. 8.16. Autocorrelation-based TSA (ATSA) was developed [14] as a more physics-oriented approach. The autocorrelation function, which is a measure of similarity, increases as the planet gears approach the sensor and decreases as the planet gear recedes from the sensor. In ATSA, the autocorrelation function is used to quantify the similarity of the measured signal as a function of the distance from the fixed sensor to the revolving planet gears; this function is then used to design the shape of the window function.

8.3.1.3 Filtering

Filtering is the process used to extract signals of interest while removing or deforming unwanted frequency components. Because a filter is a kind of system that uses a signal as an input to result in another signal as an output, the filtering process can be illustrated as shown in Fig. 8.18. The frequency components of the

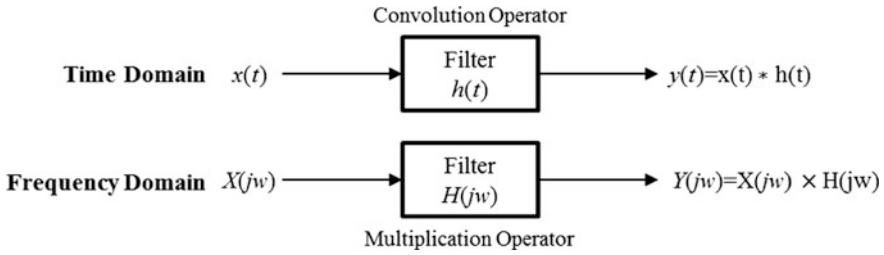


Fig. 8.18 Filtering process

input signal ($X(jw)$) are modified by the frequency characteristics of the filters ($H(jw)$). A sound equalizer is one of the most popular applications of a filter that modifies the frequency-shape of an input signal. If an equalizing filter has a high value for low frequencies and a low value for moderate or high frequencies, the filter will enhance the bass sound.

In engineered systems, frequency-selective filters and linear finite impulse response (FIR) and infinite impulse response (IIR) filters are the most widely used filters. Frequency-selective filters have unit values for the desired narrow-range frequencies and zeros for the other frequencies, as shown in Fig. 8.19, where a low-pass filter, a band-pass filter, and a high-pass filter are presented. For example, in a multi-stage rotating system, a low-pass filter, a band-pass filter, and a high-pass filter can be used to selectively analyze the health state of the low-speed shaft, middle-speed shaft, and high-speed shaft, respectively. On the other hand, a low-pass filter and a high-pass filter can be used to filter out high-frequency noise components and low-frequency undesired modulating components, respectively.

Other filtering methods, such as the moving and exponential average smoothing techniques, can also be used to filter out noise and outliers in raw sensor signals. These moving and exponential average smoothing techniques are special cases of the well-known linear finite impulse response (FIR) and infinite impulse response (IIR) filters, respectively, and are popular due to their ease of implementation. Linear filters estimate the current data point by taking a weighted sum of the current and previous measurements in a window of a finite or infinite length. Mathematically, the FIR filter can be represented as [16]

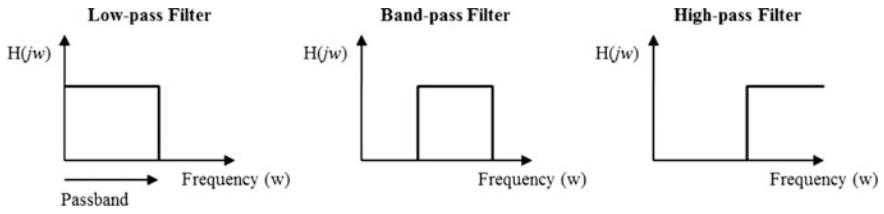


Fig. 8.19 Examples of frequency-selective filters

$$\hat{x}_k = \sum_{i=1}^{I} w_i x_{k-i+1} \tag{8.4}$$

where I is the length of the filter, and $\{w_i\}$ is the sequence of weights that define the characteristics of the filter and sum to unity. Note that when all weights are equal, the FIR filter reduces to the mean or average filter. IIR filters are linear filters with infinite filter lengths. An important IIR filter is the so-called exponentially weighted moving average, which filters the current measurement by exponentially averaging it with all previous measurements [16]

$$\hat{x}_k = a x_k + (1 - a)\hat{x}_{k-1} \tag{8.5}$$

where a is an adjustable filtering parameter between 0 and 1.

Computation of the output of a median filter can be done in two steps. First, an odd number of measurement values are sorted. Second, the median value is estimated and used as the filter output. As can be observed in Eq. (8.6), median filters use both past and future measurements for smoothing the current measurement. Despite their advantages over linear filters, median filters have several drawbacks. For example, a median filter can cause edge jitter, streaking, and may remove important details in the signals [17, 18]. This is mainly because median filters only use rank-order information of the input data and discard the input data's original temporal order information. One way to use both rank and temporal order information of sensor data is to combine a FIR filter with a median filter, which gives rise to a FIR-median hybrid filter [19]. The hybrid filter synthesizes the capability of the FIR filter to remove noise in stationary signals and achieve the desirable properties of the median filter.

Although linear filters are easy to implement and produce good filtering performance, they can smooth out features in a signal that may actually indicate the onset of a fault [17, 18]. Non-linear filters, in contrast, offer the opportunity to preserve edges in the signal, while still reducing noise and removing gross outliers. As a popular class of non-linear filters, median filters possess three important properties: (i) edge preservation, (ii) noise reduction, and (iii) robustness against gross outliers. A median filter with a length of $2I + 1$ can be represented as [17, 18]

$$\hat{x}_k = median(x_{k-I}, x_{k-I+1}, x_{k-I+2}, \ldots, x_k, \ldots, x_{k+I-2}, x_{k+I-1}, x_{k+I}) \tag{8.6}$$

8.3.1.4 Signal Normalization

As multiple sensor signals may have significantly different scales, the process of normalizing the sensor signals is important to ensuring robust fault detection and classification. An example of data normalization on a single sensor signal is shown in Fig. 8.20, where the raw measurements acquired by a sensor are normalized roughly between 0 and 1.

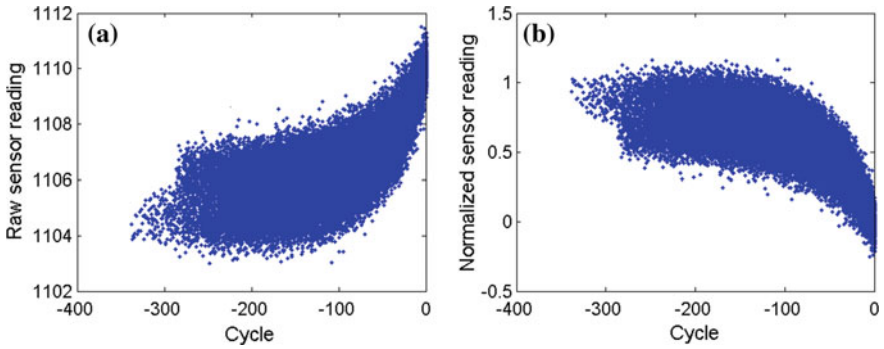


Fig. 8.20 Raw (**a**) and normalized (**b**) sensor signals

8.3.2 Feature Extraction

Feature extraction can be any computer operation or series of operations performed on preprocessed sensor data to gain insightful, health-relevant information. Feature extraction processes the sensor data in the time, frequency, and joint time-frequency domains to generate a set of features that may indicate the presence of various faults.

MATLAB® DSP System Toolbox

The Digital Signal Processing (DSP) System Toolbox provides algorithms and tools for processing digital signals in MATLAB® and Simulink®. It allows signal processing systems to be designed and prototyped using signal processing algorithms and components in the Simulink® block format. Figure 8.21 shows the Library contained in the DSP system toolbox (type "***dsplib***" in the MATLAB command window to open this library).

Feature Extraction in the Time Domain

A variety of statistical tools can be applied to time domain signals to acquire characteristic features. The following is an incomplete list of such tools:

- *Distribution Fitting*: Find the best distribution fit of the input data.
- *Histogram*: Generate a histogram of an input or sequence of inputs.
- *Autocorrelation*: Compute an autocorrelation of vector inputs.
- *Correlation*: Compute a cross-correlation of two inputs.
- *Max./Min.*: Find the maximum or minimum values in an input or sequence of inputs.
- *Mean*: Find the mean value of an input or sequence of inputs.
- *Standard Deviation*: Find the standard deviation of an input or sequence of inputs.
- *Variance*: Compute the variance of an input or sequence of inputs.
- *RMS*: Compute the root-mean-square (RMS) value of an input or sequence of inputs.

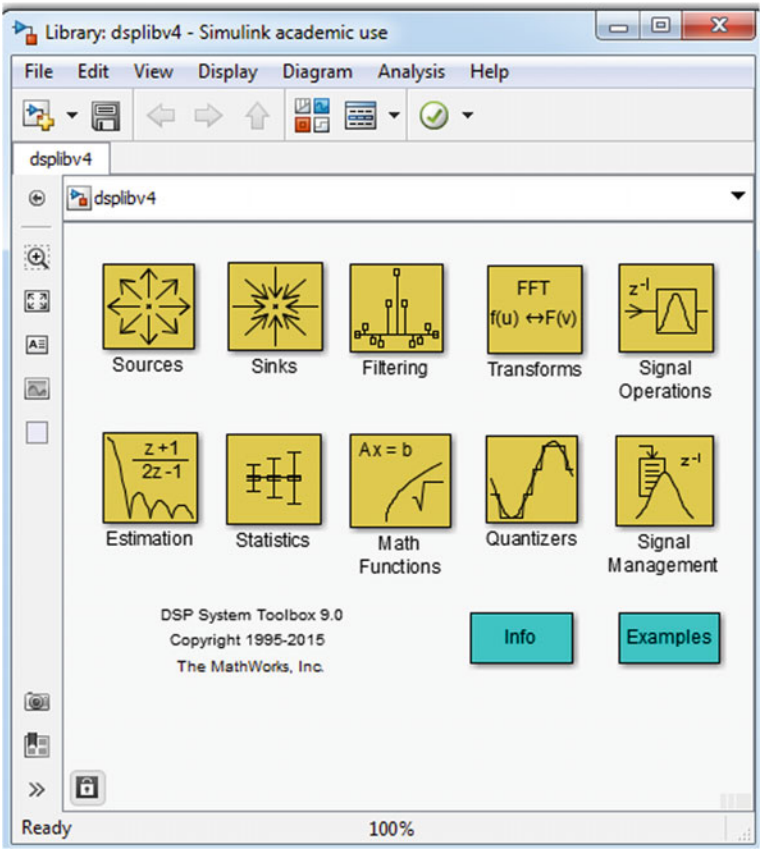


Fig. 8.21 MATLAB® DSP signal processing library

- *Sort*: Sort input elements by value.

Representative features in the time domain are presented in Table 8.5. Among the eight time-domain features, max (t1), mean (t2), and root mean square (t3) are related to the energy of a signal. Standard deviation (t4), skewness (t5) and kurtosis (t6) are directly related to the second, third, and fourth standardized moments, and respectively measure the dispersion, lopsidedness, and heaviness of the tail of the signal distribution. The last three features, namely crest factor (t7), shape factor (t8), and impulse factor (t9), measure different aspects (e.g., sharpness) of the shape of a sinusoidal wave.

Example 8.3 Data processing in a MATLAB® DSP System Toolbox
First, build the data processing block diagram shown in Fig. 8.22. As the sinusoid signal shown in Fig. 8.23a is being processed, the RMS signal in

Table 8.5 Representative features in the time domain

Index	Features	Description
t1	Max (x_{max})	$\max(\lvert x_i \rvert)$
t2	Mean ($\bar{x}$)	$\frac{1}{m}\sum_{i=1}^{m}\lvert x_i \rvert$
t3	Root mean square (x_{rms})	$\sqrt{\frac{\sum_{i=1}^{m} x_i^2}{m}}$
t4	Standard deviation (σ)	$\sqrt{\frac{\sum_{i=1}^{m}(x_i-\bar{x})^2}{m-1}}$
t5	Skewness	$\frac{\sum_{i=1}^{m}(x_i-\bar{x})^3}{(m-1)\sigma^3}$
t6	Kurtosis	$\frac{\sum_{i=1}^{m}(x_i-\bar{x})^4}{(m-1)\sigma^4}$
t7	Crest (peak) factor	$\frac{x_{max}}{x_{rms}}$
t8	Shape (waveform) factor	$\frac{x_{rms}}{\bar{x}}$
t9	Impulse (pulse) factor	$\frac{x_{max}}{\bar{x}}$

Fig. 8.23b can be obtained. RMS signals are usually used to detect changes in machine vibrations.

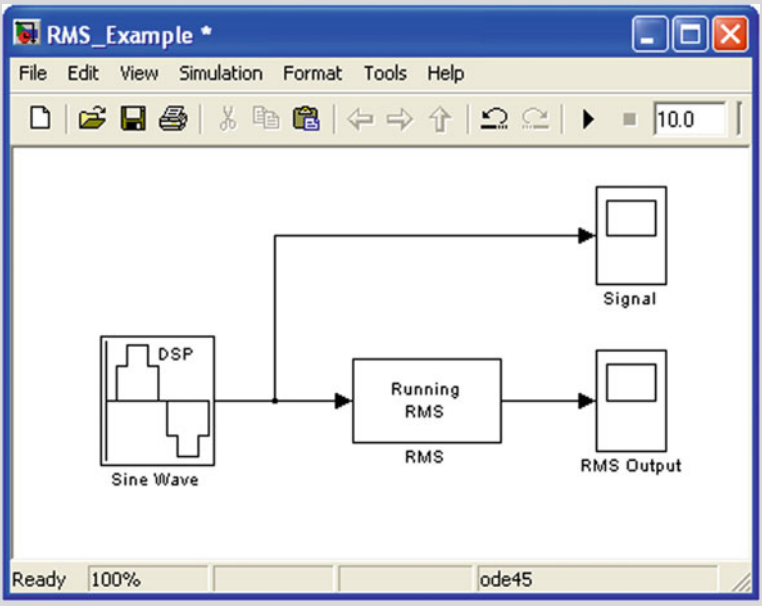


Fig. 8.22 Signal processing example: RMS block diagram

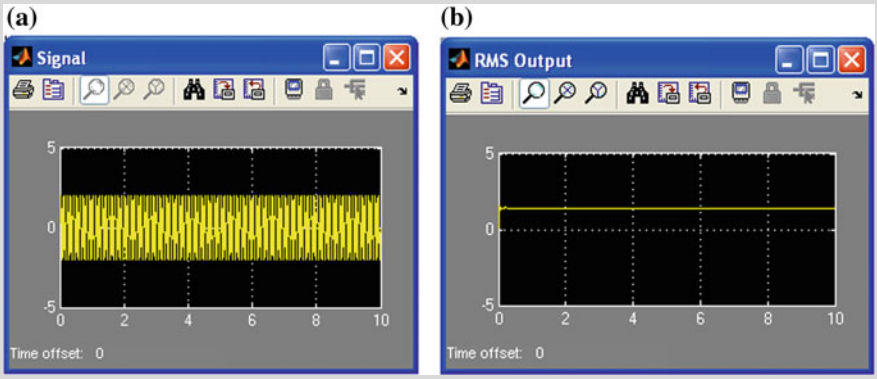


Fig. 8.23 Sample sinusoid signal (**a**) and RMS signal (**b**)

Feature Extraction in the Frequency Domain

When features extracted in the time domain do not appear to be informative, feature extraction in the frequency domain can be pursued. Fourier analysis converts a time-series signal from the original time domain to a representation in the frequency domain. This means that it generates a description of the distribution of the energy in the signal as a function of frequency. This is normally displayed as a plot of amplitude (y-axis) against frequency (x-axis) called a *power spectrum*. Fast Fourier transform (FFT), which rapidly performs Fourier analysis, has been widely used in many applications in engineering, science, and mathematics. Representative features that can be extracted from the power spectrum of a signal are presented in Table 8.6. In the table, f and $s(f)$ are the frequency and the power spectrum of the frequency, respectively. Among the three frequency domain features, frequency center (f1) and root mean square frequency (f2) are respectively the arithmetic and quadratic means of the frequency. Root variance frequency (f3) measures the dispersion of the power spectrum.

Consider a rotating bearing as an example. As the fault in the bearing progresses over the bearing's lifetime, the magnitudes of the vibration responses at the characteristic frequencies generally increase (see Fig. 8.24), and thus provide health-relevant information that can be used for health diagnostics and prognostics [20].

Table 8.6 Representative features in the frequency domain

Index	Features	Description
f1	Frequency center (FC)	$\frac{\int f \times s(f) df}{\int s(f) df}$
f2	Root mean square frequency (RMSF)	$\left[\frac{\int f^2 \times s(f) df}{\int s(f) df}\right]^{1/2}$
f3	Root variance frequency (RVF)	$\left[\frac{\int (f-FC)^2 \times s(f) df}{\int s(f) df}\right]^{1/2}$

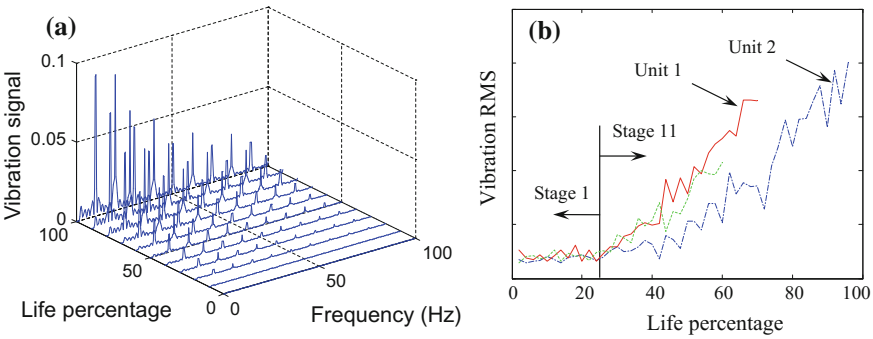


Fig. 8.24 Life-cycle evolution of vibration spectra (**a**) and RMS (**b**) with an inner race defect on a rotating bearing. Reprinted (adapted) with permission from Ref. [20]

Example 8.4 Fast Fourier transform

The MATLAB codes shown below generate a time domain signal, the sum of three sinusoid signals with resonant frequencies at 50, 120, and 200 Hz, and corrupt the signal with a Gaussian noise (see Fig. 8.25a). Then, the codes perform FFT on the signal and obtain a frequency-domain spectrum (see Fig. 8.25b) of the time-domain signal. Note that this example can also be

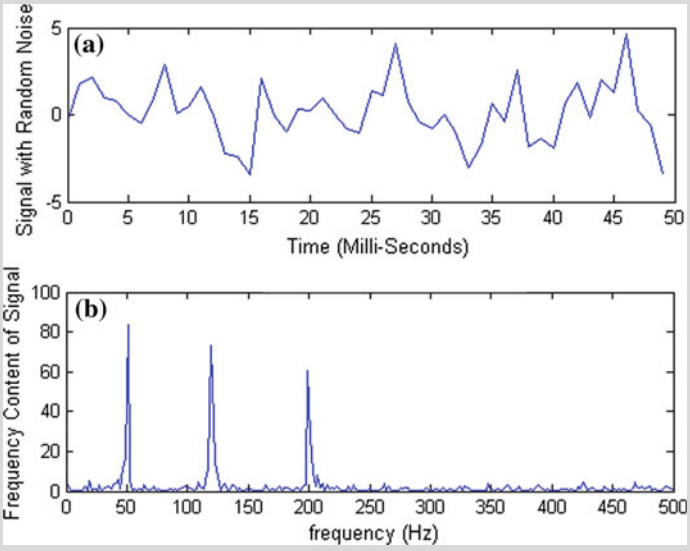


Fig. 8.25 Time domain signal (**a**) and frequency domain signal (**b**)

done by building a FFT block diagram using the MATLAB® DSP System Toolbox.

```
t = 0:0.001:0.6;
x = sin(2*pi*50*t)+sin(2*pi*120*t)+sin(2*pi*200*t);
y = x + randn(size(t));
figure(1)
subplot(2,1,1)
plot(1000*t(1:50),y(1:50))
xlabel('Time (Milli-Seconds)')
ylabel('Signal with Random Noise')

subplot(2, 1, 2)
Y = fft(y, 512);
Fy = Y.* conj(Y)/512;
f = 1000*(0:256)/512;
plot(f, Fy(1:257))
xlabel('frequency (Hz)');
ylabel('Frequency Content of Signal');
```

Feature Extraction in the Joint Time-Frequency Domain

There is a tradeoff between the resolution in the frequency domain and that in the time domain. A good resolution in the frequency domain may imply a poor resolution in the time domain. Similarly, a good resolution in the time domain may imply a poor resolution in the frequency domain. Although frequency domain representations, such as the power spectrum of a signal, often provide useful spectral information, the representations do not show how the frequency content of the signal evolves over time. Hence, it is important to introduce the time variable into the frequency domain analysis in order to represent the change of the spectral content over time. Joint Time-Frequency Analysis (JFTA) comprises a set of time-frequency transforms that map a one-dimensional time domain signal into a two-dimensional representation over both time and frequency [21]. JTFA builds a bridge between the time and frequency domain representations in that it provides both the frequency content of a signal and the change of the frequency content over time. The time-frequency transforms in JTFA are particularly useful for the representation of nonstationary signals containing multiple time-varying frequencies.

There are a number of different time-frequency transforms available for JTFA. The simplest transform is the Short Time Fourier Transform (STFT), which divides a longer time signal into shorter segments of equal length and then applies FFT repeatedly to these shorter segments [22]. This process reveals the Fourier spectrum on each of the shorter segments over a certain period. The changing spectra can be plotted as a function of time on a 3-D graph or a 2-D 1/2 representation (energy represented as colors with varying intensities of red, green, and blue light). Other

time-frequency transforms that can yield an estimate of the energy in a given time-frequency domain include wavelet transform [23, 24], Wigner distribution, Choi-Williams distribution, and spectrogram [25].

8.3.3 Feature Selection

As mentioned earlier, we can extract a large number of features (in the time, frequency, and time-frequency domains) from raw sensor signals. Among these features, only a subset of the features is relevant and should be used to build a health diagnostic model. Thus, we need to select the most relevant and unique features, while removing most irrelevant and redundant features from the data to improve the performance of the diagnostic model.

Feature Selection Using Non-adaptive Approaches

One method for selecting features for fault identification is to apply engineered flaws, similar to the ones expected in actual operating conditions, to systems and develop an initial understanding of the features that are sensitive to the expected fault. The flawed system can be used to identify the features that are sensitive enough to distinguish between the fault-free and faulty system. The use of analytical tools, such as experimentally validated finite element models, can be a great asset in this process. In many cases, analytical tools are used to perform numerical experiments where flaws are introduced through computer simulation.

Damage accumulation testing, during which significant structural components of the system under study are degraded by subjecting them to realistic loading conditions, can also be used to identify appropriate features. This process may involve induced-damage testing or accelerated degradation testing (e.g., fatigue testing, corrosion growth, and temperature cycling) to acquire feature data under certain types of damage states in an accelerated fashion. Insight into the appropriate features can be gained from several types of analytical and experimental studies, as described above, and is usually the result of information obtained from some combination of these studies.

Feature Selection Using Adaptive Approaches

In addition to the non-adaptive approaches mentioned above, adaptive approaches can also be used for selecting relevant features. As shown in Fig. 8.26, an adaptive approach typically consists of four major components, namely (i) feature subset generation, (ii) performance evaluation, (iii) stopping criteria check, and (iv) online testing. In the training phase, a certain search strategy generates candidate feature subsets, of which each subset is evaluated according to a diagnostic performance measure and compared with the previous best one with respect to this measure. A new, better subset replaces the previous best subset. This is repeated

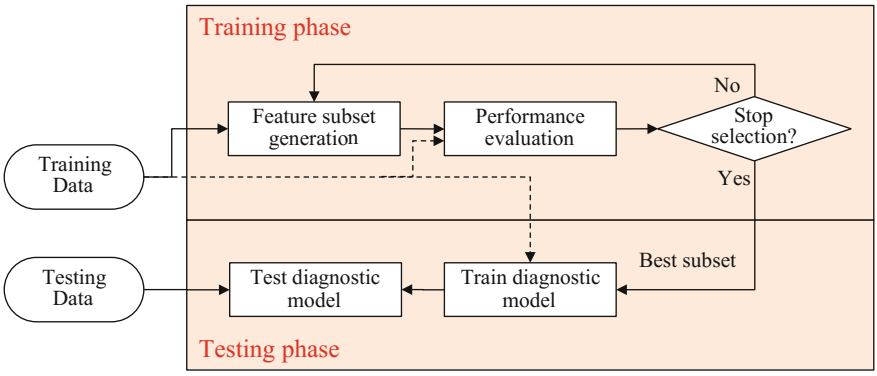


Fig. 8.26 Flowchart for an adaptive feature selection process

until a stopping criterion is satisfied. In the testing phase, the performance of the selected subset of features is evaluated with testing data not used in the feature selection process.

Feature Selection Using Deep Learning

One of the main issues with PHM of complex engineered systems is the lack of labeled data (i.e., data acquired from an operating system whose health state is known) as well as the cost of labeling unlabeled data (e.g., by performing additional diagnostics to assess the health of an operating system). Thus, there has been interest in exploring the use of unlabeled data as a way to improve prediction accuracy in fault diagnostics and failure prognostics. The availability of large volumes of this kind of data in many complex systems makes it an appealing source of information. Recently, deep learning methods have made notable advances in the fields of speech recognition [26, 27], computer vision [28, 29], and natural language processing [30, 31]. The unique ability of deep learning to automate learning of high-level, complex features from large volumes of unlabeled data makes it attractive for porting to the feature extraction/selection toolbox of a PHM practitioner. In particular, the practitioner could investigate the use of deep belief networks (DBNs) [32, 33], built as a stack of Restricted Boltzmann Machines (RBMs) on top of each other (see Fig. 8.27), to address the challenges of feature discovery when dealing with large amounts of unlabeled monitoring and inspection data.

In a DBN, the features are learned in a layer-by-layer manner, and the features learned by one-layer RBM become the input data for training the next layer of the RBM. This hierarchical multi-level learning extracts more abstract and complex features at a higher level, based on the less abstract features/data in the lower level (s) of the learning hierarchy. The bottom-layer RBM is trained with the preprocessed monitoring and inspection data, and the activation probabilities of hidden units are treated as the input data for training the upper-layer RBMs. Once the network is trained, the top layer's output becomes highly representative of deep

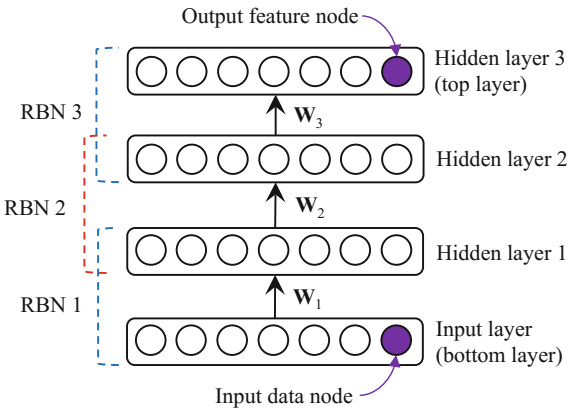


Fig. 8.27 General structure of a deep belief network

features (see Fig. 8.27) that can be used for fault diagnostics and failure prognostics. For the purpose of fault diagnostics, conventional classification models (see Sect. 8.3.4) can be trained, leveraging the deep features, with the aid of small amounts of labeled data.

8.3.4 Health Diagnostics

Fault diagnosis aims at determining the fault type, location, and severity based on the extracted feature. This task can be treated as a classification problem. The algorithms used in health classification usually fall into two categories: *supervised classification* and *unsupervised classification*. These two categories are illustrated in Fig. 8.28. When labeled data are available from both the fault-free and faulty systems, the statistical pattern recognition algorithms fall into the general classification referred to as supervised classification. Note that supervised classification belongs to a broader category, supervised learning, which also includes supervised regression (often useful

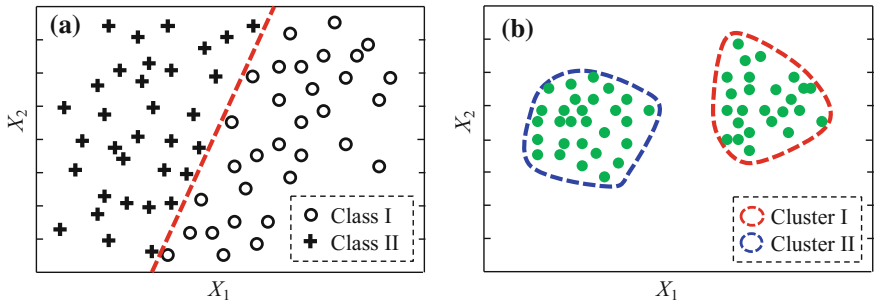


Fig. 8.28 Supervised classification (**a**) and unsupervised classification (**b**)

for health prognostics). Unsupervised classification refers to classification algorithms that are applied to data not containing examples from the faulty system. Unsupervised classification only aims to find the patterns in the data and create clusters to derive relationships inherent in the data. Outlier or novelty detection is the primary class of algorithm applied in unsupervised learning applications.

Supervised Fault Classification

Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis (LDA), also known as Fisher Discriminant Analysis (FDA), is a supervised classification technique that computes the directions of projection (or linear discriminants) that maximize the separation between multiple classes and minimize the variance within the classes [34, 35]. Suppose an input data set D, which contains m feature-class pairs, can be organized as $\{(\mathbf{x}_1, c_1), \ldots, (\mathbf{x}_m, c_m)\}$, where $\mathbf{x}_i$ is the ith feature vector of dimension N, and c_i is the class label of $\mathbf{x}_i$ indicating the class to which $\mathbf{x}_i$ belongs. Given the feature-class data set D, LDA aims to create a linear combination of the N features, which yields the largest differences between the means of the classes and the minimum variance within each class. Multi-class LDA is based on the analysis of two separation matrices: the with-class separation matrix and the between-class separation matrix. Mathematically, the between-class separation matrix is given by

$$\mathbf{S}_b = \sum_{j=1}^{M_c} m_j \left(\boldsymbol{\mu}_j - \boldsymbol{\mu}\right)\left(\boldsymbol{\mu}_j - \boldsymbol{\mu}\right)^{\mathrm{T}} \tag{8.7}$$

where M_c is the number of classes, m_j is the number of features in class j, $\boldsymbol{\mu}_j$ is the mean vector of the original features in the jth class, and $\boldsymbol{\mu}$ is the mean vector of the features in the entire M_c class. Similarly, the within-class separation can be defined as

$$\mathbf{S}_w = \sum_{i=1}^{m} \left(\mathbf{x}_i - \boldsymbol{\mu}_{c_i}\right)\left(\mathbf{x}_i - \boldsymbol{\mu}_{c_i}\right)^{\mathrm{T}} \tag{8.8}$$

Let $\mathbf{w}$ represent the project vector that maps the original N-dimensional features onto a one-dimensional space (or a line). The projected features can be expressed as $y_i = \mathbf{w}\cdot\mathbf{x}_i$, for $i = 1, \ldots, m$. Then, multi-class LDA can be formulated as an optimization problem in search of $\mathbf{w}$ that maximizes the ratio of the between-class separation to the within-class separation, as

$$\hat{\mathbf{w}} = \arg\max_{\mathbf{w}} \frac{\mathbf{w}^{\mathrm{T}}\mathbf{S}_b\mathbf{w}}{\mathbf{w}^{\mathrm{T}}\mathbf{S}_w\mathbf{w}} \tag{8.9}$$

If $\mathbf{S}_w$ is a nonsingular matrix, the solutions to the optimization problem in Eq. (8.9) are the eigenvectors of $\mathbf{S}_w^{-1}\mathbf{S}_b$ [34]. Generally, $\mathbf{S}_w^{-1}\mathbf{S}_b$ has at most $M_c - 1$ nonzero

generalized eigenvectors that can be used to discriminate between the classes. Figure 8.29 shows two classes (i.e., $M_c = 2$), marked by red dots and blue crosses. For each class, the underlying (but unknown) distribution of x_1 is shown by the dotted curve. As shown in the figure, LDA projects the original two-dimensional features onto a one-dimensional line (or linear discriminant), and compared to the original distributions, the projected distributions show better separation between the classes.

Back-Propagation Neural Networks

Among different types of supervised artificial neural network techniques available, the back-propagation neural network (BNN) is the most commonly used one. A BNN is a supervised learning technique with a basic neural network structure that has three types of layers: an input layer, an output layer, and hidden layers [36–40]. One of the unique features of a BNN is that the errors calculated from the output layer are back-propagated to train the hidden layers of the network. The size of the input layer is determined by the dimensionality of the diagnostics problem, while the size of the output layer changes based on the number of different classes of the

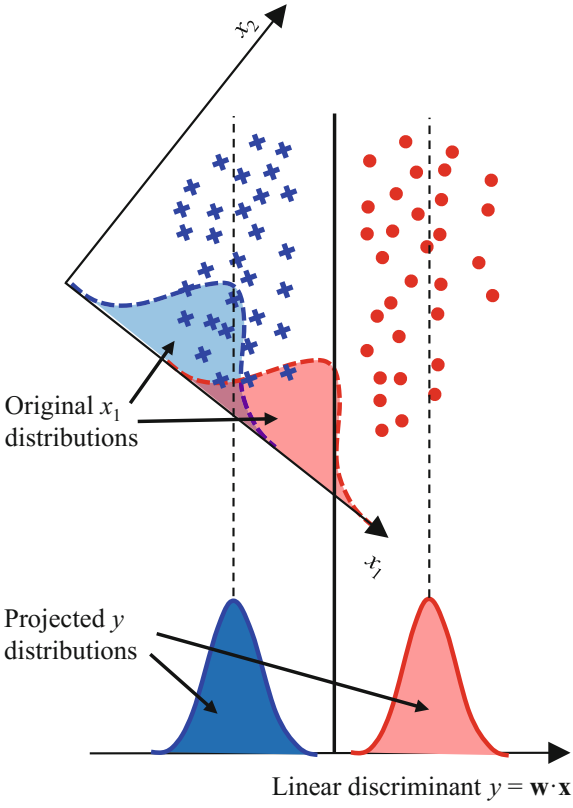


Fig. 8.29 Schematic of a two-class LDA

diagnostics problem. The number of hidden layers and number of neurons vary depending on the complexity of the problem. The model input is fed through the input layer of the network and is connected to hidden layers by synaptic weights. The number of hidden layers between the input and output layers can vary based on the complexity of the problem. Each network layer is connected to the next layer through synaptic weights in a hierarchical form. The training of the neural network aims at learning the relationship between the input layer and the output layer by adjusting the weights and bias values of each neuron in the network for each training pattern. The BNN model is trained by optimizing the synaptic weights and biases of all neurons until the maximum number of epochs is reached. The number of epochs is defined as the number of times a training algorithm uses the entire training data set. The trained BNN diagnostics model provides classification classes as an outcome, when sensor data is provided as an input to the model.

Support Vector Machine

In addition to network-based learning techniques like BNN, kernel-based machine learning techniques can also be used as member algorithms for health diagnostics. Support vector machine (SVM) is one of the most popular kernel-based machine learning techniques for classification. The following section briefly introduces SVM for classification.

With the organized input data $\{(\mathbf{x}_1, c_1), (\mathbf{x}_2, c_2), \ldots, (\mathbf{x}_m, c_m)\}$, SVM constructs the optimal separating hyper-plane that maximizes the margin between the separating hyper-plane and the data [37, 41–52]. Without any loss of generality, consider a two-class case for which the optimal separating hyper-plane and the maximum margin are shown in Fig. 8.30. For the linearly separable, two-class SVM shown in Fig. 8.30, the optimal hyper-plane separating the data can be expressed as

$$y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}} \cdot \mathbf{x} + b = 0 \tag{8.10}$$

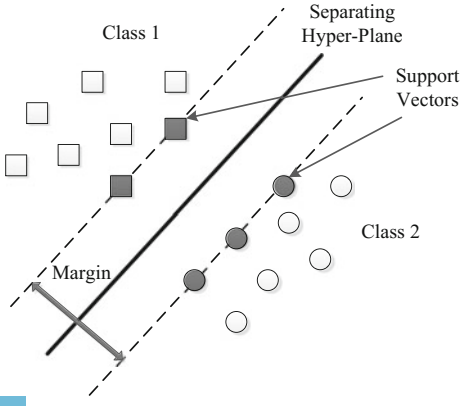


Fig. 8.30 Schematic of a two-class SVM

where $\mathbf{w}$ is a normal vector that is perpendicular to the hyper-plane and b is the offset of the hyper-plane. The parameter $b/\|\mathbf{w}\|$ determines the offset of the hyper-plane from the origin along the normal vector $\mathbf{w}$. The learning of SVM optimizes $\mathbf{w}$ and b in order to maximize the margin (or distance) between the parallel hyper-planes that are as far apart as possible while still separating the data, as shown in Fig. 8.30.

The optimization problem will eventually yield a set of optimized $\mathbf{w}$ and b that define different classification margins [42]. The optimization problem and the corresponding hyper-plane constraint for non-linear separable classes can be formulated as

$$\begin{aligned} &\text{minimize} && \tfrac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C\sum_{i=1}^{m}\xi_i \\ &\text{subject to} && y_i(\mathbf{w}^{\mathrm{T}}\cdot\mathbf{x}_i + b) \geq 1 - \xi_i \\ & && \xi_i \geq 0,\ i = 1, 2, \ldots, m \end{aligned} \tag{8.11}$$

where the regularization parameter C specifies the error penalty and ξ_i is a slack variable defining the error. If the Lagrangian multipliers are introduced, the optimization problem in Eq. (8.11) is transformed to a dual-quadratic optimization problem and expressed as

$$\begin{aligned} &\text{minimize} && L_D = \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \mathbf{x}_i\cdot\mathbf{x}_j \\ &\text{subject to} && \sum_{i=1}^{m}\alpha_i y_i = 0 \end{aligned} \tag{8.12}$$

After solving the optimization problem shown above, the solution of $\mathbf{w}$ can be expressed as

$$\mathbf{w} = \sum_{i=1}^{m}\alpha_i y_i \mathbf{x}_i \tag{8.13}$$

During the test phase, we determine on which side of the separating hyper-plane a test instance $\mathbf{x}$ lies and assign the corresponding class. The decision function can be expressed mathematically as $\mathrm{sgn}(\mathbf{w}^{\mathrm{T}}\cdot\mathbf{x} + b)$. Thus, the diagnostic SVM results provide different classification classes as a solution when a set of preprocessed sensor data is provided as an input.

Unsupervised Fault Classification

Mahalanobis Distance

Unsupervised statistical inference can be used for classifying health-relevant input features into different HSs based on their relative statistical distances. The Mahalanobis distance classifier is one of these classification techniques. In statistics, the MD is a distance measure based on the correlations between variables, by which

different patterns can be identified and analyzed. The MD gauges the similarity of an unknown sample set to a known one. Unlike the Euclidean distance method, the MD considers the correlations of the data set and is scale-invariant. The MD measure shows the degree of dissimilarity between the measured data point $\mathbf{x}_f$ and a reference training set (μ) with the covariance matrix S, as shown in Eq. (8.14).

$$D(\mathbf{x}_f) = \sqrt{(\mathbf{x}_f - \boldsymbol{\mu}_f)^T \mathbf{S}_f^{-1} (\mathbf{x}_f - \boldsymbol{\mu}_f)} \tag{8.14}$$

where $\mathbf{x}_f = (x_1, x_2, \ldots, x_F)^{\mathrm{T}}$ is a F-dimensional data vector, and $\boldsymbol{\mu}$ and $\mathbf{S}$ are respectively the mean vector and covariance matrix of the reference training data set. The MD is often used to detect outliers within multi-dimensional input samples, especially in the development of linear regression models. Due to its straightforwardness and ease of implementation, the MD model has been used for health diagnostics. MD health diagnostics considers the correlation between different input variables and determines the system HS based on the minimum MD values of the testing sample, compared to training samples from different HSs. Note that, in Sect. 8.2.1, MD is employed to classify different HSs for the development of sensor networks for health monitoring.

Self-organizing Maps

The methodologies discussed above were different machine learning processes where the target HS classes are known. If the different health conditions, and their functional relationships with the system input parameters, are not clearly known, possible health conditions of the system can then be determined using an unsupervised learning process that segregates the data based on the possible health conditions. The self-organizing map (SOM) is a type of artificial neural network that is trained using unsupervised learning to produce a two-dimensional discretized representation of the input space of the training samples. The SOM uses a neighborhood function to preserve the topological properties of the input space and determine the closest unit distance to the input vector [17]; this is then used to construct class boundaries graphically on a two-dimensional map. The SOM training utilizes competitive learning. When a training example is fed to the SOM, its Euclidean distance to all weight vectors is computed and the neuron with the weight vector most similar to the input vector $\mathbf{x}$ will be identified as the best matching unit (BMU). The weights of the BMU and neurons close to it in the SOM lattice are adjusted towards the input vector. Moreover, the magnitude of the change decreases with time and distance from the BMU. The weight vectors of the BMU and its topological neighbors are fine-tuned to move closer to the input vector space [36]. The learning rule for updating a weight vector $\mathbf{w}$ can be expressed as:

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + \alpha(t)\, h(n_{BMU}, n_i, t)(\mathbf{x} - \mathbf{w}_i(t)) \tag{8.15}$$

where $\mathbf{w}_i(t+1)$ is the updated weight vector, $\mathbf{w}_i(t)$ is the weight vector from the previous iteration, $\alpha(t)$ is the monotonically decreasing learning coefficient with $0 < \alpha < 1$, and $h(n_{BMU}, n_i, t)$ is the neighborhood function that decreases monotonically with an increase in the distance between the BMU n_{BMU} and the neuron n_i in the lattice. The Gaussian function is a common choice for the neighborhood function. Regardless of the function form, the $h(n_{BMU}, n_i, t)$ decreases over the training time. During the training process, one sample pattern is chosen from the input data X arbitrarily, and the distance between the sample point and the initial weight vector of the SOM is determined using the distance measure. Thus, through the learning, the input data are transformed into different HS clusters, and the overlapping of the different clusters is determined to be the misclassification.

Health Index

Successful implementation of health prognostics (see Sect. 8.4) often requires the derivation of a single health measure that quantifies the health condition of an operating system. This health measure is called a health index. In general, health indices can be categorized into two types: (i) a Physics Health Index (PHI) or (ii) a Virtual Health Index (VHI).

Physics Health Index (PHI)

A PHI uses a dominant physical signal as a direct health measure and is thus applicable only if sensor signals are directly related to physics-of-failure. In the literature, most engineering applications of health prognostics are based on various PHIs, such as the battery impedance [47], the magnitude of the vibration signal [48], and the radio frequency (RF) impedance [49]. However, the application of a PHI is limited to cases where sensor signals directly related to physics-of-failure are available. Mapping of a multitude of heterogeneous sensor signals to a dominant physical signal is getting more and more difficult with the growing complexity of engineered systems and sensor networks.

Virtual Health Index (VHI)

A VHI is applicable even if sensor signals are not directly related to system physics-of-failure. VHIs have a potential to overcome the limitation of PHIs described above. Multi-dimensional sensor signals can be transformed into a one-dimensional VHI using advanced data processing techniques, such as weighted averaging methods [43], the Mahalanobis distance [44], flux-based methods [45], or a linear data transformation method [46]. Let's consider the linear data transformation method. Suppose there are two multi-dimensional sensor data sets that represent the *system failed* and *system healthy* states, $\mathbf{Q}_0$ of $M_0 \times D$ matrix and $\mathbf{Q}_1$ of $M_1 \times D$ matrix, respectively. M_0 and M_1 are the data sizes for system failed and system healthy states, respectively, and D is the dimension of each data set. With these two data matrices, a transformation matrix $\mathbf{T}$ can be obtained to transform the multi-dimensional sensor signals into a one-dimensional VHI as

$$\mathbf{T} = \left(\mathbf{Q}^T\mathbf{Q}\right)^{-1}\mathbf{Q}^T\mathbf{S}_{\text{off}} \tag{8.16}$$

where $\mathbf{Q} = [\mathbf{Q}_0;\ \mathbf{Q}_1]$, $\mathbf{S}_{\text{off}} = [\mathbf{S}_0,\ \mathbf{S}_1]^{\text{T}}$, $\mathbf{S}_0$ is a $1 \times M_0$ zero vector and $\mathbf{S}_1$ is a $1 \times M_1$ unity vector. This transformation matrix $\mathbf{T}$ can transform any multi-dimensional signals from the offline learning or online prediction process to the normalized VHI as $\mathbf{H} = \mathbf{Q}_{\text{off}} \cdot \mathbf{T}$ or $\mathbf{H} = \mathbf{Q}_{\text{on}} \cdot \mathbf{T}$, where $\mathbf{Q}_{\text{off}}$ and $\mathbf{Q}_{\text{on}}$ are the offline and online multi-dimensional sensor data sets, respectively. If we assume the data sizes for $\mathbf{Q}_{\text{off}}$ and $\mathbf{Q}_{\text{on}}$ are, respectively, M_{on} and M_{off} (i.e., $\mathbf{Q}_{\text{off}}$ of $M_{\text{off}} \times D$ matrix and $\mathbf{Q}_{\text{on}}$ of $M_{\text{on}} \times D$ matrix), $\mathbf{H}$ will be a column vector of the size M_{off} or M_{on}. The VHI can also be denoted as $h(t_i)$ for $i = 1, \ldots, M$ (for the offline case) or for $i = 1, \ldots, M_{\text{on}}$ (for the online case), varying approximately between 0 and 1.

8.4 The Health Prognostics Function

Upon the detection and classification of a fault via the health reasoning function, the health prognostics function predicts the time remaining before the fault progresses to an unacceptable level, in other words, the remaining useful life (RUL). Figure 8.31 shows a typical paradigm of the health prognostics function, which first utilizes the sensor signal to produce the system degradation signal through signal processing and then leverages the degradation signal to perform diagnostics of the system's current health condition and further predict the system's RUL and reliability.

In general, two categories of approaches have been developed that enable continuous updating of system degradation and RUL distribution: (i) model-based approaches, and (ii) data-driven approaches. These two approaches are graphically compared in Fig. 8.32. The application of general, model-based prognostic approaches relies on the understanding of system physics-of-failure (PoF) and underlying system degradation models. The basic idea is to identify the parameters of the PoF-based degradation model in the online process. As practical engineered systems generally consist of multiple components with multiple failure modes, understanding all potential physics-of-failure and their interactions in a complex system is almost impossible. In such cases, the data-driven approaches for system

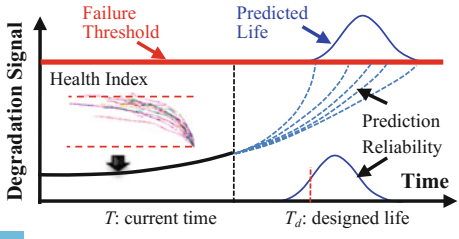


Fig. 8.31 Health prognostics for unit-wise reliability prediction

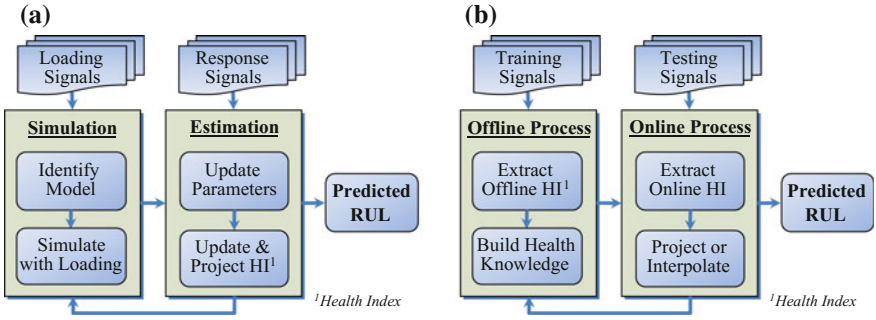


Fig. 8.32 Model-based (**a**) and data-driven (**b**) prognostics

health prognostics are more desirable; these data-driven approaches are mainly based on massive sensor data with a lessened requirement for knowledge of inherent system failure mechanisms. Data-driven prognostic approaches generally require sensor feature extraction and statistical pattern recognition for the offline training process, and interpolation, extrapolation, or machine learning for online life prediction.

8.4.1 Model-Based Prognostics

Model-based prognostic approaches use PoF-based models to represent the degradation behavior of a system for RUL prediction. PoF-based models have been investigated in the literature to capture various degradation phenomena in engineered systems, such as the side reaction in lithium-ion batteries [53, 54] and damage-based crack propagation in wire bond interconnects in power electronics [55, 56]. Sensor data contain rich information about system degradation behavior, and model-based prognostics incorporates new sensor information to update PoF-based models. Among the various approaches available to incorporate these evolving sensor data, Bayesian updating is the most widely used. Existing Bayesian updating approaches can be classified into two categories: simulation approaches [57–65] and analytical approaches [66–68]. Commonly used simulation approaches include iterative Markov Chain Monte Carlo (MCMC) methods (e.g., Metropolis-Hastings [62, 63] and Gibbs Sampling [64, 65]), and non-iterative methods (e.g., importance sampling [57–59] and rejection sampling [60, 61]). In the following sections, an iterative MCMC method and a non-iterative importance sampling method will be described in the context of model-based prognostics.

Iterative Bayesian Updating with Markov Chain Monte Carlo (MCMC)

Figure 8.33 illustrates the use of a Markov-Chain Monte Carlo (MCMC) method for non-conjugate Bayesian updating of a degradation model and RUL distribution. In this example, the system degradation model takes the following exponential form

$$S(t_i) = S_0 + \delta \cdot \exp\left(\alpha t_i^2 + \beta t_i + \varepsilon(t_i) - \frac{\sigma^2}{2}\right) \tag{8.17}$$

where $S(t_i)$ represents the degradation signal at time t_i; S_0 is a known constant; δ, α, and β are stochastic model parameters representing the uncertainty of generator operating conditions, and ε is a random error term modeling possible sensor noise that follows a zero-mean Gaussian distribution with standard deviation σ.

Non-iterative Bayesian Updating with Particle Filter

To make the discussion more concrete, consider a dynamic nonlinear discrete-time system described by a state-space model. In this context, a simplified state-space model is defined as

$$\begin{aligned} &\text{Transition:} && \mathbf{x}_i = f(\mathbf{x}_{i-1}) + \mathbf{u}_i \\ &\text{Measurement:} && \mathbf{y}_i = g(\mathbf{x}_i) + \mathbf{v}_i \end{aligned} \tag{8.18}$$

where $\mathbf{x}_i$ is the vector of (hidden) system states at time $t_i = i{\cdot}\Delta t$, Δt is a fixed time step between two adjacent measurement points, and i is the index of the measurement time step, respectively; $\mathbf{y}_i$ is the vector of system observations (or measurements); and $\mathbf{u}_i$ is the vector of process noise for the states; $\mathbf{v}_i$ is the vector of measurement noise; and $f(\cdot)$ and $g(\cdot)$ are the state transition and measurement functions, respectively. With the system defined, we aim to infer the system states $\mathbf{x}$ from the noisy observations $\mathbf{y}$.

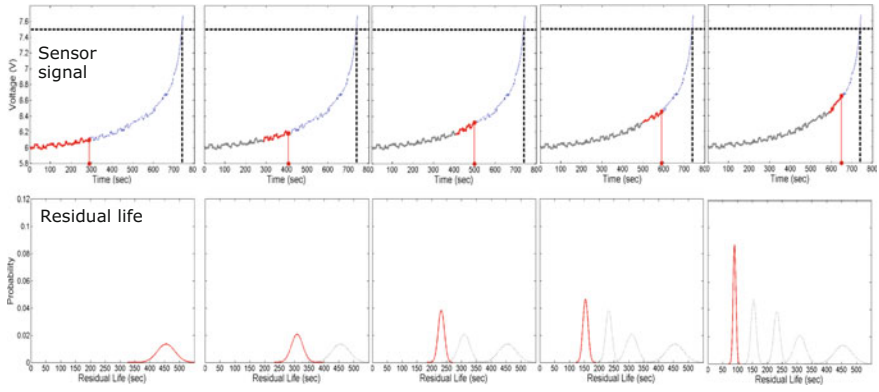


Fig. 8.33 Updating of a degradation model and RUL distribution

Fundamentals of Particle Filter

In a Bayesian framework, the posterior probability distribution of the states given the past observations, $p(\mathbf{x}_k|\mathbf{y}_{1:k})$, constitutes a statistical solution to the inference problem described in Eq. (8.18). Recursive Bayesian filtering enables a continuous update of the posterior distribution with new observations. One way to implement recursive Bayesian filtering is by using Kalman filtering, where all the variables in the dynamic system are assumed to follow Gaussian distributions whose statistical information can be fully described by the means and covariance matrix. The most widely used methods for Kalman filtering include Kalman filter [69], extended Kalman filter [70, 71], and unscented Kalman filter [72]. If the Gaussian assumption does not hold, or the state transition or measurement function is not explicitly known, simulation-based methods, such as particle filters (or the SMC methods) [73, 74] are more suitable for state estimation. Unlike Kalman filters, particle filters do not rely on the Gaussian assumption. Instead, they approximate the posterior distribution of the states based on a set of particles and their associated weights, both of which are continuously updated as new observations arrive, expressed as

$$p(\mathbf{x}_i|\mathbf{y}_{1:i}) \approx \sum_{i=1}^{N_P} w_i^j \delta\left(\mathbf{x}_i - \mathbf{x}_i^j\right) \tag{8.19}$$

where $\left\{\mathbf{x}_i^j\right\}_{j=1}^{N_P}$ and $\left\{w_i^j\right\}_{j=1}^{N_P}$ are the particles and weights estimated at the ith measurement time step, respectively; N_P is the number of particles; and δ is the Dirac delta function. The standard particle filter algorithm follows a standard procedure of sequential importance sampling and resampling (SISR) to recursively update the particles and their associated weights [73]:

(1) Initialization ($i = 0$)
For $j = 1, 2, \ldots, N_P$, randomly draw state samples $\mathbf{x}_0{}^j$ from the prior distribution $p(\mathbf{x}_0)$.
(2) For $i = 1, 2, \ldots$

(a) Importance Sampling
For $j = 1, 2, \ldots, N_P$, randomly draw samples from the proposed importance density $\mathbf{x}_i^j \sim q\left(\mathbf{x}_i|\mathbf{x}_{0:i-1}^j, \mathbf{y}_{1:i}\right)$. The standard SISR particle filter employs the so-called transmission prior distribution $q\left(\mathbf{x}_i|\mathbf{x}_{0:i-1}^j, \mathbf{y}_{1:i}\right) = p\left(\mathbf{x}_i|\mathbf{x}_{i-1}^j\right)$.
For $j = 1, 2, \ldots, N_P$, evaluate the importance weights

$$w_i^j = w_{i-1}^j \frac{p\left(\mathbf{y}_i|\mathbf{x}_i^j\right)p\left(\mathbf{x}_i^j|\mathbf{x}_{i-1}^j\right)}{q\left(\mathbf{x}_i|\mathbf{x}_{0:i-1}^j, \mathbf{y}_{1:i}\right)} \tag{8.20}$$

For $j = 1, 2, \ldots, N_P$, normalize the importance weights

$$\tilde{w}_i^j = w_i^j \left[\sum_{j=1}^{N_P} w_i^j \right]^{-1} \tag{8.21}$$

(b) Selection (Resampling)
Multiply/suppress samples $\{\mathbf{x}_i^j\}_{j=1}^{N_P}$ with respect to high/low importance weights to obtain N_P random samples $\{\mathbf{x}_i^j\}_{j=1}^{N_P}$ with equal weights N_P^{-1}.

(c) Posterior Distribution Approximation using Eq. (8.19).

Application of Particle Filter to RUL Prediction

Next, consider the application of a particle filter to predict the RUL of an engineered system. Take a lithium-ion (Li-ion) rechargeable battery as an example [75]. Capacity fade over time/cycling is the primary degradation mode of a Li-ion battery cell. Given the readily available measurements (i.e., voltage, current, and temperature) from a cell operating under a typical use condition and a discrete-time state space model that describes the capacity fade behavior of the cell, the task of Li-ion battery prognostics is to estimate the capacity of the cell at every charge/discharge cycle and predict its RUL, i.e., how long the cell is expected to operate before its capacity falls below an unacceptable level (or a failure threshold). Model-based prognostics of a Li-ion battery is illustrated in Fig. 8.34, where a battery cell has been cycled (charged/discharged) for 300 times and we want to predict the RUL of this cell. This online task of RUL prediction can be accomplished in two sequential steps: (1) capacity estimation, which infers the capacity of the cell from the voltage, current, and temperature at every charge/discharge cycle; and (2) model updating and extrapolation, which updates an empirical capacity fade model with the capacity measurements (i.e., the capacity estimates from Step 1) and extrapolates the updated model to predict future capacity fade and RUL. The rest of this section presents the application of particle filter to accomplish the second step of this online task.

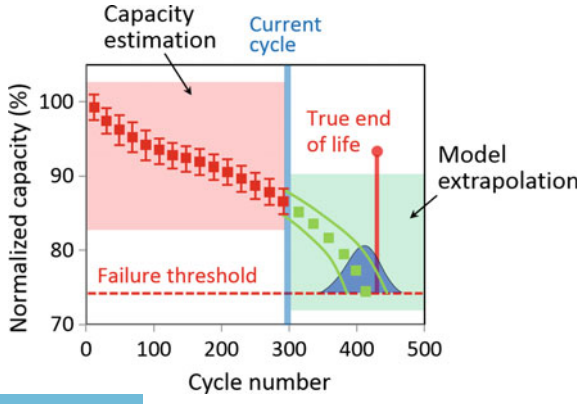


Fig. 8.34 Schematic of model-based prognostics of Li-ion battery (for simplicity, the updating of capacity fade model at the current cycle is not shown)

Assume the underlying capacity fade model can be expressed as a hybrid of linear and exponential functions

$$c_i = \frac{C_i}{C_0} = 1 - \alpha[1 - \exp(-\lambda i)] - \beta i \tag{8.22}$$

where C_i is the capacity of the cell at the ith cycle, C_0 is the initial capacity, α is the coefficient of the exponential component of capacity fade, λ is the exponential capacity fade rate, β is the coefficient of the linear component of capacity fade, and c_i is the normalized capacity at the ith cycle. Researchers have reported that the exponential function captures the active material loss [76] and the hybrid of the linear and exponential functions was reported to provide a good fit to three years' cycling data [77]. Here, we treat the normalized capacity c and capacity fade rates α, λ, and β as the state variables, i.e., $\mathbf{x} \equiv [c, \alpha, \lambda, \beta]^{\mathrm{T}}$. The system transition and measurement functions can then be written as [75].

$$\begin{aligned} &\text{Transition:} && c_i = 1 - \alpha_{i-1}[1 - \exp(-\lambda_{i-1} i)] - \beta_{i-1} i + u_i, \\ &&& \alpha_i = \alpha_{i-1} + r_{1,i},\ \lambda_i = \lambda_{i-1} + r_{2,i},\ \beta_i = \beta_{i-1} + r_{3,i} \\ &\text{Measurement:} && y_i = c_i + v_i \end{aligned} \tag{8.23}$$

Here, y_i is the capacity measurement (or estimate) at the ith cycle; and u, r_1, r_2, r_3, and v are the Gaussian noise variables with zero means.

To perform the RUL prediction, we learn and track the capacity fade behavior of the cell at every charge/discharge cycle. The learning and tracking are done by updating the parameters α, λ, and β of the capacity fade model in Eq. (8.22) with the capacity measurement via the use of particle filter. After the resampling step of particle filter at the ith cycle, the posterior probability distribution of the normalized capacity is approximated as

$$p(c_i|y_{1:i}) \approx \frac{1}{N_P} \sum_{i=1}^{N_P} \delta\left(c_i - c_i^j\right) \tag{8.24}$$

where c_i^j is the jth resampled particle of c_i. The normalized capacity l cycles in the future can be predicted by extrapolating the capacity fade model with the updated parameters, expressed as

$$p(c_{i+l}|y_{1:i}) \approx \frac{1}{N_P} \sum_{i=1}^{N_P} \delta\left(c_{i+l} - c_{i+l}^j\right) \tag{8.25}$$

where

$$c_{i+l}^j = 1 - \alpha_i^j\left[1 - \exp\left(-\lambda_i^j (i+l)\right)\right] - \beta_i^j (i+l) \tag{8.26}$$

Here, we define the normalized capacity at 78.5% of the failure threshold. Then the RUL (in cycles) can be obtained for each particle as the number of cycles between the current cycle i and the end-of-life (EOL) cycle

$$L_i^j = \text{root}\left[\alpha_i^j\left[1 - \exp\left(-\lambda_i^j i\right)\right] + \beta_i^j i = 0.215\right] - i \tag{8.27}$$

Finally, the RUL distribution can be built based on these particles, expressed as

$$p(L_i|y_{1:i}) \approx \frac{1}{N_P}\sum_{i=1}^{N_P}\delta\left(L_i - L_i^j\right) \tag{8.28}$$

This completes the derivation of the RUL distribution.

It can be observed from this application that as a battery cell degrades every charge/discharge cycle, we feed in the capacity measurement for that cycle, which then allows us to "on the fly" learn how the battery is degrading by updating an underlying capacity fade model. Once the capacity fade model is updated, we can then extrapolate the model to the failure threshold for calculating the RUL. This in fact can be viewed as the general procedure of model-based prognostics.

8.4.2 Data-Driven Prognostics

Data-driven prognostic techniques utilize monitored operational data related to system health. The major advantage of data-driven approaches is that they can be deployed more quickly and often at a lower cost, as compared to other approaches. In addition, data-driven techniques can provide system-wide coverage. The principal disadvantage is that data-driven approaches may have wider confidence intervals than other approaches and they require a substantial amount of data for training. Three approaches can be used for online RUL prediction in data-driven approaches: interpolation, extrapolation, and machine learning.

Interpolation-Based Approach

The basic idea of this data-driven prognostic approach is to compare the similarity between the partial degradation data ($\mathbf{h}_r$) from an online[1] system unit and the offline[2] health degradation curve ($\mathbf{h}_p$) and to determine a time-scale initial health condition (T_0) through an optimum fitting [46, 78, 79]. As shown in Fig. 8.35, the optimum fitting process basically moves the online health index data (or normalized degradation data) $\mathbf{h}_r$ along the time axis of a predictive health degradation curve $\mathbf{h}_p$ to find the optimum time-scale initial health state (T_0) that best matches $\mathbf{h}_r$ with $\mathbf{h}_p$ with respect to the sum of squared differences (*SSD*).

[1]The term "online" indicates a state where a (testing) system unit is operating in the field and its RUL is unknown and needs to be predicted.

[2]The term "offline" indicates a state where a (training) system unit is operating in the lab or field and often runs to failure (thus, its RUL at any time is known) prior to the operation of any system units online.

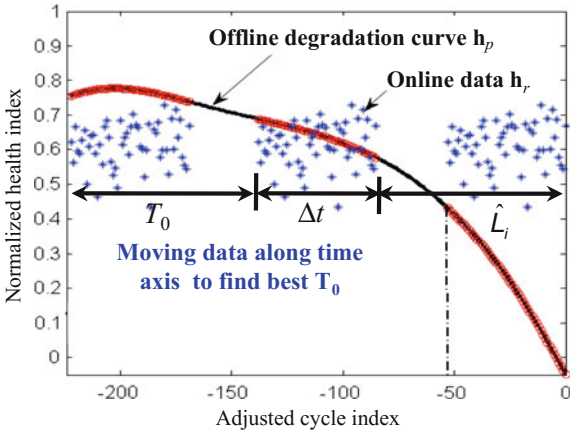


Fig. 8.35 Similarity-based interpolation for RUL prediction

This optimum fitting can be formulated as

$$\begin{aligned} &\text{Minimize} \quad SSD = \sum_{j=1}^{M_s} \left(h_r(t_j) - h_p(t_j + T_0)\right)^2 \\ &\text{subject to} \quad T_0 \in [0, L - \Delta t] \end{aligned} \tag{8.29}$$

where $h_r(t_j)$ and $h_p(t_j)$ are the online and offline health index data at t_j, respectively; M_s is the length of the online health index data; T_0 is the time-scale initial health condition; Δt is the time span ($= t_{Ms} - t_1$) of the online health index data; and L is the time span of a predictive health degradation curve, i.e., the life span of an offline system unit. Once T_0 is determined from the optimization (Eq. 8.29), the projected RUL of an online system unit based on a given predictive health degradation curve can be calculated as

$$RUL = L - \Delta t - T_0 \tag{8.30}$$

Repeating the optimum fitting process on K predictive health degradation curves from K different offline system units gives K RUL estimates (L_i^P for $i = 1, \ldots, K$). Then, the predictive RUL is a weighted sum in terms of different projected RULs (L_i for $i = 1, \ldots, K$) of an online unit as

$$L = \frac{1}{W} \sum_{i=1}^{K} (W_i \cdot L_i) \quad \text{where} \quad W = \sum_{i=1}^{K} W_i \tag{8.31}$$

where L_i is the projected RUL on the ith offline predictive health degradation curve and W_i is the ith similarity weight. A similarity weight W_i can be defined as the inverse of the corresponding SSD_i, i.e., $W_i = (SSD_i)^{-1}$. This definition ensures that a greater similarity gives a greater weight.

Extrapolation-Based Approach

Unlike the interpolation-based approach, the extrapolation-based approach employs the training data set not for comparison with the testing data set but rather for obtaining prior distributions of the degradation model parameters. The testing data set is then used to update these prior distributions. An RUL estimate can be obtained by extrapolating the updated degradation model to a predefined failure threshold (see Fig. 8.36). Bayesian linear regression, Kalman filter, or particle filter can be employed for construction and updating of the degradation model.

Machine Learning-Based Approach

In contrast to the interpolation- or extrapolation-based approaches, the machine learning-based approach does not involve any visible manipulation of the offline and online data; rather, it requires the training of a prognostics model using the offline data. One such model is the recurrent neural network (RNN) model, which is capable of learning nonlinear dynamic temporal behavior due to the use of an internal state and feedback. A first-order, simple RNN is an example of multi-layer perceptron (MLP) with feedback connections (see Fig. 8.37). The network is composed of four layers, namely, the input layer I, recurrent layer R, context layer C, and output layer O. Units of the input layer and the recurrent layer are fully connected through the weights $\mathbf{W}^{RI}$, while units of the recurrent layer and output layer are fully connected through the weights $\mathbf{W}^{OR}$. Through the recurrent weights $\mathbf{W}^{RC}$, the time delay connections link current recurrent units $\mathbf{R}^{(t)}$ with the context units $\mathbf{C}^{(t)}$ holding recurrent units $\mathbf{R}^{(t-1)}$ in the previous time step. The net input of the ith recurrent unit can be computed as

$$\tilde{R}_i^{(t)} = \sum_j W_{ij}^{RI} I_j^{(t)} + \sum_j W_{ij}^{RC} R_j^{(t-1)} \tag{8.32}$$

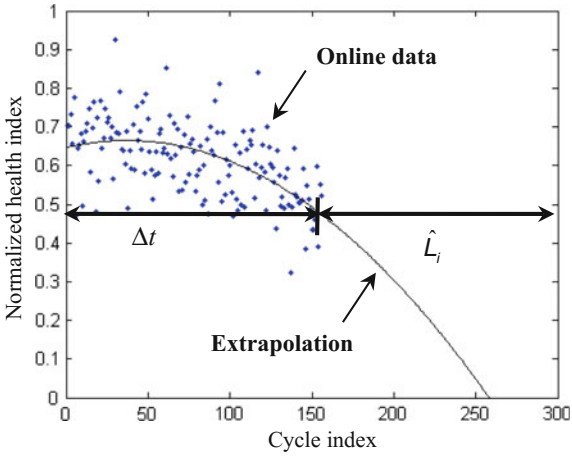


Fig. 8.36 Extrapolation-based RUL prediction

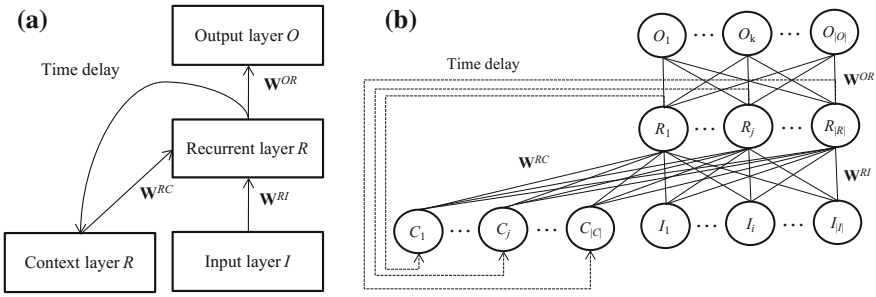


Fig. 8.37 Simplified (**a**) and more detailed representation (**b**) of Elman's simple RNN. Reprinted (adapted) with permission from Ref. [79]

Given the logistic sigmoid function as the activation function f, the output activity of the ith recurrent unit can then be computed as

$$R_i^{(t)} = f\left(\tilde{R}_i^{(t)}\right) = \left[1 + \exp\left(-\tilde{R}_i^{(t)}\right)\right]^{-1} \tag{8.33}$$

The net input and output activity of the ith output unit can be computed, respectively, as

$$\tilde{O}_i^{(t)} = \sum_j W_{ij}^{OR} R_j^{(t)} \tag{8.34}$$

and

$$O_i^{(t)} = f\left(\tilde{O}_i^{(t)}\right) = \left[1 + \exp\left(-\tilde{O}_i^{(t)}\right)\right]^{-1} \tag{8.35}$$

In health prognostics, the inputs to the RNN are the normalized sensor data set $\mathbf{Q}^N$ and the outputs are the RULs associated with the data set. The RNN training process calculates the gradients of network weights with respect to the network performance and updates the network weights in search of the optimum weights with the minimum prediction error.

Other machine learning techniques that can be used for health prognostics include artificial intelligence-based techniques, feed-forward neural networks, the decision tree method, support vector machine (SVM), relevance vector machine (RVM), k-nearest neighbor (KNN) regression, fuzzy logic, and others.

Example 8.5 NASA Aircraft Engine Prognostics
The objective of the problem in this example is to predict the number of remaining operational cycles before failure in the testing data set for a NASA aircraft engine. The prognostics framework consists of the following functions (see Fig. 8.38): (i) health sensing for sensor signal acquisition, (ii) health reasoning for health index construction (feature extraction), and (iii) a health prognostics function for interpolation-based RUL prediction. These functions will be explained in detail in the following subsections.

Health Sensing Function

The data set used for this example is the one provided by the 2008 IEEE PHM Challenge problem. This data set consists of multivariate time series signals that are collected from a NASA aircraft engine dynamic simulation process (see Fig. 8.39). Each time series signal comes from a different degradation instance of the dynamic simulation of the same engine system. The data for each cycle of each unit include the unit ID, cycle index, 3 values for an operational setting and 21 values for 21 sensor measurements. The sensor data were contaminated with measurement noise. In addition, different engine units start with different initial health conditions and manufacturing

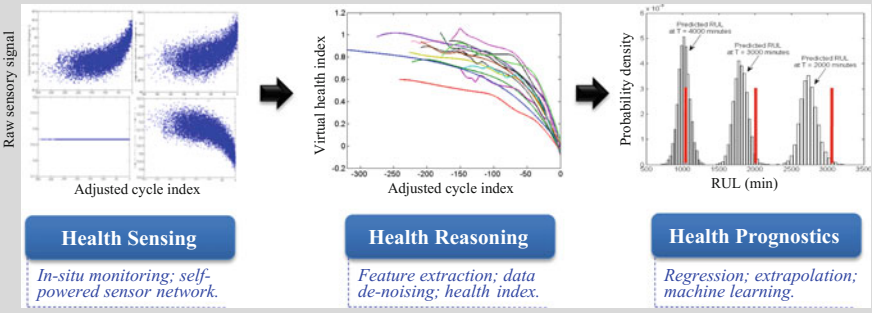


Fig. 8.38 The three main PHM functions for NASA aircraft engine prognostics

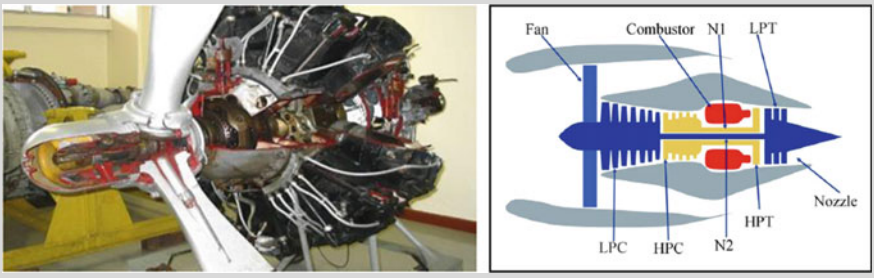


Fig. 8.39 NASA aircraft engine and C-MAPSS simulation model

variations that are unknown. The 21 sensor signals were obtained from six different operation regimes. The whole data set was divided into training and testing subsets, each of which consists of 218 engine units.

Health Reasoning Function

To account for the different initial degradation conditions, an adjusted cycle index is proposed as $C_{adj} = C - C_{f,}$ where C is the operational cycle of the training data for an engine unit and C_f is the cycle-to-failure of an engine unit. A cycle index 0 indicates engine unit failure; whereas, negative cycle indices are realized prior to failure. Among the 21 sensor signals, some signals contain no or little degradation information for engine units; other signals do contain degradation information. To improve the *RUL* prediction accuracy and efficiency, seven relevant signals (2, 3, 4, 7, 11, 12, and 15) were selected by screening all 21 sensor signals according to the degradation behaviors.

Based on the seven sensor signals, a normalized health index was constructed to represent the health degradation process of the engine. This normalization process is realized by using a linear transformation with the sensor data representing *system failure* and *system healthy* states. The dots in Fig. 8.40 represent the normalized health index data obtained from the training data set of an offline engine unit.

The randomness of the health index data is mainly due to the measurement noise from the signals. Thus, a stochastic regression technique, namely relevance vector machine (RVM) regression, can be used to model the VHI data in a stochastic manner. The RVM is a Bayesian representation of a generalized sparse linear model, which shares the same functional form as the

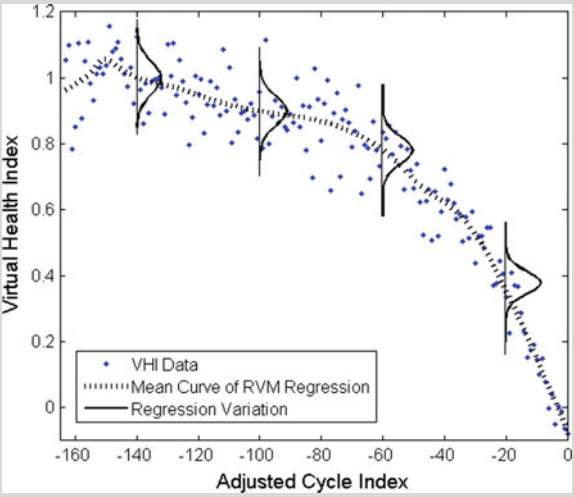


Fig. 8.40 VHI and the RVM regression

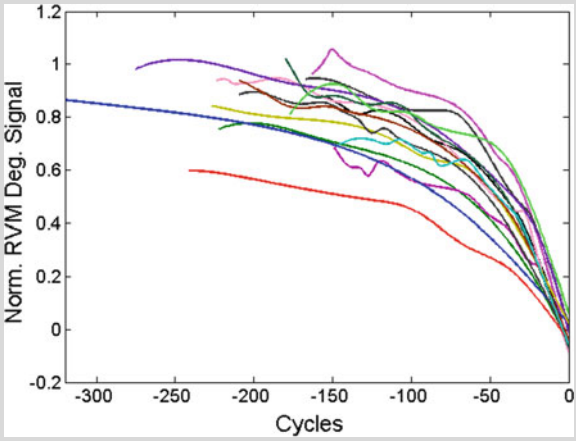


Fig. 8.41 Background degradation knowledge from the RVM

support vector machine (SVM). In this example, the linear spline kernel function was used as a basis function for the RVM. The RVM was used to build the predictive health degradation curves ($h_i^p(t)$, $i = 1, \ldots, 218$) for 218 offline engine units. The regression model gives both the mean and the variation of the predictive health degradation curve, as shown in Fig. 8.40. These predictive health degradation curves for the offline units altogether construct the background health knowledge, which characterizes the system degradation behavior. Later, this background knowledge can be used for modeling the predictive *RUL* distributions of online engine units. Degradation curves built for the offline units are exemplified in Fig. 8.41.

Health Prognostics Function

The online prediction process employed the testing data set obtained from 218 online system units. As explained earlier, the optimum fitting was employed to determine a time-scale initial health degradation state (T_0) with the training data set for an online engine unit, while minimizing the SSE between the online health data $h(t_j)$ and the predictive health degradation data $h^p(t_j)$, as shown in Fig. 8.42. It should be noted that the offline learning process generates different predictive health degradation curves from K identical offline units. Repeating this process provided different projected *RUL*s (RUL_i for $i = 1, \ldots, 218$) on different predictive health degradation curves. The projected *RUL*s can be used to predict the *RUL* of an online unit through a weighted-sum formulation.

From 218 offline engine units, the same number of the predictive health degradation curves and projected *RUL*s was obtained for each online engine unit. Likewise, the same number of similarity weights was sought for each online engine unit, based on the inverse of the SSE. A weighted-sum

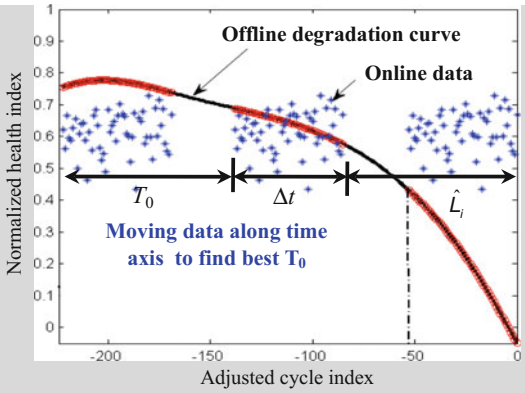


Fig. 8.42 Extrapolation-based RUL prediction

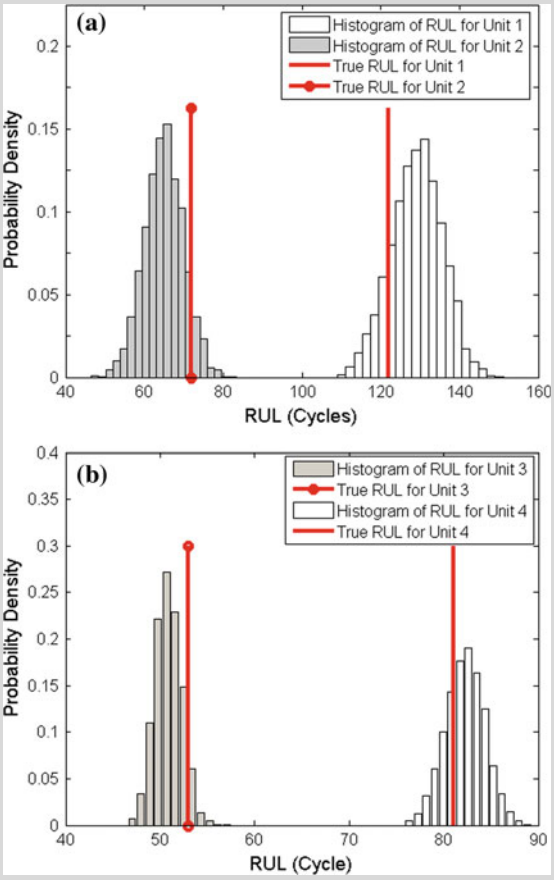


Fig. 8.43 Predicted *RUL* histograms with true RULs for **a** units 1 and 2, and **b** units 3 and 4

formulation was then used to predict the *RUL* for each online engine unit as a function of the projected *RUL*s, while considering the first 50 largest similarity weights. Note that $h_i^p(t_i)$ are stochastically modeled using RVM regression. Thus, the resulting similarity weights were modeled in a statistical manner, as was the *RUL* of the online unit. Using the mean and covariance matrices of the relevance vector coefficients for the RVM regression, the random samples of the coefficients result in the random samples of the similarity weights for the projected *RUL*s of the engine unit. The randomness of the similarity weights and projected *RUL*s is then propagated to the predictive *RUL* of the engine unit through the weighted-sum formulation. Figure 8.43 shows the *RUL* histogram and the true value with the testing data set for the first four online engine units.

8.4.3 *Uncertainty Management in Health Prognostics*

Uncertainty management is of great importance for health prognostics as it (i) identifies sources of uncertainty that significantly contribute to the uncertainty in the RUL prediction and (ii) provides decision-makers with statistical information about the predicted RULs. Uncertainty management in health prognostics has been discussed in various publications [78–81]. The three major uncertainty-related activities are quantification, propagation, and management [82]. These activities are detailed in the following subsections.

Uncertainty Quantification

This first activity aims to identify and characterize the various sources of uncertainty that may affect the RUL prediction. It is important that these sources of uncertainty be accounted for in models and algorithms for health prognostics. The sources of uncertainty often present in PHM applications include (i) sensor noise and bias, (ii) inadequacy of the sensor network in health sensing, (iii) signal filtering and resampling, (iv) data reduction (loss of information resulting from feature extraction and selection) in health reasoning, and (v) future usage uncertainty and algorithm inadequacy in health prognostics. Figure 8.44 shows various sources of uncertainty in each of the PHM functions. Often, uncertainty quantification characterizes each of these uncertainties using probability distributions and these quantified uncertainties facilitate subsequent activity, i.e., uncertainty propagation.

Uncertainty Propagation

The second activity, namely uncertainty propagation, accounts for all quantified uncertainties in health prognostics. Specifically, it propagates the uncertainties associated with the PHM functions through a set of models and algorithms to

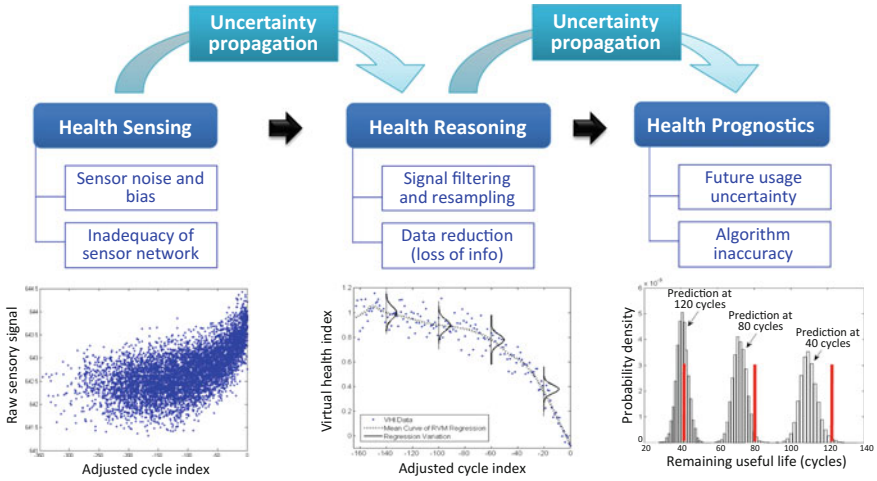


Fig. 8.44 Sources of uncertainty and uncertainty propagation in PHM

estimate the total uncertainties in future states and in the RUL. Figure 8.44 shows uncertainty propagation from the PHM function at one level to the one at the next level. An arbitrary assignment of the distribution type to RUL is often erroneous; thus, the true probability distribution of the RUL needs to be estimated through rigorous uncertainty propagation of the various sources of uncertainty through the models and algorithms. Such a distribution may not share similarity with any of the commonly used distribution types. It is important to understand that uncertainty propagation is significantly important and challenging in the context of prognostics, since the focus is on predicting the future unknown behavior of a system.

Uncertainty Management

The third activity is uncertainty management, and it has two primary focuses. First, it focuses on reducing the uncertainty in the predicted RUL and increasing the confidence in condition-based maintenance during real-time operation. This can be done by partitioning uncertainty to quantify the contributions of uncertainties from individual sources to the uncertainty in the predicted RUL. For example, if sensor noise and bias are identified to be significant contributors to the uncertainty in the RUL prediction, a better RUL prediction (with less uncertainty) can be achieved by improving the quality of the sensors. The second focus of uncertainty management is on addressing how uncertainty-related information can assist in making decisions about when and how to maintain the system. Since the uncertainty in the RUL prediction cannot be eliminated, it is important to take into account the uncertainty and make optimum decisions under acceptable risk.

8.5 PHM Demonstration: Prognostics of Electric Cooling Fan

This demonstration applies a data-driven prognostic approach, similarity-based interpolation (see Sect. 8.4.2), to the health prognostics of electronic cooling fan. Cooling fans are one of the most critical parts in most electronic products and in cooling towers of many chemical plants. This demonstration aims to illustrate data-driven prognostics with electric cooling fan units and has two essential elements:

(1) MATLAB .mat files (DuFan.mat, DuFan_test.mat, and RUL_True_Test.mat) with the vibration-based health index data of 20 training fan units (DuFan.mat) that have run to failure and 24 testing fan units (DuFan_test.mat) whose RULs are assumed to be unknown and need to be estimated and whose actual RULs are in fact known (RUL_True_Test.mat) and can be used as the ground truth for evaluation of prognostic accuracy;
(2) MATLAB.m files (SBI_FanDemo.m, Life_Pred_SBI_FanDemo.m, RVM_FanDemo.m, and Kernel_Linearspline_FanDemo.m) with the codes for predicting the RULs of the 24 testing fan units.

   (i) SBI_FanDemo.m: main code that contains all steps for data-driven prognostics with relevance vector machine (offline training) and similarity-based interpolation (online prediction).
   (ii) Life_Pred_SBI_FanDemo.m: supporting code that implements online prediction based on similarity-based interpolation.
   (iii) RVM_FanDemo.m: supporting code that contains a simple implementation of relevance vector machine regression (a more sophisticated and stable version of relevance vector machine can be found at the following web link: http://www.miketipping.com/downloads.htm).
   (iv) Kernel_Linearspline_FanDemo.m: supporting code that builds a linear spline kernel function for the implementation of relevance vector machine.

Arranging the two .mat files and four .m files in the same folder and running the main code SBI_FanDemo.m in MATLAB should allow this PHM demonstration to be executed. In what follows, the basics and MATLAB implementations of offline training and online prediction are briefly described.

Feature Extraction

This demonstration involves the use of the RMS of the vibration spectral responses at the first five resonance frequencies. It first extracts the RMS from the fan vibration signals as the relevant feature and then normalizes the feature to derive a health index for the electric cooling fan prognostics. Figure 8.45 shows the health index data of three fan units to demonstrate the degradation behavior of these units.

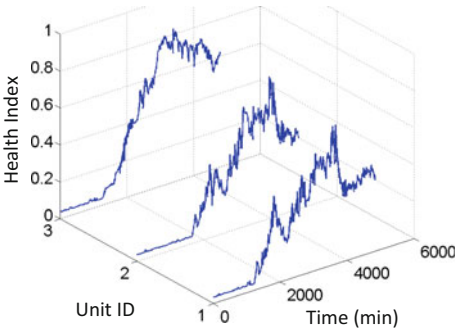


Fig. 8.45 Sample degradation signals from electric cooling fan testing. Reprinted (adapted) with permission from Ref. [79]

The vibration-based health index is highly random and non-monotonic, but gradually increases as the bearing in the fan degrades over time. For the electric cooling fan prognostics, the first 20 fan units are employed for the training dataset in the offline training (Step 1) process, while the rest are used to produce the testing dataset in the online prediction (Step 2) process.

In this demonstration, the vibration-based health index data of the 20 training and 24 testing fan units are pre-computed and stored in the MATLAB.mat files, DuFan.mat and DuFan_test.mat. Step 0.1 in the main code SBI_FanDemo loads these pre-computed training and testing data sets.

```
%% Step 0.1: Load training and testing data sets
load('DuFan.mat','DuFan')              % Load data 'DuFan' of 24 training
                                       % units that have run to failure.
                                       % Data from one fan unit consist of
                                       % time in min (Column 1) and health
                                       % index (Column 2). One cycle is
                                       % defined as 10 min. Health index is
                                       % defined as RMS of the vibration
                                       % spectral responses at the first
                                         five
                                       % resonance frequencies
load('DuFan_test.mat','DuFan_test') % Load data 'DuFan_test' of 24
                                         testing
                                       % units whose RULs are unknown
                                         and need
                                       % to be predicted. 'DuFan_test'
                                       % contains partial
                                         degradation data
                                       % from the testing units.
```

Health Prognostics

RUL prediction with similarity-based interpolation involves two sequential steps, offline training and online prediction. In offline training, we have the degradation data from the 20 offline units. An offline unit is tested to failure in the lab. So we have a complete degradation trajectory from beginning of life (BOL) to end of life (EOL). We overlay the degradation data from the 20 units on the same graph ("Offline data" in Fig. 8.46). The intent of offline training is to construct a fitted curve to represent the trajectory.

In this demonstration, machine learning-based regression (i.e., RVM regression) is used to fit a degradation curve to the data for each offline unit. By doing so, we transform these offline data into 20 degradation curves, as shown in Fig. 8.46 ("Background degradation curves"). These curves will be used to predict the RUL of an online unit. The lines of code for Step 1 in the MATLAB main code SBI_FanDemo.m implement this curve fitting on all 20 offline training units with RVM regression.

```
%% Step 1: Offline training (to build offline degradation curves)
nTrain = length(D_deg);
for ki = 1:nTrain
    fprintf('\n Currently running RVM regression on training unit %d
    \n',ki);

    %% Step 1.1: Extract training data from training unit ki
    x = D_deg{ki}(:,1);
    t = D_deg{ki}(:,2);
```

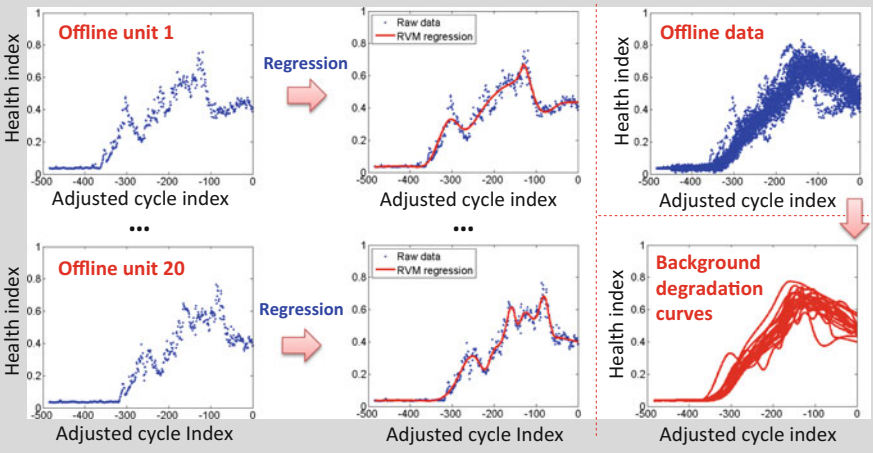


Fig. 8.46 Offline training (Step 1) for electric cooling fan prognostics

```
    %% Step 1.2: Train an RVM regression model with extracted Data
    [~,mou,~] = RVM_FanDemo(x,t);
    MUW{ki} = mou;
    clear x t afa mou cov
end
```

Unlike an offline unit, an online unit is operating in the field and has not failed yet. So we only get partial degradation data (see the blue dots in Fig. 8.47) from the online unit. Now, we have two pieces of information: (1) the degradation curves from the offline units; and (2) the partial degradation data from the online unit. The objective is to predict the RUL of this online unit.

Online RUL prediction using similarity-based interpolation is implemented in the MATLAB supporting code Life_Pred_SBI_FanDemo.m. In the implementation, an outer "for" loop is used to run RUL prediction on all testing units (e.g., 24 testing fan units in this demonstration) and an inner "for" loop is used to run RUL prediction of each testing unit based on all training units (e.g., 20 training fan units in this demonstration).

```
% Run an outer for loop to predict RULs for all testing units
for ku = 1:nTest
    fprintf('\n\n Currently making prediction for testing unit %d \n
\n',ku);
    % Run an inner for loop over all training units
    for ki = 1:nTrain
        ...
    end
        ...
end
```

Two steps are involved in RUL prediction of a testing unit. First, we predict the RUL of the online unit based on the 1st offline unit. The prediction involves the optimization process in Eq. (8.29). In this process, we move the online data along the time axis to find the best match with the degradation curve. After the optimum match is found, we will have a predicted life and a sum squared error of this match. Repeating this process for all the other offline units, we can get 20 life predictions. The inner "for" loop in the MATLAB supporting code Life_Pred_SBI_FanDemo.m implements this first step.

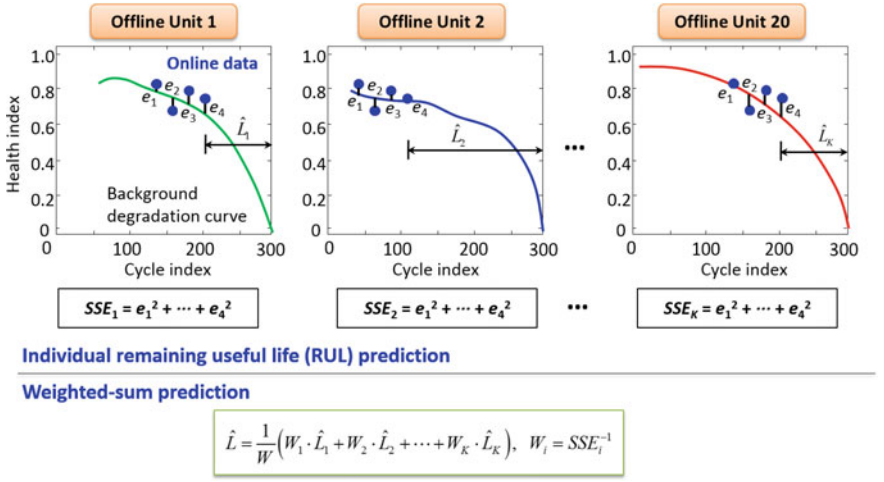


Fig. 8.47 Online prediction (Step 2) for electric cooling fan prognostics

```
for ki = 1:nTrain
 % Predict RUL of testing unit ku using data from training unit ki
 L1 = size(D_deg_test{ku},1); % Length of testing data
 L2 = size(D_deg{ki},1);      % Length of training data
 Wei_Rul = [];
 X = D_deg{ki}(:,1);          % Extract cycle index data of training
                              % unit ki
 W = MUW{ki};                 % Extract RVM model weights of training
                              % unit ki
 if L1 < L2
     ndif = L2-L1;
     % Use matrix operation to accelerate computation
     xpool = (1-L2):1:0;      % Define adjusted cycle index of
                              % training unit ki
     % Construct design matrix Phi and calcuate prediction by RVM
     % regression model
     Phi_pred = Kernel_Linearspline_FanDemo(xpool,X);
     T_pred = Phi_pred*W;
```

```
        % List all ranges of adjusted cycle index of testing unit ku
        IndexExp = repmat(1:1:L1,ndif+1,1) + repmat((0:1:ndif)',1,L1);
        T_predExp = T_pred(IndexExp);

        % Compute sum of squared differences (i.e., Weight) for each
        % range of adjusted cycle index of testing unit ku
        DataExp = repmat(D_deg_test{ku}(:,2)',ndif+1,1);
        Weight = sum((T_predExp-DataExp).^2,2);
        RUL = ndif + 1 - [1:1:ndif+1]';
        Wei_Rul = [Weight, RUL];
        clear T_predExp DataExp T_pred Fei_pred Weight RUL
    end

    if isempty(Wei_Rul)==0
        % Identify RUL that produces the best match between online data
        % (testing unit ku) and offline degradation curve (training
        % unit ki)
        L_Pred = min(Wei_Rul(:,1));
        Pred_Results(ki,:) = Wei_Rul(Wei_Rul(:,1)==L_Pred,:);
        clear Wei_Rul
    else
        Pred_Results(ki,:) = [-1,-1];
    end
end
RUL_Prediction{ku} = Pred_Results;
```

Next, we aggregate all the 20 predictions in a weighted-sum form (see Eq. 8.31). This gives the final predicted RUL. The weight is inversely proportional to the error of the match. That is, a larger weight will be assigned if the degradation curve of a certain offline unit shows better agreement with the online data. The lines of code right below the inner "for" loop in the MATLAB supporting code Life_Pred_SBI_FanDemo.m implement this second step.

```
    di_nn = find(RUL_Prediction{ku}(:,2) > -1);
    if isempty(di_nn)
```

```
    % If the number of cycles in the testing data is larger than that
    % in the the training data, set the predicted RUL to 0
    RUL_Mean(ku) = 0;
else
    % Aggregate all nTrain RUL predictions in a weighted-sum form
    RUL_Prediction_temp = RUL_Prediction{ku}(di_nn,:);
    [~,di] = sortrows(RUL_Prediction_temp,1);
    di = di(1:min(length(di),50));
    RUL_Mean(ku) = sum(RUL_Prediction_temp(di,1).^(-1).*....
        RUL_Prediction_temp(di,2))./....
        sum(RUL_Prediction_temp(di,1).^(-1));
end
clear Pred_Results RUL_Prediction_temp
```

RUL Prediction Results

Executing the MATLAB main code SBI_FanDemo.m produces the predicted RULs of all 24 testing fan units, which are plotted along with the true RULs in Fig. 8.48.

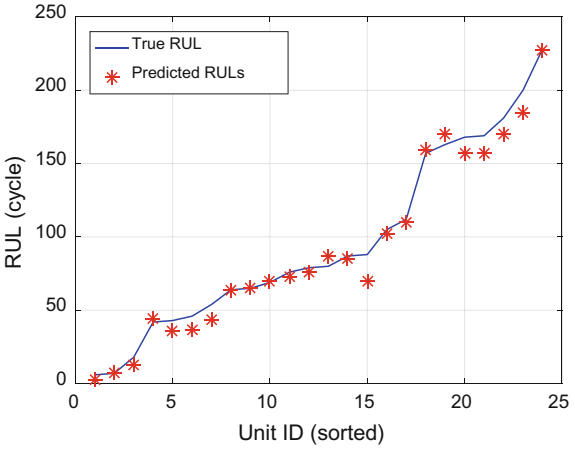


Fig. 8.48 True and predicted RULs in electric cooling fan prognostics

8.6 Exercises

8.1 Consider a basic model of an isotropic rotor with a rubbing condition. Assume that the rotor with a lumped mass system is supported by bearings, which can be modeled as springs and dampers. The motion behavior of the rotor system can be ideally modeled using a second-order ordinary differential equation as

$$M\ddot{x}(t) + C\dot{x}(t) + Kx(t) = mr\omega^2\cos(\omega t + \delta) - F_N(\cos\theta - \mu\sin\theta)$$

where M, C, and K are mass, damping, and stiffness, respectively, m, r, and δ are unbalance mass, radius, and angular orientation, respectively, ω is a rotational speed, x is a lateral displacement of the disk as a function of time t, F_N indicates the normal force, μ is a friction coefficient, and θ is the direction angle of the rubbing part. Assuming that the contact occurs at a specific rotating angle, the normal force F_N can be modeled as

$$F_N = K_N(t)x(t) = K_r\left[a_0 + \sum_{n=1}^{m} a_n\cos(n\omega t)\right]x(t),$$
$$\text{where } a_0 = \phi/2\pi, a_n = 2(-1)^n\sin(n\pi/2)/n\pi$$

where ϕ is the angle of contact, K_N is the rubbing stiffness, and K_r is the amplitude of the rubbing stiffness. The profile of K_N as a function of ϕ is graphically shown in Fig. 8.49. The values of the parameters are listed in Table 8.7.

(1) Find the response, $x(t)$, in $1 \le t \le 2$ for each of the following three cases: (i) $K_r/K = 0.2$, (ii) $K_r/K = 0.35$, and (iii) $K_r/K = 0.4$.
(2) Derive the amplitude ratio of two times the fundamental frequency ($2\times$) to the fundamental frequency ($1\times$) in a frequency domain as a feature for all three cases in (1). Assume that the amplitude of two times the fundamental frequency ($2\times$) is the sum of amplitudes between 1.98 and 2.02 times the fundamental frequency, and the amplitude of the fundamental frequency is the sum of amplitudes between 0.98 and 1.02 times the fundamental frequency.
(3) Calculate the remaining useful life (RUL) of the rotor, assuming that cases (i), (ii), and (iii) are the responses at day 1, 65, and 100,

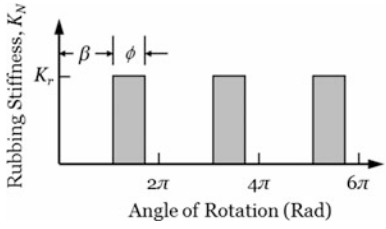


Fig. 8.49 Profile of rubbing stiffness as a function of angle of rotation

Table 8.7 Parameters and their values in Problem 8.1

Parameters	Values
Mass, M	1.6×10^6 kg
Damping coefficient, C	5.66×10^2 kg s^{-1}
Stiffness, K	5.0×10^5 N/m
Unbalance mass, m	1.0×10^{-3} kg
Unbalance radius, r	2.0×10^{-2} m
Rotational speed, ω	5.0×10^3 rpm
Angular orientation of unbalance mass, δ	0°
Angle of contact, ϕ	150°
Angle at the start of rubbing, β	0°
Friction coefficient, μ	0.6

respectively. Assume that the present day is day 100, and the threshold of the frequency feature is 0.9. Use a single-term exponential function to build a prognosis model.

8.2 Consider an application of model-based and data-driven prognostics to a Li-ion battery. The objective of this application is to predict the RUL of a Li-ion battery undergoing repeated charge and discharge cycling. The failure threshold of capacity fade is defined as the discharge capacity of a cell falling below 80% of its initial (BOL) capacity. Assume the capacity fade behavior of the battery can be represented by the following empirical model [83]

$$C_i = \alpha_1 \exp(\lambda_1 i) + \alpha_2 \exp(\lambda_2 i)$$

where C_i is the capacity (Ah) at the ith cycle, α_1 and α_2 are the coefficients of the exponential components of capacity fade, and λ_1 and λ_2 are the exponential capacity fade rates. Suppose the model parameters α_1, λ_1, α_2, and λ_2 are independent random variables whose statistical information is summarized in Table 8.8.

(1) Generate a synthetic data set of capacity fade from 10 cells through the following steps:

Step 1: Generate 10 sets of random realizations of the model parameters based on their distributions.

Table 8.8 Statistical information of the parameters of the capacity fade model for Problem 8.2

Parameter	α_1	λ_1	α_2	λ_2
Distribution	Normal	Normal	Normal	Normal
Mean (fitted value [83])	−9.860E−07	5.752E−02	8.983E−01	−8.340E−04
Standard deviation	9.860E−08	5.752E−03	8.983E−02	8.340E−05

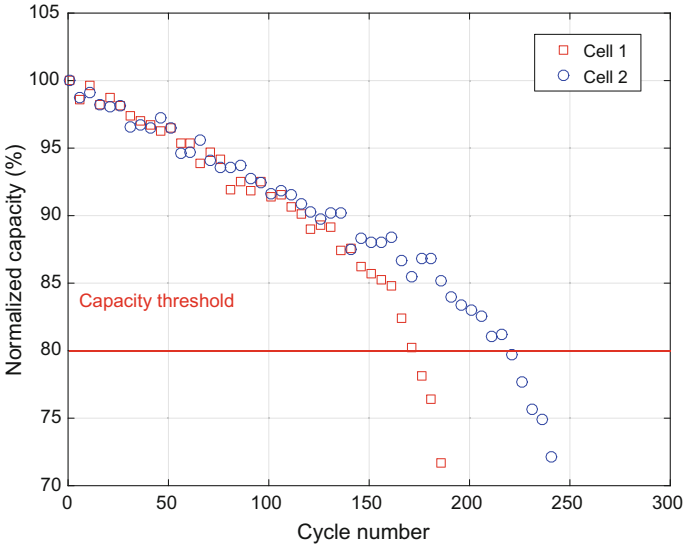


Fig. 8.50 Synthetic data of capacity fade from two cells. For ease of visualization, capacity measurements are plotted every 5 cycles for both cells

Step 2: Produce a capacity vs. cycle trajectory from cycle 1 to cycle 300 for each set of parameters.
Steps 3: Corrupt each trajectory by adding a white Gaussian noise with a mean of 0 Ah and a standard deviation of 0.005 Ah.

An example synthetic data set of capacity fade from 2 cells is shown in Fig. 8.50, where the normalized capacity at the ith cycle c_i is computed as $c_i = C_i/C_0$, with C_0 being the initial capacity.

(2) Here, treat the capacity C as the state variable and the capacity fade rates α_1, λ_1, α_2,and λ_2 as the model parameters. The system transition and measurement functions can then be written as

$$\begin{aligned}
&\text{Transition:} \quad C_i = \alpha_{1,i-1}\exp\left(-\lambda_{1,i-1}i\right) + \alpha_{2,i-1}\exp\left(-\lambda_{2,i-1}i\right) + u_i,\\
&\qquad\qquad\quad \alpha_{1,i} = \alpha_{1,i-1} + r_{1,i},\ \lambda_{1,i} = \lambda_{1,i-1} + r_{2,i},\\
&\qquad\qquad\quad \alpha_{2,i} = \alpha_{2,i-1} + r_{3,i},\ \lambda_{2,i} = \lambda_{2,i-1} + r_{4,i},\\
&\text{Measurement:} \quad y_i = C_i + v_i
\end{aligned}$$

Here, y_i is the capacity measurement at the ith cycle; u, r_1, r_2, r_3, r_4, and v are the Gaussian noise variables with zero means. Use the standard particle filter method to predict the RULs of each cell at cycles 50 and 150. Compare the prediction accuracy at these two cycles for each cell.

References

1. Bond, L. J. (2015). From NDT to prognostics: Advanced technologies for improved quality, safety and reliability, Invited Keynote. In *12th Far East NDT Forum*, Zhuhai, China, May 29–31, 2015.
2. Bond, L. J., Doctor, S. R., Jarrell, D. B., & Bond, J. W. D. (2008). Improved economics of nuclear plant life management. In *Proceedings of the 2nd IAEA International Symposium on Nuclear Power Plant Life Management*, International Atomic Energy Agency, Shanghai, China, IAEA Paper IAEA-CN-155-008KS.
3. Global industry estimates based on "Industrial Internet: Pushing the Boundaries of Minds & Machines". November 26, 2012.
4. Wang, P., Youn, B. D., & Hu, C. (2014). A probabilistic detectability-based sensor network design method for system health monitoring and prognostics. *Journal of Intelligent Material Systems and Structures*. https://doi.org/10.1177/1045389X14541496.
5. Wang, P., Wang, Z., Youn, B. D., & Lee, S. (2015). Reliability-based robust design of smart sensing systems for failure diagnostics using piezoelectric materials. *Computers & Structures, 156,* 110–121.
6. Youn, B. D., & Xi, Z. (2009). Reliability-based robust design optimization using the eigenvector dimension reduction (EDR) method. *Structural and Multidisciplinary Optimization, 37*(5), 475–492.
7. Youn, B. D., Choi, K. K., Du, L., & Gorsich, D. (2007). Integration of possibility-based optimization and robust design for epistemic uncertainty. *ASME Journal of Mechanical Design, 129*(8).
8. Adjiman, C. S., Androulakis, I. P., & Floudas, C. A. (2000). Global optimization of mixed-integer nonlinear problems. *AIChE Journal, 46*(9), 1769–1797.
9. Wei, J., & Realff, J. (2004). Sample average approximation methods for stochastic MINLPs. *Computers & Chemical Engineering, 28*(3), 333–346.
10. *New technology captures freely available vibration energy to power wireless sensor*. https://energy.gov/eere/amo/vibration-power-harvesting. Accessed February 25, 2018.
11. McFadden, P. D. (1987). A revised model for the extraction of periodic waveforms by time domain averaging. *Mechanical Systems and Signal Processing, 1,* 83–95.
12. Jardine, A. K. S., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing, 20,* 1483–1510.
13. Bechhoefer, E., & Kingsley, M. (2009). A review of time synchronous average algorithms. In *Annual Conference of the Prognostics and Health Management Society*, San Diego, CA.
14. Ha, J. M., Youn, B. D., Oh, H., Han, B., & Jung, Y. (2016). Autocorrelation-based time synchronous averaging for health monitoring of planetary gearboxes in wind turbines. *Mechanical Systems and Signal Processing, 70–71,* 161–175.
15. McFadden, P. D. (1989). Interpolation techniques for time domain averaging of gear vibration. *Mechanical Systems and Signal Processing, 3,* 87–97.
16. Strum, R. D., & Kirk, D. E. (1989). *First principles of discrete systems and digital signal processing*. Reading, MA: Addison-Wesley.
17. Yin, L., Yang, M., Gabbouj, M., & Neuvo, Y. (1996). Weighted median filters: A tutorial. *IEEE Transactions on Circuits and Systems, 40*, 157–192.
18. Ganguli, R. (2002). Noise and outlier removal from jet engine health monitoring signals using weighted FIR median hybrid filters. *Mechanical Systems and Signal Processing, 16*(6), 867–978.
19. Neerjarvi, J., Varri, A., Fotopoulos, S., & Neuvo, Y. (1993). Weighted FMH filters. *Signal Processing, 31*, 181–190.
20. Hu, C., Youn B. D., Kim, T. J., & Wang, P. (2015). Semi-supervised learning with co-training for data-driven prognostics. *Mechanical Systems and Signal Processing, 62–63*, 75–90.

21. Sejdić, E., Djurović, I., & Jiang, J. (2009). Time-frequency feature representation using energy concentration: An overview of recent advances. *Digital Signal Processing, 19*(1), 153–183.
22. Gröchenig, K. (2001). *Foundations of time-frequency analysis*. Boston: Birkhäuser.
23. Mallat, S. G. (1999). *A wavelet tour of signal process* (2nd ed.). San Diego: Academic Press.
24. Daubechies, I. (1992). *Ten lectures on wavelets*. Philadelphia: Society for Industrial and Applied Mathematics.
25. Cohen, L. (1995). *Time-frequency analysis*. Englewood Cliffs, NJ: Prentice Hall.
26. Dahl, G., Ranzato, M., Mohamed, A.-R., & Hinton, G. E. (2010). Phone recognition with the mean-covariance restricted boltzmann machine. In *Advances in neural information processing systems* (pp. 469–477). New York: Curran Associates, Inc.
27. Hinton, G., Deng, L., Yu, D., Mohamed, A.-R., Jaitly, N., Senior, A., et al. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine, 29*(6), 82–97.
28. Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation, 18*(7), 1527–1554.
29. Krizhevsky, A., Sutskever, I., & Hinton, G. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (Vol. 25, pp. 1106–1114). New York: Curran Associates, Inc.
30. Mikolov, T., Deoras, A., Kombrink, S., Burget, L., & Cernocký, J. (2011). Empirical evaluation and combination of advanced language modeling techniques. In *INTERSPEECH*, ISCA (pp. 605–608).
31. Socher, R., Huang, E. H., Pennin, J., Manning, C. D., & Ng, A. (2011) Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Advances in neural information processing systems* (pp. 801–809). New York: Curran Associates, Inc.
32. Hinton, G. E., Osindero, S., & Teh, Y. (2006). A fast learning algorithm for deep belief nets. *Neural Computation, 18,* 1527–1554.
33. Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and Trends in Machine Learning, 2*(1).
34. Fisher, R. A. (1938). The statistical utilization of multiple measurements. *Annals of Eugenics, 8,* 376–386.
35. Fukunaga, K. (1990). *Introduction to statistical pattern recognition* (2nd ed.). USA: Academic Press.
36. Huang, R., Xi, L., Li, X., Richard Liu, C., Qiu, H., & Lee, J. (2007). Residual life predictions for ball bearings based on self-organizing map and back propagation neural network methods. *Mechanical Systems and Signal Processing, 21,* 193–207.
37. Samanta, B. (2004). Gear fault detection using artificial neural networks and support vector machines with genetic algorithms. *Mechanical Systems and Signal Processing, 18,* 625–644.
38. Srinivasan, S., Kanagasabapathy, P., & Selvaganesan, N. (2007). Fault diagnosis in deaerator using neural networks. *Iranian Journal of Electrical and Computer Engineering, 6,* 62.
39. Saxena, A., & Saad, A. (2007). Evolving an artificial neural network classifier for condition monitoring of rotating mechanical systems. *Applied Soft Computing, 7,* 441–454.
40. Yang, B. S., Hwang, W. W., Kim, D. J., & Chit Tan, A. (2005). Condition classification of small reciprocating compressor for refrigerators using artificial neural networks and support vector machines. *Mechanical Systems and Signal Processing, 19,* 371–390.
41. Saimurugan, M., Ramachandran, K. I., Sugumaran, V., & Sakthivel, N. R. (2011). Multi component fault diagnosis of rotational mechanical system based on decision tree and support vector machine. *Expert Systems with Applications, 38*(4), 3819–3826.
42. Ge, M., Du, R., Zhang, G., & Xu, Y. (2004). Fault diagnosis using support vector machine with an application in sheet metal stamping operations. *Mechanical Systems and Signal Processing, 18,* 143–159.
43. Xue, F., Bonissone, P., Varma, A., Yang, W., Eklund, N., & Goebel, K. (2008). An instance-based method for remaining useful life estimation for aircraft engines. *Journal of Failure Analysis and Prevention, 8*(2), 199–206.

44. Nie, L., Azarian, M. H., Keimasi, M., & Pecht, M. (2007). Prognostics of ceramic capacitor temperature-humidity-bias reliability using mahalanobis distance analysis. *Circuit World, 33*(3), 21–28.
45. Baurle, R. A., & Gaffney, R. L. (2008). Extraction of one-dimensional flow properties from multidimensional data sets. *Journal of Propulsion and Power, 24*(24), 704–714.
46. Wang, T., Yu, J., Siegel, D., & Lee, J. (2008). A similarity-based prognostics approach for remaining useful life estimation of engineered systems. In *International Conference on Prognostics and Health Management*, Denver, CO, October 6–9, 2008.
47. Saha, B., Goebel, K, Poll, S., & Christophersen, J. (2009). Prognostics methods for battery health monitoring using a Bayesian framework. *IEEE Transaction on Instrumentation and Measurement, 58*(2), 291–296.
48. Gebraeel, N. Z., Lawley, M. A., Li, R., & Ryan, J. K. (2005). Residual-life distributions from component degradation signals: A Bayesian approach. *IIE Transactions on Reliability, 37*(6), 543–557.
49. Kwon, D., Azarian, M., & Pecht, M. (2008). Detection of solder joint degradation using RF impedance analysis. In *IEEE Electronic Components and Technology Conference*, Lake Buena Vista, FL, 27–30 May (pp. 606–610).
50. Abbasion, S., Rafsanjani, A., Farshidianfar, A., & Irani, N. (2007). Rolling element bearings multi-fault classification based on the wavelet denoising and support vector machine. *Mechanical Systems and Signal Processing, 21,* 2933–2945.
51. Sun, J., Rahman, M., Wong, Y., & Hong, G. (2004). Multiclassification of tool wear with support vector machine by manufacturing loss consideration. *International Journal of Machine Tools and Manufacture, 44,* 1179–1187.
52. Geramifard, O., Xu, T. X., Pang, C., Zhou, J., & Li, X. (2010). Data-driven approaches in health condition monitoring—A comparative study. In *8th IEEE International Conference on Control and Automation (ICCA)* (pp. 1618–1622).
53. Ramadass, P., Haran, B., Gomadam, P. M., White, R., & Popov, B. N. (2004). Development of first principles capacity fade model for Li-ion cells. *Journal of the Electrochemical Society, 151,* A196.
54. Santhanagopalan, S., Zhang, Q., Kumaresan, K., & White, R. E. (2008). Parameter estimation and life modeling of lithium-ion cells. *Journal of the Electrochemical Society, 155*(4), A345–A353.
55. Yang, L., Agyakwa, P., & Johnson, C. M. (2013). Physics-of-failure lifetime prediction models for wire bond interconnects in power electronic modules. *IEEE Transactions on Device and Materials Reliability, 13*(1), 9–17.
56. Shao, J., Zeng, C., & Wang, Y. (2010). Research progress on physics-of-failure based fatigue stress-damage model of solderjoints in electronic packing. In *Proceedings of Prognostics and Health Management Conference*, January 10–12, 2010 (pp. 1–6).
57. Gordon, N. J., Salmond, D. J., & Smith, A. F. M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Radar and Signal Processing, IEE Proceedings, F140*(2), 107–113.
58. Kitagawa, G. (1996). Monte carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics, 5*(1), 1–25.
59. Arulampalam, M., Maskell, S., Gordon, N., & Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing, 50*(2), 174–189.
60. Tavare, S., Balding, D., Griffiths, R., & Donnelly, P. (1997). Inferring coalescence times from DNA sequence data. *Genetics, 145,* 505–518.
61. Weiss, G., & von Haeseler, A. (1998). Inference of population history using a likelihood approach. *Genetics, 149,* 1539–1546.
62. Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equations of state calculations by fast computing machines. *Journal of Chemical Physics, 21*(6), 1087–1092.

63. Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika, 57*(1), 97–109.
64. Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 6*(6), 721–741.
65. Liu, J. S. (1994). The collapsed Gibbs sampler in bayesian computations with applications to a gene regulation problem. *Journal of the American Statistical Association, 89*(427), 958–966.
66. Tierney, L., & Kadane, J. B. (1986). Accurate approximations for posterior moments and marginal distributions. *Journal of the American Statistical Association, 81,* 82–86.
67. Tierney, L., Kass, R. E., & Kadane, J. B. (1989). Approximate marginal densities of nonlinear functions. *Biometrika, 76*(3), 425–433.
68. Azevedo-Filho, A., & Shachter, R. (1994). Laplace's method approximations for probabilistic inference in belief networks with continuous variables. In R. Mantaras & D. Poole (Eds.), *Uncertainty in artificial intelligence*. San Francisco, CA: Morgan Kauffman.
69. Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering, 82*(1), 35–45.
70. Jazwinski, A. H. (1970). *Stochastic processes and filtering theory*. San Diego, CA: Academic.
71. Sorenson, H. W. (Ed.). (1985). *Kalman filtering: Theory and application*. Piscataway, NJ: IEEE.
72. Julier, S. J., & Uhlmann, J. K. (2004). Unscented filtering and nonlinear estimation. *Proceedings of IEEE, 92*, 401–422.
73. Arulampalam, S., Maskell, S., Gordon, N., & Clapp, T. (2002). A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. *IEEE Transaction on Signal Processing, 50*(2), 174–188.
74. Cappe, O., Godsill, S. J., & Moulines, E. (2007). An overview of existing methods and recent advances in sequential Monte Carlo. *IEEE Proceedings, 95*(5), 899–924.
75. Hu, C., Jain, G., Tamirisa, P., & Gorka, T. (2014). Method for estimating capacity and predicting remaining useful life of lithium-ion battery. *Applied Energy, 126,* 182–189.
76. Honkura, K., Takahashi, K., & Horiba, T. (2011). Capacity-fading prediction of lithium-ion batteries based on discharge curves analysis. *Journal of Power Sources, 196*(23), 10141–10147.
77. Brown, J., Scott, E., Schmidt, C., & Howard, W. (2006). A practical longevity model for lithium-ion batteries: De-coupling the time and cycle-dependence of capacity fade. In *208th ECS Meeting*, Abstract #239.
78. Wang, P., Youn, B. D., & Hu, C. (2012). A generic probabilistic framework for structural health prognostic and uncertainty management. *Mechanical Systems and Signal Processing, 28,* 622–637.
79. Hu, C., Youn, B. D., Wang, P., & Yoon, J. T. (2012). Ensemble of data-driven prognostic algorithms for robust prediction of remaining useful life. *Reliability Engineering & System Safety, 103*, 120–135.
80. Orchard, M., Kacprzynski, G., Goebel, K., Saha, B., & Vachtsevanos, G. (2008). Advances in uncertainty representation and management for particle filtering applied to prognostics. In *International Conference on Prognostics and Health Management, 2008. PHM 2008*, October 2008 (pp. 1–6).
81. Tang, L., Kacprzynski, G., Goebel, K., & Vachtsevanos, G. (2009). Methodologies for uncertainty management in prognostics. In *2009 IEEE Aerospace Conference*, March 2009 (pp. 1–12).
82. Sankararaman, S., & Goebel, K. (2015). Uncertainty in prognostics and systems health management. *International Journal of Prognostics and Health Management, 6*, Special Issue on Uncertainty in Prognostics and Health Management.
83. He, W., Williard, N., Osterman, M., & Pecht, M. (2011). Prognostics of lithium-ion batteries based on Dempster-Shafer theory and the Bayesian Monte Carlo method. *Journal of Power Sources, 196*(23), 10314–10321.

Chapter 9
Case Studies: Prognostics and Health Management (PHM)

Prognostics and health management (PHM) technology has been successfully implemented into engineering practice in diverse settings. This chapter presents case studies that explain successful PHM practices in several engineering applications: (1) steam turbine rotors, (2) wind turbine gearboxes, (3) the core and windings in power transformers, (4) power generator stator windings, (5) lithium-ion batteries, (6) fuel cells, and (7) water pipelines. These examples provide useful findings about the four core functions of PHM technology, contemporary technology trends, and industrial values.

9.1 Steam Turbine Rotors

Steam turbines in power plants are large and complex mechanical rotating systems. Generally, rigid couplings are used to connect three to five stages of shafts, as shown in Fig. 9.1. Steam turbine systems are typically composed of a high-pressure (HP) turbine, an intermediate-pressure (IP) turbine, two low-pressure (LP) turbines, a generator, and an exciter. Each shaft is supported by two journal bearings. An oil film between the journal bearing and the turbine shaft prevents direct contact between the rotor and the stator. These bearings ensure the turbine system operates steadily. Although turbines are designed to operate in a stable condition, various uncertainties exist, such as operation uncertainty, manufacturing variability, and installation uncertainty. In adverse conditions, various anomaly states can be found in a turbine system, for example unbalance, misalignment, rubbing, and oil whirl, among others. The work described here as a PHM case study examined a number of steam turbine systems from four power plant sites in Korea.

C. Hu et al., *Engineering Design under Uncertainty and Health Prognostics*,
Springer Series in Reliability Engineering,
https://doi.org/10.1007/978-3-319-92574-5_9

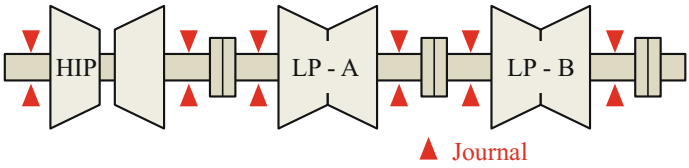


Fig. 9.1 Three-stage steam turbine diagram

9.1.1 Designing the Health Sensing Function

Vibration signals are commonly used for health diagnostics and prognostics of steam turbines. Even under normal conditions, a bit of unbalance is present in the turbine rotors, resulting in a certain level of vibration when rotating. As anomaly states begin developing, the level of vibration increases. Sophisticated analysis of vibration signals is essential for robust diagnostics and prognostics of the turbine rotors. In addition to vibration sensors, temperature and pressure signals can also be acquired for PHM of turbine rotors. Data acquisition must be carefully designed and care must be taken to account for the data-sampling rate and frequency, logging power and storage, and other factors.

Proximity sensors are widely used for condition monitoring of journal bearing rotor systems in turbines. The sensors directly measure gap displacement signals between the turbine shaft and the sensor. These measurements are represented by vibration signals that are acquired via the AC component of the signals. The DC component of the signals represents the absolute position of the turbine shaft centerline. Since the gap information provides information about the dynamic behavior of the turbine system, the state of the turbine can be accurately determined by using vibration signals to analyze the behavior of the turbines [1, 2].

To robustly detect potential anomaly states, the number and location of proximity sensors must be carefully considered. Adding more sensors requires increased data analysis loads and data storage capacity. On the other hand, the high-temperature environment in steam turbines limits sensor placement. Sensors are often placed between the coupling and the bearing seal to limit the effect of the high-temperature steam. Considering these practical aspects, the sensors are typically placed at two positions for each turbine stage, adjacent to the journal bearings, as illustrated Fig. 9.2. For example, in a three-stage steam turbine, vibration signals are acquired from six different axial positions. For each axial location, two proximity sensors are placed orthogonally to obtain the orbit trace of the turbine centerline. In total, twelve proximity sensors are used for PHM in a three-stage steam turbine.

Other signal measurement specifications—period, sampling rate, and duration—must also be carefully determined when designing the sensing function. The period between the signal measurements should be short enough to detect abrupt degradation while still being long enough to minimize the burden of data storage. Signals should be measured when an event (e.g., anomaly state, high vibration)

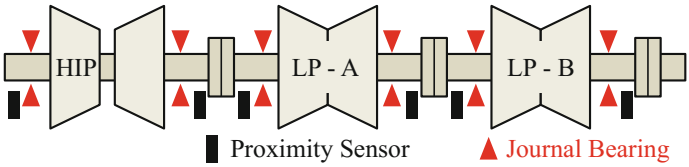


Fig. 9.2 Axial locations of proximity sensors. Reprinted (adapted) with permission from Ref. [3]

unexpectedly occurs. When determining the sampling rate, general considerations include the harmonics in the frequency response, computational capacity, and the frequency resolution required for anomaly detection. The sampling rate of signals must be high enough to perform accurate analysis of vibration signals while preventing the leakage of fast Fourier transform (FFT). The duration of the signal measurement should be carefully determined while considering the available computation ability and the capacity of the data acquisition (DAQ) hardware. In this case study, the measurement period, sampling duration, and rate were set as every twelve hours (day and night), ten minutes per day, with a sampling frequency of 2048 Hz.

Given the vibration signals, a tachometer signal can be measured, which has a pulse per revolution. This signal can provide absolute phase information for the vibration signals and the revolutions per minute (rpm) of the turbine. Moreover, the tachometer signal works as a reference for use when vibration signals are resampled. Using this reference, the signals can be synchronized with other vibration signals.

9.1.2 Designing the Health Reasoning Function

The objective of the reasoning function is to extract the health data of steam turbines using measured vibration signals. This requires some key reasoning steps, including: preprocessing, feature extraction, feature selection, and classification. These are the steps required for the supervised machine learning method.

Preprocessing

Vibration signals must be processed before signal features are extracted. The rpm of a steam turbine varies between 3600 ± 20 rpm; however, larger variations can occur due to uncertainties. These variations in rpm may lead to inconsistent reasoning results because the fixed sampling rate is likely to yield different sampled data at given time intervals. This uncertainty can be controlled by applying *phase synchronized resampling*, also known as *angular resampling*, to the acquired vibration signals, as shown in Fig. 9.3 [4, 5]. Using the tachometer peaks as the starting point of a revolution, the signals can be resampled to have an equal number

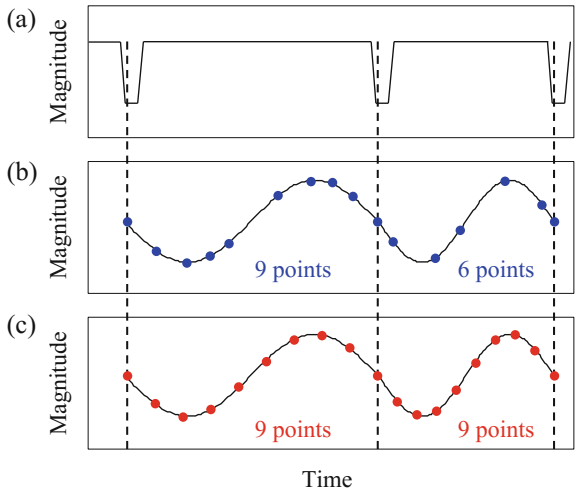


Fig. 9.3 Resampling vibration signals; **a** tachometer signal, **b** raw signal, and **c** resampled signal

of points per revolution. The resampled vibration signals will then give consistent results, despite rpm variations.

Feature Extraction

Next, the resampled vibration signals are used to extract candidate features. Based on the information from complete revolutions, candidate features can be extracted with minimal noise in the order domain. Since steam turbines rotate mostly at a steady state, time- and frequency-domain features can be used. Candidate features include eight time-domain features and eleven frequency–domain features. The extracted features are presented in Tables 9.1 and 9.2 [6, 7].

Among the eight time-domain features, max, mean, and root-mean-square (RMS) are related to the energy of the vibration. Skewness and kurtosis, which are the third and fourth statistical moments, respectively, represent the statistical characteristics. The last three features—crest factor, shape factor, and impulse

Table 9.1 Time-domain features

Index	Features	Description
t1	Max	$\text{Max}(\lvert X\rvert)$
t2	Mean	$\text{Mean}(\lvert X\rvert)$
t3	RMS	$\sqrt{\frac{\sum X_i^2}{N}}$
t4	Skewness	$\frac{\sum (X_i-\bar{X})^3}{(N-1)s^3}$
t5	Kurtosis	$\frac{\sum (X_i-\bar{X})^4}{(N-1)s^4}$
t6	Crest factor	$\frac{\text{Max}}{RMS}$
t7	Shape factor	$\frac{RMS}{Mean}$
t8	Impulse factor	$\frac{\text{Max}}{Mean}$

Table 9.2 Frequency-domain features

Index	Features	Description
f1	FC	$\frac{\int f \times s(f) df}{\int s(f) df}$
f2	RMSF	$\left[\frac{\int f^2 \times s(f) df}{\int s(f) df}\right]^{1/2}$
f3	RVF	$\left[\frac{\int (f-FC)^2 \times s(f) df}{\int s(f) df}\right]^{1/2}$
f4	$0.5x/1x$	$\sqrt{s(f_{0.5x})}/\sqrt{s(f_{1x})}$
f5	$2x/1x$	$\sqrt{s(f_{2x})}/\sqrt{s(f_{1x})}$
f6	$(1x \sim 10x)/1x$	$\left\{\sum_{n=1}^{10} \sqrt{s(f_{nx})}\right\}/\sqrt{s(f_{1x})}$
f7	$(0 \sim 0.39x)/1x$	$\left\{\int_0^{0.39x} \sqrt{s(f)} df\right\}/\sqrt{s(f_{1x})}$
f8	$(0.4x \sim 0.49x)/1x$	$\left\{\int_{0.4x}^{0.49x} \sqrt{s(f)} df\right\}/\sqrt{s(f_{1x})}$
f9	$(0.51x \sim 0.99x)/1x$	$\left\{\int_{0.51x}^{0.99x} \sqrt{s(f)} df\right\}/\sqrt{s(f_{1x})}$
f10	$(3x \sim 5x)/1x$	$\left\{\int_{3x}^{5x} \sqrt{s(f)} df\right\}/\sqrt{s(f_{1x})}$
f11	$(3x, 5x, 7x, 9x)/1x$	$\left\{\sum_{n=1}^{4} \sqrt{s(f_{(2n+1)x})}\right\}/\sqrt{s(f_{1x})}$

factor—are related to the shape of the sinusoidal wave (e.g., sharpness). Frequency-domain features are calculated from the power spectrum of the vibration signals. In Table 9.2, f and $s(f)$ denote the frequency and the power spectrum of frequency f, respectively. Among the eleven features, frequency center (FC) and root-mean-square frequency (RMSF) are related to the main frequency. Root-variance frequency (RVF) denotes the variance of the power spectrum. Other frequency features represent the ratios of harmonics/sub-harmonics to the fundamental frequency.

As stated above, the features are extracted based on complete revolutions. Time-domain features are extracted based on one revolution, while the frequency-domain features are based on multiple revolutions. To observe signals that change in a short time period, one-revolution-based time-domain features have been found to be more effective. On the other hand, the resolution of the power spectrum is an important factor for extraction of frequency-domain features. Resolution can be enhanced by using data from multiple revolutions; however, the number of revolutions is limited to sixty because the data for a large number of revolutions can be burdensome to the DAQ hardware. Sixty revolutions are recommended to minimize the frequency leakage problem for turbines operating at 3600 rpm [3].

Omnidirectional regeneration (ODR) can be applied to enhance the robustness of diagnosis. Diagnosis results for directional anomaly states (e.g., misalignment, rubbing) depend on the direction of the sensors. The ODR method can generate vibration signals in an arbitrary direction, which makes the diagnosis procedure

robust regardless of directionality. Researchers have also reported that the performance of the diagnosis can be enhanced by using the ODR method without extra sensors [8].

Feature Selection and Classification

The feature selection process is of great importance when the number of features is large. This process determines the optimal feature subset from a set of candidate features. Each feature possesses a different separation ability to distinguish the health states of the steam turbine. To obtain robust reasoning for steam turbines, a genetic algorithm is integrated with different separability measures, such as probability-of-separation (PoS) or correlation coefficient [3, 9]. A genetic algorithm can be used to randomly generate subsets of features, and the degrees of separability can be measured for the subsets using either the PoS or a correlation coefficient method. Subset generation is repeated until a predefined criterion is satisfied. Note that PoS is a separability measure that quantifies the degree of separation between two classes.

Once the optimal feature subset is determined, classification continues using the support vector machine (SVM) method to minimize structural risk [10, 11]. Through an optimization process, hyper-planes that separate multiple states are trained using the data from known states. Using the trained classifier, the unknown states of a turbine can be predicted.

9.1.3 Designing the Health Prognostics and Management Functions

The prognostics function predicts the remaining useful life (RUL) of a steam turbine. To develop the prognostics function, a health index is defined for each anomaly state, as shown Table 9.3. V_{pp} is the peak-to-peak value of the vibration signal, clearance denotes the *clearance* of an oil seal in a steam turbine, and f8 indicates one of the frequency-domain features shown in Table 9.2. Each health index, defined as a function of the extracted features, considers certain characteristics of the anomaly conditions. The indices are normalized between zero and one. A simple example is an unbalanced state in a turbine. The health index of unbalance is the peak-to-peak level divided by the clearance level. After the health indices are defined, the RUL can be calculated by tracing the trend of the indices, as presented in Fig. 9.4. The blue points in the figure are used to establish the RUL model. The model is used for estimating the RUL of the steam turbine, as shown by the green dotted line.

An accurately calculated RUL provides the basis for condition-based monitoring (CBM) of steam turbines. CBM of steam turbines helps steam turbine operators schedule maintenance actions based on the analyzed condition of the turbines. Operators can minimize unnecessary maintenance actions and can prevent catastrophic turbine failures by reliably predicting turbines' health conditions. It has

Table 9.3 Definition of an anomaly index for each health state

Health state	Anomaly index
Unbalance	$\frac{V_{pp}}{clearance}$
Rubbing	$\frac{V_{pp}}{clearance}(2 - Aspect\ Ratio\ of\ 1X\ Orbit)$
Oil whirl	$\frac{V_{pp}}{clearance}(2 - f8)$
Misalignment	$\frac{V_{pp}}{clearance}(2 - Aspect\ Ratio\ of\ 1X\ Orbit)$

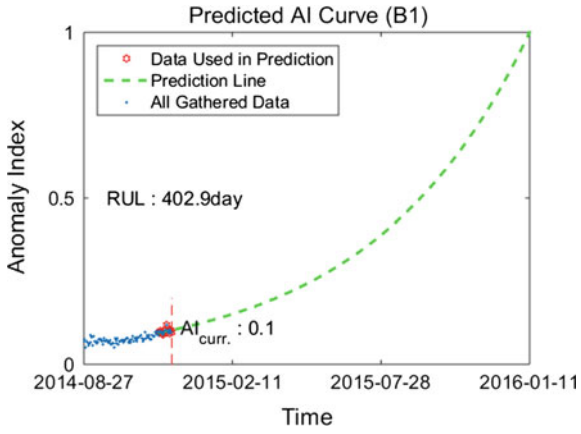


Fig. 9.4 Example of prognostics

been reported in [12] that CBM can substantially save on overall operation and maintenance (O&M) costs.

9.2 Wind Turbine Gearboxes

The drivetrain of a wind turbine consists primarily of the main bearing, drive shaft, gearbox, and generator, as shown in Fig. 9.5. The main bearing supports the blades, and the gearbox connected to the drive shaft increases the rotating speed. Rotating energy from the high-speed shaft (i.e., the output shaft of the gearbox) is transferred to the generator. Wind turbines generally operate in harsh environmental conditions, such as random and non-stationary wind profiles. Moreover, offshore wind turbines are exposed to highly varying sea gusts with high salt concentrations. Thus, the rotating parts in the drivetrain of a wind turbine are prone to a variety of mechanical failures, such as fatigue, wear, and corrosion. Among the drive-train components of a wind turbine, the gearbox is known to have the highest risk due to its potential for an excessively long downtime and its expensive replacement cost. This section discusses the four main PHM functions for wind turbine gearboxes.

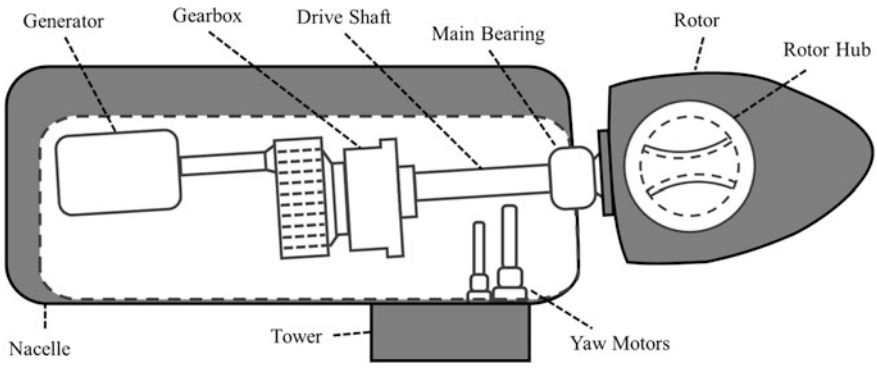


Fig. 9.5 Diagram of a wind turbine

9.2.1 Designing the Health Sensing Function

Wind turbines are equipped with two kinds of DAQ systems, including SCADA (Supervisory Control and Data Acquisition) and CMS (Condition Monitoring System). The main purpose of both SCADA and CMS is to provide data for integrated performance and health management in wind turbines. Although SCADA was originally designed for performance management, recent studies have shown that SCADA data can also serve as a precursor to represent the health state of a wind turbine; data can also be used to enhance the performance of PHM [13].

SCADA basically measures four kinds of signals, including environmental data (e.g., wind speed and direction), operational data (e.g., power and rotor speed), controlling data (e.g., yaw and pitch control) and response data (e.g., temperatures of the main bearing and gearbox shaft). Data collected at a very low frequency, for example once every 10 min, make it possible to evaluate the performance behavior of a wind turbine. Some failure types of the gearbox affect the overall system performance, thus leading to anomaly conditions such as increased temperature of the lubrication oil or unexpected speed fluctuations due to irregular load conditions. Thus, the use of SCADA data as a means for performing PHM is an emerging trend.

CMS requires various kinds of signal analysis, such as vibration analysis, oil debris analysis, and noise analysis, to enhance PHM capabilities. Among the various analysis methods available, a recent study described that typical mechanical failure can be most sensitively detected using vibration analysis [14]. DNV GL, one of the most important organizations for certification of wind turbines, enacted a regulation that every system should acquire high-frequency vibration signals (in addition to SCADA) for the purpose of health assessment in wind turbines. DNV GL certification established a guideline regarding the necessary number of vibration sensors that should be installed on the drivetrain of a wind turbine. According to the certification, at least four vibration sensors, with a sampling rate of

more than 10 kHz, should be installed on each gearbox for condition monitoring purposes. A typical wind turbine gearbox consists of multiple stages of gear sets, including at least one planetary stage. Vibration sensors should be mounted near locations where a high load is applied. In general, it is suggested that sensors be positioned at each gear stage of the gearbox. When a planetary gearbox is of interest for PHM, DNV GL certification suggests placing the sensors in "the area of the ring gear of the planetary gear stage" and at "the level of the sun gear in the 1st stage of the planetary gear stage" where vibration signals from gear meshing can be effectively captured.

Recent studies showed that the integrated measurement of SCADA signals and vibration signals can enhance the performance of PHM for wind turbine gearboxes. For example, to reduce the effects of speed variations within the system, several commercial CMSs are attempting to analyze the vibration signals only when a wind turbine operates under nominally constant rotational speed and torque. This can be achieved by adaptively measuring vibration signals, while continuously monitoring the performance of wind turbines via SCADA signals.

9.2.2 Designing the Health Reasoning Function

Preprocessing

Wind turbine gearboxes consist of multiple stages of gear sets and many bearings. The effects of the gearbox components are mixed together, and are measured using an accelerometer. Thus, to enhance fault detectability, preprocessing techniques should be used on vibration signals. As discussed in Sect. 8.3.1 of Chap. 8, one of the most widely used preprocessing techniques is time synchronous averaging (TSA). TSA serves as a signal separator that facilitates selective analysis of a particular vibration signal that is estimated to originate from a single component of interest in the gearbox [15]. Figure 9.6 illustrates the TSA procedure. In TSA, the signal is separated by gear rotating frequency and ensemble averaged. The deterministic signal, which is synchronously regular with the gear rotation, remains. However, irregular or noise signals asymptotically converge to zero as the number of averages in the ensemble increases. Using the signal processed with TSA, additional analysis can be performed by defining residual signals (RES) and difference signals (DIF) that represent the energy of sidebands and noise, respectively. RES can be calculated from TSA by filtering out regular components, including the fundamental gear mesh frequency and its harmonics. Information about pure sidebands can be observed from RES signals. DIF is calculated from RES by filtering out sidebands. In the normal state, DIF should be ideally a white Gaussian noise because there are no regular or irregular meshing components in the frequency domain. When a fault occurs, an increase in the energy of sidebands and unexpected frequency components can be detected well using RES and DIF.

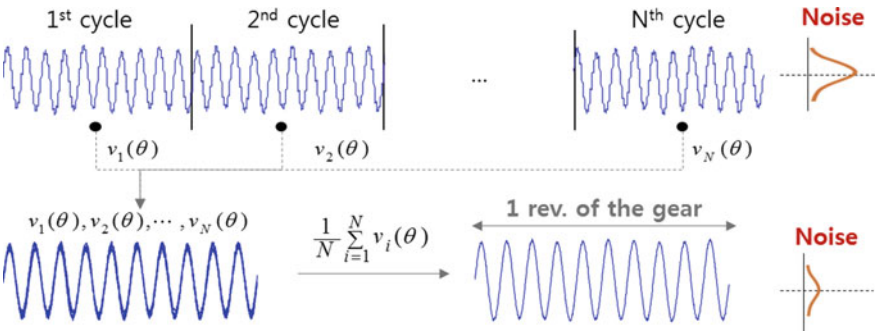


Fig. 9.6 Procedures for TSA

Feature Extraction

Two feature extraction techniques are discussed here, based on: (a) SCADA and (b) vibration. First, the SCADA-based technique can be performed by monitoring power quality and temperature trends. Both the data-driven approach and the model-based approach have four essential steps: (1) define a data-driven or physics-based model that estimates the output signal (i.e., power or temperature) of a gearbox based on the input signals (e.g., speed, torque, and power) under the normal state, (2) continuously measure the input signals and calculate the estimated output signals using the established model (while simultaneously collecting the measured output signals), (3) calculate residuals, which are defined as the difference between the estimated and the measured output signals, and (4) correlate the residuals and the health state. As an example, Fig. 9.7 illustrates the health reasoning process of a wind turbine gearbox where the gearbox oil outlet temperature was selected as the reference SCADA data. Applicable modeling techniques include physics-based modeling techniques, such as temperature modeling using a heat generation model, and data-driven modeling techniques, such as principal component analysis (PCA), neural networks (NN), auto-associative kernel regression (AAKR), and the nonlinear state estimate technique (NSET).

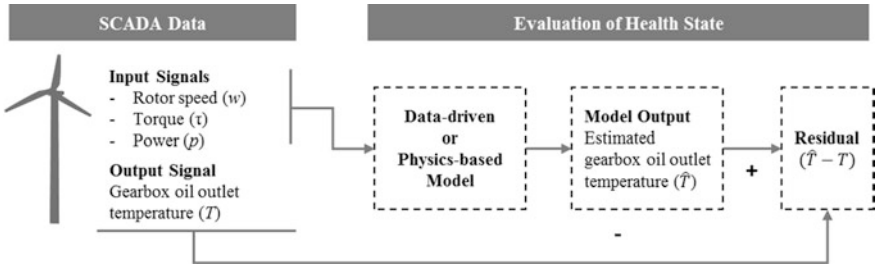


Fig. 9.7 Health reasoning for a wind turbine gearbox using temperature monitoring

Second, vibration-based feature extraction can be performed using time-domain analysis and/or frequency-domain analysis along with preprocessed signals (i.e., TSA, RES, and DIF signals extracted from the preprocessed vibration signals) [16]. Time-domain features acquired from the preprocessed signals can be categorized into three types: kinetic energy, statistics, and waveform. Unbalance in a rotor system can be detected by examining features related to kinetic energy, such as root mean squares (RMS). Sometimes, statistical information (i.e., skewness and kurtosis) of the DIF signal is useful for anomaly detection in gears. For detection of abrupt peaks caused by a gear defect, waveform features (e.g., crest factor (CF), which is defined as the peak-to-peak value divided by the RMS of the pre-processed signals) can be used. In general, time-domain features are rarely used solely, but are instead typically combined with frequency-domain features. Frequency domain features can be obtained by examining a combination of spectral components of the pre-processed signals, which can be calculated through FFT (Fast Fourier Transform) analysis. Energy contained in the fundamental gear mesh frequency and its harmonics, along with the sidebands, is known to be a reliable feature in the frequency-domain. For example, tooth defects can cause amplitude and frequency modulation in the signals, thus increasing the energy of sidebands.

Fault Diagnostics

After extracting the features, health classification and health evaluation can continue using the same process described in Sect. 9.1.2 for a rotor system. An anomaly state in a wind turbine gearbox can be suspected when feature data acquired from an in-service gearbox unit deviates from that expected from a normal state. Various pattern recognition techniques can be applied to classify anomaly states from the normal state of the gearbox. Recently, vibration signals from CMS and operating data from SCADA have been analyzed together for robust health reasoning of wind turbine gearboxes. This technique is suggested for certification of wind turbines (e.g., ISO, IEC, and GL).

9.2.3 Designing the Health Prognostics and Management Functions

Schedule-based maintenance is a conventional activity designed to assure reliability of wind turbines. However, schedule-based maintenance is challenging because an unexpected failure of a wind turbine can cause a huge amount of economic loss. Recently, to overcome this challenge, condition-based maintenance (CBM) of wind turbines using PHM has been proposed.

In CBM, a health prognostics function predicts the remaining useful life (RUL) of the system by understanding the temporal behavior of the health features. The Hidden Markov model, neural networks, and particle filter methods are the most widely used techniques for modeling the temporal behavior of the health

features, thus enabling timely and preventive maintenance and decision making (e.g., CBM [17]).

For effective maintenance decisions, it is of great importance to understand the lead-time needed for maintenance of a target system. Wind turbine maintenance cannot occur instantly after predicting an anomaly state of the turbines due to numerous practical difficulties, such as remote location of the turbines, part procurement and supply, availability of maintenance crews and facilities, site accessibility, etc. This problem is even worse when wind turbines are located offshore. Thus, the lead-time to prepare for suggested maintenance must be carefully analyzed. Moreover, cost analysis becomes an important element when dealing with a large-scale wind farm comprised of a fleet of wind turbines. Because even a single maintenance action is accompanied by large expenses for a sea vessel with a crane, crews, skilled technicians, and so on, maintenance schedules should be optimized to minimize the overall operation and maintenance (O&M) cost. Thus, the health management function should make optimal maintenance decisions while accounting for the lead-time and cost structure. Additional details on these considerations can be found in [18], which contains a general theoretical background, and in [19] which examines applications in wind farms.

9.3 Power Transformers

A power transformer is an electrical system that increases or decreases the voltage of alternating current (AC). Power transformers should be monitored and maintained properly due to the catastrophic consequences of a failure and their high failure rate. Transformer failures can be categorized into three groups: mechanical, electrical, and chemical. Among them, mechanical failure is rarely researched, despite the high chance (approximately 40%) of mechanical transformer failure [20]. The following subsections present the four core functions exercised for PHM of power transformers.

9.3.1 Designing the Health Sensing Function

Common mechanical failures of power transformers include loosening, cracking, and wear at mechanical joints. These failures are mainly due to constant vibrations in the transformer. Careful analysis of vibration can help detect mechanical failures in advance. Transformer vibration mostly comes from the winding and the core. Winding vibration is mechanically produced by an electrodynamic force, and its acceleration a_w is proportional to the square of loading current I as

$$a_w \propto I^2 \propto I_0^2 \cos^2 4\pi ft \tag{9.1}$$

where I_0 and f are the current amplitude and AC frequency, respectively. The core vibration is generated by a phenomenon called magnetostriction, and its acceleration a_c is proportional to the square of loading voltage U with an amplitude U_0 as

$$a_c \propto U^2 \propto U_0^2 \cos^2 4\pi ft \tag{9.2}$$

Both vibration sources have twice the AC frequency as the fundamental frequency. It is known that the core vibration has harmonic frequency components as well, due to magnetization hysteresis and the complexity of the core structure [21]. Thus, transformer vibration is a key physical quantity that can be used for PHM of power transformers.

In order to measure the vibration of a transformer, acceleration sensors are often installed on the outer surface of the transformer (see Fig. 9.8). Vibration is propagated into the sensors through the insulating oil inside the transformer. These sensors cannot be installed inside the transformer due to the insulating oil and high electro-magnetic field.

As shown in Fig. 9.8, transformer vibration is measured at numerous locations, specifically 32–162 locations, depending on the transformer's size and type. Measurement can be problematic due to (1) costly sensor installation and maintenance, (2) management and prohibitively large processing times due to the amount

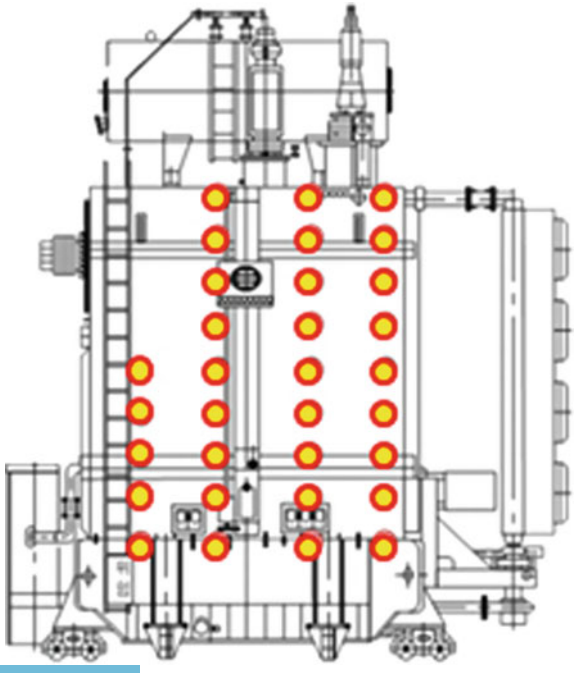


Fig. 9.8 Sensor locations on the power transformer's front surface

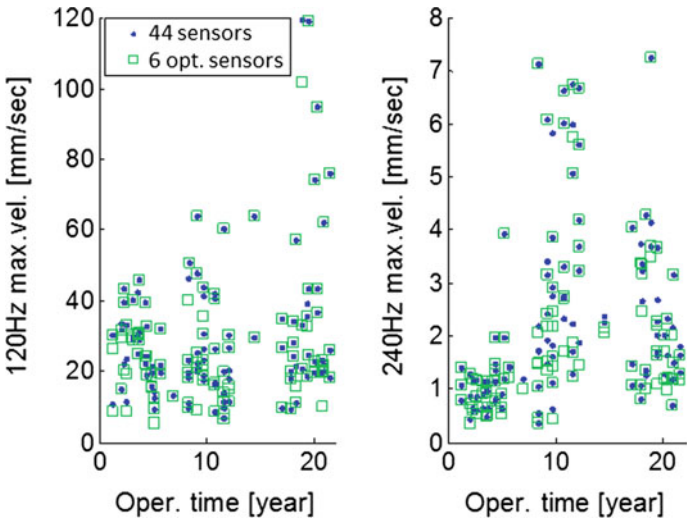


Fig. 9.9 Comparison of fully-loaded sensors and optimally positioned sensors

of data, and (3) acquisition of unnecessary data. Thus, sensor networks must be optimally designed for cost-effective PHM. Measured or simulated vibration data can be used to optimize design of the sensor network. Decision variables include the type, number, and locations of sensors [22]. It is not easy for the simulation-based approach to take into account various uncertainty sources, (e.g., operating conditions, maintenance, and manufacturing). Thus, first, six optimal positions are found using measured vibration data. Given the fundamental and harmonic frequencies found as features of vibration signals in the transformers, the six sensors show the equivalent behavior of the features, compared to the fully loaded sensors, as presented in Fig. 9.9.

9.3.2 Designing the Health Reasoning Function

Two health measures are defined here: the fundamental frequency and the harmonic frequency. They are referred to as the fundamental health measure (*FHM*) and the harmonic health measure (*HHM*), respectively. As addressed in Sect. 9.3.1, they play an important role in assessing the health condition of the core and the winding. Any mechanical failure in the core can increase the vibration energy in both the fundamental and harmonic frequencies; whereas, one in the winding can only affect the fundamental frequency. Mathematically, the two health measures, *FHM* and *HHM*, can be expressed as

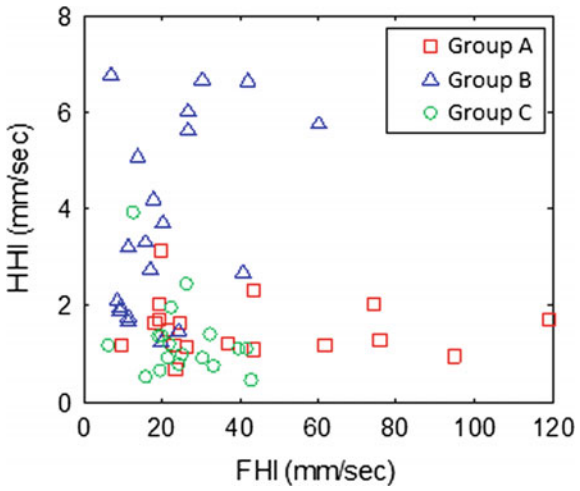


Fig. 9.10 Two health measures for power transformers

$$\boldsymbol{FHM} = \mathbf{max}_{i\in\{k\}}\, \boldsymbol{S_i^{fund}} \tag{9.3}$$

$$\boldsymbol{HHM} = \mathbf{max}_{i\in\{k\}}\, \boldsymbol{S_i^{harm}} \tag{9.4}$$

where S_i^{fund} and S_i^{harm} are the spectral responses of the vibration signals at the fundamental (120 Hz) and harmonic frequencies (240 Hz) measured at the ith sensor, and $\{k\}$ is a sensor set obtained from the sensor position optimization process. Figure 9.10 shows the two health measures for three groups of power transformers. The different groups are defined in terms of age: Group A is 21 years old, Group B is 14 years old, and Group C is 6 years old. From the spread of the two health measures, it can be inferred that (1) a faulty winding in Group A leads to a high *FHM*, (2) a faulty core in Group B leads to high *HHM*, and (3) no fault findings in Group C indicate any low-health measures. Later, it was confirmed that the transformers in Groups A and B had been replaced according to field experts' decisions, and the expected mechanical faults of these replaced transformers were also observed.

9.3.3 Designing the Health Prognostics Function

This subsection briefly discusses strategies for predicting the RUL of power transformers in light of potential mechanical failures. Determining the threshold for mechanical failure of a transformer is an important step for the health prognostics task. This study defines the threshold based upon historical data and experts' opinion. A health degradation model can be developed using the health data acquired from many transformers over different operational periods. The RUL is

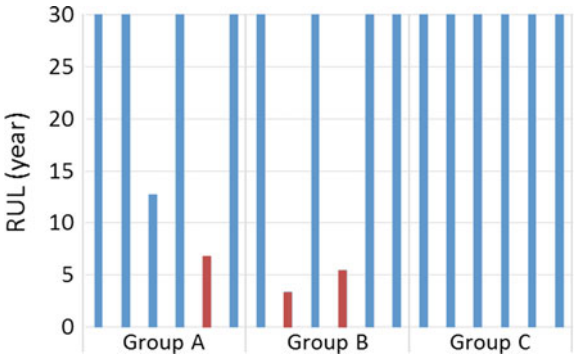


Fig. 9.11 Predicted RULs of power transformers in Groups A, B, and C

then predicted using the similarity-based interpolation (SBI) method [23]. Figure 9.11 shows the RUL prediction results, which can be verified from the fact that the RULs of Groups A and B are smaller than those of Group C.

9.4 Power Generators

Power generators are key components in power plants. Power generators convert kinetic energy into electrical energy. A power generator generally consists of a stator and a rotor, as shown in Fig. 9.12a. Reliable operation of power generators is essential because unexpected breakdowns of a generator can lead to power plant shutdown and substantial related economic and societal losses. Typically, the stator winding, which is composed of slender copper strands, is one of the most vulnerable locations of the power generator. In an anomaly condition, the exterior insulation of the stator winding can deteriorate due to moisture from inner coolant channels, as shown in Fig. 9.12b. This water absorption in stator winding insulation is an indirect reason for catastrophic failure, as shown in Fig. 9.12c. This section describes how a smart health reasoning system for power generator windings can mitigate downtime due to moisture absorption.

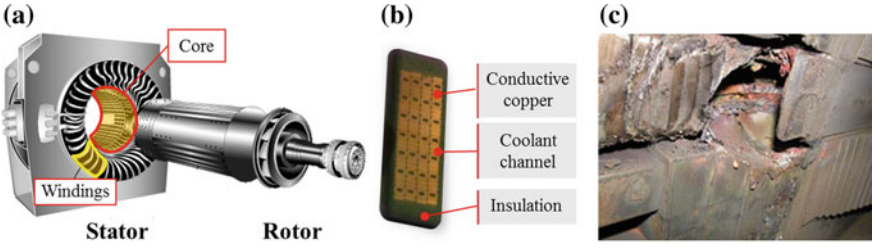


Fig. 9.12 **a** Structure of a power generator, **b** cross-sectional view of a winding, and **c** catastrophic failure of a power generator

9.4.1 Designing the Health Sensing Function

In a water-cooled power generator, coolant water flows into the water channels of the winding. Sometimes, leakage occurs and water is absorbed into and remains in the winding insulation. Leakage can be caused by various operational stresses, such as mechanical vibration, thermal shock, and crevice corrosion. The water that remains in the insulation degrades the winding insulation [24], which can cause the insulation to break down and ultimately cause the power generator to fail. For this reason, electric companies or manufacturing companies assess the health condition of the winding insulation using water absorption detectors. A water absorption detector infers the presence of water in the insulation by measuring capacitance of the insulation [25]. Because the relative static permittivity (or the dielectric constant) of water is higher than that of mica (which is what is generally used as the insulation material), wet insulation has a higher capacitance C, based upon the following equation:

$$C = \varepsilon_r \varepsilon_0 \frac{A}{t} \tag{9.5}$$

where A is the measurement area, t is the distance between the plates, ε_0 is the electric constant ($\varepsilon_0 \approx 8.854$ pF-m^{-1}), and ε_r is the relative static permittivity of the material.

Capacitance measurements as health data provide valuable information that can be used to infer the amount of water absorption in a stator winding. Thus, health-relevant information about the winding can be extracted from capacitance data. The power generators employed in the study described here have forty-two windings and are water-cooled. As shown in Fig. 9.13, the assembly slot for both the top and bottom groups contains ten measurement points. Each measurement point can be modeled as a random variable, X_1–X_{10}.

The capacitance data acquired at these measurement points were modeled with statistically correlated random variables, X_i. One way to measure the correlation between two random variables is to use the Pearson product-moment correlation

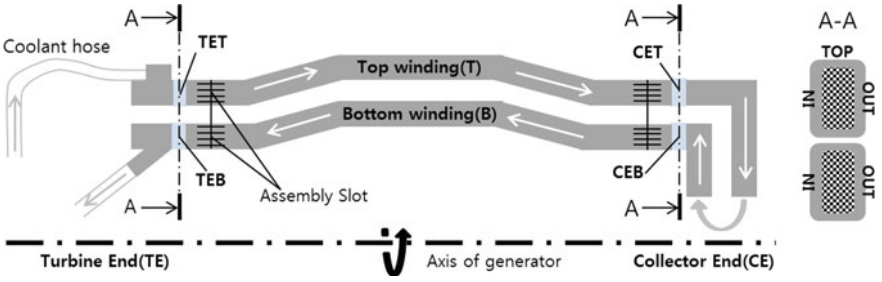


Fig. 9.13 Structure diagram of a water-cooled power generator with a 2-path system. Reprinted (adapted) with permission from Ref. [26]

coefficient. Table 9.4 summarizes the correlation coefficients for the ten random variables in a matrix form. The highlighted values in this table are the correlation coefficients between the measurement variables within the same group (e.g., the CET group). One can observe two features from the highlighted values: (1) there is a statistically positive correlation, and (2) there is a higher degree of correlation within the same group. These features indicate that the two or three capacitance data from the same group tend to behave with linear dependence. Based upon the measurement location and the correlation features, the measurement points with high correlation can be conceived as individual data groups, such as CET, CEB, TET, and TEB. This implies that one entire dataset for ten random variables would be split into four groups, each of which consists of two or three random variables.

9.4.2 Designing the Health Reasoning Function

Although the capacitance data are relevant to the health condition of the stator winding, the high dimensionality and non-linearity of the data make it difficult to infer the health condition easily and precisely. To address this situation, this section introduces the definition of a new health index, namely Directional Mahalanobis Distance (DMD). Traditional Mahalanobis distance (MD) is a relative health measure that quantifies the deviation of a measured data point from a clustered data center, which is generally a populated mean (μ) of a dataset [27]. MD degenerates multi-dimensional data (X) to a one-dimensional distance measure, while taking into account the statistical correlation between random variables, as shown in Fig. 9.14.

As compared to the Euclidean distance, the MD measure possesses a few unique advantages: (1) MD transforms a high-dimensional dataset that is complicated to handle into a one-dimensional measure capable of easy comprehension and quick computation. (2) MD is robust to differing scales of the measurements, as MD values are calculated after normalizing the data. (3) By taking into account the correlation of the dataset, MD is sensitive to inter-variable changes in multivariate measurements. However, MD has its own limitation in that it is a direction-independent health measure in the random capacitance space. In other words, two capacitance measurements with the same MD values, but one with a higher capacitance value and the other with a lower value, are treated equally, although they most likely imply two different levels of moisture absorption. Thus, Directional Mahalanobis Distance (DMD) is proposed to overcome this limitation of MD. DMD can be expressed as follows:

$$\boldsymbol{DMD}\left(\widetilde{\boldsymbol{X}}_i\right) = \sqrt{\left(\widetilde{\boldsymbol{X}}_i - \boldsymbol{\mu}\right)^T \boldsymbol{\Sigma}^{-1}\left(\widetilde{\boldsymbol{X}}_i - \boldsymbol{\mu}\right)} \tag{9.6}$$

$$\widetilde{\boldsymbol{X}}_{n,i} = \begin{cases} \boldsymbol{X}_{n,i}, & \textbf{if } \boldsymbol{X}_{n,i} > \boldsymbol{\mu}_n \\ \boldsymbol{\mu}_n, & \text{otherwise} \end{cases} \tag{9.7}$$

Table 9.4 Correlation coefficient matrix (symmetric) for ten random variables. Reprinted (adapted) with permission from Ref. [26]

Correlation matrix		CET			CEB		TET			TEB	
		TOP (X_1)	OUT (X_2)	IN (X_3)	OUT (X_4)	IN (X_5)	TOP (X_6)	OUT (X_7)	IN (X_8)	OUT (X_9)	IN (X_{10})
CET	TOP (X_1)	1									
	OUT (X_2)	**0.4761**	1								
	IN (X_3)	**0.4194**	**0.5503**	1							
CEB	OUT (X_4)	0.0849	0.1572	0.1354	1						
	IN (X_5)	−0.039	0.1686	0.0765	**0.3445**	1					
TET	TOP (X_6)	0.3341	0.1553	0.1868	0.0343	−0.052	1				
	OUT (X_7)	0.1972	0.2506	0.2729	0.0879	0.0171	**0.4377**	1			
	IN (X_8)	0.2295	0.1423	0.3296	0.0082	0.0457	**0.4269**	**0.4900**	1		
TEB	OUT (X_9)	0.0438	−0.128	−0.097	0.0186	−0.114	0.0887	−0.010	−0.003	1	
	IN (X_{10})	0.0354	−0.040	−0.004	0.0457	0.0870	−0.048	0.1084	0.0215	**0.3385**	1

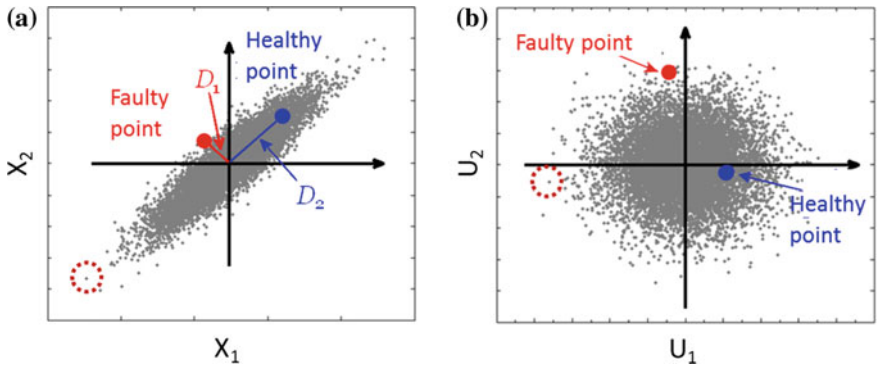


Fig. 9.14 Healthy and faulty points located in the **a** original space and **b** transformed space. Reprinted (adapted) with permission from Ref. [28]

where $X_{n,i}$ denotes the raw capacitance data at the nth measurement location of the ith bar unit, μ_n is the mean of the capacitance data at the nth measurement location, and $\widetilde{X}_{n,i}$ denotes the processed capacitance data. Through the projection process in Fig. 9.15b, the absolutely healthy data would be ignored in the subsequent transformation. The data projection underscores the need for consideration of the direction in the health reasoning process of the measurement data. This leads to the unique capability of the proposed index that makes use of the distance and degradation direction as a health measure, as shown in Fig. 9.15c.

Based upon the maintenance strategies for the stator winding and field experts' opinions, three health grades were proposed, as summarized in Table 9.5: (1) faulty condition (or the presence of water absorption), (2) warning condition (or close to water absorption), and (3) healthy condition (or no water absorption).

9.4.3 *Designing the Health Prognostics and Management Functions*

Figure 9.16 shows the scatter plot of DMD against the operating period. The circles mark the data obtained from the faulty (water absorbed) windings. In particular, two data points marked with red circles represent two failed windings that caught fire in 2008. Most of the circled data points were classified as "faulty" or "warning." This indicates that the proposed health grade system properly defines the health condition of the generator stator windings as it relates to water absorption. The circles in the healthy zone are maintenance cases due to reasons other than water absorption.

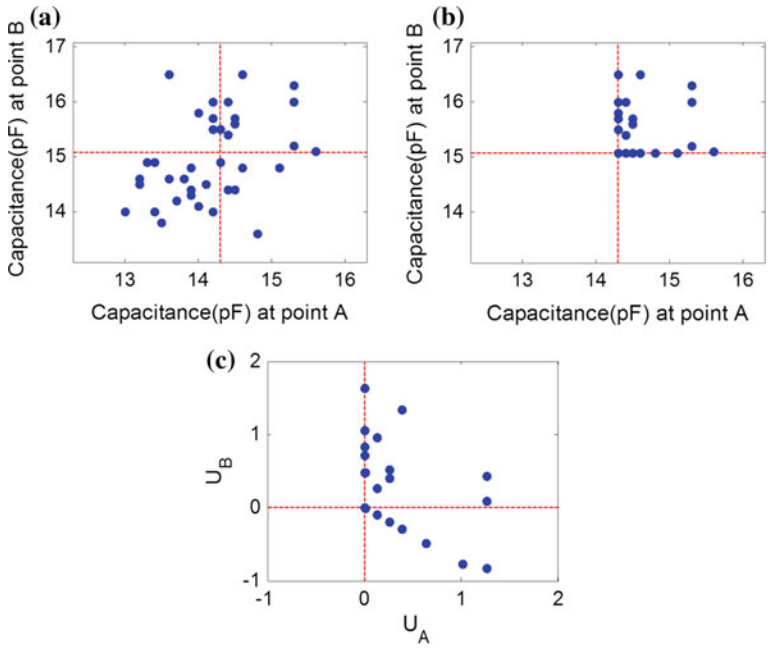


Fig. 9.15 Scatter plots **a** before the projection, **b** after projection, and **c** after transformation. Reprinted (adapted) with permission from Ref. [28]

Table 9.5 Definition of the health grades and suggested actions

Health grade	Range		Suggested actions
	Top	Bottom	
Faulty	$DMD^2 \geq 25$	$DMD^2 \geq 16.7$	Immediate replacement
Warning	$15 \leq DMD^2 < 25$	$10 \leq DMD^2 < 16.7$	Frequent inspection
Healthy	$DMD^2 < 15$	$DMD^2 < 10$	No schedule

The proposed health grade system and the suggested maintenance actions are expected to make it possible to carry out condition-based system maintenance. By providing power plant maintenance personnel with a quantitatively established maintenance guideline, the facility maintenance process would become much more systematic and objective.

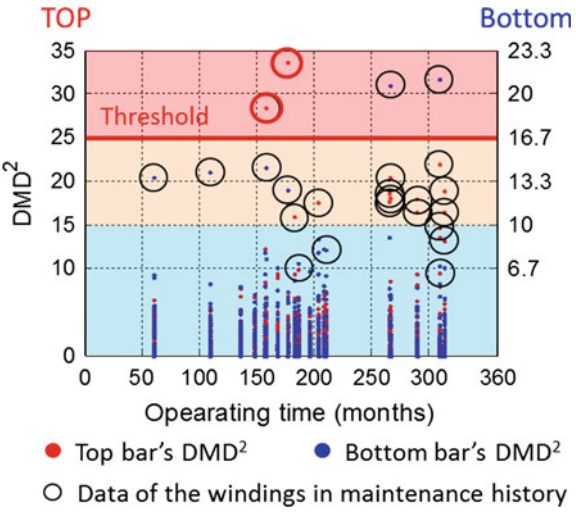


Fig. 9.16 Scatter plot of DMD against the operating time

9.5 Lithium-Ion Batteries

Lithium-ion (Li-ion) battery technology has played a critical role in realizing wide-scale adoption of hybrid and electric vehicles; it also offers great promise for emerging applications in smart grids and medical devices. Compared to conventional batteries, Li-ion batteries have a higher energy density. This feature contributes to their success but also raises safety concerns. For example, a Li-ion battery pack could rupture or even explode under short-circuit conditions. Due to their widespread use, failures of Li-ion batteries could result in enormous economic losses and/or catastrophic events. Over the past two decades, real-time health reasoning techniques have been developed and deployed in battery management systems (BMSs) to monitor the health condition of batteries in operation. Based on voltage, current, and temperature measurements acquired from the battery, these techniques estimate two performance indicators of the battery: state of charge (SOC) and state of health (SOH). Recently, health prognostic techniques have been developed, based on health reasoning, to predict RUL, i.e., when the battery is likely to fail.

9.5.1 Designing the Health Sensing Function

Most BMSs are capable of measuring three operating characteristics of Li-ion batteries: (terminal) voltage, current, and temperature. It is important to be able to measure these characteristics in real-time for two reasons. First, these sensory measurements serve as essential inputs for real-time health diagnostics and prognostics. Second, the performance of Li-ion batteries is dependent on each of these

operating characteristics. Specifically, Li-ion batteries must operate within a safe and reliable operating area, defined by simultaneously acceptable voltage, current, and temperature windows. When any of these are outside acceptable operating ranges there may be rapid degradation of battery performance and/or sudden catastrophic failure (e.g., fire, rupture, or explosion). As an example, we will next examine the effects of operating voltage on battery degradation. Current and temperature extremes can also result in failure.

The usual operating voltage window of most Li-ion batteries (e.g., $LiCoO_2$/graphite or LCO/C, $LiNi_{0.8}Co_{0.15}Al_{0.05}O_2$/graphite or NCA/C, and $LiCo_xNi_yMn_zO_2$/graphite or NCM/C) is between 2.5 and 4.2 V. If the operating voltage during charge exceeds the upper voltage limit (e.g., 4.2 V), the resulting overcharge condition will accelerate the oxidation and decomposition of the electrolyte components and the dissolution of the active material(s) in the positive electrode [29, 30]. Along with the damage induced to the electrolyte and positive electrode, the excessive charge current will cause lithium ions to be deposited as metallic lithium on the surface of the negative electrode. This is known as lithium plating. This degradation mechanism accelerates capacity fade, and may ultimately result in an internal short circuit between the two electrodes. If the operating voltage falls well below the lower voltage limit (e.g., 2.5 V), Li-ion cells will suffer from progressive loss of the negative electrode materials. Specifically, the copper current collector of the negative electrode will be dissolved into the electrolyte, which will cause delamination of the electrode materials from the current collector. When this occurs, the delamination causes a significant loss of active material and an increase of internal resistance.

9.5.2 Designing the Health Reasoning Function

In a Li-ion battery cell, the SOC quantifies the remaining charge of a cell relative to its fully charged capacity. The SOC of a cell changes very rapidly and, depending on the use condition, may traverse the entire range 100–0% within minutes. Capacity is an important SOH indicator [31, 32] that determines the maximum amount of charge that a fully charged battery can deliver. In contrast to the rapidly varying behavior of the SOC, the cell capacity tends to vary more slowly; it typically decreases 1.0% or less in a month with regular use. Given a discrete-time dynamic model that describes the electrical behavior of a cell and knowledge of the measured electrical signals, the health reasoning function aims to estimate the SOC and the capacity of the cell in a dynamic environment at every charge/discharge cycle. Using this information, it predicts how long the cell is expected to last before the capacity fade reaches an unacceptable level. The subsequent sections describe one case study that demonstrates this process.

Discrete-Time Cell Dynamic Model

SOC estimation is essential for capacity estimation. In order to estimate the SOC in a dynamic environment, we need a cell dynamic model that relates the SOC to the cell terminal voltage. Here we can use a simplified equivalent circuit model (or lumped parameter model), as shown in Fig. 9.17, which considers the effects of OCV, series resistance (R_s), diffusion resistance (R_d), and diffusion capacitance (C_d) [34]. The model expresses the cell terminal voltage as

$$V_k = \mathrm{OCV}(\mathrm{SOC}_k) - i_k \cdot R_s - V_{d,k} \tag{9.8}$$

where OCV is the open circuit voltage, i is the current, R_s is the series resistance, V_d is the diffusion voltage, and k is the index of the measurement time step. Since there is a strong correlation between the SOC and OCV, the SOC can be estimated from the OCV of the cell. The state transition equation of the diffusion voltage can be expressed as

$$V_{d,k+1} = V_{d,k} + \left(i_k - \frac{V_{d,k}}{R_d}\right) \cdot \frac{\Delta t}{C_d} \tag{9.9}$$

where R_d is the diffusion resistance, C_d the diffusion capacitance, and Δt is the length of the measurement interval. The time constant of the diffusion system can be expressed as $\tau = R_d C_d$. Note that, after a sufficiently long duration (e.g., 5τ) with a constant current i_k, the system reaches a final steady state with a final voltage $V_d = i_k \cdot R_d$ and the cell terminal voltage becomes $V_k = \mathrm{OCV}(\mathrm{SOC}_k) - i_k(R_s + R_d)$.

Given the measured electrical signals (i.e., cell current and terminal voltage) and the cell dynamic model, we can estimate the SOC by using one of the existing

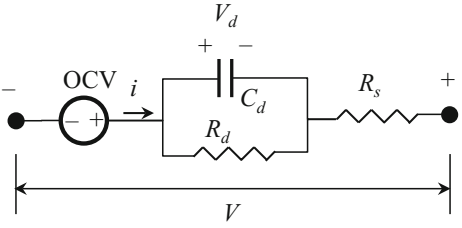


Fig. 9.17 A Li-ion battery equivalent circuit model (or lumped parameter model): open circuit voltage (OCV), series resistance (R_i), diffusion resistance (R_d), and diffusion capacitance (C_d). Reprinted (adapted) with permission from Ref. [33]

approaches, such as the extended/unscented Kalman filter [31, 32, 34, 35] and the coulomb counting technique [36].

Capacity Estimation with State Projection

Recent literature reports a variety of approaches for estimating the capacity and/or internal resistance of Li-ion batteries. In general, these approaches can be categorized into (1) adaptive filtering approaches [31, 32, 34–42], (2) coulomb counting approaches [36, 43], (3) neural network approaches [44–46], and (4) kernel regression approaches [47–50]. In what follows, we elaborate on a capacity estimation method that utilizes the SOC estimates before and after the state projection to estimate the capacity. Based on a capacity estimate C_k, the state projection scheme projects the SOC through a time span $L\Delta t$, expressed as [35]

$$\mathrm{SOC}_{k+L} = \mathrm{SOC}_k + \frac{\int_{t_k}^{t_{k+L}} i(t)dt}{C_k} \tag{9.10}$$

The effect of the capacity on the projected SOC is graphically explained in Fig. 9.18, where the projected SOCs with larger/smaller-than-true capacity estimates exhibit positive/negative deviations from their true values under a constant current discharge. This observation has two implications: (1) the capacity significantly affects the SOC estimation and inaccurate capacity estimation leads to inaccurate SOC estimation; and (2) the SOCs before and after the state projection, if accurately estimated based on the voltage and current measurements, can be used along with the net charge flow to back-estimate the capacity. The second implication can be mathematically expressed as

$$C_k = \frac{\int_{t_k}^{t_{k+L}} i(t)dt}{\mathrm{SOC}_{k+L} - \mathrm{SOC}_k} \tag{9.11}$$

At the start of every state projection (i.e., at the time t_k), an accurate SOC estimate is needed. This estimate will then be projected through the projection time span $L\Delta t$ according to the state projection equation in Eq. 9.11. Upon the completion of every state projection (i.e., at the time t_{k+L}), we also need to have an accurate SOC estimate to complete the capacity estimation. It is important to note that accuracy in the SOC estimation is a key factor that affects accuracy in the capacity estimation. In applications where the SOC estimates contain large measurement or estimation noise, the state projection expressed by Eq. 9.11 will result in inaccurate and biased capacity estimates, as also noted in [39]. In order to maintain an acceptable level of accuracy in capacity estimation in the presence of inaccurate SOC estimates, a large cumulated charge (i.e., the numerator in Eq. 9.11) is needed to compensate for the inaccuracy in the SOC estimation.

Li-ion cells for this case study were constructed in hermetically sealed prismatic cases between 2002 and 2012 and subjected to *full depth of discharge* cycling with a nominal weekly discharge rate (C/168 discharge) at 37 °C [33]. The cycling data from four 2002 cells was used to verify the effectiveness of the method for capacity

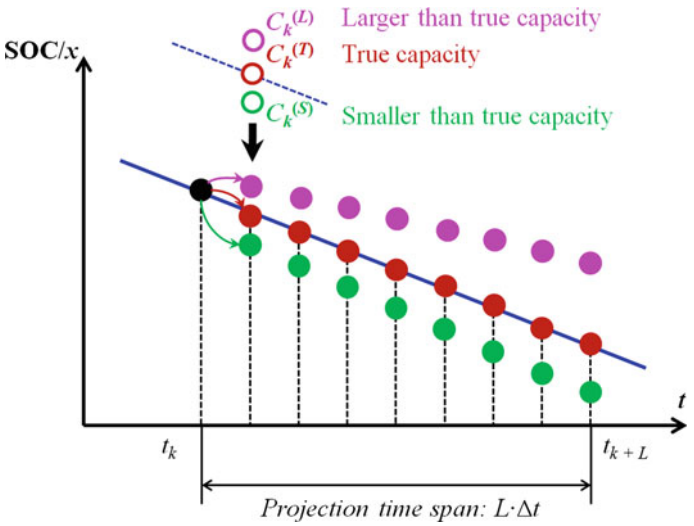


Fig. 9.18 Effect of capacity on state projection (assuming a constant current discharge). Reprinted (adapted) with permission from Ref. [33]

estimation. The cell discharge capacity was estimated based on the state projection scheme. In this study, an unknown SOC (or 1 − DOD) at a specific OCV level was approximated based on the cubic spline interpolation with a set of known OCV and SOC values (see the measurement points and interpolated curve in Fig. 9.19a). As shown in Fig. 9.19a, the state projection zone spans an OCV range of 4.0–3.7 V. In Fig. 9.19b, the net flow charge in this state projection zone was plotted as a function of cell discharge capacity for four test cells (cells 1–4) at eight different cycles spanning the whole 10-year test duration. The graph shows that the net flow charge is a linear

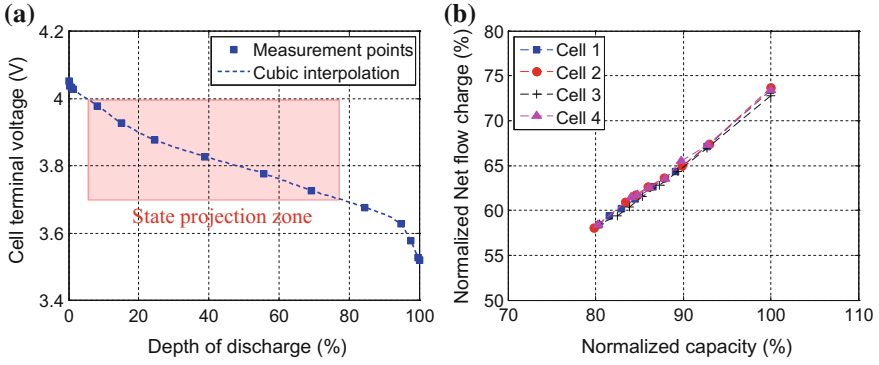


Fig. 9.19 **a** Plot of OCV as a function of DOD with state projection zone and **b** plot of normalized net flow discharge as a function of normalized discharge capacity. Reprinted (adapted) with permission from Ref. [33]

function of the cell discharge capacity. This observation suggests that a linear model can be generated to relate the capacity to the current integration. In fact, this linear model is exactly the one given in Eq. 9.11. With the SOCs at 4.0 and 3.7 V, derived based on the OCV-SOC relationship, and the net flow charge calculated by coulomb counting, the cell discharge capacity can be computed based on Eq. 9.11.

The capacity estimation results for the first two cells (i.e., Cells 1 and 2) are shown in Fig. 9.20. This figure shows that the capacity estimation method closely tracks the capacity fade trend throughout the cycling test for both cells. Table 9.6 summarizes the capacity estimation errors for the four cells. Here, the root mean square (RMS) and the maximum errors are formulated as

$$\varepsilon_{RMS} = \sqrt{\frac{1}{N_c}\sum_{i=1}^{N_c}\left(\Delta\hat{C}_i - \Delta C_i\right)^2}, \quad \varepsilon_{Max} = \max_{1 \le i \le N_c}\left|\Delta\hat{C}_i - \Delta C_i\right| \tag{9.12}$$

where N_C is the number of charge/discharge cycles; and ΔC_i and $\Delta\hat{C}_i$ are respectively the measured and estimated normalized capacities at the ith cycle. Observe that the average error is less than 1% for any of the four cells and the maximum error is less than 3%. The results suggest that the state projection method is capable of producing accurate and robust capacity estimation in the presence of cell-to-cell manufacturing variability.

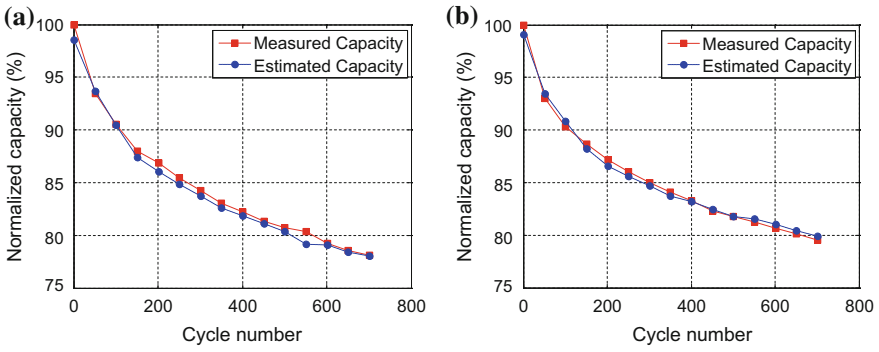


Fig. 9.20 Capacity estimation results of **a** cell 1 and **b** cell 2; results are plotted every 50 cycles for ease of visualization. Reprinted (adapted) with permission from Ref. [33]

Table 9.6 Capacity estimation results for the 4 test cells. Reprinted (adapted) with permission from Ref. [33]

Cell	Cell 1	Cell 2	Cell 3	Cell 4
RMS error (%)	0.52	0.51	0.88	0.52
Maximum error (%)	2.38	2.91	2.10	2.90

9.5.3 *Designing the Health Prognostics Function*

An RUL estimate for Li-ion batteries refers to the available service time or number of charge/discharge cycles left before the performance of the system degrades to an unacceptable level. Research on battery health prognostics has to date been mainly conducted by researchers in the prognostics and health management (PHM) society. For example, a Bayesian framework with a particle filter was proposed for prognostics (i.e., RUL prediction) of Li-ion batteries based on impedance measurements [51]. In order to eliminate the need for prognostics to rely on impedance measurement equipment, researchers have developed various model-based approaches that predict RUL by extrapolating a capacity fade model [52–55]. In addition, particle filters (or the sequential Monte Carlo methods) described in Sect. 8.4.1 can be used for online updating of a capacity fade model for lithium-ion batteries and for prediction of RUL using the updated model [33].

The RUL is used as the relevant metric for determining the state of life (SOL) of Li-ion batteries. Based on the capacity estimates obtained from the state projection scheme, the Gauss-Hermite particle filter technique is used to project the capacity fade trend to the end of life (EOL) value for the RUL prediction [33]. Here the EOL value is defined as 78.5% of the BOL discharge capacity. Figure 9.21a shows the capacity tracking and RUL prediction from the GHPF at cycle 200 (or 3.1 years). The figure shows that the predicted PDF of the life provides a slightly conservative solution and includes the true EOL cycle (i.e., 650 cycles or approximately 9.4 years). Figure 9.21b plots the RUL predictions from the GHPF at multiple cycles throughout the lifetime. The graph shows that, as we keep updating the RUL distribution throughout the battery lifetime, the prediction tends to converge to the true value as the battery approaches its EOL cycle.

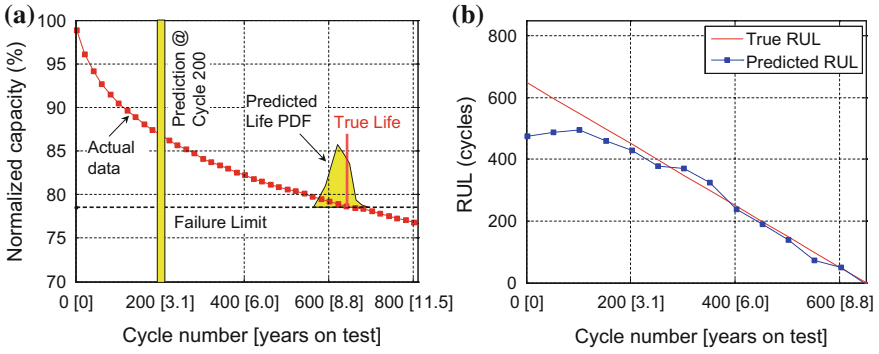


Fig. 9.21 RUL prediction results of cell 1; Figure **a** plots the capacity tracking and RUL prediction provided by the GHPF at cycle 200 (results are plotted every 20 cycles for ease of visualization) and **b** plots the RUL predictions from the GHPF at multiple cycles throughout the lifetime. Reprinted (adapted) with permission from Ref. [33]

9.6 Fuel Cells

Although fuel cells are promising candidates for future power generation, commercialization of fuel cells remains a future prospect due to safety concerns. Thus, accurate evaluation of the state of health (SOH) of fuel cells is necessary to enable condition-based maintenance to prevent impending catastrophic failures of the fuel cells. In this case study, the fuel cell is modeled by an equivalent circuit model (ECM) whose parameters are the health indicators. The reasoning function deals with the task of estimating the ECM parameters. The reasoning function can be attained by either measuring or estimating the impedance spectrum of the fuel cell. In the prognostic function, the ECM parameters are estimated for a future time with the help of voltage estimation.

9.6.1 Overview of Fuel Cells

This section provides a basic overview of fuel cells. The basic structure of a fuel cell contains an anode, a cathode, and electrolyte, as shown in Fig. 9.22. The hydrogen is supplied from the anode side. It reacts with the oxygen coming from the cathode side, and this reaction produces water. This overall reaction is made up of two reactions.

$$\begin{aligned} \mathbf{H_2} &\rightarrow \boldsymbol{2H^+ + 2e^-} \\ \boldsymbol{\tfrac{1}{2}O_2 + 2H^+ + 2e^-} &\rightarrow \boldsymbol{H_2O} \end{aligned} \tag{9.13}$$

The first reaction, separation of a hydrogen molecule into hydrogen ions and electrons, occurs at the interface between the anode and the electrolyte. The hydrogen ions then move to the cathode through the electrolyte to complete the reaction. Since the electrolyte only allows the ions to pass, the electrons transfer through an external wire and provide energy to the load on the way to the cathode. This process continues as long as the hydrogen fuel supply remains.

During operation of a fuel cell, individual components of the fuel cell are subject to degradation. The degradation rate accelerates during particular working steps,

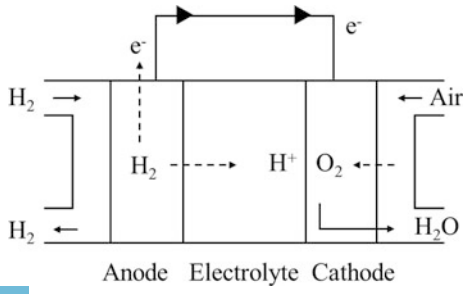


Fig. 9.22 Basic structure of a fuel cell

such as during the transportation of the reactants, and during charge transfers. It also varies based on the rate of the electrochemical reactions. Variation in the degradation rates of fuel cells is due to the varied degradation of each component in the fuel cell; membrane, catalyst layer, gas diffusion layer, and bipolar plates. For example, the membrane can undergo mechanical, thermal, and electrochemical degradation due to non-uniform stresses during the assembly or as cycling hydration induces mechanical stress on the membrane. Also, increased operating temperatures due to reactant crossover via pinholes or perforations decompose the membrane. Further, the radicals generated by the undesirable side reaction diminish the membrane. In the electrocatalyst layer, the detachment, dissolution of the catalyst, and growth in catalyst particle reduce the catalytic area and, thus, reduce the catalyst activity. Also, corrosion can occur in the gas diffusion layer and bipolar plates. Degradation of individual components, of course, occurs in a combined manner. The electrochemical side reaction weakens the mechanical strength of the membrane. Mechanical degradation brings about local pinholes and results in thermal degradation due to the crossover of the reactant, which again enhances the side reaction rate. This complex degradation must be prevented before it ends up resulting in a catastrophe, such as an explosion. Prevention starts from knowing the current status, and furthermore by predicting the future status of the fuel cells.

9.6.2 Designing the Health Reasoning Function

In general, diagnostics of a fuel cell is conducted using in-situ characterization techniques. There are several characterization techniques used to estimate the properties of the fuel cells, including: the kinetic property, ohmic property, mass transport property, and so on. Well-known techniques include the polarization curve, current interrupt method, and the electrochemical impedance spectroscopy (EIS) method. Among the methods, the EIS test, when combined with the ECM, gives abundant information about the fuel cells. This section briefly explains the use of the EIS method for diagnosing fuel cells.

The EIS test measures the impedance, which is the ratio of the time-dependent voltage to the time-dependent current, at various frequencies. According to the fuel cell's properties, various shapes of impedance plots are obtained. For example, Fig. 9.23 shows a typical impedance spectrum of a fuel cell. The radii of the three semi-circles correspond to the anode activation loss, the cathode activation loss, and the mass transport loss, respectively. Also, the amount of the curve shift from the origin designates the ohmic loss. The impedance spectrum can be interpreted in a quantitative way with the help of equivalent circuit modeling. The equivalent circuit model is used to describe the impedance spectral behavior. The anode and cathode activation losses are represented by the parallel connection of the resistor and the capacitor. R_A and R_C indicate the kinetic resistance of the electrochemical reaction, and C_A and C_C show the charge separation at the interface between the electrode and the electrolyte. The ohmic resistance is represented as R_o, and the mass

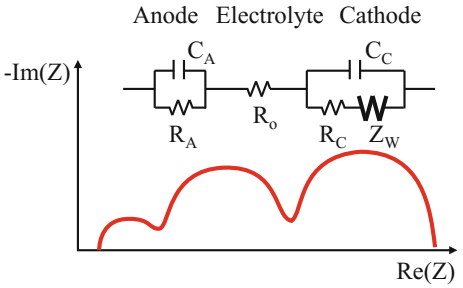


Fig. 9.23 Structure of a fuel cell

transport is modeled using the Warburg element, Z_W. Measuring the impedance spectrum and fitting the ECM gives the parameters of the ECM; these parameters tell us the main problem that the fuel cell is experiencing, and how much the fuel cell suffers from it.

In spite of the robust features of the EIS method, obtaining the impedance spectrum of a fuel cell requires an expensive impedance analyzer and stable measurement conditions. These requirements make it difficult for EIS to be applicable for online use. Thus, broadband current interruption techniques have been developed to overcome the shortcomings of the EIS method. These methods utilize current interrupting waveforms, such as pseudo-random binary sequence (PRBS) and multi-sine signals, imposed on the operating current. They estimate the impedance spectrum by analyzing the voltage response to the interrupting current. These methods reduce the measurement time and extract information similar to that found through EIS measurements.

9.6.3 Designing the Health Prognostics Function

The health indicator defined and estimated in the reasoning stage can be extrapolated to predict the future status of the fuel cell. This section explains the process for life prognosis using predictions of the health parameters of the equivalent circuit.

One way of predicting the health parameters is to combine the health parameters with a physical property of the fuel cell and induce its health parameters [56]. In fuel cells, voltage is a well-known lumped health indicator, and it can be related to the health parameters of the equivalent circuit. An example is shown in Fig. 9.24.

Figure 9.24a shows the equivalent circuit of the fuel cell and its parameters, Fig. 9.24b shows the voltage degradation curve, and Fig. 9.24c depicts the relation between one of the model parameters and the voltage. From this relation, and with the help of the voltage prediction, the health parameters for future status can be estimated.

The voltage is modeled through two degradation models: reversible and irreversible. The reversible degradation is temporary and recoverable. An example of a reversible degradation is water clogging in flow channels. Irreversible degradation

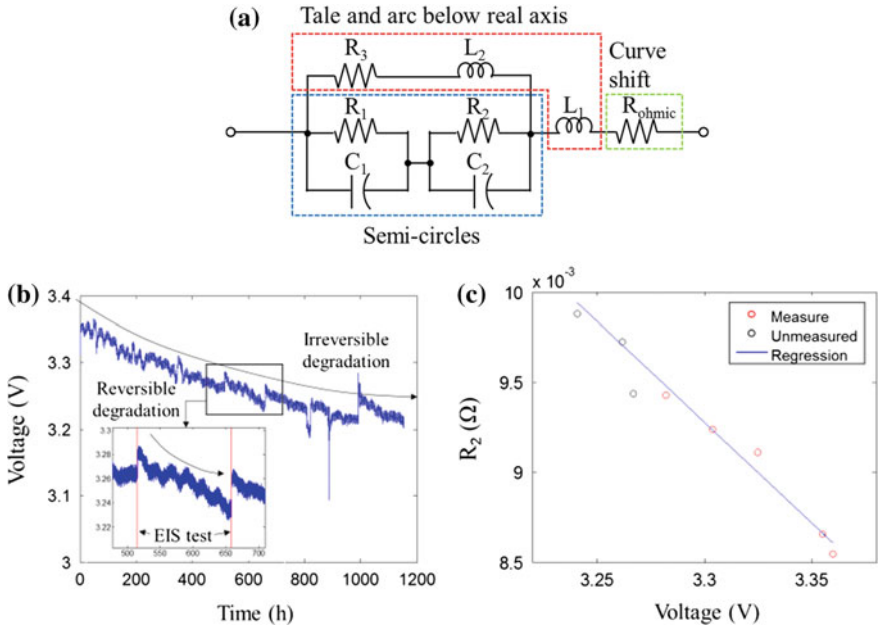


Fig. 9.24 **a** Equivalent circuit model, **b** measured voltage, **c** relationship between the voltage and ECM parameter. Reprinted (adapted) with permission from Ref. [56]

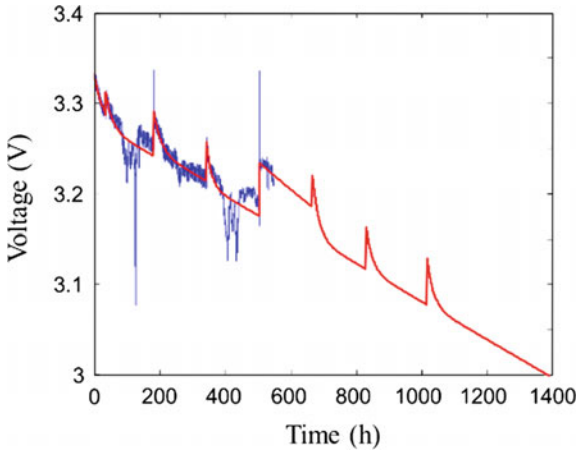


Fig. 9.25 Prediction of voltage (red) with voltage data (blue). Reprinted (adapted) with permission from Ref. [56] (color figure online)

is permanent damage to the fuel cell, such as a melted membrane. Reversible and irreversible degradation are modeled by exponential and linear models, respectively, as shown in Fig. 9.25.

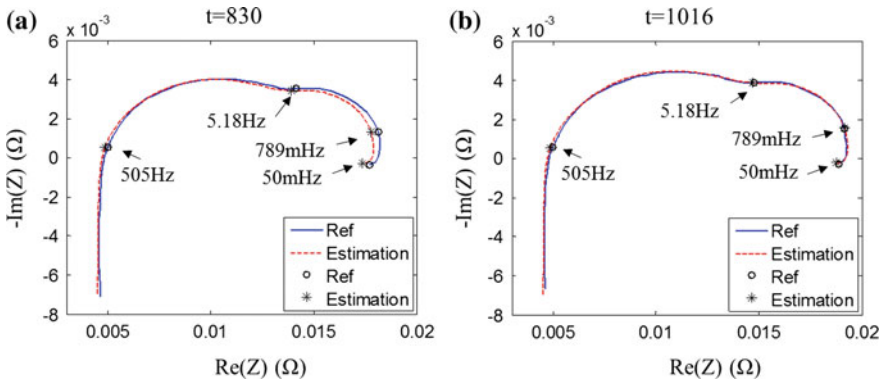


Fig. 9.26 Predicted impedance spectrum results of FC2 at time **a** 830, and **b** at time 1016 h. Reprinted (adapted) with permission from Ref. [56]

Next, the predicted voltage is used for estimating the health parameters. The reconstructed impedance spectrum using the estimated parameters is compared with the measured impedance spectrum and the comparison results are shown in Fig. 9.26.

9.7 Pipelines

Pipelines are essential infrastructures in our modern society. Pipelines play crucial roles in transporting various fluids, such as oil, water, and gas, from storage to users. Leaks in pipelines result in economic, environmental, and social problems. Thus, accurate detection and quick maintenance should be done to prevent these problems. However, due to the characteristics of pipelines, the detection of leaks is not easy. For example, accessibility of the pipeline is generally limited due to installation conditions, which involve long distances that often include underground, underwater, or alpine locations. To overcome these challenges, various leak detection techniques, such as ground penetrating radar (GPR) [57], leak noise correlators (LNC) [58], and pig-mounted acoustic (PMA) sensing [59], have been developed over many years. In this section, a real-time and remote monitoring detection method is introduced. This method uses a time-domain reflectometry (TDR) technique that can stochastically detect multiple leaks using forward and inverse models [60]. This method was validated through a case study that was performed in a water distribution system. The water distribution system is the most typical pipeline system. This TDR-based method is expected to be applicable to not only water pipelines but also to other pipelines in general.

9.7.1 Designing the Health Sensing Function

TDR was originally proposed as a method for locating faults on a transmission line, such as electrical open, short, or chafe. The principle of TDR is similar to RADAR, which finds the location of an object by measuring a reflected radio wave. Likewise, a TDR device propagates an incident pulse along the transmission line. When the pulse meets any fault on the transmission line, the pulse is reflected at that location and returns to the device. Thus, by measuring the travel time of the pulse from departure to arrival, the fault location can be estimated by multiplying the propagation velocity of the pulse. The cause of the reflection is the nonconformity of impedance in the transmission line. The change of impedance is caused by a fault, such as a short, open, or chafe of the transmission line. The shape and degree of reflection are represented by the reflection coefficient, Γ, as shown in Fig. 9.27.

The reflection coefficient, Γ, is expressed as:

$$\Gamma_1 = \frac{Z_F - Z_0}{Z_F + Z_0} \tag{9.14}$$

where Z_0 is the characteristic impedance of the transmission line and Z_F is the fault impedance. If Γ is less than zero, a wave shape of the reflected pulse is upside down against the incident pulse, which indicates an electrical short. If the Γ is 1, the wave shape of the reflected pulse is same as that of the incident pulse, which specifically indicates an electrical open condition, as shown in Fig. 9.28.

Similarly, using these electric characteristics, TDR-based leak detection methods have been developed by correlating water leakage with an electric short in a transmission line. To generate an electric short at the leak position, a *leak detector* is used, as shown in Fig. 9.29. The device is connected to a transmission line attached to the pipeline. The leak detector consists of two copper plates and a plastic case with holes. As leaking water contacts the copper plates, a pulse signal is reflected due to the resulting electric short.

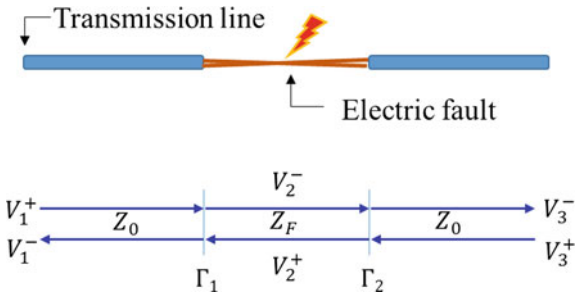


Fig. 9.27 An impedance disparity and the reflected coefficient

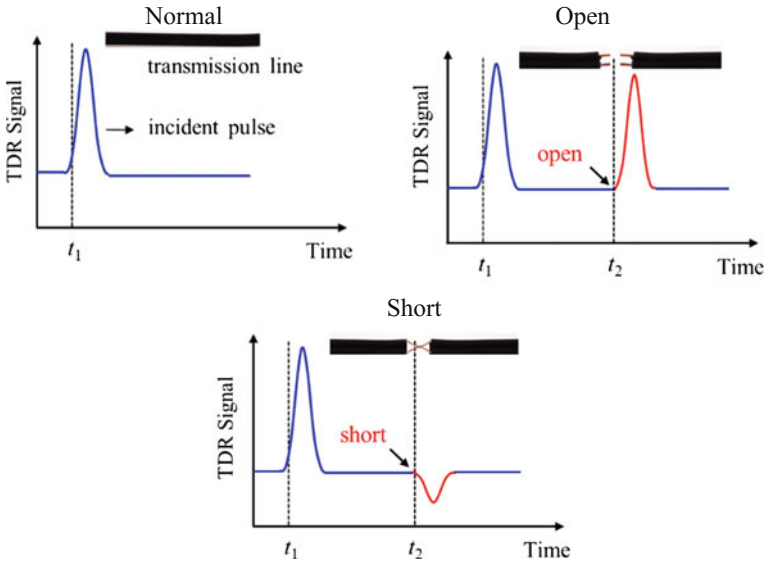


Fig. 9.28 TDR signals according to transmission line conditions

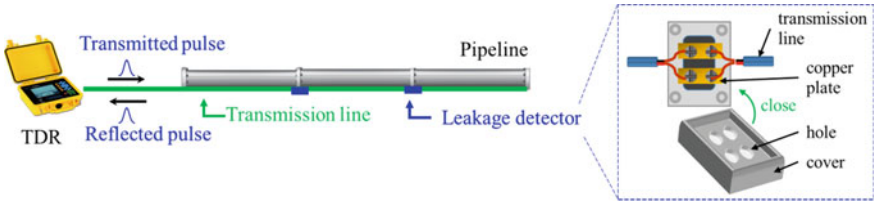


Fig. 9.29 Concept of pipeline leak detection using a TDR and leak detectors

9.7.2 Designing the Health Reasoning Function

Even with the measured TDR signals, analysis of the reflected pulse signal for accurate leak detection is not a simple matter. This is because ambient noise and overlapped pulses can result from multiple leaks. Thus, both forward and inverse models can be applied to accurately interpret the measured TDR signal. This model-based leak detection method consists of three steps: (1) emulating possible TDR signals through the forward model, (2) estimating the correlation of the emulated signal with the measured one, and (3) determining the number of leaks and their locations, as shown in Fig. 9.30. The first step is to simulate the TDR signals for all possible leak situations using a forward model. The second step is to calculate the correlation between the simulated and measured TDR signals using a likelihood metric. Third, Bayesian inference is used to determine the number of leaks, of which the simulated signal gives the maximum likelihood.

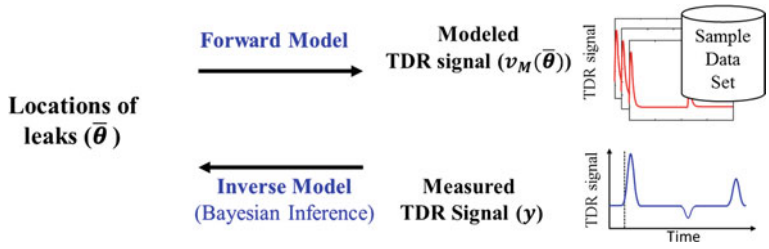


Fig. 9.30 Framework for the model-based leak detection method

The forward model produces simulated TDR signals as a function of model input parameters, such as the physical properties of the transmission line and leak information (e.g., the number of leaks and their locations). The forward model is generally derived using RLCG circuit theory or S-parameters. It is known that the S-parameter model is computationally far more efficient than the RLCG signal modeling method.

The inverse model employs Bayesian inference, which infers the posterior distributions of the model parameters from the prior distributions. The prior distribution of a leak's location assumes a uniform distribution. Then, for all possible leak situations, the likelihood function in Eq. 9.15 can be calculated by comparing the simulated TDR signals (v_m) with the measured TDR signal (y) [61]. Bayesian inference is employed to determine the number of leaks and their locations. The simulated signal yields the highest value of $\Pr(\theta|y)$, and θ is the leak location.

$$\mathbf{Pr}(\boldsymbol{y}|\boldsymbol{\theta}) = (\boldsymbol{2\pi\sigma_M})^{-m/2} \boldsymbol{exp}\left(-\frac{1}{\boldsymbol{2\sigma_M}} \parallel \boldsymbol{y} - \boldsymbol{v_m} \parallel^2\right) \tag{9.15}$$

Lab-scale experiments were carried out for validation of this leak detection system. The test bed consisted of a 10 m pipeline, a transmission line, a leak detector, and a TDR device, as shown in Fig. 9.31a. The experiment was conducted with three water leaks present, specifically at 5.5, 6, and 8 m. Figure 9.31b displays the signal acquired from the experiment. As shown in the figure, this signal does not easily identify leaks or locations.

Figure 9.31c displays the three leaks and their locations that were correctly identified using Bayesian inference. It shows the marginal PDFs of the leak locations.

9.8 Summary

Four core functions in PHM have been successfully applied to engineering applications. This chapter presented several case studies that explain successful PHM practices: (1) steam turbine rotors, (2) wind turbine gearboxes, (3) the core and

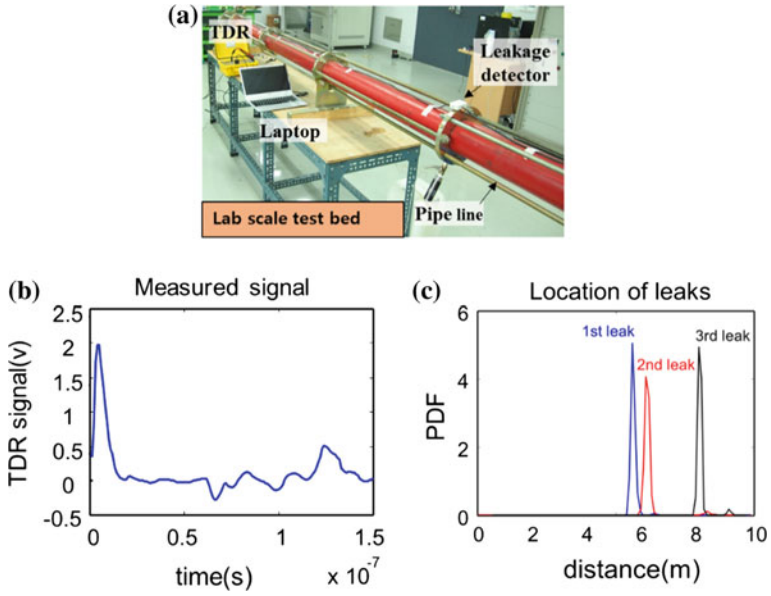


Fig. 9.31 **a** Test bed, **b** measured TDR signal, and **c** result of the Bayesian inference

windings in power transformers, (4) power generator stator windings, (5) lithium-ion batteries, (6) fuel cells, and (7) water pipelines. These examples provide useful findings about the four core functions of PHM technology, contemporary technology trends, and industrial values. PHM offers great economic value to various industries through condition-based maintenance (CBM), which helps operators schedule maintenance actions based on the analyzed conditions of engineered systems. Operators can minimize unnecessary maintenance actions and can prevent catastrophic system failures by reliably predicting a system's health condition. CBM can substantially save on overall operation and maintenance (O&M) costs.

References

1. Gupta, K. (1997). Vibration—A tool for machine diagnostics and condition monitoring. *Sadhana, 22,* 393–410.
2. Lin, J., & Qu, L. S. (2000). Feature extraction based on Morlet wavelet and its application for mechanical fault diagnosis. *Journal of Sound and Vibration, 234,* 135–148.
3. Jeon, B., Jung, J., Youn, B., Kim, Y.-W., & Bae, Y.-C. (2015). Datum unit optimization for robustness of a journal bearing diagnosis system. *International Journal of Precision Engineering and Manufacturing, 16*, 2411–2425.
4. Bonnardot, F., El Badaoui, M., Randall, R., Daniere, J., & Guillet, F. (2005). Use of the acceleration signal of a gearbox in order to perform angular resampling (with limited speed fluctuation). *Mechanical Systems and Signal Processing, 19,* 766–785.

5. Villa, L. F., Reñones, A., Perán, J. R., & de Miguel, L. J. (2011). Angular resampling for vibration analysis in wind turbines under non-linear speed fluctuation. *Mechanical Systems and Signal Processing, 25*, 2157–2168, 8.
6. Han, T., Yang, B.-S., Choi, W.-H., & Kim, J.-S. (2006). Fault diagnosis system of induction motors based on neural network and genetic algorithm using stator current signals. *International Journal of Rotating Machinery, 2006*.
7. Yang, B.-S., & Kim, K. J. (2006). Application of Dempster-Shafer theory in fault diagnosis of induction motors using vibration and current signals. *Mechanical Systems and Signal Processing, 20,* 403–420.
8. Jung, J.H., Jeon, B.C., Youn, B.D., Kim, M., Kim, D. & Kim, Y., (2017). Omnidirectional regeneration (ODR) of proximity sensor signals for robust diagnosis of journal bearing systems. *Mechanical Systems and Signal Processing*, *90*, 189–207.
9. Guo, B., Damper, R. I., Gunn, S. R., & Nelson, J. D. B. (2008). A fast separability-based feature-selection method for high-dimensional remotely sensed image classification. *Pattern Recognition, 41*, 1653–1662, 5.
10. Abbasion, S., Rafsanjani, A., Farshidianfar, A., & Irani, N. (2007). Rolling element bearings multi-fault classification based on the wavelet denoising and support vector machine. *Mechanical Systems and Signal Processing, 21,* 2933–2945.
11. Widodo, A., & Yang, B.-S. (2007). Support vector machine in machine condition monitoring and fault diagnosis. *Mechanical Systems and Signal Processing, 21,* 2560–2574.
12. Heng, A., Zhang, S., Tan, A. C. C., & Mathew, J. (2009). Rotating machinery prognostics: State of the art, challenges and opportunities. *Mechanical Systems and Signal Processing, 23*, 724–739, 4.
13. Zaher, A., McArthur, S., Infield, D., & Patel, Y. (2009). Online wind turbine fault detection through automated SCADA data analysis. *Wind Energy, 12,* 574–593.
14. Tchakoua, P., Wamkeue, R., Ouhrouche, M., Slaoui-Hasnaoui, F., Tameghe, T. A., & Ekemb, G. (2014). Wind turbine condition monitoring: State-of-the-art review, new trends, and future challenges. *Energies, 7,* 2595–2630.
15. Samuel, P. D., Conroy, J. K., & Pines, D. J. (2004). Planetary transmission diagnostics. *NASA CR, 213068*.
16. Lebold, M., McClintic, K., Campbell, R., Byington, C., & Maynard, K. (2000). Review of vibration analysis methods for gearbox diagnostics and prognostics. In *Proceedings of the 54th Meeting of the Society for Machinery Failure Prevention Technology* (p. 16).
17. Lau, B. C. P., Ma, E. W. M., & Pecht, M. (2012). Review of offshore wind turbine failures and fault prognostic methods. In *2012 IEEE Conference on Prognostics and System Health Management (PHM)* (pp. 1–5).
18. Feldman, K., Jazouli, T., & Sandborn, P. A. (2009). A methodology for determining the return on investment associated with prognostics and health management. *IEEE Transactions on Reliability, 58,* 305–316.
19. Nilsson, J., & Bertling, L. (2007). Maintenance management of wind power systems using condition monitoring systems—Life cycle cost analysis for two case studies. *IEEE Transactions on Energy Conversion, 22,* 223–229.
20. Hu, C., Wang, P., Youn, B. D., Lee, W.-R., & Yoon, J. T. (2012). Copula-based statistical health grade system against mechanical faults of power transformers. *IEEE Transactions on Power Delivery, 27,* 1809–1819.
21. Shengchang, J., Yongfen, L., & Yanming, L. (2006). Research on extraction technique of transformer core fundamental frequency vibration based on OLCM. *IEEE Transactions on Power Delivery, 21,* 1981–1988.
22. Wang, P., Youn, B. D., Hu, C., Ha, J. M., & Jeon, B. (2014). A probabilistic detectability-based sensor network design method for system health monitoring and prognostics. *Journal of Intelligent Material Systems and Structures*, 1045389X14541496.
23. Hu, C., Youn, B. D., Wang, P., & Yoon, J. T. (2012). Ensemble of data-driven prognostic algorithms for robust prediction of remaining useful life. *Reliability Engineering & System Safety, 103,* 120–135.

24. Inoue, Y., Hasegawa, H., Sekito, S., Sotodate, M., Shimada, H., & Okamoto, T. (2003). Technology for detecting wet bars in water-cooled stator windings of turbine generators. In *Electric Machines and Drives Conference, 2003. IEMDC'03. IEEE International* (pp. 1337–1343).
25. Kim, H. S., Bae, Y. C., & Kee, C. D. (2008). Wet bar detection by using water absorption detector. *Journal of Mechanical Science and Technology, 22,* 1163–1173.
26. Park, K. M., Youn, B. D., Yoon, J. T., Hu, C., Kim, H. S., & Bae, Y. C. (2013). Health diagnostics of water-cooled power generator stator windings using a Directional Mahalanobis Distance (DMD). In *2013 IEEE Conference on Prognostics and Health Management (PHM)* (pp. 1–8).
27. Wang, Y., Miao, Q., & Pecht, M. (2011). Health monitoring of hard disk drive based on Mahalanobis distance. In *Prognostics and System Health Management Conference (PHM-Shenzhen), 2011* (pp. 1–8).
28. Youn, B. D., Park, K. M., Hu, C., Yoon, J. T., Kim, H. S., Jang, B. C., et al. (2015). Statistical health reasoning of water-cooled power generator stator bars against moisture absorption. *IEEE Transactions on Energy Conversion, 30*(4), 1376–1385.
29. Arora, P., White, R. E., & Doyle, M. (1998). Capacity fade mechanisms and side reactions in lithium-ion batteries. *Journal of the Electrochemical Society, 145,* 3647–3667.
30. Vetter, J., Novák, P., Wagner, M., Veit, C., Möller, K.-C., Besenhard, J., et al. (2005). Ageing mechanisms in lithium-ion batteries. *Journal of Power Sources, 147,* 269–281.
31. Plett, G. L. (2004). Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 3. State and parameter estimation. *Journal of Power Sources, 134,* 277–292.
32. Plett, G. L. (2006). Sigma-point Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 2: Simultaneous state and parameter estimation. *Journal of Power Sources, 161,* 1369–1384.
33. Hu, C., Jain, G., Tamirisa, P., & Gorka, T. (2014). Method for estimating capacity and predicting remaining useful life of lithium-ion battery. *Applied Energy, 126,* 182–189.
34. Lee, S., Kim, J., Lee, J., & Cho, B. (2008). State-of-charge and capacity estimation of lithium-ion battery using a new open-circuit voltage versus state-of-charge. *Journal of Power Sources, 185,* 1367–1373.
35. Hu, C., Youn, B. D., & Chung, J. (2012). A multiscale framework with extended Kalman filter for lithium-ion battery SOC and capacity estimation. *Applied Energy, 92,* 694–704.
36. Ng, K. S., Moo, C.-S., Chen, Y.-P., & Hsieh, Y.-C. (2009). Enhanced coulomb counting method for estimating state-of-charge and state-of-health of lithium-ion batteries. *Applied Energy, 86,* 1506–1511.
37. Chiang, Y.-H., Sean, W.-Y., & Ke, J.-C. (2011). Online estimation of internal resistance and open-circuit voltage of lithium-ion batteries in electric vehicles. *Journal of Power Sources, 196,* 3921–3932.
38. He, W., Williard, N., Chen, C., & Pecht, M. (2013). State of charge estimation for electric vehicle batteries using unscented Kalman filtering. *Microelectronics Reliability, 53,* 840–847.
39. Plett, G. L. (2011). Recursive approximate weighted total least squares estimation of battery cell total capacity. *Journal of Power Sources, 196,* 2319–2331.
40. Schmidt, A. P., Bitzer, M., Imre, Á. W., & Guzzella, L. (2010). Model-based distinction and quantification of capacity loss and rate capability fade in Li-ion batteries. *Journal of Power Sources, 195,* 7634–7638.
41. Verbrugge, M. (2007). Adaptive, multi-parameter battery state estimator with optimized time-weighting factors. *Journal of Applied Electrochemistry, 37,* 605–616.
42. Xiong, R., Sun, F., Chen, Z., & He, H. (2014). A data-driven multi-scale extended Kalman filtering based parameter and state estimation approach of lithium-ion olymer battery in electric vehicles. *Applied Energy, 113,* 463–476.
43. Waag, W., & Sauer, D. U. (2013). Adaptive estimation of the electromotive force of the lithium-ion battery after current interruption for an accurate state-of-charge and capacity determination. *Applied Energy, 111,* 416–427.

44. Bai, G., Wang, P., Hu, C., & Pecht, M. (2014). A generic model-free approach for lithium-ion battery health management. *Applied Energy, 135,* 247–260.
45. Kim, J., Lee, S., & Cho, B. (2012). Complementary cooperation algorithm based on DEKF combined with pattern recognition for SOC/capacity estimation and SOH prediction. *IEEE Transactions on Power Electronics, 27,* 436–451.
46. Eddahech, A., Briat, O., Bertrand, N., Delétage, J.-Y., & Vinassa, J.-M. (2012). Behavior and state-of-health monitoring of Li-ion batteries using impedance spectroscopy and recurrent neural networks. *International Journal of Electrical Power & Energy Systems, 42,* 487–494.
47. Pattipati, B., Sankavaram, C., & Pattipati, K. (2011). System identification and estimation framework for pivotal automotive battery management system characteristics. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 41*, 869–884.
48. Nuhic, A., Terzimehic, T., Soczka-Guth, T., Buchholz, M., & Dietmayer, K. (2013). Health diagnosis and remaining useful life prognostics of lithium-ion batteries using data-driven methods. *Journal of Power Sources, 239,* 680–688.
49. Widodo, A., Shim, M.-C., Caesarendra, W., & Yang, B.-S. (2011). Intelligent prognostics for battery health monitoring based on sample entropy. *Expert Systems with Applications, 38,* 11763–11769.
50. Hu, C., Jain, G., Zhang, P., Schmidt, C., Gomadam, P., & Gorka, T. (2014). Data-driven method based on particle swarm optimization and k-nearest neighbor regression for estimating capacity of lithium-ion battery. *Applied Energy, 129,* 49–55.
51. Saha, B., Goebel, K., Poll, S., & Christophersen, J. (2009). Prognostics methods for battery health monitoring using a Bayesian framework. *IEEE Transactions on Instrumentation and Measurement, 58,* 291–296.
52. Ng, S. S., Xing, Y., & Tsui, K. L. (2014). A naive Bayes model for robust remaining useful life prediction of lithium-ion battery. *Applied Energy, 118,* 114–123.
53. Wang, D., Miao, Q., & Pecht, M. (2013). Prognostics of lithium-ion batteries based on relevance vectors and a conditional three-parameter capacity degradation model. *Journal of Power Sources, 239,* 253–264.
54. Liu, J., Saxena, A., Goebel, K., Saha, B., & Wang, W. (2010). *An adaptive recurrent neural network for remaining useful life prediction of lithium-ion batteries*. DTIC Document 2010.
55. Saha, B., & Goebel, K. (2009). Modeling Li-ion battery capacity depletion in a particle filtering framework. In *Proceedings of the Annual Conference of the Prognostics and Health Management Society*, 2009 (pp. 2909–2924).
56. Kim, T., Oh, H., Kim, H., & Youn, B. D. (2016). An online-applicable model for predicting health degradation of PEM fuel cells with root cause analysis. *IEEE Transactions on Industrial Electronics, 63*(11), 7094–7103.
57. Demirci, S., Yigit, E., Eskidemir, I. H., & Ozdemir, C. (2012). Ground penetrating radar imaging of water leaks from buried pipes based on back-projection method. *NDT and E International, 47,* 35–42.
58. Gao, Y., Brennan, M., Joseph, P., Muggleton, J., & Hunaidi, O. (2004). A model of the correlation function of leak noise in buried plastic pipes. *Journal of Sound and Vibration, 277,* 133–148.
59. McNulty, J. (2001). An acoustic-based system for detecting, locating and sizing leaks in water pipelines. In *Proceedings of the 4th International Conference on Water Pipeline Systems: Managing Pipeline Assets in an Evolving Market.* New York, UK, 2001.
60. Kim, T., Woo, S., Youn, B. D., & Huh, Y. C. (2015). TDR-based pipe leakage detection and location using Bayesian inference. In *2015 IEEE Conference on Prognostics and Health Management (PHM)* (pp. 1–5).
61. Schuet, S., Timucin, D., & Wheeler, K. (2011). A model-based probabilistic inversion framework for characterizing wire fault detection using TDR. *IEEE Transactions on Instrumentation and Measurement, 60,* 1654–1663.