ABSTRACT

Title of Document: ADVANCES IN SYSTEM RELIABILITY-BASED DESIGN AND PROGNOSTICS AND HEALTH MANAGEMENT (PHM) FOR SYSTEM RESILIENCE ANALYSIS AND DESIGN.

Chao Hu, Doctor of Philosophy, 2011

Directed By: Dr. Byeng D. Youn,
Assistant Professor,
Department of Mechanical Engineering.

Failures of engineered systems can lead to significant economic and societal losses. Despite tremendous efforts (e.g., $200 billion annually) denoted to reliability and maintenance, unexpected catastrophic failures still occurs. To minimize the losses, reliability of engineered systems must be ensured throughout their life-cycle amidst uncertain operational condition and manufacturing variability. In most engineered systems, the required system reliability level under adverse events is achieved by adding system redundancies and/or conducting system reliability-based design optimization (RBDO). However, a high level of system redundancy increases a system's life-cycle cost (LCC) and system RBDO cannot ensure the system reliability when unexpected loading/environmental conditions are applied and

unexpected system failures are developed. In contrast, a new design paradigm, referred to as resilience-driven system design, can ensure highly reliable system designs under any loading/environmental conditions and system failures while considerably reducing systems' LCC.

In order to facilitate the development of formal methodologies for this design paradigm, this research aims at advancing two essential and co-related research areas: Research Thrust 1 – system RBDO and Research Thrust 2 – system prognostics and health management (PHM). In Research Thrust 1, reliability analyses under uncertainty will be carried out in both component and system levels against critical failure mechanisms. In Research Thrust 2, highly accurate and robust PHM systems will be designed for engineered systems with a single or multiple time-scale(s). To demonstrate the effectiveness of the proposed system RBDO and PHM techniques, multiple engineering case studies will be presented and discussed. Following the development of Research Thrusts 1 and 2, Research Thrust 3 – resilience-driven system design will establish a theoretical basis and design framework of engineering resilience in a mathematical and statistical context, where engineering resilience will be formulated in terms of system reliability and restoration and the proposed design framework will be demonstrated with a simplified aircraft control actuator design problem.

ADVANCES IN SYSTEM RELIABILITY-BASED DESIGN AND PROGNOSTICS AND HEALTH MANAGEMENT (PHM) FOR SYSTEM RESILIENCE ANALYSIS AND DESIGN

By

Chao Hu

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2011

Advisory Committee:
Assistant Prof. Byeng D. Youn (Chair/Advisor)
Prof. Bongtae Han (Co-Advisor)
Prof. Mohammad Modarres
Associate Prof. Michel Cukier
Associate Prof. Radu Balan (Dean's Representative)

Acknowledgements

First, I would like to express my sincere gratitude and appreciation to my advisor, Dr. Byeng D. Youn, for his insightful guidance and persistent support on my research throughout my graduate studies. I sincerely appreciate his dedication and guidance that endowed me with the confidence to explore previously uncharted scientific territory, the diligence and rigorous altitude to advance cutting-edge research, and the writing and presentation skills to communicate my research outcomes. I have a strong belief that all that I learned from him will have a significant impact on my future career development.

I would also like to express my thanks to my dissertation committee members, Dr. Bongtae Han as my co-advisor, Dr. Mohammad Modarres, Dr. Michel Cukier, and Dr. Radu Balan, for their valuable comments and suggestions, without which this dissertation cannot be completed.

I am also thankful to my colleagues and friends, Dr. Soobum Lee, Dr. Zhimin Xi, Dr. Pingfeng Wang, Mr. Byungchang Jung, Ms. Lulu Wang, Mr. Taejin Kim, Ms. Jisun Kim, Mr. Heonjun Yoon, Mr. Jungtak Yoon, and Mr. Chulmin Cho, for sharing their friendly support over the past few years. I would also like to express my thanks to Dr. Jaesik Chang in PCTEST Engineering Laboratory, for his help and support during my summer internship.

Finally, my special and sincere thanks must go to my parents and my sister for their love and support during my studies.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

1.1 Motivation

Failures of engineered systems can lead to significant economic and societal losses. Despite tremendous efforts (e.g., $200 billion annually) denoted to reliability and maintenance, unexpected catastrophic failures still occurs. As an example, Ameren Corp. experienced an unexpected power transformer fire in Missouri in 2002 as shown in Figure 1-1, which caused $3 million property damage and $2 million business interruption. UPS Corp. also lost their shipping flight due to Lithium-ion (Li-ion) battery explosion. To minimize the losses, reliability of engineered systems must be ensured throughout their life-cycle amidst uncertain operational condition and manufacturing variability. In most engineered systems, the required system reliability level under adverse events[1] is achieved by adding system redundancies and/or conducting system reliability-based design optimization (RBDO). However, a high level of system redundancy increases a system's life-cycle cost (LCC) while system RBDO cannot ensure the system reliability when unexpected loading/environmental conditions are applied and unexpected system failures are developed. It is well-conceived that engineered systems designed under the conventional design paradigm are typically passive and thus non-responsive under unexpected loading/environmental conditions.

---

[1] Adverse events could include the failure of components due to internal hazards (e.g., degradation) and/or external hazards (e.g., harsh operational conditions) that occur during the mission of the systems.

(a) (b)

Figure 1-1 Disastrous system failures: (a) unexpected fire of a GSU power transformer, MO, in 2002 and (b) unexpected system failures of Li-ion battery fire in UPS Flight 1307, PA, in 2006

Recently, prognostics and health management (PHM) has been successful in lowering system maintenance by enabling proactive maintenance decisions. A new design paradigm, referred to as resilience-driven system design, capitalizes on PHM technology at an early design stage to adaptively ensuring high system reliability under adverse conditions. In contrast to conventional design paradigm, resilience-driven system design can ensure highly reliable system designs under any loading/environmental conditions and system failures while considerably reducing systems' LCC. As a solution to overcome design deficiencies due to unexpected loading/environmental conditions and system failures, this new design paradigm enables the transformation of passively reliable (or vulnerable) engineered systems into adaptively reliable (or resilient) engineered systems while considerably reducing systems' LCC. There is, however, no theoretical basis and design framework of engineering resilience to facilitate the development of formal methodologies for resilient engineered system design and to advance conventional engineered systems to resilient engineered systems.

1.2 Overview and Significance

This research aims at advancing two essential and co-related research areas for system resilience analysis and design: Research Thrust 1 – system RBDO and Research Thrust 2 – system PHM to facilitate the development of formal methodologies for resilient engineered system design. System RBDO is capable of designing engineered systems with sufficiently high reliability at the early stage of their lifecycle, whereas capitalizing on PHM technology at the early design stage may enable the transformation of passively reliable (or vulnerable) conventional systems into adaptively reliable (or resilient) engineered systems while considerably reducing systems' LCCs. In addition to advancing the above two research areas, this research also makes the first attempt to establish a theoretical basis and design framework of engineering resilience in a mathematical and statistical context, which will be elaborated in Research Thrust 3 – resilience-driven system design. Specifically, the research scope in this dissertation is to develop technical advances in the following three research thrusts:

Research Thrust 1: System RBDO (Reliability Analysis)

Research Thrust 1 addresses research challenges in system RBDO, specifically, reliability analysis, to design engineered systems with sufficiently high reliability at the early stage of their lifecycle. Reliability analyses under uncertainty will be carried out in both component and system levels against critical failure mechanisms. For component reliability analysis, the adaptive-sparse polynomial chaos expansion (PCE) and asymmetric dimension-adaptive tensor-product (ADATP) methods are proposed to efficiently address high-order interactions in system responses. For system reliability analysis, the generalized complementary intersection method (GCIM) is developed to provide a unique system reliability estimate regardless of the system structures (series, parallel, and mixed systems).

Research Thrust 2: System PHM

Research Thrust 2 aims at designing highly accurate and robust PHM systems for engineered systems with a single or multiple time-scale(s). A PHM system generally comprises of sensor selection, sensor network configuration, data acquisition, data processing, and remaining useful life (RUL) prediction with prognostic algorithms. This thrust places the main focus on the design of PHM algorithms for data processing, and accurate RUL prediction in real-time. For PHM in a single time-scale, this research achieves two technical advances: (i) an ensemble data-driven prognostic approach with weight optimization and *k*-fold cross validation (CV) is proposed to enhance the prediction accuracy and robustness of sole member algorithms; and (ii) a co-training data-driven prognostic approach is developed to exploit the suspension data to achieve highly-confident health prognostics with the lack of failure data. For PHM in multiple time-scales, this research proposes a multiscale framework with extended Kalman filter (EKF) and applies this framework to Li-ion battery systems for efficient and accurate state of charge (SOC) and capacity estimation.

Research Thrust 3: Resilience-Driven System Design

Following the development of Research Thrusts 1 and 2, Research Thrust 3 – resilience-driven system design will establish a theoretical basis and design framework of engineering resilience in a mathematical and statistical context.

Regarding the construction of the theoretical basis of engineering resilience, this research formulates engineering resilience in terms of system reliability and restoration. Regarding the development of the design framework of engineering resilience, this research proposes a resilience-driven system design framework composed of three hierarchical design tasks, namely the resilience allocation problem (RAP) as a top-level design problem, and the system RBDO and PHM design as two bottom-level design problems, and demonstrates this design framework with a simplified aircraft control actuator design problem.

The proposed system resilience analysis and design methodologies are expected to make significant contributions to current knowledge in resilient engineered systems. This advanced knowledge will be applicable to a broad class of engineered system design problems. It is also believed that this innovative research will enable a transformative shift in engineered system design from reliability-based to resilience-driven system design. The theoretical basis and design framework of engineering resilience gained from this research will facilitate the development of formal methodologies for resilient engineered system design.

1.3 Dissertation Layout

The dissertation is organized as follows. Chapter 2 reviews the current state of knowledge on system RBDO (specifically, reliability analysis), system PHM and resilience concept, and presents the main challenges in these research areas. Chapter 3 presents the adaptive-sparse polynomial chaos expansion (PCE) and asymmetric dimension-adaptive tensor-product (ADATP) methods for component reliability analysis. Chapter 4 discusses the generalized complementary intersection method (GCIM) for system reliability analysis for series, parallel and mixed systems. Chapter 5 presents an ensemble data-driven prognostic approach with weight optimization and *k*-fold cross validation (CV) and a co-training data-driven prognostic approach for PHM in a single time-scale, and discusses a multiscale framework with extended Kalman filter (EKF) for PHM in multiple time-scales. Chapter 6 presents the resilience-driven system design framework and demonstrates this design framework with a case study on the system resilience analysis and design of an aircraft control actuator. Finally, Chapter 7 summarizes the contributions of the research work and gives insights on future works.

Chapter 2: Literature Review

In this section, current state-of-art knowledge for each of the three thrusts will be reviewed: (i) reliability analysis methodologies; (ii) prognostics and health management (PHM) methodologies, and (iii) resilience concept. Literatures on each of these three aspects are discussed in one subsection and challenges are identified at the end.

2.1 Reliability Analysis

In the past few decades, reliability analysis has been widely recognized as of great importance in the development of engineered systems. Hence, various methods have been developed to assess the engineered system reliability while taking into account various uncertainty sources (e.g., material properties, loads, geometric tolerances). This section reviews these methods by classifying them into two categories: (a) component reliability analysis, and (b) system reliability analysis.

2.1.1 Component Reliability Analysis

Engineered systems are typically composed of multiple components. To predict the reliability at the system level, reliability analysis is often firstly conducted at the component level, referred to as component reliability analysis. The component reliability can be 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, loads, geometric tolerances). In order to formulate the component reliability in a mathematical framework, random variables are often used to model uncertainty sources in engineered systems. The component reliability can then be formulated as a multi-dimensional integration of a performance function over a safety region

$$R = \int \cdots \int_{\Omega^S} f\left(\mathbf{x}\right) d\mathbf{x} \tag{2.1}$$

where R denotes the component reliability; $f(\mathbf{x})$ denotes the joint probability density function (PDF) of the vector of random variables; $\mathbf{x} = (x_1, x_2, \ldots, x_N)^T$ models uncertainty sources such as material properties, loads, geometric tolerances; the safety domain $\mathbf{\Omega}^S$ is defined by the limit-state function as $\mathbf{\Omega}^S = \{\mathbf{x}: g(\mathbf{x}) < 0\}$; $g(\mathbf{x})$ is a component performance function. The concept of component reliability analysis in a two-dimensional case is illustrated in Figure 2-1. The dashed lines represent the contours of the joint PDF of the two random variables x_1 (operational factors) and x_2 (manufacturing tolerance). The basic idea of reliability analysis is to compute the probability that $\mathbf{X}$ is located in the safety region $\mathbf{\Omega}^S = \{\mathbf{x}: g(\mathbf{x}) < 0\}$.

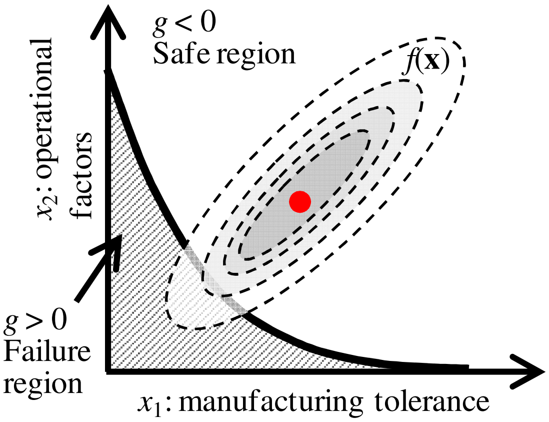


Figure 2-1 Concept of component reliability analysis.

In practice, however, it is extremely difficult to perform the multi-dimensional numerical integration when the number of random variables is relatively large. The search for efficient computational procedures to estimate the reliability has resulted in a variety of numerical and simulation methods such as the first- or second-order reliability method (FORM/SORM) [1-3], direct or smart Monte Carlo simulation (MCS) [4-7], dimension reduction (DR) method [8-10], stochastic spectral method [11-24], and stochastic collocation method [26-39]. This section gives a brief review of these approaches.

First- or Second-Order Reliability Method

Among many reliability analysis methods, the first- or second-order reliability method (FORM or SORM) [1-3] is most commonly used. The FORM/SORM uses 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 safety regions of a response. Some major challenges of the FORM/SORM include (i) it is very expensive to build the probability density function (PDF) of the response, (ii) large errors can be generated for highly nonlinear performance functions, and (iii) system design can be expensive when employing a large number of the responses.

Direct or Smart MCS

The direct or smart MCS provides an alternative way for multi-dimensional integration [4-7]. Although the direct MCS [4] produces accurate results for reliability analysis and allows for relative ease in the implementation, it demands a prohibitively large number of simulation runs. Thus, it is often used for the purpose of a benchmarking in reliability analysis. To alleviate the computational burden of the direct MCS, researchers have developed various smart MCS methods, among which the (adaptive) importance sampling methods [5-7] are the most popular. The idea behind the (adaptive) importance sampling is that certain values of the input random variables in a simulation have more impact on the parameter being estimated than others. If these “important” values are emphasized by sampling more frequently, then the estimator variance can be reduced. Hence, an important step in the (adaptive) importance sampling is to choose a distribution which "encourages" the important values and the simulation outputs are weighted by the likelihood ratio. Despite the improved efficiency than the direct MCS, these methods are still computationally expensive, especially for component performance functions with very high or very low reliabilities, where an extremely large sample size is required to achieve good accuracy in predicting those highly tailed performance PDFs.

Dimension Reduction Method

Recently, the dimension reduction (DR) method [8,9] has been proposed and is known to be a sensitivity free method for reliability analysis. This method uses an additive decomposition of a response that simplifies a single multi-dimensional integration to multiple one-dimensional integrations by the univariate DR (UDR) method [8] or to multiple one- and two-dimensional integrations by the bivariate DR (BDR) method [9]. The eigenvector dimension reduction (EDR) method [10] improves numerical efficiency and stability of the UDR method with the ideas of eigenvector samples and stepwise moving least squares method with no extra expense. Results of the DR-family methods are given in the form of statistical moments. To further predict the reliability or PDF of the response, PDF generation techniques must be involved, which could increase numerical error in reliability prediction. Furthermore, performance functions with strong tri- and higher-variate interactions among random inputs require tri- and higher-variate dimension decompositions for accurate reliability analysis [9]. In such cases, the computational effort could become prohibitively large for high input dimensions, thus making the decomposition strategy infeasible.

Stochastic Spectral Method

The stochastic spectral method [11] is an emerging technique for reliability analysis of complex engineered systems. This method uses a number of response samples and generates a stochastic response surface approximation with multi-dimensional polynomials over a sample space of random variables. 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 original Hermite polynomial chaos basis was proposed by Wiener [12] for modeling stochastic response with Gaussian input random variables. Xiu and Karniadakis [13] extended the method under the Askey polynomial scheme to non-Gaussian random variables (e.g., gamma, uniform, and beta), referred to as the generalized PCE. The wavelet basis [14] and multi-element generalized PCE [15] were developed to further extend the generalized PCE to use the polynomial basis functions that are not globally smooth. For the estimation of small failure probability, shifted and windowed Hermite polynomial chaos were proposed to enhance the accuracy of a response surface in the failure region [16]. In recent papers [17-19], researchers have applied the PCE method to various engineering reliability problems. Although the PCE method is considered to be accurate, the primary drawback of the PCE method is the curse of dimensionality, which substantially increases the computational cost as the number of random variables increases. To alleviate the difficulty, many adaptive algorithms were recently developed. The authors in 20 proposed an adaptive multi-element generalized PCE, where an error indicator based on the decay rate of local variance was used for an h-adaptive refinement. Its collocation-based counterpart, the multi-element probabilistic collocation method, used the tensor product or sparse grid collocation [26] in each random element [21]. A more recent version of the multi-element probabilistic collocation method incorporates the ANOVA (Analysis-of-Variance) decomposition to truncate the PCE at a certain dimension to further enhance the computational efficiency [22]. In addition to the multi-element PCE, a sparse polynomial chaos approximation was introduced as an alternative to tensor-product polynomial bases [23] and a sparse stochastic collocation method based on this sparse basis was recently developed in [24]. Although these adaptive algorithms alleviate the curse of dimensionality to some degree, more efforts are still needed to fully resolve this difficulty. As demonstrated by Lee [25], the implementation of the PCE method becomes inconvenient in engineering design practice since the PCE order cannot be predetermined for black-box-type problems.

Stochastic Collocation Method

The stochastic collocation (SC) method is another stochastic expansion technique that approximates a multi-dimensional random function using function values given at a set of collocation points. A comparison between the SC and PCE methods for uncertainty quantification (UQ) was discussed in [38], where the SC method was reported to consistently outperform the PCE method. In the SC method, the great improvement in reducing the curse of dimensionality in numerical integration was accomplished by Smolyak [26], who introduced the concept of the sparse grid. Since then, the sparse grid has been applied to high dimensional integration [27] and interpolation [28], UQ in reliability analysis [38] and design [39], and PDEs with deterministic inputs [29] and random inputs [30-32]. 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. Recently, the so called dimension-adaptive tensor-product (DATP) quadrature method introduced the concept of the generalized sparse grid and considered the dimensional importance indicated by an error estimator to adaptively refine the collocation points for efficient multi-dimensional integration [33]. Klimke [34] further developed this work for hierarchical interpolation by using either piecewise multi-linear basis functions or Lagrangian polynomials. In this method, all the dimensions in the random space are not considered as of equal importance and the adaptive sampling scheme automatically detects the highly nonlinear dimensions and adaptively refines the collocation points in those dimensions. In [37], a priori and a posteriori procedures are included to update a weight vector for different stochastic dimensions, which combines the advantages of conventional and dimensional-adaptive approaches. As demonstrated in [35,36], the application of the DATP method in stochastic problems is promising.

Summary and Discussion

As discussed above, a wide variety of numerical and simulation methods have been developed for efficient and accurate uncertainty propagation and reliability analysis of engineered components/systems subject to various engineering uncertainties. Recently, researchers are paying more attention to advanced sensitivity-free reliability analysis methods such as the PCE and SC methods. Despite advances in these advanced methods, critical challenges still remain unresolved. In the PCE method, the curse of dimensionality, which substantially increases the computational cost as the number of random variables increases, imposes severe limitations on the practical use for reliability analysis. The SC method greatly reduces the curse of dimensionality by using the sparse grid and can even achieve a substantially higher convergence rate by using its generalized counterpart (i.e., the dimension-adaptive tensor-product method). However, the dimension-adaptive algorithm treats the positive and negative axial directions in a multi-dimensional cube as of equal importance and thus may not be approximate for engineering cases where not only different dimensions but also two opposite directions (positive and negative) within one dimension often demonstrate a large difference in response nonlinearity. The above difficulties must be fully resolved to make the advanced reliability analysis methods generally applicable to the practical engineering reliability analysis and design problems.

2.1.2 System Reliability Analysis

System reliability analysis aims at analyzing the probability of system success while considering multiple system performances or failure modes (e.g., fatigue, corrosion, fracture). Let us first stake a look at the concept of system reliability analysis with two performance functions (i.e., fatigue safety g_1 and wear safety g_1) and two random variables (i.e., operational factors X_1 and manufacturing tolerance

X_2), as shown in Figure 2-2. In this figure, we have two limit state functions $g_1 = 0$ and $g_2 = 0$ which divides the input random space into four subspaces $\{g_1 < 0 \;\&\; g_2 < 0\}$, $\{g_1 < 0 \;\&\; g_2 > 0\}$, $\{g_1 > 0 \;\&\; g_2 < 0\}$, $\{g_1 > 0 \;\&\; g_2 > 0\}$. Component reliability analysis aims at quantifying the probability that a random sample $\mathbf{x}$ falls into the component safety region (i.e., $\{g_1 < 0\}$ or $\{g_2 < 0\}$) while system reliability analysis (assuming a series system) aims at quantifying the probability that a random sample $\mathbf{x}$ falls into the system safety region (i.e., $\{g_1 < 0 \;\&\; g_2 < 0\}$). Clearly, the component reliability (for $\{g_1 < 0\}$ or $\{g_2 < 0\}$) is larger than the system reliability since the component safety region has a larger area than the system safety region by the area of an intersection region $\{g_1 < 0 \;\&\; g_2 > 0\}$ or $\{g_1 > 0 \;\&\; g_2 < 0\}$.

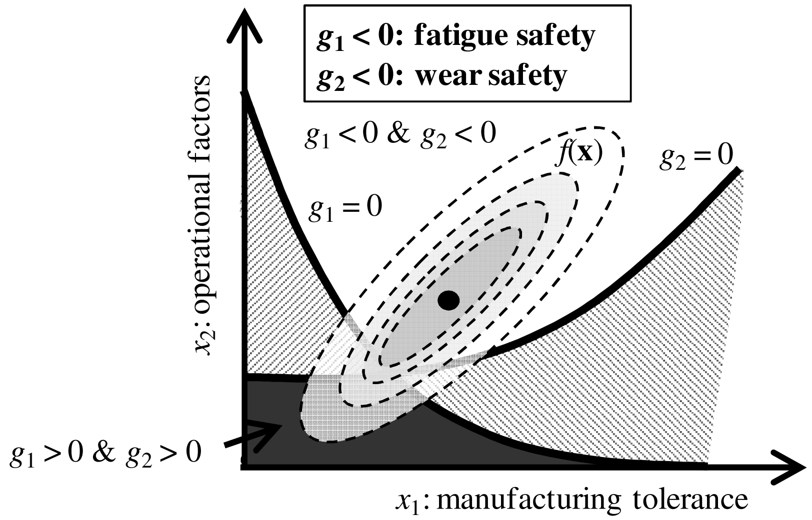


Figure 2-2 Concept of system reliability analysis (two performance functions)

The aforementioned discussion leads to a mathematical definition of system reliability as a multi-dimensional integration of a joint probability density function over a system safety region, expressed as

$$R_{sys} = \int \cdots \int_{\Omega^S} f(\mathbf{x})\, d\mathbf{x} \tag{2.2}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_N)^T$ models uncertainty sources such as material properties, loads, geometric tolerances; $f_{\mathbf{X}}(\mathbf{x})$ denotes the joint PDF of this random vector; $\mathbf{\Omega}^S$ denotes the system safety domain and, for this example, reads $\mathbf{\Omega}^S = \{\mathbf{x}: g_1(\mathbf{x}) < 0\ g_2(\mathbf{x}) < 0\}$. We can see that this formula bears a striking resemblance to that of component reliability analysis. The only difference between these two formulae lies in the definition of the safety domain. For component reliability analysis, the safety domain can be defined in terms of a single limit-state function as $\mathbf{\Omega}^S = \{\mathbf{x}: g(\mathbf{x}) < 0\}$. For system reliability analysis involving *nc* performance functions, the safety domains must be defined in terms of these *nc* limit-state function.

We note that, in practice, it is extremely difficult to perform the multi-dimensional numerical integration for system reliability analysis in Eq. (2.2) due to the high nonlinearity and complexity of the system safe domain. Compared with tremendous advances in component reliability analysis, the research in system reliability analysis has been stagnant, mainly due to two technical difficulties. First, it hard to derive an explicit formula for system reliability for given system redundancy. Second, even if system reliability is given explicitly, most numerical methods cannot effectively assess system reliability with high efficiency and accuracy. Nevertheless, in what follows, Nevertheless, in the subsequent sections, we will introduce the most widely used bounds methods for system reliability analysis as well as a recently developed point estimation method.

First- and Second-Order Bounds Methods

Based on the well known Boolean bounds, Ang and other researchers [40] developed the first-order system reliability bounds for serial and parallel systems in 1960's and 1980's. Consider a serial system with m components, the first-order bounds for the probability of system failure P_{fs} can be expressed as

$$\max_i \left[P\left(\overline{E_i}\right) \right] \leq P_{fs} \leq \min \left[\sum_{i=1}^{m} P\left(\overline{E_i}\right),\ 1 \right] \tag{2.3}$$

It should be noted that an upper system reliability bound is given with the assumption that all system safety events are perfectly dependent and that a lower system reliability bound is obtained by assuming that all system safety events are mutually exclusive. Since the first-order system reliability bounds are usually too wide to be of practical use, the application of the first-order bound method is very limited. Thus, the need for narrower bounds results in the second-order bound method proposed by Ditlevsen and Bjerager [41] in Eq. (2.4).

$$\begin{aligned} & P\left(\overline{E_1}\right) + \sum_{i=2}^{n} \max \left\{ \left[P\left(\overline{E_i}\right) - \sum_{j=1}^{i-1} P\left(\overline{E_i}\,\overline{E_j}\right) \right],\ 0 \right\} \leq P_{fs} \\ & \leq \min \left\{ \left[\sum_{i=1}^{n} P\left(\overline{E_i}\right) - \sum_{i=2}^{n} \max_{j<i} P\left(\overline{E_i}\,\overline{E_j}\right) \right],\ 1 \right\} \end{aligned} \tag{2.4}$$

The above bounds, also known as Ditlevsen's bounds, are most widely used for system reliability analysis. Other equivalent forms of Ditlevsen's bounds can be found in Refs [42-45]. Although second-order bounds method can give fairly narrow system reliability bounds generally assuming given the system input uncertainty information, evaluation of these bounds usually suffers from numerical errors, since most of numerical methods cannot evaluate probabilities of second or higher order joint events effectively considering system coupling effects between all different modes. Besides, the second order method is not able to carry out system reliability assessment for a mixed system which does not follow a simple series or parallel system structure.

Linear Programming Bounds Method

Recently, Song and Der Kiureghian formulated system reliability to a Linear Programming (LP) problem, referred to as the LP bounds method [46]. The LP bounds method treats the system reliability as the objective function and obtains the lower and upper bound through minimizing and maximizing the objective function accordingly. The LP bounds method is able to calculate optimal bounds for system reliability with the component reliabilities and/or probabilities of joint failure events as provided input information. However, it is known that the LP bound method can suffer when an approximate LP algorithm is used for over-constrained problems. Besides, it is extremely sensitive to the accuracy of the given input information, which are the probabilities of the first-, second-, and high-order joint safety events. To assure high accuracy of the LP bound method, the input probabilities must be given very accurately and the problem must not be over-constrained.

Complementary Intersection Method (CIM)

Most recently, Youn and Wang [47] introduced a novel concept of the complementary intersection (CI) event and proposed the complementary intersection method (CIM) for system reliability analysis of series systems. The CIM provides not only a unique formula for system reliability but also an effective numerical method to evaluate the system reliability with high efficiency and accuracy. The CIM decomposes the probabilities of high-order joint failure events into the probabilities of CI events. This probability decomposition allows the use of advanced reliability analysis methods for evaluating the probabilities of the second-order (or higher) joint failure events efficiently. To facilitate system reliability analysis for large-scale systems, the CI-matrix can be built to store the probabilities of the first- and second-order CI events. The CIM with the EDR method was reported to outperform other methods for system reliability analysis in terms of efficiency and accuracy [47]. However, the application of the CIM is limited to series systems only. A fundamentally sound framework must be established to extend the applicability of the CIM to parallel and mixed systems.

Summary and Discussion

Although the second-order bounds method and LP bounds method can give fairly narrow system reliability bounds given the probabilities of the joint safety events, the evaluation of these bounds may suffer from numerical errors since most numerical methods are not capable of evaluate the probabilities of second- or higher-order joint safety events effectively while considering the complex coupling between the component safety events. Besides, these bounds methods cannot provide continuous system reliability estimate with respect to the system input random variables. The CIM resolved these difficulties in system reliability analysis of series systems. However, these difficulties still remain unresolved in system reliability analysis of parallel and mixed systems.

2.2 Prognostics and Health Management (PHM)

As discussed above, tremendous research efforts have been devoted to the physics-based reliability analysis under uncertainties during the design stage of engineered systems. Recently, Research on real-time diagnosis and prognosis which interprets data acquired by distributed sensor networks, and utilizes these data streams in making critical decisions provides significant advancements across a wide range of applications. Maintenance and life-cycle management is one of the beneficiary application areas because of the pervasive nature of design and maintenance activities throughout the manufacturing and service sectors. Maintenance and life-cycle management activities constitute a large portion of overall costs in many industries [48]. These costs are likely to increase due to the rising competition in today's global economy. For instance, in the manufacturing and service sectors, unexpected breakdowns can be prohibitively expensive since they immediately result in lost production, failed shipping schedules, no operational service, repair cost, and poor customer satisfaction. In order to reduce and possibly eliminate such problems, it is necessary to accurately assess current system health condition and precisely predict the remaining useful life (RUL) of operating components, subsystems, and systems. This section reviews the current state-of-the-art PHM technology that can predict system's health condition in: (a) a single time-scale and (b) in multiple time-scales. In the former case, all the system health-relevant information (e.g., health condition, measurable physical quantities) of interest tend to vary in the same time-scale. In the latter case, the system exhibits a time-scale separation. In other words, certain system health-relevant responses (e.g., health condition) of interest tend to vary very slowly as opposed to other system health-relevant responses (e.g., measurable physical quantities) that vary very fast.

2.2.1 PHM in a Single Time-Scale

Many engineered systems do not exhibit time-scale separations and their health prognostics can thus be done in a single time-scale. In general, technical approaches for health monitoring and prognostics of these engineered systems can be categorized into model-based approaches [49-51], data-driven approaches [52-56] and hybrid approaches [61-63].

Model-Based Prognostics

The application of general model-based prognostics approaches relies on the understanding of system physics-of-failure and underlying system degradation models. In the literature, various stochastic degradation models have been developed to model degradation behaviors of different engineered systems. Luo et al [49] developed a model-based prognostic technique that relies on an accurate simulation model for system degradation prediction and applied this technique to a vehicle suspension system. Gebraeel presented a degradation modeling framework for RUL predictions of rolling element bearings under time-varying operational conditions [50] or in the absence of prior degradation information [51]. As practical engineered systems generally consist of multiple components with multiple failure modes, understanding all potential physics-of-failures and their interactions for a complex system is almost impossible.

Data-Driven Prognostics

With the advance of modern sensor systems as well as data storage and processing technologies, the data-driven approaches for system health prognostics, which are mainly based on the massive sensory data with less requirement of knowing inherent system failure mechanisms, have been widely used and become popular. A good review of data-driven prognostic approaches was given in [52]. Data-driven prognostic approaches generally require the sensory data fusion and feature extraction, statistical pattern recognition, and for the life prediction, the interpolation [53,54], extrapolation [55], or machine learning [56] and so on.

The data-driven prognostic approaches mentioned in the above literature survey belongs to the category of supervised learning which relies on a large amount of failure data for the offline training in order to achieve good accuracy for the online prediction. Here, failure data refer to condition monitoring data collected from the very beginning of an engineered system’s lifetime till the occurrence of its failure. Unfortunately, in many engineered systems, only very limited failure data are available since running systems to failure can be a fairly expensive and lengthy process. In contrast, we can easily obtain a large amount of suspension data. By suspension data, we mean condition monitoring data acquired from the very beginning of an engineered system’s lifetime till planned inspection or maintenance when the system is taken out of service. The lack of failure data and plenty of suspension data with rich information on the degradation trend makes it essentially critical and quite possible to utilize suspension data in order to improve supervised data-driven prognostics and achieve more accurate remaining useful life (RUL) prediction. However, the utilization of suspension data for data-driven prognostics is still in infancy. The very few relevant works we are aware of are the survival probability-based approaches [57-59] and life-percentage-based approach [60]. The former approaches use conditional monitoring data as inputs to an artificial neural network (ANN) [57] or relevance vector machine [58,59] which then gives the survival probability as the output. As pointed out in [60], the drawback of these approaches lies in the fact that the outputs cannot easily be converted to equivalent RULs for practical use. In contrast, the latter approach employs the condition monitoring data and operation time as inputs to an ANN which then produces the life percentage as the output.

Hybrid Prognostics

Hybrid approaches attempt to take advantage of the strength from data-driven approaches as well as model-based approaches by fusing the information from both approaches. Garga et al. [61] described a data fusion approach where domain knowledge and predictor performance are used to determine weights for different state-of-charge predictors. Goebel et al. [62] employed a Dempster-Shafer regression to fuse a physics-based model and an experience-based model for prognostics. Saha et al. [63] combined the offline relevance vector machine (RVM) with the online particle filter for battery prognostics. Similar to model-based approaches, the application of hybrid approaches is limited to the cases where sufficient knowledge on system physics-of-failures is available.

Summary and Discussion

The traditional data-driven prognostic approach is to construct multiple candidate algorithms using a training data set, evaluate their respective performance using a testing data set, and select the one with the best performance while discarding all the others. This approach has three shortcomings: (i) the selected standalone algorithm may not be robust, i.e., it may be less accurate when the real data acquired after the deployment differs from the testing data; (ii) it wastes the resources for constructing the algorithms that are discarded in the deployment; (iii) it requires the testing data in addition to the training data, which increases the overall expenses for the algorithm selection. Thus, an accurate yet robust data-driven prognostic approach must be developed to overcome these drawbacks.

In addition to enhancing the prognostic accuracy and robustness, another important issue as mentioned above is to achieve highly-confident health prognostics when we only have very limited failure data but a large amount of suspension data. Although the approach described in [60] is capable of enhancing the accuracy in RUL prediction with suspension data, it still suffers from the follows drawbacks: (i) it simply uses all suspension data regardless of the quality and usefulness; and (ii) the only criteria to determine the RUL of a suspension unit is the minimization of a validation error in the offline training, which could lead to a largely incorrect RUL estimate or even a physically unreasonable estimate (i.e., less than or equal to zero) of that unit. Thus, a smart data-driven prognostic approach should be developed which can selectively utilize the suspension data as well as effectively predict the RUL of a utilized suspension unit.

2.2.2 PHM in Multiple Time-Scales

Different physical principles governing system health-relevant responses over time require health monitoring and prognostics in different time-scales. Challenges exist in PHM for such systems. For example, if we are interested in the evolution of the system health condition over a large time-scale and measurable physical quantities evolve in small time-scales, we must estimate the system health condition using the measurements for all the small time steps within one large time step, which can be very inefficient and produce large variance in the estimate.

As a typical engineered system with time-scale separation, the Li-ion battery system is considered for the demonstration of PHM for systems with multiple time-scales. Lithium-ion (Li-ion) batteries are the rechargeable batteries most commonly used in hybrid electric vehicles (HEV), battery electric vehicles and consumer portable electronics. Among these applications, the HEV environment is particularly harsh, imposing many stringent load requirements on the battery cells [64]. The harsh operation conditions and demanding requirements require the incorporation of resilience into a battery system. Typical parameters indicative of a battery system's conditions are the state of charge (SOC), state of health (SOH) and state of life (SOL). The resilience must be the hallmark capability of a battery system that can be used to optimally maintain the SOC, and the current and future health conditions (SOH/SOL), and to provide this information for decision-making on cell balancing and maintenance (see Figure 2-3). To this end, advanced prognostic methods must be incorporated into the battery management system (BMS) to accurately estimate the SOC, SOH and SOL to manage the battery health and to maximize the useful lifetime of the pack. This section reviews current state-of-art methods for SOC and SOH/SOL estimation.

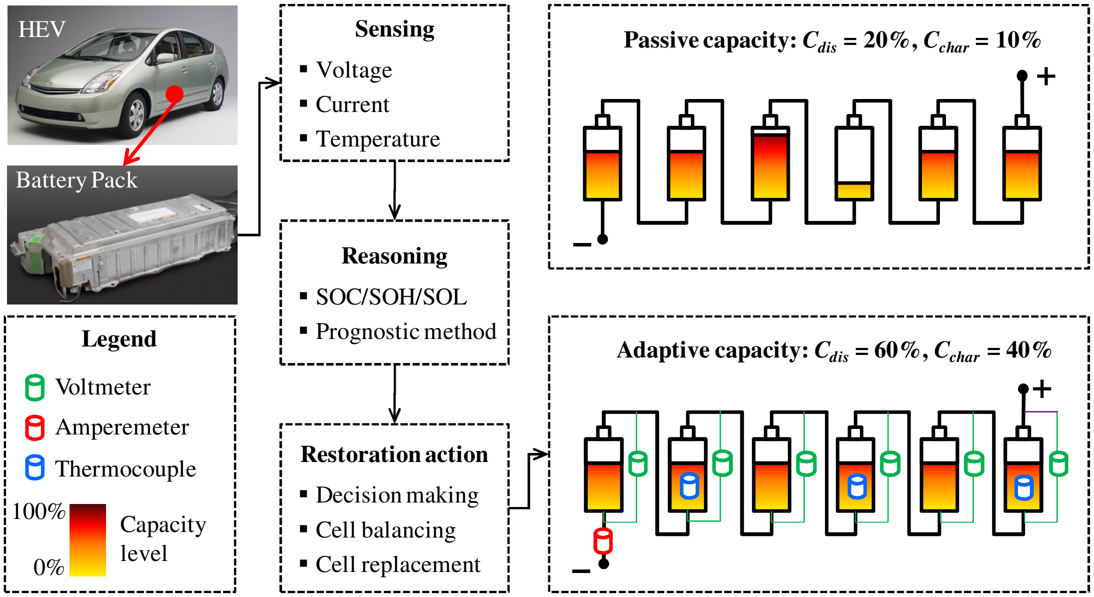


Figure 2-3 Resilience process of an HEV battery pack. Capitalizing on the engineering resilience enables the optimum cell equalization/replacement, thus maximizing the charge/discharge capacity restoration.

State of Charge (SOC) Estimation

Numerous approaches have been proposed to estimate the SOC of battery cells. These include coulomb-counting methods [65,66], impedance measurements [67-70], open-circuit voltage (OCV) measurements [71,72], Electro-Motive Force (EMF) measurements [73,74], adaptive systems employing Kalman filters [75-79], fuzzy logic [80,82] and neural networks [82,83]. An extensive review of most of these methods can be found in Piller [84]. The coulomb-counting procedure is easy to implement but it suffers from an initial value error and accumulated errors due to current measurements and charge lost [85]. The open-circuit voltage (OCV) measurement is inexpensive and accurate for predefined circumstances but it needs a long rest time and thus is not suitable for real-time applications [86]. Fuzzy logic and neural network methods have been found to be applicable only if training conditions are similar to testing ones. To compensate for the shortcomings of the aforementioned methods, Plett [75-78] proposed an adaptive system based on an extended or sigma point Kalman filter and an extension of circuit analog models. Accurate SOC estimation results were reported in the urban dynamometer drive scheme (UDDS) tests.

State of Health (SOH) and State of Life (SOL) Estimation

As a battery cell ages, the cell capacity and resistance directly limit the pack performance through capacity fade and power fade, respectively [76]. These two degradation parameters are often used to quantify the cell state of health (SOH). Thus, it is important to accurately estimate these parameters to monitoring the present battery SOH and to predict the remaining useful life (RUL). Recent literature reports various approaches to estimate the SOH with a focus on the capacity estimation. Joint/dual extended Kalman filter (EKF) [76] and unscented Kalman filter [78] with an enhanced self-correcting model were proposed to simultaneously estimate the SOC, capacity and resistance. To improve the performance of joint/dual estimation, adaptive measurement noise models of the Kalman filter were recently developed to separate the sequence of SOC and capacity estimation [87]. A physics-based single particle model was used to simulate the life cycling data of Li-ion cells and to study the physics of capacity fade [88,89]. In the PHM society, a Bayesian framework combining the relevance vector machine (RVM) and particle filter was proposed for prognostics (i.e., RUL prediction) of Li-ion battery cells [63]. More recently, the particle filter with an empirical circuit model was used to predict the remaining useful lives for individual discharge cycles as well as for cycle life [90].

Summary and Discussion

In general, existing PHM techniques for engineered systems with time-scale separation simultaneously estimate all the system health-relevant responses of interest in the small scale. The joint estimation presents two difficulties: (i) it requires tremendous computational efforts since the vast majority of computational time must be spent evolving the fast time-varying health-relevant responses of the system while we may be primarily interested in the slowly time-varying health-relevant responses; (ii) it may provide noisy estimates of the slowly time-varying health-relevant responses. For example, commonly used joint/dual EKF for battery SOC and capacity estimation suffers from the lack of accuracy and efficiency in the capacity estimation. Thus, a prognostic approach that takes into account time-scale separation must be developed to achieve high-fidelity and high-efficiency health monitoring and prognostics for engineered systems with time-scale separation.

2.3 Resilience Concept

In recent years, research on resilience has been widely conducted in ecology [91-93], psychology [94-97], economics and organizational science [98-100], and others to improve the ability of systems or people to respond to and quickly recover from catastrophic events. In contrast to the resilience research in several non-engineering fields, resilience in engineering design remains almost untouched. There is still a great need for a theoretical basis that furnishes a better understanding of how engineered systems achieve resilience, as well as enables the development of a generic resilience principle widely applicable to the field of engineering design. In what follows, a brief review of the research on resilience in non-engineering fields will be provided with an aim to extract useful information for the theoretical basis of engineering resilience.

In ecology, resilience is loosely defined as “the ability of the system to maintain its function when faced with novel disturbance” [91]. The current research on resilience applied to ecosystems mainly focuses on the analysis of ecosystem resilience using complex adaptive systems (CAS) theory [92]. As an extension of traditional systems theory, the CAS theory enables analysis of the role of adaptation in system resilience through specifically modeling how individual variation and changes in that variation lead to system-level responses [92]. Furthermore, natural selection and evolution play an important role in shaping ecological response to disturbance, which provides new insight to the understanding of resilience [93]. In psychology, resilience is defined as a dynamic process that individuals exhibit positive behavioral adaptation when they encounter significant adversity [94]. The process of resilience involves both the exposure of adversity and the positive adaptation to that adversity. Extensive research has been conducted to understand the protective factors that contribute to people’s adaptation to adverse conditions, e.g., bereavement [95], terrorist attacks [96], or urban poverty [97].

In economics and organizational science, resilience of an organization can be defined as its intrinsic ability to keep or recover a stable state, thereby allowing it to continue operations after a disruption or in presence of continuous stress [98,99]. In economics, resilience can be improved by adding redundancy or increasing flexibility. While investments in redundancy means a pure increase in cost, investments in flexibility yields many competitive advantages in day-to-day operations [98]. Of particular interest here is the characterization of a resilient organization with the following three steps within the context of a perturbation, as shown in Figure 2-4 [100]. Firstly, the organization should continuously monitor key state variables indicative of its health condition to anticipate the occurrence of a perturbation. Secondly, upon the occurrence of a perturbation, the system should conduct the situation assessment and identify an optimal way to reorganize itself and keep operations. Thirdly, after the occurrence of a perturbation, the organization should analyze various alternative ways of functioning and learn from the past experiences to determine the most relevant state variables for the first step and to enhance its capability to cope with perturbations.

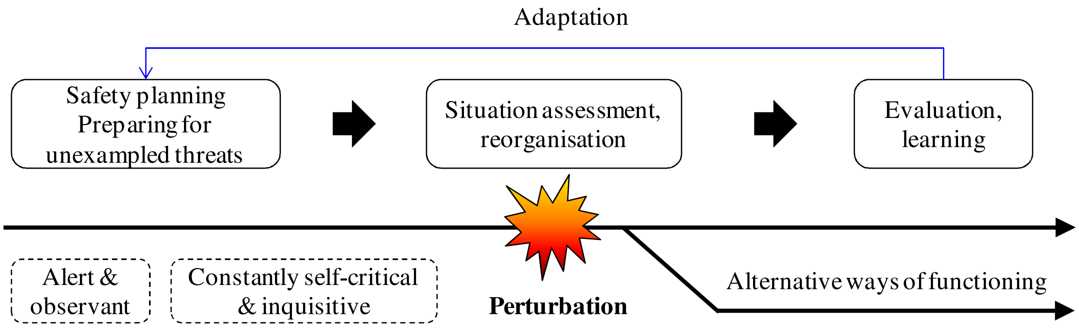


Figure 2-4 Resilient (Proactive) organization.

In contrast to the aforementioned developments of resilience in many non-engineering fields, resilience in engineering design has rarely been studied. One possible reason is that PHM, which is essential to make engineered systems resilient, has only recently received critical attention from the research community. It is fair to say, therefore, that there is still a great need for a theoretical basis that furnishes a better understanding of how engineered systems achieve resilience, as well as enables the development of an engineering resilience principle readily applicable to engineering design.

Chapter 3: Component Reliability Analysis

Engineered systems are subject to various uncertainties and variabilities such as physical uncertainties and variabilities, model errors (or uncertainties) and statistical uncertainties. Thus, system reliability analysis, as an essential step in system RBDO, must be able to systematically handle these uncertainties and variabilities in engineered systems. Component reliability analysis has been recognized as an essential element to successfully conducting system reliability analysis. However, advanced component reliability analysis methods suffer from either the curse of dimensionality or the lack of accuracy. To address this challenge, this research thrust identifies two research solutions as: (i) adaptive-sparse polynomial chaos expansion (PCE) method, and (ii) asymmetric dimension-adaptive tensor-product (ADATP) method. These research solutions are detailed in subsequent sections.

3.1 Adaptive-Sparse Polynomial Chaos Expansion (PCE) Method

To resolve the curse of dimensionality in the PCE method, as identified in the literature review, this research presents an adaptive-sparse polynomial chaos expansion (adaptive-sparse PCE) method for reliability analysis and design of complex engineered systems. To overcome the curse of dimensionality of the PCE method, this research first proposes an adaptive-sparse expansion scheme. This scheme automatically detects the most significant bivariate terms and adaptively builds the sparse PCE with the minimum number of bivariate basis functions. Moreover, the adaptive-sparse scheme offers the additional capability of automatically adjusting the PCE order to optimize the accuracy of the stochastic response surface. The PCE model resulting from the adaptive-sparse scheme is expected to achieve an optimal compromise between the UDR and BDR (more accurate than the UDR and more efficient than the BDR). To make the proposed method computationally tractable for engineering design, the projection technique used in the EDR method is employed to effectively compute the expansion coefficients. Moreover, a copula theory is successfully integrated to the proposed adaptive-sparse PCE method, which enables the designer to handle nonlinear correlation of input random variables. The adaptive-sparse PCE method is expected to give good accuracy and efficiency for highly nonlinear responses containing only a specific part of bivariate terms with significant interactions.

3.1.1 Review of Polynomial Chaos Expansion (PCE) Method

In the following sections, 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 $f_i(x_{\mathrm{i}}): \Gamma_i \rightarrow \mathbb{R}^+$, where the support Γ_i is a one-dimensional random space of x_i . Under the assumption of independence, the probabilistic characteristics of the random variables $\mathbf{x}$ can be completely defined by the joint PDF $f(\mathbf{x}) = f_1(x_1)\cdot f_2(x_2)\cdots f_N(x_N)$ with the support $\Gamma = \Gamma_1\cdot\Gamma_2\cdots\Gamma_N$. 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 N-dimensional space and one-dimensional space $g: \mathbb{R}^N \rightarrow \mathbb{R}$. 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 an 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 number of function evaluations.

Generalized PCE Method

The original Hermite polynomial chaos, also termed as the homogeneous chaos, was derived from the original theory of Wiener [12] for the spectral representation of any second-order stochastic response in terms of Gaussian random variables. To improve the expansion convergence rate, Xiu and Karniadakis [13] 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 3-1. In the finite dimensional random space Γ, a second-order stochastic response g can be expanded in a convergent series of generalized polynomial chaos basis as

$$\begin{aligned} g(\mathbf{x}) = {} & c_0\psi_0 + \sum_{i_1=1}^{\infty} c_{i_1}\psi_1\left(\zeta_{i_1}(\mathbf{x})\right) + \sum_{i_1=1}^{\infty}\sum_{i_2=1}^{i_1} c_{i_1 i_2}\psi_2\left(\zeta_{i_1}(\mathbf{x}),\zeta_{i_2}(\mathbf{x})\right) \\ & + \sum_{i_1=1}^{\infty}\sum_{i_2=1}^{i_1}\sum_{i_3=1}^{i_2} c_{i_1 i_2 i_3}\psi_3\left(\zeta_{i_1}(\mathbf{x}),\zeta_{i_2}(\mathbf{x}),\zeta_{i_3}(\mathbf{x})\right) + \cdots \end{aligned} \tag{3.1}$$

where $\psi_n\left(\zeta_{i_1}(\mathbf{x}),\zeta_{i_2}(\mathbf{x}),\ldots,\zeta_{i_n}(\mathbf{x})\right)$ 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 [105], the polynomial chaos expansion in Eq. (3.1) converges in the L_2 sense.

Table 3-1 Types of random inputs and corresponding polynomial chaos basis

	Random variable	Polynomial chaos	Support
Continuous	Gaussian	Hermite	(-inf, +inf)
	Gamma (Exponential)	Generalized Laguerre (Laguerre)	[0,+inf)
	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*}

For the purpose of notational convenience, Eq. (3.1) is often rewritten as

$$g(\mathbf{x})=\sum_{i=0}^{\infty}s_i\Phi_i\left(\boldsymbol{\zeta}(\mathbf{x})\right),\qquad \boldsymbol{\zeta}=\{\zeta_1,\zeta_2,\ldots\} \tag{3.2}$$

where there exists a one-to-one mapping between the polynomial basis functions ψ_n and Φ_i, and the PCE coefficients s_i and $c_{i_1,\ldots,i_r}$.

The orthogonality of the Askey-chaos can be expressed as

$$E\left[\Phi_i\Phi_j\right]=\delta_{ij}E\left[\Phi_i^2\right] \tag{3.3}$$

where δ_{ij} is the Kronecker's delta and $E[\cdot]$ is the expectation operator. Considering all N-dimensional polynomials of degree not exceeding p gives the truncated PCE as follows (with P denoting the number of unknown PCE coefficients):

$$g(\mathbf{x})=\sum_{i=0}^{P-1}s_i\Phi_i(\boldsymbol{\zeta}),\qquad \mathbf{x}=\{x_1,x_2,\ldots,x_N\},\ \boldsymbol{\zeta}=\{\zeta_1,\zeta_2,\ldots\zeta_N\} \tag{3.4}$$

In the above summation, the number of unknown PCE coefficients P is

$$P = \binom{N+p}{p} = \frac{(N+p)!}{N!\,p!} \tag{3.5}$$

Determination of PCE Coefficients

In this study, the reliability analysis for the performance function *g* under random inputs **x** is of our interest. Since the uncertainty of a stochastic response *g* can be fully characterized by the PCE coefficients in Eq. (3.2), an efficient and accurate numerical procedure to compute the coefficients is essential for reliability analysis.

Based on the orthogonality of the polynomial chaos, the projection method [106,107] can be used as a non-intrusive approach to compute the expansion coefficients of a response. Pre-multiplying both sides of Eq. (3.2) by $\Phi_j(\zeta)$ and taking the expectation gives the following equation

$$E\left[g(\mathbf{x})\Phi_j(\zeta)\right] = E\left[\sum_{i=0}^{\infty} s_i \Phi_i(\zeta)\Phi_j(\zeta)\right] \tag{3.6}$$

Due to the orthogonality of the polynomial chaos, Eq. (3.6) takes the form

$$s_j = \frac{E\left[g(\mathbf{x})\Phi_j(\zeta)\right]}{E\left[\Phi_j^2(\zeta)\right]} \tag{3.7}$$

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 the full tensorization of one-dimensional Gaussian quadrature [107], the crude MCS [108], or the Smolyak sparse grid [26,27,33,36]. The relative merits and disadvantages of these approaches are discussed below:

Approach 1. The full tensorization of one-dimensional Gaussian quadrature exhibits fast convergence for smooth integrand. 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. The crude MCS is robust and has a convergence rate that is independent of the dimension N asymptotically [108]. However, the convergence is very slow (as $1/\sqrt{M}$). Thus, accurate results require a large number of function evaluations which may incur intolerable computational burden, especially for complex engineered systems that are computationally intensive.

Approach 3. The sparse grid collocation based on the Smolyak algorithm [26] offers an alternative way for the multidimensional integration [27]. Compared with the fully tensorized quadrature, it also achieves fast convergence for smooth integrand but with much lower computational cost. Recently, adaptive algorithms [33,36] 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".

3.1.2 Adaptive-Sparse Scheme

The aim of this section is to develop an adaptive-sparse scheme for obtaining the minimum number of bivariate terms. In order to make the adaptive process computationally efficient and convergent, the adaptive-sparse scheme takes advantage of the PCE as the projection basis due to the inherent characteristics of orthogonal polynomials. For highly nonlinear responses containing only a specific part of bivariate terms with significant interactions, the PCE model resulting from the adaptive-sparse scheme is expected to achieve an optimal compromise between the UDR and BDR (more accurate than the UDR and more efficient than the BDR).

This scheme mainly consists of two loops to determine: (outer loop) the number (q) of the most significant bivariate terms (denoted as a set $\mathbf{B}$), and (inner loop) the optimal expansion order p. The detailed procedures are listed as follows:

<u>Initialization</u>: (a) Initialize $p = 2$, $q = 0$, $\mathbf{B} = \emptyset$, and set the convergence criteria ε_1 and ε_2 for the outer and inner loops, respectively.

(b) Compute the values of the performance function $g(\mathbf{x})$ at the univariate sample points: $g(\boldsymbol{\mu})$, $g(x_k^{(ik)}, \boldsymbol{\mu}^k)$, for $i_k = 1,2,\ldots, M_1$, $k = 1,2,\ldots, N$, where the superscript i_k denotes the corresponding sample point for x_k, M_1 the number of univariate sample points in each dimension, and $\boldsymbol{\mu}^k$ the mean vector of input random variables excluding x_k.

(c) With the function values obtained in step (b), construct a 2nd order PCE by computing the coefficients of univariate polynomial terms while setting the other coefficients to zero. The method for computing the PCE coefficients are detailed in the subsequent section.

<u>Outer loop</u>: (d) Compute the values of $g(\mathbf{x})$ at the $N(N-1)/2$ bivariate sample points which correspond to $N(N-1)/2$ pairs of variables: $g(x_k^{\mathrm{t}}, x_l^{\mathrm{t}}, \boldsymbol{\mu}^{k,l})$, for $k,l = 1,2,\ldots,N$, $k < l$, where $\boldsymbol{\mu}^{k,l}$ denotes the mean vector of input random variables excluding x_k and x_l. Based on the function values, compute the error indicators for all $N(N-1)/2$ bivariate terms. Note

that, for computing the error indicators, we do not require all the bivariate sample points that are used to compute the coefficients for each bivariate term but only pick one sample point for each bivariate term. An error indicator for testing the bivariate interaction between k^{th} and l^{th} input variables $[x_k, x_l]$ is defined as

$$e_{kl} = \frac{\left| \hat{g}_{(u)}\left(x_k^t, x_l^t, \boldsymbol{\mu}^{k,l}\right) - g\left(x_k^t, x_l^t, \boldsymbol{\mu}^{k,l}\right) \right|}{\max\limits_{1 \le i_j \le M_1, 1 \le j \le N} \left\{ g\left(x_j^{(i_j)}, \boldsymbol{\mu}^j\right) \right\} - \min\limits_{1 \le i_j \le M_1, 1 \le j \le N} \left\{ g\left(x_j^{(i_j)}, \boldsymbol{\mu}^j\right) \right\}} \tag{3.8}$$

where $\hat{g}_{(\mathrm{u})}(x_k^{\mathrm{t}}, x_l^{\mathrm{t}}, \boldsymbol{\mu}^{k,l})$ is the functional approximation at $(x_k^{\mathrm{t}}, x_l^{\mathrm{t}}, \boldsymbol{\mu}^{k,l})$ by a response approximation method using the function values at the univariate sample points. For the response approximation, we use the stepwise moving least squares (SMLS) method of which the details can be found in the author's previous work [10]. In this study, we apply $x_k^{\mathrm{t}} = \mu_k + 3\sigma_k$ and $x_l^{\mathrm{t}} = \mu_l + 3\sigma_l$, where μ_k and μ_l denote the means, and σ_k and σ_l denote the standard deviations of x_k and x_l. The numerator in Eq. (3.8) can be treated as the absolute univariate approximation error induced by the bivariate interaction, while the denominator can be treated as a normalization factor. The error indicator is a crucial part of the outer loop in the adaptive-sparse algorithm. A larger error indicator implies a stronger interaction between a given pair of variables. The pairs of variables with stronger interaction are given a higher priority in the algorithm since the inclusion of the pairs is likely to reduce a numerical error more significantly in probability analysis.

(e) Add the bivariate term $[x_k^{q+1}, x_l^{q+1}]$ with the $(q+1)^{\text{th}}$ largest error indicator to the bivariate set: $\mathbf{B} = \mathbf{B} \cup \{[x_k^{q+1}, x_l^{q+1}]\}$ and increase the number of bivariate terms: $q = q + 1$. Compute the function values of $g(\mathbf{x})$ at the bivariate sample points corresponding to the bivariate term $[x_k^{q+1}, x_l^{q+1}]$.

(f) With the function values obtained in step (e), compute the coefficients of bivariate polynomial terms in the constructed PCE model. The method for computing the PCE coefficients are detailed in the subsequent section.

Inner Loop: (g.1) If $q = 1$, we intend to determine the optimum PCE order through a convergence analysis. For this purpose, we need an error estimate to assess the performance of the constructed p^{th} order PCE (or stochastic response surface) $\hat{g}_p$. We prefer an efficient error estimate of which the evaluation only requires the already obtained response values at the sample points $\mathbf{x}^{(i)}$, for $1 \le i \le M$, where M is the total number of sample points. In this study, we use the coefficient of determination R^2, which can be defined based on the residual sum of squares e_{RSS} and total sum of squares e_{TSS} as

$$R^2\left(\hat{g}_p\right) = 1 - \frac{e_{RSS}\left(\hat{g}_p\right)}{e_{TSS}} \tag{3.9}$$

where

$$e_{RSS}\left(\hat{g}_p\right) = \frac{1}{M}\sum_{i=1}^{M}\left(g\left(\mathbf{x}^{(i)}\right) - \hat{g}_p\left(\mathbf{x}^{(i)}\right)\right)^2 \tag{3.10}$$

and

$$e_{TSS} = \frac{1}{M}\sum_{i=1}^{M}\left(g\left(\mathbf{x}^{(i)}\right) - \bar{g}\right)^2; \quad \bar{g} = \frac{1}{M}\sum_{i=1}^{M} g\left(\mathbf{x}^{(i)}\right) \tag{3.11}$$

We note that the cross-validation-based errors [109], which have been widely used in the machine learning technique to evaluate the model performance, can also be used as error estimates and deserve future studies.

(g.2) Increase the PCE order: $p = p + 1$.

(g.3) Repeat the steps (g.1) and (g.2) until R^2 converges to within a relative tolerance of ε_2.

Postprocessor: (h) Compute the reliability value based on the constructed PCE model. The numerical method for estimating the reliability are detailed in the subsequent section.

(i) Repeat the steps from (e) to (h) until the value of reliability converges to within a relative tolerance of ε_1.

The completion of the adaptive-sparse algorithm entails the optimal determination of the set **B** of bivariate terms and the PCE order p. The resultant PCE model should guarantee the most accurate and cost-effective fit among all bivariate PCE models.

3.1.3 Decomposition-Based Projection Method

This section presents a decomposition-based projection method for efficiently computing the expansion coefficients of an optimum set of uni- and bivariate polynomial terms. The proposed method attempts to further reduce the computational cost of the projection method.

Uni- and Bivariate Dimension Reduction

Let $\boldsymbol{\mu}^i$ = denote the mean vector of input random variables excluding x_i, and let $\boldsymbol{\mu}^{i_1,i_2}$ denote the mean vector of input random variables excluding x_{i_1} and x_{i_2}. Depending on the levels of the decomposition, the uni- and bivariate decomposed responses [9] can be expressed as, respectively,

$$g_1(\mathbf{x}) = \sum_{i=1}^{N} g\left(x_i, \boldsymbol{\mu}^i\right) - (N-1) g(\boldsymbol{\mu}) \tag{3.12}$$

and

$$\begin{aligned} g_2(\mathbf{x}) = & \sum_{1 \le i_1 < i_2 \le N} g\left(x_{i_1}, x_{i_2}, \boldsymbol{\mu}^{i_1,i_2}\right) \\ & -(N-2)\sum_{i=1}^{N} g\left(x_i, \boldsymbol{\mu}^i\right) + \frac{(N-1)(N-2)}{2} g(\boldsymbol{\mu}) \end{aligned} \tag{3.13}$$

It is important to note that the univariate decomposed response g_1 in Eq. (3.12) contains the univariate terms $g(x_i, \boldsymbol{\mu}^i)$ of any order in the Taylor series expansion and, similarly, the bivariate decomposed response g_2 in Eq. (3.13) has all bivariate terms in the Taylor series expansion. Thus, the approximations in Eqs. (3.12) and (3.13) should not be viewed as first- or second-order Taylor series expansion nor do they represent a limited degree of nonlinearity in $g(\mathbf{x})$. In fact, the residual error of a univariate approximation to a multidimensional integration of a system response over a symmetric domain contains only even-order terms of dimension two and higher since the integrations of odd-order terms become zeros for a symmetric integration domain and was reported to be far less than that of a second-order Taylor expansion method for probability analysis [8].

Formulation of Decomposition-Based Projection Method

To compute the coefficient of any n^{th}-order univariate polynomial term $\Psi_n(\zeta_k(\mathbf{x}),\ldots,\ \zeta_k(\mathbf{x}))$ in Eq. (3.1), which corresponds to a univariate polynomial term $\Phi_j(\zeta_k)$ in Eq. (3.2), the proposed decomposition-based projection method uses the univariate decomposed response in Eq. (3.12) [9]. The expansion coefficients can be obtained by projecting the univariate terms onto $g(\mathbf{x})$ as

$$\begin{aligned} s_j^k &= \frac{E\left[g(\mathbf{x})\Phi_j(\zeta_k)\right]}{E\left[\Phi_j^2(\zeta_k)\right]} \\ &\cong \frac{\sum_{i=1}^{N} E\left[g\left(x_i,\boldsymbol{\mu}^i\right)\Phi_j(\zeta_k)\right] - (N-1)\,g(\boldsymbol{\mu})\,E\left[\Phi_j(\zeta_k)\right]}{E\left[\Phi_j^2(\zeta_k)\right]} \\ &= \frac{E\left[g\left(x_k,\boldsymbol{\mu}^k\right)\Phi_j(\zeta_k)\right]}{E\left[\Phi_j^2(\zeta_k)\right]} \end{aligned} \tag{3.14}$$

Similarly, the coefficient of any n^{th}-order bivariate polynomial term $\Psi_n(\zeta_k(\mathbf{x}),\ldots,\ \zeta_l(\mathbf{x}))$ in Eq. (3.1), which corresponds to a bivariate polynomial term $\Phi_j(\zeta_k,\ \zeta_l)$ in Eq. (3.2), can be computed using the decomposition-based projection method. This method makes use of the bivariate decomposed response in Eq. (3.13) [9]. The expansion coefficients can be obtained by projecting the bivariate terms onto $g(\mathbf{x})$ as

$$\begin{aligned} s_j^{k,l} &= \frac{E\left[g(\mathbf{x})\Phi_j(\zeta_k,\zeta_l)\right]}{E\left[\Phi_j^2(\zeta_k,\zeta_l)\right]} \\ &\cong \frac{\left\{ \begin{aligned} &\sum_{1\le i_1<i_2\le N} E\left[g\left(x_{i_1},x_{i_2},\boldsymbol{\mu}^{i_1,i_2}\right)\Phi_j(\zeta_k,\zeta_l)\right] - (N-2)\sum_{i=1}^{N} E\left[g_i\left(x_i,\boldsymbol{\mu}^i\right)\Phi_j(\zeta_k,\zeta_l)\right] \\ &+\frac{(N-1)(N-2)}{2} E\left[g(\boldsymbol{\mu})\Phi_j(\zeta_k,\zeta_l)\right] \end{aligned} \right\}}{E\left[\Phi_j^2(\zeta_k,\zeta_l)\right]} \\ &= \frac{E\left[g\left(x_k,x_l,\boldsymbol{\mu}^{k,l}\right)\Phi_j(\zeta_k,\zeta_l)\right]}{E\left[\Phi_j^2(\zeta_k,\zeta_l)\right]} \end{aligned} \tag{3.15}$$

It is noted that Eqs. (3.14) and (3.15) require only one-dimensional and two-dimensional integrations and are computationally more efficient than performing the *N*-dimensional integration. Thus, the computational cost in calculating the coefficient of any uni- or bivariate polynomial term is substantially reduced by using the decomposition-based projection method. Similarly, the decomposition-based projection can be extended to compute the coefficients of tri- and higher-variate polynomial terms. However, the coefficient of any tri- or higher-variate polynomial term is treated as zero in this study. This is because of the following two facts: (i) for most engineering problems, considering the interaction between two variables (i.e., the bivariate interaction) is sufficient to yield very accurate statistical results [9], and (ii) the calculations of tri- and higher-variate polynomial coefficients require a substantially larger amount of computational effort, which may make the method computationally intolerable.

Numerical Procedure of Decomposition-Based Projection Method

The numerical integration is required to evaluate the first-order moments in Eqs. (3.14) and (3.15). The most straightforward and efficient way is to directly use the Gaussian quadrature, where Gauss-Hermite, Gauss-Legendre, and Gauss-Jacobi quadrature rules determine the integration points and associated weights for a random variable following Gaussian, Uniform, and Beta distributions, respectively. However, the direct numerical integration may have instability and inaccuracy problems for highly nonlinear performance function g and for high PCE orders while maintaining reasonable efficiency. To enhance the stability and accuracy of the one- and two-dimensional integrations in Eqs. (3.14) and (3.15), we first use the stepwise moving least squares (SMLS) method [10] to construct uni- and bivariate response approximations with the response values evaluated at the predefined samples points, and then carry out the Gaussian quadrature integrations with a large number of integration points (or a large quadrature order) from the approximate response. Note that the uni- and bivariate sample points used to construct the response approximations should not be confused with the integration points in the Gaussian quadrature. Thus, even if the PCE order p is increased, the numbers of uni- and bivariate sample points may not necessarily be increased as long as the response approximations by the SMLS are sufficiently accurate. We believe this is an innovative way to enhance the efficiency in computing the coefficients of high order PCE terms. More detailed information regarding the SMLS and Gaussian quadrature for integrations can be found in the author's previous work [10].

3.1.4 Copula for Nonlinear Correlation Modeling

In many structural reliability analysis and design problems, it is highly probable that the input random variables such as material properties and fatigue properties are correlated [110]. In this case, the reliability analysis and design requires a joint CDF for the exact transformation of the correlated random variables into uncorrelated standard normal random variables. However, it requires an infinite amount of data to acquire the true joint CDF. In contrast, a copula only requires marginal CDFs and a dependence structure to formulate an approximate joint CDF. Thus, the selection of dependence structure and formulation of the joint CDF can be done with a limited amount of data [110].

Introduction of Copula

In statistics, a copula is defined by Roser [111] as "a function that joins or couples multivariate joint distribution functions to their one-dimensional marginal distribution functions", or "multivariate distribution functions whose one-dimensional margins are uniform on the interval [0,1]".

Let F be an N-dimensional cumulative distribution function (CDF) with continuous marginal CDFs F_1, F_2,…, F_N. Then according to Sklar's theorem, there exists a unique N-copula C such that

$$F\left(x_1, x_2, \ldots, x_N\right) = C\left(F_1\left(x_1\right), F_2\left(x_2\right), \ldots, F_N\left(x_N\right)\right) \tag{3.16}$$

It then becomes clear that a copula formulates a joint CDF with the support of separate marginal CDFs and a dependence structure. The copula is capable of constructing the joint CDF in real applications with different types of marginal CDFs or dependence structures. Various general types of dependence structures can be represented, corresponding to various copula families, such as Gaussian, Clayton, Frank, and Gumbel. Let $u_i = F_i(x_i)$, i = 1, 2,…, N, a N-dimensional Archimedean copula is defined as

$$C\left(u_1, u_2, \cdots, u_N \mid \alpha\right) = \Psi_\alpha^{-1}\left(\sum_{i=1}^{N} \Psi_\alpha\left(u_i\right)\right) \tag{3.17}$$

where Ψ_α denotes a generator function with a correlation parameter α and satisfies the following conditions:

$$\Psi_\alpha(1) = 0; \ \lim_{u \to 0} \Psi_\alpha(u) = \infty; \ \frac{d}{du}\Psi_\alpha(u) < 0; \ \frac{d^2}{du^2}\Psi_\alpha(u) > 0 \tag{3.18}$$

Let $\Psi_\alpha(u) = u^\alpha - 1$ and $N = 2$, then we formulate a bivariate Clayton copula as

$$C\left(u_1,u_2 \mid \alpha\right)=\left(u_1^{-\alpha}+u_2^{-\alpha}-1\right)^{-1/\alpha} \tag{3.19}$$

More detailed information on copula families can be found in References [110,111].

Rosenblatt Transformation

The Rosenblatt transformation has been used extensively for mapping the correlated random variables onto the independent standard normal variables. The successive conditioning procedures for a vector of correlated random variables are defined as [112]

$$\begin{aligned} z_1 &= \varphi^{-1}\left[F_1\left(x_1\right)\right] \\ z_2 &= \varphi^{-1}\left[F_2\left(x_2 \mid x_1\right)\right] \\ &\vdots \\ z_N &= \varphi^{-1}\left[F_N\left(x_N \mid x_1, x_2, \cdots, x_{N-1}\right)\right] \end{aligned} \tag{3.20}$$

where z_1, z_2,…, z_N denote the independent standard random variables after the transformation, $\varphi^{-1}(\cdot)$ denotes the inverse CDF of a standard normal variable, $F_i(x_i|x_1, x_2,\ldots, x_{i-1})$ denotes the CDF of x_i conditioned on $X_1 = x_1$, $X_2 = x_2$,…, $X_{i-1} = x_{i-1}$, and can be expressed as

$$F_i\left(x_i \mid x_1, x_2, \cdots, x_{i-1}\right)=\frac{\int_{-\infty}^{x_i} f_i\left(x_1, x_2, \cdots, x_{i-1}, \tau\right) d\tau}{f_{i-1}\left(x_1, x_2, \cdots, x_{i-1}\right)} \tag{3.21}$$

where $f_i(x_1, x_2,\ldots, x_i)$ denotes the marginal joint PDF of x_1, x_2,…, x_i.

To use the Rosenblatt transformation for the purpose of reliability analysis and design, the joint CDF of input random variables should be available. However, it is very difficult to obtain the joint CDF in real applications. In contrast, a copula can easily formulate an approximate joint CDF based on separate marginal CDFs and correlation parameters, which can be practically obtained from limited experimental data [110]. The Rosenblatt transformation for a bivariate copula is given as

$$\begin{aligned} z_1 &= \varphi^{-1}\left[u_1\right] = \varphi^{-1}\left[F_1\left(x_1\right)\right] \\ z_2 &= \varphi^{-1}\left[C\left(u_2 \mid u_1\right)\right] = \varphi^{-1}\left[C\left(F_2\left(x_2\right) \mid F_1\left(x_1\right)\right)\right] \end{aligned} \tag{3.22}$$

where

$$\begin{aligned} C\left(u_2 \mid u_1\right) = P\left(U_2 \le u_2 \mid U_1 = u_1\right) &= \lim_{\Delta u_1 \to 0} \frac{C\left(u_1 + \Delta u_1, u_2\right) - C\left(u_1, u_2\right)}{\Delta u_1} \\ &= \frac{\partial C\left(u_1, u_2\right)}{\partial u_1} \end{aligned} \tag{3.23}$$

After the Rosenblatt transformation, the independent standard random variables are used as the Gaussian input variables for the generalized PCE with Hermite polynomial basis. A vehicle side-crash example in the case study section illustrates the feasibility of the proposed method.

3.1.5 Reliability and Sensitivity Analysis

Reliability Analysis

Once the uni- and bivariate PCE coefficients are calculated, an approximate function of the original implicit performance function g is obtained as

$$\hat{g}\left(\mathbf{x}\right) = g\left(\boldsymbol{\mu}\right) + \sum_{k=1}^{N}\sum_{j} s_j^k \Phi_j\left(\zeta_k\left(\mathbf{x}\right)\right) + \sum_{k,l=1;k<l}^{N}\sum_{j} s_j^{k,l} \Phi_j\left(\zeta_k\left(\mathbf{x}\right), \zeta_l\left(\mathbf{x}\right)\right) \tag{3.24}$$

The above expression can be viewed as an explicit mapping $\hat{g}: \mathbb{R}^N \to \mathbb{R}$, which approximates the exact implicit mapping g: $\mathbb{R}^N \to \mathbb{R}$. Thus, any probabilistic characteristics of $g(\mathbf{x})$, including statistical moments, reliability, and PDF, can be easily estimated by performing MCS. For example, any r^{th} moment can be calculated as

$$\begin{aligned} m_r &\cong \int \hat{g}^r(\mathbf{x}) f(\mathbf{x}) d\mathbf{x} \\ &= E\left(\hat{g}^r(\mathbf{x})\right) = \lim_{ns \to \infty} \frac{1}{ns} \sum_{k=1}^{ns} \hat{g}^r\left(\mathbf{x}^{(k)}\right) \end{aligned} \tag{3.25}$$

where m_{r} is the r^{th} moment of the performance function g; $f(\mathbf{x})$ is the joint PDF; $\mathbf{x}^{(k)}$ is the k^{th} realization of $\mathbf{x}$; and ns is the sampling size. It is noted that, although MCS is used to compute the moments due to its convenience, it is not required since moments of a PCE can be analytically obtained. Low-order moments (e.g., mean and variance) have simple analytical forms while high-order moments, for which orthogonality cannot be fully exploited, possess complicated forms. For reliability calculation, let us define an approximate safe domain for the performance function g as

$$\hat{\Omega}^S = \left\{\mathbf{x} : \hat{g}(\mathbf{x}) < 0\right\} \tag{3.26}$$

Therefore, the reliability R can also be estimated by performing MCS as

$$\begin{aligned} R &\cong \int I_{\hat{\Omega}^S}(\mathbf{x}) f(\mathbf{x}) d\mathbf{x} \\ &= E\left(I_{\hat{\Omega}^S}(\mathbf{x})\right) = \lim_{ns \to \infty} \frac{1}{ns} \sum_{k=1}^{ns} I_{\hat{\Omega}^S}\left(\mathbf{x}^{(k)}\right) \end{aligned} \tag{3.27}$$

where $I[\cdot]$ is an indicator function of safe or fail state such that

$$I_{\hat{\Omega}^S}\left(\mathbf{x}^{(k)}\right) = \begin{cases} 1, & \mathbf{x}^{(k)} \in \hat{\Omega}^S \\ 0, & \mathbf{x}^{(k)} \in \Omega \setminus \hat{\Omega}^S \end{cases} \tag{3.28}$$

It should be noted that the MCS performed here is inexpensive because it employs the explicit representation function in Eq. (3.24).

Probabilistic Sensitivity Analysis

In reliability-based design optimization (RBDO), probabilistic sensitivity analysis is required to identify the effect of the change in the parameters of random variables upon the change in reliability or moments. Since MCS is used for evaluating the statistical properties (e.g., r^{th} moment, reliability) of a response in the adaptive-sparse PCE method, this study computes the probabilistic sensitivity of the response with respect to a random variable using a finite difference method (FDM). The FDM uses the original and perturbed values of moments or reliabilities to computes their sensitivities.

The sensitivity of any r^{th} moment and reliability with respect to the j^{th} element θ_j (e.g., μ, or σ, etc.) in a vector of deterministic distribution parameters $\boldsymbol{\theta}$ is computed using Eqs. (3.29) and (3.30), respectively.

$$\frac{\partial m_r(\boldsymbol{\theta})}{\partial \theta_j} \cong \frac{m_r(\theta_j + \Delta\theta_j) - m_r(\theta_j)}{\Delta\theta_j} \tag{3.29}$$

$$\frac{\partial R(\boldsymbol{\theta})}{\partial \theta_j} \cong \frac{R(\theta_j + \Delta\theta_j) - R(\theta_j)}{\Delta\theta_j} \tag{3.30}$$

where $m_r(\boldsymbol{\theta})$ is the r^{th} moment of the constraint G (or the cost function C); $\Delta\theta_j$ is the perturbed value of θ_j. A perturbation size of 0.1% is employed in this study. It is noted that, for computing a perturbed moment or reliability, an extra MCS based on the approximate response model in Eq. (3.24) is used without extra computational cost. For the extra MCS, the random number seeds for the original MCS should be reused to reduce numerical noise and obtain a stable sensitivity estimate. As an alternative to the FDM, the score function can also be used to compute the probabilistic sensitivities [113] and we observed similar performance.

3.1.6 Computational Procedure

The overall computation procedure is shown in Figure 3-1. If nonlinear correlation exists between the random inputs **x**, the copula is employed to model the joint PDF $f(\mathbf{x})$ and the Rosenblatt transformation to transform **x** to independent standard normal variables **z**. The computations of PCE coefficients s_j^k in Eq. (3.14), require the response values (i.e., values of the performance function) at the univariate sample points: $g(\boldsymbol{\mu})$, $g(x_k^{(i_k)}, \boldsymbol{\mu}^k)$, for $i_k = 1,2,\ldots, M_1$, where the superscript i_k denotes the corresponding sample point for x_k and M_1 the number of univariate sample points in each dimension. The computations of PCE coefficients $s_j^{k,l}$ in Eq. (3.15) require the response values at the bivariate sample points: $g(x_k^{(i_{k,l})}, x_l^{(i_{k,l})}, \boldsymbol{\mu}^{k,l})$, for $i_{k,l} = 1,2,\ldots, M_2$, where the superscript $i_{k,l}$ denotes the corresponding bivariate sample points for the bivariate term $[x_k, x_l]$, and M_2 the number of bivariate sample points for each bivariate term. Thus, the total number of function evaluations for the adaptive-sparse PCE with q bivariate terms is $q(M_2 - 1) + M_1(M_1 - 1)/2 + (M_1 - 1)N + 1$. Below are several important remarks regarding the properties of the adaptive-sparse PCE.

Remark 1. The N-variate, p^{th}-order adaptive-sparse PCE is a finite sum of uni- and bivariate polynomial terms up to the p^{th} order, with the coefficient of any tri- or higher-variate polynomial term being zero. Thus, if the tri- and higher-variate interactions are negligible, the adaptive-sparse PCE gives an accurate approximation of the function g, with a lower computational effort than the conventional PCE. Otherwise, numerical error in the adaptive-sparse PCE may be stacked up due to the tri- and higher-variate interactions. More detailed error analysis will be given in the subsequent section.

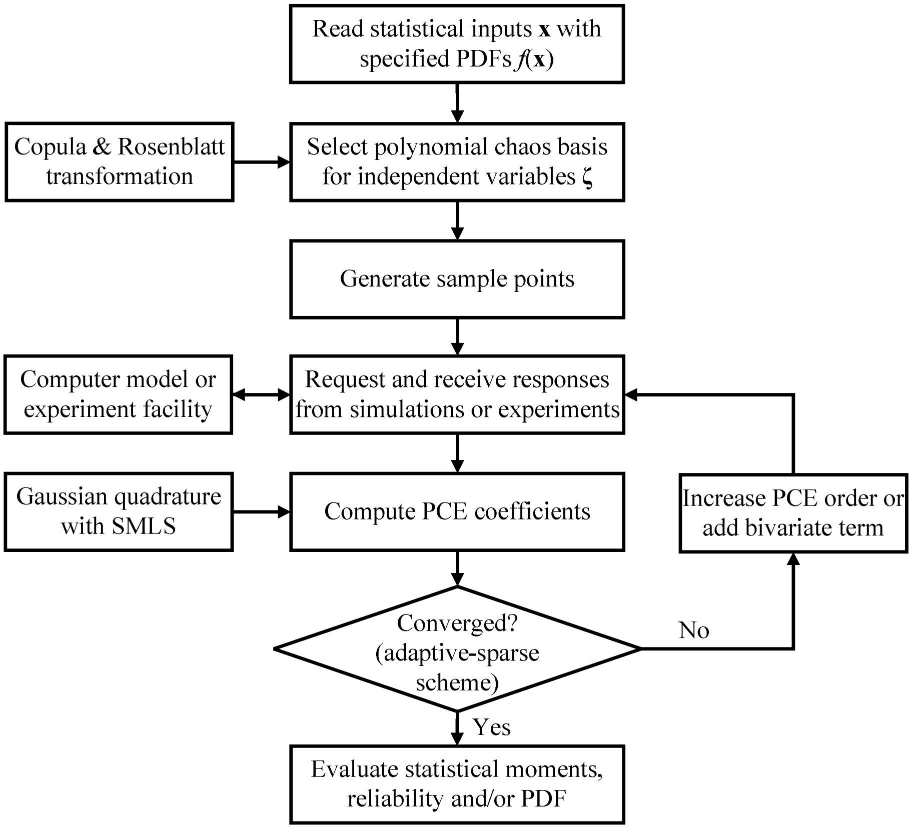


Figure 3-1 Flowchart of the adaptive-sparse algorithm.

Remark 2. The uni- and bivaraite dimension reduction methods have been extensively studied for reliability analysis and design by previous researchers [8-10,114,115]. However, no attempt has been made to optimize the number of the bivariate terms to be considered for probability analysis. The common approach either depends on the univariate dimension reduction (UDR) [8,10,114] or makes comparison with its bivairate counterpart, bivariate dimension reduction (BDR) [9,115]. The method developed here uses the error indicator in Eq. (3.8) to adaptively add the bivariate terms to the PCE model until a convergence criterion is achieved. This adaptive process takes advantage of the PCE as the projection basis. The inherent characteristics of orthogonal polynomials make the adaptive process computationally efficient and convergent. Therefore, we argue that the adaptive-sparse PCE achieves an optimal compromise between the UDR and BDR (more accurate than the UDR and more efficient than the BDR).

Remark 3. In addition to the Rosenblatt transformation, alternative transformation techniques (e.g., Nataf transformation [116]) are also capable of transforming Gaussian variables with nonlinear correlation to independent Gaussian variables. In the current study, the non-Gaussian variables with nonlinear correlation are all transformed to independent Gaussian variables. However, it may also be possible to transform the original random variables to independent non-Gaussian variables (e.g., gamma, beta) with distribution types supported by the PCE. Thus, the selections of an appropriate transformation technique and procedure are worthy of future studies.

Remark 4. Through extensive testing with many mathematical and engineering examples, we observed that the parameter setting $\varepsilon_1 = 0.01$ and $\varepsilon_2 = 0.001$ achieves a near-optimum compromise between the accuracy and efficiency. Thus we intended to make this setting as a guideline for implementing the algorithm in most engineering cases. More conservative criteria may give higher accuracy but require more computational effort. Thus, for a specific problem, the optimum ε_1 and ε_2 may vary depending on the requirements on the accuracy and efficiency.

3.1.7 Error Decomposition Scheme

The proposed adaptive-sparse PCE method integrates the adaptive-sparse scheme and the decomposition-based projection method with the PCE method. It is obvious that the approximation and numerical schemes produce associated errors in the proposed adaptive-sparse PCE method. This study therefore analyzes the approximation and numerical errors in the proposed method, which provides insights into identifying the most appropriate applications for the proposed method. There are three primary error sources: (i) a PCE truncation error (ε_P), (ii) an error due to a univariate decomposition (ε_U), (iii) an error due to a bivariate decomposition (ε_B), and (iv) an aliasing error of approximating the first-order moments in Eqs. (3.14) and (3.15) via the SMLS and Gaussian quadrature integration. The total error is a mean-squares error of the N-variate, p^{th}-order adaptive-sparse PCE with the set $\mathbf{B}$ of bivariate terms and can be decomposed as

$$\begin{aligned}
\varepsilon^2 &= \int_\Gamma \left[g(\mathbf{x}) - w^p(\zeta(\mathbf{x})) \right]^2 f(\mathbf{x})\,d\mathbf{x} \\
&\leq \int_\Gamma \left[g - w_{UB}^p \right]^2 f(\mathbf{x})\,d\mathbf{x} + \int_\Gamma \left[w_{UB}^p - w_U^p \right]^2 f(\mathbf{x})\,d\mathbf{x} \\
&\quad + \int_\Gamma \left[w_U^p - w_I^p \right]^2 f(\mathbf{x})\,d\mathbf{x} + \int_\Gamma \left[w_I^p - w^p \right]^2 f(\mathbf{x})\,d\mathbf{x} \\
&= \varepsilon_P^2 + \varepsilon_B^2 + \varepsilon_U^2 + \varepsilon_I^2
\end{aligned} \tag{3.31}$$

The detailed derivation of the first three error terms in Eq. (3.31) can be found in Appendix A.

3.1.8 Case Studies for Adaptive-Sparse PCE

Three mathematical and engineering examples are given in this section to demonstrate the effectiveness of the adaptive-sparse PCE method. The first two examples were used for studying the computational accuracy and efficiency of the proposed method for uncertainty quantification and reliability analysis. For comparison purpose, we also employ FORM as a classic reliability analysis method, and the univariate DR (UDR) method (with the Pearson PDF generation system) as a representative of the recently developed moment-based reliability methods [8-10]. In the last example, we carried out reliability-based robust design optimization

(RBRDO) for a lower control arm in a high mobility, multipurpose, wheeled vehicle (HMMWV). This case study demonstrates the feasibility of the proposed method in complex product or process design.

V6 Gasoline Engine Power Loss: Bimodal PDF

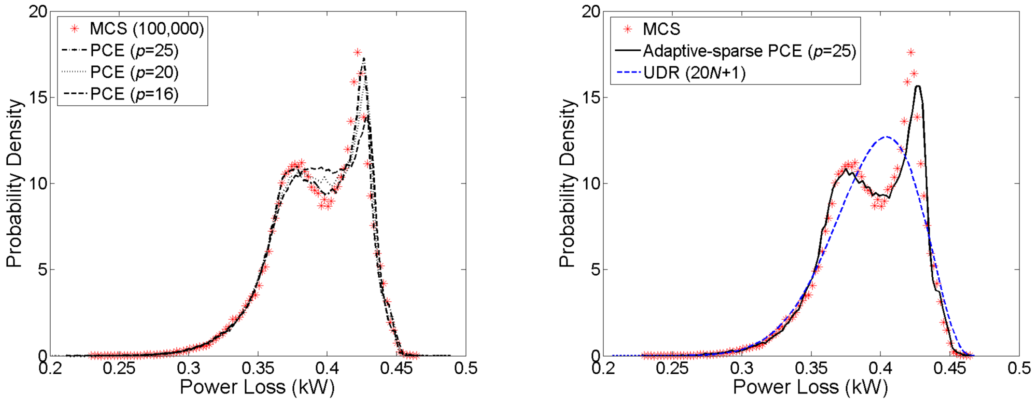
This example is the V6 gasoline engine problem studied by Lee [25]. The performance function considered in this example is the power loss due to the friction between the piston ring and the cylinder liner, oil consumption, blow-by, and liner wear rate. A ring/liner subassembly simulation model was used to compute the power loss. The simulation model has four input parameters, the ring surface roughness x_1, liner surface roughness x_2, linear Young's modulus x_3 and linear hardness x_4. Of the total four 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 [25] that the power loss has a bimodal PDF. To predict the bimodal PDF, the adaptive-sparse PCE used $M_1 = 20$ and $M_2 = 20$. As shown in Table 3-2, the adaptive-sparse expansion scheme was converged with $p = 25$ and $q = 1$. Figure 3-2 shows the PDF approximations by the 16th, 20th and 25th order PCEs with full tensorized Gaussian quadrature ($M_1 = 17$ for $p = 16$, $M_1 = 21$ for $p = 20$, and $M_1 = 26$ for $p = 25$), UDR and adaptive-sparse PCE. Both the adaptive-sparse PCE and 20th order PCE with Gaussian quadrature produce accurate approximations for the left peak and tail regions of the PDF. The 16th order PCE cannot accurately approximate this bimodal PDF (see Figure 3-2a) while the 25th order PCE gives the most accurate solution. As shown in Figure 3-2b, the UDR fails to represent the irregular shape of this PDF. The comparison results in Table 3-3 suggest that the adaptive-sparse PCE method is more accurate than the UDR method, particularly for system responses with strong bivariate interactions. The error in the probability estimation by FORM is due to the nonlinearity of the power loss function. The computational cost by the adaptive-sparse PCE method is much lower than that by the conventional PCE method with full tensorized Gaussian quadrature.

Table 3-2 Adaptive-sparse process of the adaptive-sparse PCE for the V6 engine example

	PCE order (p)	No. of bivariate terms (q)	No. FE	R^2	Reliability	Relative error (%)
Step 1	2	0	41	0.98842	0.00687	26.987
Step 2	2	1	61	0.98810	0.00717	32.532
Step 3	3	1	61	0.97922	0.00468	13.494
Step 4 ~ 23	…	…	…	…	…	...
Step 24	24	1	61	0.98197	0.00547	1.109
Step 25	25	1	61	0.98273	0.00547	1.109

Table 3-3 Probability analysis results for the V6 engine example

	adaptive-sparse PCE (p = 25)	MCS	PCE (p = 20, Gauss Quad)	20N+1 UDR	FORM
Mean (kW)	0.3935	0.3935	0.3934	0.3935	-
Std. dev. (kW)	0.0315	0.0310	0.0311	0.0314	-
Skewness	-0.5527	-0.5883	-0.5735	-0.5393	-
Kurtosis	3.0249	3.0828	3.0599	3.0974	-
Pr(PL < 0.3)	0.0056	0.0054	0.0054	0.0048	0.0057
No. FE	61	100,000	441	41	15



(a) (b)

Figure 3-2 PDF approximations by the PCEs (a), adaptive-sparse PCE and UDR (b) for the V6 engine example

<u>Side-Impact Crash Problem: Nonlinear Correlation</u>

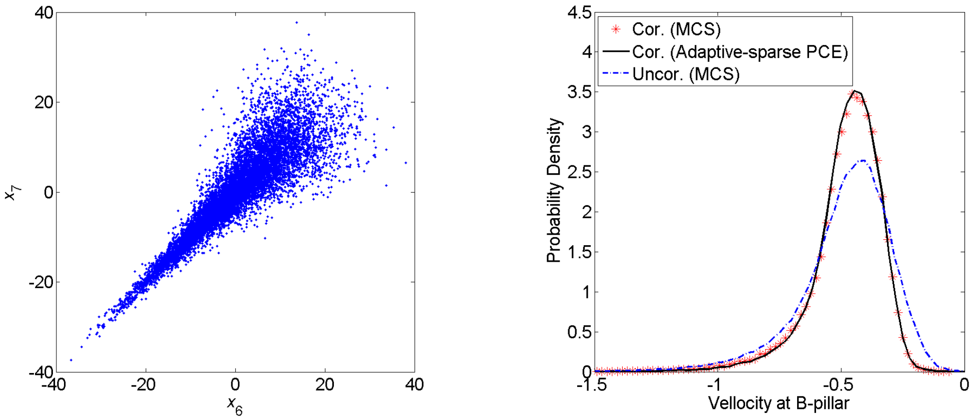
Vehicle side-impact responses [117] are considered for system performances with statistical nonlinear correlation modeled by a copula theory [110,111]. The properties of the design and random variables are shown in Table 3-4. This example considered the velocity of a front door at B-pillar. The failure is defined when the velocity exceeds the threshold value 15.7. Thus, the system performance can be expressed as

$$g(\mathbf{x}) = \begin{pmatrix} 16.45 - 0.489 x_1 x_4 - 0.843 x_2 x_3 + 0.0432 x_5 x_6 \\ -0.0556 x_5 x_7 - 0.000786 x_7^2 \end{pmatrix} - 15.7 \qquad (3.32)$$

In the study, the random variables x_6 and x_7 with the maximum variation were assumed to have a statistical nonlinear correlation described by a Clayton copula, as shown in Figure 3-3a. The rank correlation coefficient was used to quantify the nonlinear correlation. In this case, we assumed the rank correlation coefficient Kendall's τ to be 0.75 and the corresponding copula parameter to be 6.0. As discussed in Section 3.1.4, the Rosenblatt transformation is required to transform correlated input variables into uncorrelated standard normal variables. Using $M_1 = 4$ and $M_2 = 8$, the adaptive-sparse expansion scheme was converged with p = 3 and q = 1 and the bivariate term considered was [x_6, x_7], which were nonlinearly correlated. To illustrate the effect of statistical nonlinear correlation on the system response, the PDFs for both correlated and uncorrelated cases are shown in Figure 3-3b. It shows that the nonlinear correlation affects the PDF of the system performance significantly and that the adaptive-sparse PCE accurately predicted the peak and tail regions of the PDF.

Table 3-4 Input random variables for the side impact example

Random input	Distri. type	Mean	Std. dev.	Lower bound	Upper bound	Mode
x_1	Beta	1.500	0.050	1.000	1.800	-
x_2	Uniform	-	-	0.850	1.150	-
x_3	Uniform	-	-	0.699	0.999	-
x_4	Uniform	-	-	0.850	1.150	-
x_5	Triangular	-	-	0.327	0.363	0.345
x_6	Normal	0	10.000	-	-	-
x_7	Normal	0	10.000	-	-	-



(a) (b)

Figure 3-3 Scatter plot of input variables x_6 and x_7 (a), and PDF results (b)

Quantitative results are summarized in Table 3-5. To further study the effects of different correlation coefficients on the reliability estimation, we plotted in Figure 3-4 the reliabilities for increasing values of Kendall's τ. As shown in the figure, the correlation coefficients significantly affect the reliabilities and the adaptive-sparse PCE maintains consistent accuracy within ± 0.01 at all reliability levels.

Table 3-5 Probability analysis results for the side impact example ($\tau = 0.75$)

	adaptive-sparse PCE ($p = 3$)	MCS
Mean	-0.4766	-0.4813
Std. dev.	0.1408	0.1520
Skewness	-1.7109	-1.7402
Kurtosis	10.2690	9.2106
Pr($g < 0$)	0.9496	0.9437
No. FE	57	1,000,000

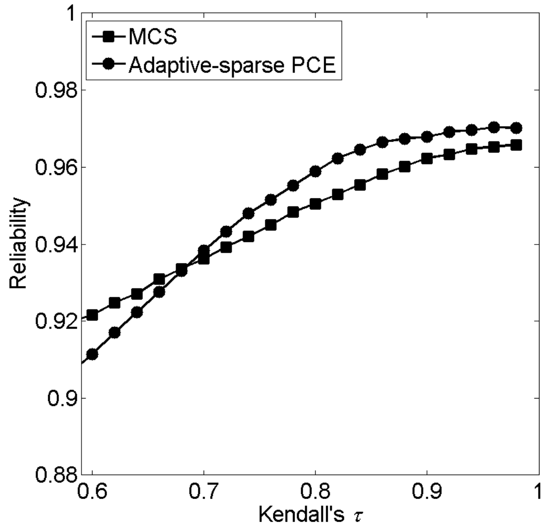


Figure 3-4 Reliabilities for increasing values of Kendall's τ

Lower Control A-Arm: RBRDO against a Fatigue Failure

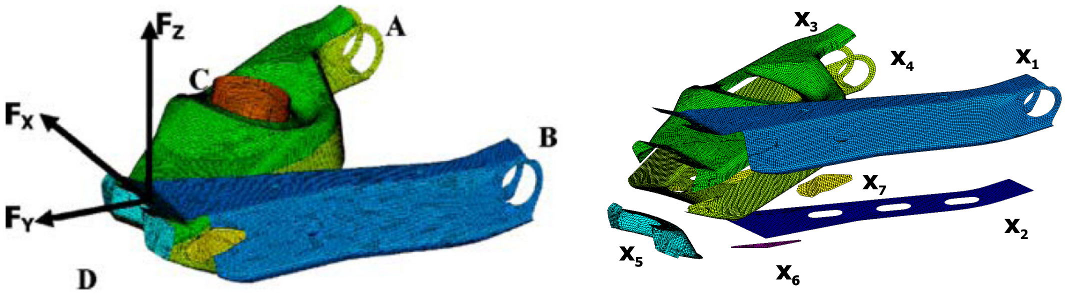
Vehicle suspension systems experience intense loading conditions throughout their service lives. Control arms act as the backbone of the suspension system, through which the majority of these loads are transmitted [118]. Therefore, it is crucial that control arms be highly reliable while its mass is minimized. The lower control-arm was modeled with plane stress elements using 54,666 nodes, 53,589 elements, and 327,961 DOFs, where all welds were modeled using rigid beam elements. Hyper-Works 8.0 was used for FE modeling and design parameterization. The loading and boundary conditions are shown in Figure 3-5a. The loading was applied at the ball-joint (point D) in three directions, and the boundary conditions were applied to simulate the bushing joints (points A and B) and the joint with a shock absorber and spring assemble (point C). This HMMWV lower control-arm model was used for RBRDO using the adaptive-sparse PCE method.

RBRDO Formulation

From a worst-case scenario analysis, 91 constraints (G_1 to G_{91}) were defined in several critical regions using the von Mises stress, as shown in Figure 3-6. With 91 stress constraints, the RBRDO is formulated as

$$\begin{aligned} &\text{Minimize} \quad Q = \mu_m + \sigma_m \\ &\text{Subject to} \quad R_i = \Pr\left(G_i\left(\mathbf{x};\mathbf{d}\right) = \frac{s_i\left(\mathbf{x};\mathbf{d}\right)}{s_y} - 1 \le 0 \right) \ge \Phi\left(\beta_i^t\right) = R_i^t,\ i = 1,\cdots,91 \\ &\qquad\qquad \mathbf{d}^{\mathbf{L}} \le \mathbf{d} \le \mathbf{d}^{\mathbf{U}} \end{aligned} \tag{3.33}$$

where, the objective function Q is the summation of the mean μ_m and standard deviation σ_m of the mass; $\mathbf{x}$ is the random vector; $\mathbf{d} = \boldsymbol{\mu}(\mathbf{x})$ is the design vector; s_i is the von Mises stress of the i^{th} constraint; s_y is the yield stress and was set to 60.9 ksi for any constraint; R_i^t is the target reliability level and was set to 99.87% for any constraint, which corresponds to a target reliability index $\beta_i^t = 3.0$. The seven design variables are the thicknesses of the seven major components of the control arm, as shown in Figure 3-5b. Three load variables (not design variables) are considered as random noisy variables. The statistical information of these random and design variables is summarized in Table 3-6 and Table 3-7, respectively.



(a) (b)

Figure 3-5 Three random load variables (a) and seven design variables (b)

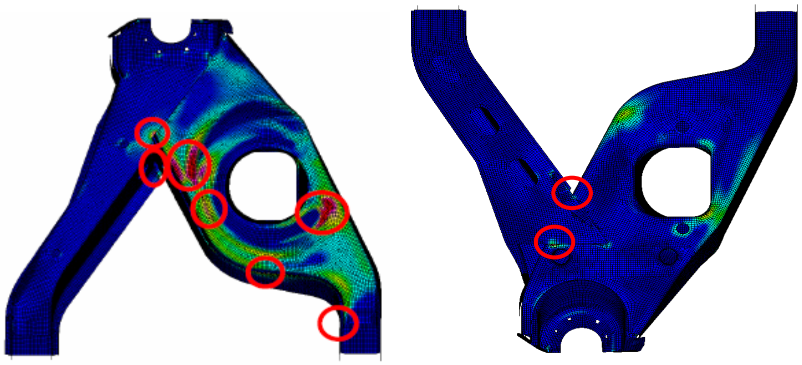


Figure 3-6 Ninety-one critical constraints of the lower control A-arm model.

Table 3-6 Random force variables for the lower control A-arm model

Random variable	Distri. type	Mean	Std. dev.
F_X	Normal	1900	95
F_Y	Normal	95	4.75
F_Z	Normal	950	47.5

Table 3-7 Design variables for the lower control A-arm model

Design variable	Distri. type	Lower bound	Initial des.	Upper bound	Std. dev.
x_1	Normal	0.100	0.120	0.500	0.006
x_2	Normal	0.100	0.120	0.500	0.006
x_3	Normal	0.100	0.180	0.500	0.009
x_4	Normal	0.100	0.135	0.500	0.007
x_5	Normal	0.150	0.250	0.500	0.013
x_6	Normal	0.100	0.180	0.500	0.009
x_7	Normal	0.100	0.135	0.500	0.007

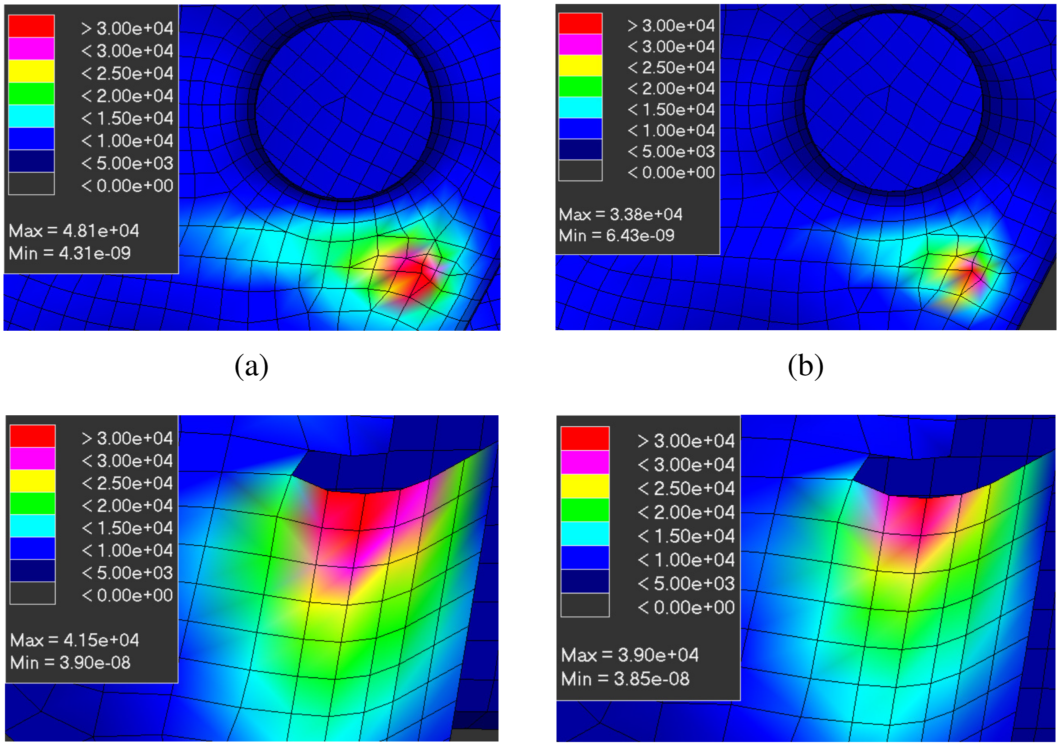
Optimization results

The adaptive-sparse PCE method with $4N + 1$ (= 41) FE analyses was carried out to evaluate the quality function, 91 reliabilities, and their sensitivities at any design iteration, without considering the bivariate polynomial basis functions. The sensitivities of the quality function and reliabilities with respect to the seven design variable were computed by using a finite difference method (FDM) at each design Iteration. The perturbed values of the quality function and reliabilities were estimated based on approximate stochastic response surfaces (PCE) with perturbed design variables, without requiring gradients of the original mass or stress functions. A perturbation size of 0.1% is employed in this study.

The design optimization problem was solved using a gradient-based optimization technique (e.g., sequential quadratic optimization). The histories of the design parameters, objective function, and reliabilities for significant constraints G_6, G_{80} and G_{87} are shown in Table 3-8. At the initial design, the constraints G_6 and G_{80} severely violated the reliability requirement. After seven design iterations, the optimum design was found where all the reliability requirements were satisfied. Overall, the adaptive-sparse PCE method required 287 FE simulations for RBRDO. After the optimization, the direct MCS with 5,000 random samples was employed to verify the reliability results at the optimum design. The reliabilities of constraints G_6, G_{80} and G_{87} were estimated by the MCS as 99.71%, 99.88%, and 99.84%, respectively, and all the other constraints were confirmed with 100% reliabilities.

Table 3-8 Design history of the lower control A-arm model

Iter.	Design Variables							R_6	R_{80}	R_{87}	Obj.
	x_1	x_2	x_3	x_4	x_5	x_6	x_7				
0	0.120	0.120	0.180	0.135	0.250	0.180	0.135	0.3235	0.0050	1.0000	31.473
1	0.100	0.142	0.150	0.164	0.150	0.500	0.100	0.9989	0.9970	0.9620	32.044
2	0.100	0.140	0.169	0.161	0.150	0.500	0.325	0.9988	0.9982	0.9998	32.875
3	0.100	0.140	0.160	0.162	0.150	0.500	0.336	0.9982	0.9986	0.9963	32.513
4	0.100	0.140	0.164	0.164	0.150	0.500	0.228	0.9988	0.9989	0.9991	32.763
5	0.100	0.140	0.162	0.164	0.150	0.500	0.224	0.9986	0.9984	0.9982	32.607
6	0.100	0.140	0.163	0.164	0.150	0.500	0.211	0.9985	0.9988	0.9991	32.697
7	0.100	0.140	0.164	0.164	0.150	0.500	0.210	0.9987	0.9989	0.9991	32.717
Opt	0.100	0.140	0.164	0.164	0.150	0.500	0.210	0.9987	0.9989	0.9991	32.717



(a) (b)

(c) (d)

Figure 3-7 Stress comparisons of initial and optimum design: (a) G_6 at initial design, (b) G_6 at optimum design, (c) G_{80} at initial design, (d) G_{80} at optimum design

The stress contours at the initial and optimum designs for constraints G_6 and G_{80} are shown in Figure 3-7. It can be seen in both constraints that the high stress areas are greatly reduced by the RBRDO process.

3.1.9 Summary

In the first part of this chapter, the adaptive-sparse PCE method is proposed for efficient component reliability analysis involving high nonlinearity or large dimension. The adaptive-sparse PCE method combines four ideas and methods: (1) an adaptive-sparse scheme to determine the number (q) of the most significant bivariate terms and PCE order (p) in the PCE model; (2) an efficient decomposition-based projection method using the SMLS method; (3) the integration of the copula system to handle nonlinear correlation of input random variables, and (4) the systematic error decomposition analysis in the proposed method. It was found in many examples that the adaptive-sparse scheme and decomposition-based projection method achieves greater accuracy and efficiency than other probability/reliability methods, including FORM/SORM and moment-based reliability methods. This high accuracy can be attributed to the consideration of significant bivariate response components and the accurate integration scheme by the SMLS method. The adaptive-sparse PCE method can also approximate a multi-modal PDF. Moreover, the proposed method is stable, unlike other probability/reliability methods, since it does not require a distribution generation system.

3.2 Asymmetric Dimension-Adaptive Tensor-Product Method

As mentioned in the literature review, the stochastic collocation (SC) method achieves great improvement in reducing the curse of dimensionality encountered in other advanced reliability analysis methods. To further enhance the computational efficiency of the SC method, this research presents an asymmetric dimension-adaptive tensor-product (ADATP) method. The proposed method leverages three ideas: (i) an asymmetric dimension-adaptive scheme to efficiently build the tensor-product interpolation considering both directional and dimensional importance, (ii) a hierarchical interpolation scheme using either piecewise multi-linear basis functions or cubic Lagrange splines, (iii) a hierarchical surplus as an error indicator to automatically detect the highly nonlinear regions in a random space and adaptively refine the collocation points in these regions. The proposed method has three distinct features for reliability analysis: (a) automatically detecting and adaptively reproducing tri- and higher-variate interactions, (b) greatly alleviating the curse of dimensionality, and (c) no need of response sensitivities. The ADATP method is expected to perform accuracy and efficient reliability analysis for highly nonlinear responses with significant tri- and higher-order interactions.

3.2.1 Review of Stochastic Collocation methods

Great attention has been paid to the stochastic collocation method for approximating a multi-dimensional random function due to its strong mathematical foundation and ability to achieve fast convergence for interpolation construction. This section reviews the stochastic collocation methods using the tensor-product grid, the conventional and generalized sparse grids, and the hierarchical interpolation scheme using multivariate hierarchical basis functions.

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 $f_i(x^{\mathrm{i}})$: $\Pi_i \rightarrow \mathbb{R}^+$, where the support Π_i is a one-dimensional random space of x^i. Under the assumption of statistical independence, the probabilistic characteristics of the random variables $\mathbf{x}$ can then be completely defined by the joint PDF $f(\mathbf{x}) = f_1(x^1)\cdot f_2(x^2)\cdot\cdots\cdot f_N(x^N)$ with the support $\Pi = \Pi_1\cdot\Pi_2\cdot\cdots\cdot\Pi_N$. If the assumption of statistical independence does not hold, that is, the random variables such as fatigue material properties (fatigue ductility coefficient and exponent) are statistically dependent, a copula [110,111] can be employed to select an appropriate dependence structure and formulate a joint CDF of the random variables based on available input data, which then allows the use of the Rosenblatt transformation [112] to transform the dependant random variables into independent standard normal random variables. A numerical investigation on how to deal with dependant random variables is provided in the subsequent case study section. 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 an 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 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 $\mathbf{\Theta} = \{\mathbf{x}_j \mid \mathbf{x}_j \in \Gamma, j = 1,\dots, M_T\}$. Suppose that we can obtain the performance function value $g(\mathbf{x}_j)$ at each collocation point $\mathbf{x}_j$. We then aim at building 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 the tensor-product grid, conventional sparse grid based on the Smolyak algorithm [26], or generalized sparse grid based on the dimension-adaptive tensor-product algorithm [33]. We begin by constructing the interpolation with the 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

$$U^i\left(g\right)=\sum_{j=1}^{m_i} a_j^i \cdot g\left(x_j^i\right) \tag{3.34}$$

with a set of support nodes

$$X^i=\left\{x_j^i \mid x_j^i \in\left[0,1\right],\ j=1,2,\ldots,m_i\right\} \tag{3.35}$$

where $i \in \mathbb{N}$ is the interpolation level, $a_j^i \in C([0,1])$ the jth interpolation nodal basis functions, x_j^i the jth support nodes and m_i the number of support nodes in the interpolation level i. Note that, by following the descriptions in References [31,34,35], we use the superscript i to denote the interpolation level during the development of stochastic collocation methods. Two widely used nodal basis functions are piecewise multi-linear basis functions and Lagrange polynomials. Here we will 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 [34]. In the case of a univariate interpolation (N = 1), the support nodes are defined as

$$m_i = \begin{cases} 1 & \text{if } i = 1 \\ 2^{i-1} + 1, & \text{if } i > 1 \end{cases}$$

$$x_j^i = \begin{cases} \dfrac{j-1}{m_i - 1} & \text{for } j = 1, \ldots, m_i \text{ if } m_i > 1 \\ 0.5 & \text{for } j = 1, \ldots, m_i \text{ if } m_i = 1 \end{cases} \tag{3.36}$$

The resulting set of the points fulfill 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 [34]

$$a_j^i = 1 \text{ for } i = 1$$

$$a_j^i = \begin{cases} 1 - (m_i - 1) \cdot \left| x - x_j^i \right|, & \text{if } \left| x - x_j^i \right| < 1/(1 - m_i) \\ 0, & \text{otherwise} \end{cases} \tag{3.37}$$

for $i > 1$. More detailed information on the one-dimensional interpolation can be found in [34].

Applying a sequence of formulas in Eq. (3.34) on 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

$$\left(U^{i_1} \otimes \cdots \otimes U^{i_N}\right)(g) = \sum_{j_1=1}^{m_1} \cdots \sum_{j_N=1}^{m_N} \left(a_{j_1}^{i_1} \otimes \cdots \otimes a_{j_N}^{i_N}\right) \cdot g\left(x_{j_1}^{i_1}, \ldots, x_{j_N}^{i_N}\right) \tag{3.38}$$

where the superscript i_k, $k = 1, \ldots, N$, denotes the interpolation level along the k^{th} dimension, U^{i_k} are the interpolation functions with the interpolation level i_k along the k^{th} dimension and the subscript j_k, $k = 1, \ldots, N$, denotes the index of a given support node in the k^{th} dimension. The number of function evaluations required by the tensor-product formula reads

$$M_T = m_1 \cdot m_2 \cdot \cdots \cdot m_N \tag{3.39}$$

Suppose that we have the same number of collocation points in each dimension, i.e., $m_1 = m_2 = \cdots = m_N \equiv m$, 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., $M_T = 3^{10} \sim 6\times10^4$, for $N = 10$). Thus, we need more efficient sampling schemes than the tensor-product grid to reduce the amount of computational effort for the multi-dimensional interpolation. The search for such sampling schemes has resulted in sparse grid methods of which the fundamentals will be 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 [26]. 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 as [28]

$$A_{q,N}(g) = \sum_{q-N+1\le|\mathbf{i}|\le q} (-1)^{q-|\mathbf{i}|} \cdot \binom{N-1}{q-|\mathbf{i}|} \cdot \left(U^{i_1} \otimes \cdots \otimes U^{i_N}\right)(g) \tag{3.40}$$

where $\mathbf{i} = (i_1, \cdots, i_N)$ is the multi-index, and $|\mathbf{i}| = i_1 + \cdots + i_N$. The above formula indicates that the Smolyak algorithm builds the multi-dimensional interpolation by considering one-dimensional functions of interpolation levels $i_1,\cdots,\ i_N$ under the constraint that the sum of these interpolation levels lies within the range $[q - N + 1, q]$. With the incremental interpolant, $\Delta^i = U^i - U^{i-1}$, $U^0 = 0$, the Smolyak formulas can be equivalently written as [28]

$$\begin{aligned} A_{q,N}(g) &= \sum_{|\mathbf{i}|\le q} \left(\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_N}\right)(g) \\ &= A_{q-1,N}(g) + \sum_{|\mathbf{i}|=q} \left(\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_N}\right)(g) \end{aligned} \tag{3.41}$$

The above formulas suggest that the Smolyak algorithm improves the interpolation by utilizing all 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}, $X_\Delta^{i+1} = X^{i+1} \backslash X^i$. 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

$$\begin{aligned} H_{q,N} &= \bigcup_{|\mathbf{i}|\le q} \left(X_\Delta^{i_1} \times \cdots \times X_\Delta^{i_N}\right) = H_{q-1,N} \cup \Delta H_{q,N} \\ \Delta H_{q,N} &= \bigcup_{|\mathbf{i}|=q} \left(X_\Delta^{i_1} \times \cdots \times X_\Delta^{i_N}\right) \end{aligned} \tag{3.42}$$

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 possibility of further reducing the number of function evaluations 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 $\mathbf{I}_{l,N} = \{\mathbf{i} \mid |\mathbf{i}| \leq l + N\}$ 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 33. An index set $\mathbf{I}$ is called admissible if for all $\mathbf{i} \in \mathbf{I}$,

$$\mathbf{i} - \mathbf{e}_k \in \mathbf{I} \text{ for } 1 \leq k \leq N,\ i_k > 1 \tag{3.43}$$

Here, $\mathbf{e}_k$ is the k^{th} unit vector. This admissibility condition still satisfies the telescopic property of the incremental interpolant $\Delta^i = U^i - U^{i\text{-}1}$. 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. (3.41) and (3.42). 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 an adaptive refinement, since possibly a larger error reduction can 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

$$\mathbf{I}_{\mathrm{F}}(\mathbf{i}) = \{\mathbf{i} + \mathbf{e}_k,\ 1 \leq k \leq N\} \tag{3.44}$$

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 have been refined are called old indices and grouped as an old index set $\mathbf{I}_\text{O}$. The overall index set $\mathbf{I}$ comprises of the active and old index sets: $\mathbf{I} = \mathbf{I}_\text{A} \bigcup \mathbf{I}_\text{O}$. For more details of the dimension-adaptive algorithm, readers are referred to References [33] and [34].

It is noted that, in the dimension-adaptive algorithm, the generalized sparse grid construction allows for an adaptive detection of the important dimensions and thus a more efficient refinement compared to the conventional sparse grid interpolation [24,25]. 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, and the dimension-adaptive algorithm may not be appropriate for this purpose.

<u>Hierarchical interpolation scheme using multivariate hierarchical basis functions</u>

For the dimension-adaptive interpolation, the hierarchical interpolation scheme provides a more convenient way for error estimation than the nodal interpolation scheme [34]. Here, we start with the derivation of 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 presented earlier, $\Delta^i = U^i - U^{i-1}$. Based on Eq. (3.34) and $U^{i-1}(g) = U^i(U^{i-1}(g))$, we can write [34]

$$\begin{aligned}\Delta^i(g) &= U^i(g) - U^i\left(U^{i-1}(g)\right) \\ &= \sum_{x_j^i \in X^i} a_j^i \cdot g\left(x_j^i\right) - \sum_{x_j^i \in X^i} a_j^i \cdot U^{i-1}(g)\left(x_j^i\right) \\ &= \sum_{x_j^i \in X^i} a_j^i \cdot \left(g\left(x_j^i\right) - U^{i-1}(g)\left(x_j^i\right)\right)\end{aligned} \tag{3.45}$$

Since for all $x_j^i \in X^{i-1}$, $g(x_j^i) - U^{i-1}(g)(x_j^i) = 0$, Eq. (3.45) can be rewritten as

$$\Delta^i(g) = \sum_{x_j^i \in X_\Delta^i} a_j^i \cdot \left(g\left(x_j^i\right) - U^{i-1}(g)\left(x_j^i\right) \right) \tag{3.46}$$

Since $X^i \subset X^{i+1}$, the number of grid points in $X_\Delta{}^i$ reads

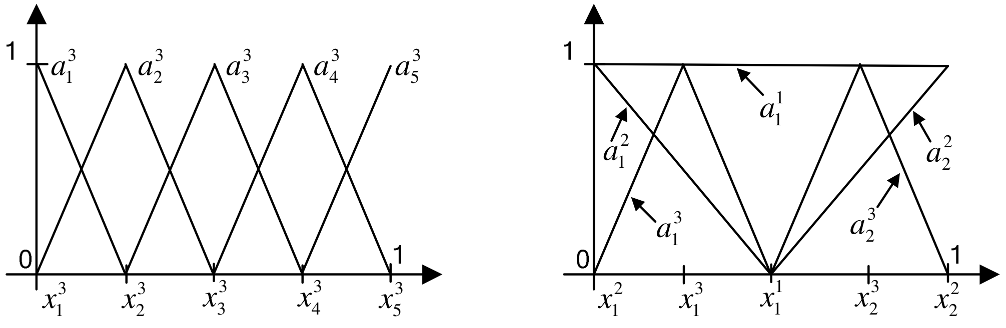
$$m_\Delta^i = m^i - m^{i-1} \tag{3.47}$$

By denoting the j^{th} element of $X_\Delta{}^i$ by x_j^i, Eq. (3.46) can be rewritten as

$$\Delta^i(g) = \sum_{j=1}^{m_\Delta^i} a_j^i \cdot \underbrace{\left(g\left(x_j^i\right) - U^{i-1}(g)\left(x_j^i\right) \right)}_{w_j^i} \tag{3.48}$$

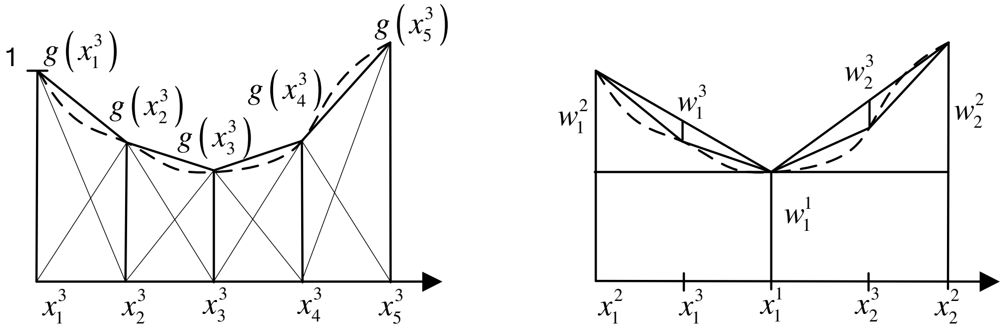
Here, w_j^i is defined as the hierarchical surplus, which indicates the interpolation error of a previous interpolation at the node x_j^i of the current interpolation level i. The bigger the hierarchical surpluses, the larger the interpolation errors. For smooth performance functions, the hierarchical 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 [34]. Figure 3-8 shows the comparison between the hierarchical and nodal basis functions with piecewise linear spline and Clenshaw-Curtis grid [34]. Figure 3-9 illustrates the comparison between the hierarchical and nodal interpolation. Based on the Smolyak formula in Eq. (3.41), a multivariate hierarchical interpolation formula can be obtained as [34]

$$\begin{aligned} A_{q,N}(g) &= A_{q-1,N}(g) + \Delta A_{q,N}(g) \\ &= A_{q-1,N}(g) \\ &\quad + \sum_{|\mathbf{i}|=q} \sum_{\mathbf{j}} \underbrace{\left(a_{j_1}^{i_1} \otimes \cdots \otimes a_{j_N}^{i_N} \right)}_{a_{\mathbf{j}}^{\mathbf{i}}} \cdot \underbrace{\left(g\left(x_{j_1}^{i_1}, \cdots, x_{j_N}^{i_N}\right) - A_{q-1,N}(g)\left(x_{j_1}^{i_1}, \cdots, x_{j_N}^{i_N}\right) \right)}_{w_{\mathbf{j}}^{\mathbf{i}}} \end{aligned} \tag{3.49}$$



(a) (b)

Figure 3-8 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.



(a) (b)

Figure 3-9 Nodal (a) and hierarchical (b) interpolations in 1D

3.2.2 Asymmetric Dimension-Adaptive Tensor-Product Method

As an attempt to enhance the adaptive feature of the dimension-adaptive algorithm, we, for the first time, introduce the concept of the directional sparse grid (DSG) which allows for the considerations of both directional and dimensional importance. Furthermore, a hierarchical interpolation scheme using cubic Lagrange splines is proposed for eliminating numerical inaccuracy of the high-order Lagrange interpolation as well as maintaining the smoothness property of the polynomial interpolation. The hierarchical ADATP interpolation and UQ and reliability analysis using the proposed ADATP method will also be presented in subsequent sections.

Directional Sparse Grid (DSG)

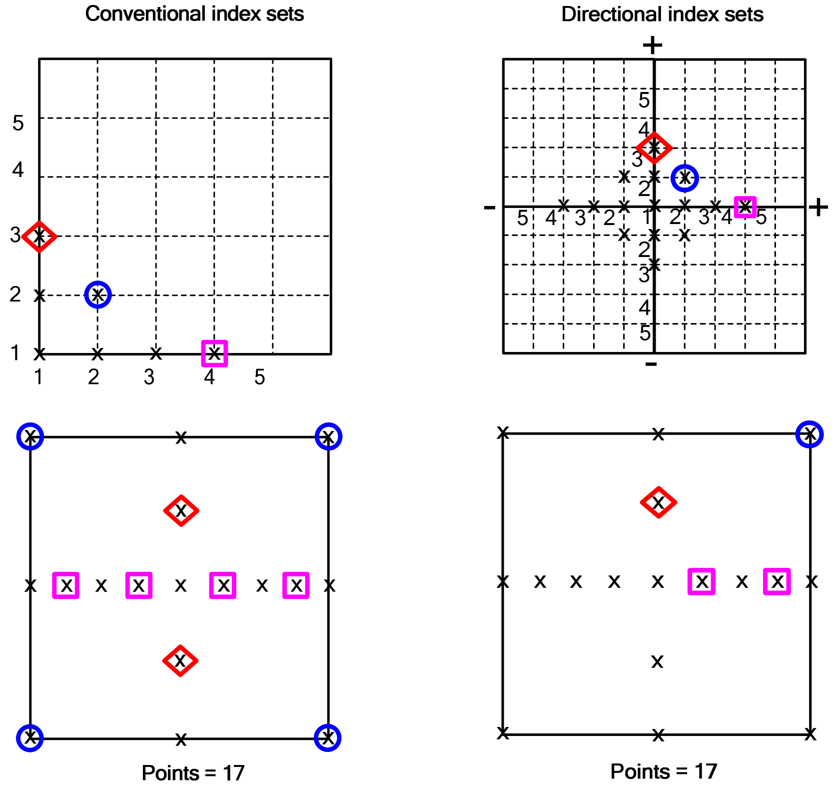
For the construction of the directional sparse grid (DSG), a conventional index i in the case of the univariate interpolation is decomposed into positive and negative directional index (DI) sets as

$$\mathbf{I}^{\mathrm{D}} = \left\{ i^{+},\ i^{-} \right\} \tag{3.50}$$

where the positive DI i^{+} corresponds to the DSG which belong to the index i and whose values are greater than the value (0.5) of the center grid point, and the negative DI i^{-} corresponds to the DSG which belong to the index i and whose values are smaller than 0.5. For the multivariate case ($N > 1$), we obtain a tensor-product formula of DI sets for a multi-index $\mathbf{i}$ as

$$\mathbf{I}^{\mathrm{D}} = \mathbf{I}_1^{\mathrm{D}} \times \cdots \times \mathbf{I}_N^{\mathrm{D}} \tag{3.51}$$

where, $\mathbf{I}_k^{\mathrm{D}} = \{i_k^{+},\ i_k^{-}\}$, $1 \leq \mathrm{k} \leq N$. Here, the forward neighborhood of a multi-dimensional DI $\mathbf{i}^{\mathrm{d}} \in \mathbf{I}^{\mathrm{D}}$ is defined as the N indices $\{\mathbf{i}^{\mathrm{d}} + \mathbf{e}_k^{+/-}\}$, $1 \leq \mathrm{k} \leq N$, and the sign of k^{th} directional unit vector $\mathbf{e}_k^{+/-}$ is the same with that of the k^{th} index element $i_{\mathrm{k}}^{\mathrm{d}}$ (i_{k}^{+} or i_{k}^{-}) of $\mathbf{i}^{\mathrm{d}}$. If $i_{\mathrm{k}}^{\mathrm{d}}$ is equal to 1, i.e., the corresponding collocation point is located at 0.5, both the positive and negative directional unit vectors are employed to obtain the forward neighborhood in that dimension. Figure 3-10 shows the conventional multi-index and the proposed DI for a 2D interpolation with the same set of the collocation points. From this figure, it is observed that the proposed DI divides the conventional index space into the four quadrants. In subsequent sections, it will be seen that this division allows for an adaptive refinement of the collocation points in these quadrants. In general, it is noted that the DI divides the conventional multi-index space into 2^N N-hyperoctants for more precise refinement.



(a) (b)

Figure 3-10 Conventional (a) and directional (b) index sets in 2D. Top row: index sets including (1, 3) and (1^+, 3^+) denoted by◇, (2, 2) and (2^+, 2^+) denoted by O, (4, 1) and (4^+, 1^+) denoted by □. Bottom row: corresponding collocation points.

Hierarchical interpolation scheme using cubic Lagrange splines

For the dimension-adaptive interpolation, the hierarchical interpolation scheme provides a more convenient way for the error estimation than the nodal interpolation scheme. In the case of singularities or discontinuities in the random space, the piecewise multi-linear basis function provides a strong local support for the adaptive algorithm. The detailed information regarding the selection of grid type and numerical scheme can be found in [34]. The Clenshaw-Curtis grid with equidistant nodes is recommended for piecewise multi-linear basis functions and is thus utilized in the ADATP method. In the case of a smooth function, the polynomial interpolation provides a faster error decay with increasing numbers of grid points than the piecewise multi-linear interpolation. However, the high-order Lagrange interpolation may give an inaccurate estimation of the performance function between collocation points due to severe oscillation, especially when the grid points are asymmetrically distributed with respect to the center point. To avoid this numerical inaccuracy and take advantage of the polynomial interpolation, a hierarchical interpolation scheme using cubic Lagrange splines is proposed for the ADATP method.

Choice of Sparse Grid Type

For the Lagrange interpolation, it is known that the Chebyshev-Gauss-Lobatto grid is a good choice due to its Chebyshev-based node distribution and its nesting characteristic [34]. However, this type of grid may not be appropriate for local adaptivity without a global support provided by Lagrange polynomials. In contrast, the Clenshaw-Curtis grid with equidistant nodes is more suitable for a local support provided by the cubic Lagrange spline function. In addition, it possesses the nesting characteristic. Thus, we propose to use the Clenshaw-Curtis grid as collocation points. In the case of a univariate interpolation ($N = 1$), the support nodes are defined in Eq. (3.36). As mentioned earlier, the resulting set of the points fulfill the nesting property $X^i \subset X^{i+1}$, and therefore $H_{q-1,N} \subset H_{q,N}$.

Univariate Nodal Basis Functions

The interpolation for smooth functions can be improved by replacing piecewise multi-linear basis functions by cubic Lagrange splines. The univariate nodal basis functions for cubic Lagrange splines can be expressed as [119]

$$a_j^i = 1 \text{ for } i = 1$$

$$a_j^i = \begin{cases} \prod\limits_{\substack{l=j-1 \\ l\neq j}}^{j+2} \dfrac{x - x_l^i}{x_j^i - x_l^i} & \text{if } x \in \left[x_j^i, x_{j+1}^i\right], \quad j = 2,\ldots,m_i - 2 \\ \prod\limits_{l=j+1}^{j+2} \dfrac{x - x_l^i}{x_j^i - x_l^i} & \text{if } x \in \left[x_j^i, x_{j+1}^i\right], \quad j = 1 \\ \prod\limits_{\substack{l=j-1 \\ l\neq j}}^{j+1} \dfrac{x - x_l^i}{x_j^i - x_l^i} & \text{if } x \in \left[x_j^i, x_{j+1}^i\right], \quad j = m_i - 1 \\ 0 & \text{otherwise} \end{cases} \tag{3.52}$$

for $i > 1$. In Eq. (3.52), the function value at endpoint x_j^i, $j = 1, m_i$, is not given. Then the polynomial on the interval $[x_2^i, x_3^i]$ is extended to the interval $[x_1^i, x_2^i]$ and the polynomial on the interval $[x_{m_i-2}^i, x_{m_i-1}^i]$ to the interval $[x_{m_i-1}^i, x_{m_i}^i]$. We observed that these extensions caused negligible sacrifice of the interpolation accuracy on the intervals $[x_1^i, x_2^i]$ and $[x_{m_i-1}^i, x_{m_i}^i]$.

Asymmetric Dimension-Adaptive Tensor-Product (ADATP) Interpolation

Based on the proposed concepts of the DI and DSG, the overall procedure of the ADATP interpolation is briefly summarized in Table 3-9. The relative error indicator used in the interpolation scheme can be defined for a DI **i** (see Figure 3-10b) as

$$\varepsilon_r(\mathbf{i}) = \frac{1}{(g_{\max} - g_{\min}) M_{\mathbf{i}}} \sum_{\mathbf{j}} \left| w_{\mathbf{j}}^{\mathbf{i}} \right| \tag{3.53}$$

where $w_{\mathbf{j}}^{\mathbf{i}}$ are the hierarchical surpluses of the collocation points $\mathbf{X}^{\mathbf{i}} = X_{\Delta}^{i1} \times \cdots \times X_{\Delta}^{iN}$, with $\mathbf{j} = (j_1,\ldots, j_N)$, $j_k = 1,\ldots, m_{\Delta}^{ik}$, $1 \le k \le N$, and $M_{\mathbf{i}} = m_{\Delta}^{i1} \cdot m_{\Delta}^{i2} \cdot \cdots \cdot m_{\Delta}^{iN}$. It is noted that, for simplicity, we use $\mathbf{i} = (i_1, \cdots, i_N)$ instead of $\mathbf{i}^{\mathrm{d}} = (i_1^{\mathrm{d}}, \cdots, i_N^{\mathrm{d}})$ to denote a multi-dimensional DI and that the term "index" in the description of the ADATP method refers to the DI. The pseudo code for the ADATP algorithm is given in Appendix B. Under the proposed scheme of asymmetric sampling, we expect that the error decay be at least as fast as that of the dimension-adaptive tensor-product interpolation.

Table 3-9 Procedure of the proposed ADATP interpolation

STEP 1	Set an initial interpolation level l $(q - N) = 0$; set the initial old index set $\mathbf{I}_{\mathrm{O}} = \emptyset$ and the initial active index set $\mathbf{I}_{\mathrm{A}} = \{\mathbf{i}\}$, where the initial active DI $\mathbf{i} = (1,\ldots,1)$ is the center point $(0.5,\ldots,0.5)$; set an initial relative error indicator $\varepsilon_r(\mathbf{i}) = 1$
STEP 2	Select a trial index set $\mathbf{I}_{\mathrm{T}}$ (from $\mathbf{I}_{\mathrm{A}}$) with the error indicator greater than a relative error threshold value ε_{C}; move the active index set $\mathbf{I}_{\mathrm{A}}$ to the old index set $\mathbf{I}_{\mathrm{O}}$. If $\mathbf{I}_{\mathrm{T}} = \emptyset$, go to STEP 7
STEP 3	Select and remove the trial index $\mathbf{i}_{\mathrm{t}}$ with the largest error indicator from $\mathbf{I}_{\mathrm{T}}$; 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 $\mathbf{I}_{\mathrm{F}}$ of $\mathbf{i}_{\mathrm{t}}$ and add $\mathbf{I}_{\mathrm{F}}$ to the active index set $\mathbf{I}_{\mathrm{A}}$
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

Uncertainty Quantification (UQ) and Reliability Analysis

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 hierarchical basis functions at collocation points. Thus, any probabilistic characteristics of $g(\mathbf{x})$, including statistical moments, reliability, and

PDF, can be easily estimated by performing MCS. For example, any r^{th} moment can be calculated as

$$\begin{aligned} \beta_r &\cong \int \hat{g}^r(\mathbf{x}) f(\mathbf{x}) d\mathbf{x} \\ &= E\left(\hat{g}^r(\mathbf{x})\right) = \lim_{ns\to\infty} \frac{1}{ns} \sum_{j=1}^{ns} \hat{g}^r\left(\mathbf{x}_j\right) \end{aligned} \tag{3.54}$$

where β_{r} is the r^{th} moment of the performance function $g(\mathbf{x})$; $f(\mathbf{x})$ is the joint PDFs; $\mathbf{x}_j$ is the j^{th} realization of $\mathbf{x}$; and ns is the sampling size. For reliability estimation, let us define an approximate safe domain for the performance function g as

$$\hat{\Omega}^S = \left\{\mathbf{x} : \hat{g}(\mathbf{x}) < 0\right\} \tag{3.55}$$

Therefore, the reliability R can also be estimated by performing MCS as

$$\begin{aligned} R &\cong \int I_{\hat{\Omega}^S}(\mathbf{x}) f(\mathbf{x}) d\mathbf{x} \\ &= E\left(I_{\hat{\Omega}^S}(\mathbf{x})\right) = \lim_{ns\to\infty} \frac{1}{ns} \sum_{j=1}^{ns} I_{\hat{\Omega}^S}\left(\mathbf{x}_j\right) \end{aligned} \tag{3.56}$$

where $I[\cdot]$ is an indicator function of safe or fail state such that

$$I_{\hat{\Omega}^S}\left(\mathbf{x}_j\right) = \begin{cases} 1, & \mathbf{x}_j \in \hat{\Omega}^S \\ 0, & \mathbf{x}_j \in \Omega \setminus \hat{\Omega}^S \end{cases} \tag{3.57}$$

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. We also note that the way of approximating the response function over the entire (truncated) input domain allows for the derivation of any probabilistic characteristics (e.g., statistical moments, reliability, and PDF) based on the same set of collocation points and can be used for design problems that require both moment estimation and reliability analysis such as reliability-based robust design optimization [43-45].

Remarks

Several important remarks regarding the properties of the ADATP method are listed as follows.

Remark 1 – Potential benefits: In the proposed method, the DI divides the conventional multi-index space into 2^N N-hyperoctants, thus enabling asymmetric refinements among these hyperoctants. Therefore, for a performance function with unequal degrees of nonlinearity in 2^N N-hyperoctants, the ADATP method is expected to outperform the DATP method in terms of efficiency.

Remark 2 – Determination of an interpolation domain: It is noted that when the random space Π is unbounded, e.g. in the case of Gaussian random variables, we need to truncate it to a bounded one Γ that achieves a nearly full coverage of the original random space Π. Without loss of generality, we consider the case of N-dimensional independent standard Gaussian random vector **x**, the probability that a realization of the original random space Π belongs to the truncated random space $\Gamma^* = [-\lambda, \lambda]^N$ can be expressed as

$$\begin{aligned} P\left(\Gamma^* \mid \Pi\right) &= \int I_{\Gamma^*}\left(\mathbf{x}\right) f\left(\mathbf{x}\right) d\mathbf{x} = \lim_{ns\to\infty} \frac{1}{ns} \sum_{j=1}^{ns} I_{\Gamma^*}\left(\mathbf{x}_j\right) \\ &= \left[\Phi(\lambda) - \Phi(-\lambda)\right]^N \end{aligned} \tag{3.58}$$

In the current study, we used $\lambda = 3.5$, which, for example, gives $P(\Gamma^* | \Pi) = 0.9954$ for $N = 10$. We note that the truncation has a negative effect on the accuracy of reliability analysis, especially for problems with small numbers of random variables (i.e., small N in Eq. (3.58)) and low probabilities of failure, and that the truncated interpolation domain for a specific problem should be carefully determined based on the understanding of this problem. More conservative criteria may guarantee higher accuracy but creates a larger interpolation domain that requires more computational effort. Since the goal in this work is to develop an asymmetric interpolation scheme for uncertainty qualification and reliability analysis, this paper does not address the issues regarding how a choice of the interpolation domain affects numerical accuracy and efficiency of reliability analysis and how to achieve an optimum λ for a given problem.

Remark 3 – Discretion on an error threshold: We also note that the relative error threshold ε_{C} greatly affects the convergence rate and accuracy of the asymmetric dimension-adaptive sampling. A larger ε_{C} leads to faster error decay but results in a lower level of interpolation accuracy. A zero threshold, as an extreme case, results in a conventional sparse grid construction. In the current study, we used $\varepsilon_{\mathrm{C}} = 0.10$ for most engineering cases. Under the hierarchical interpolation scheme, ε_{C} allows a user to put a preference between the convergence rate and accuracy.

3.2.3 Case Studies for ADATP

Three mathematical and engineering examples are given in this section to demonstrate the effectiveness of the ADATP method. The first mathematical example was designed to compare the performances of the ADATP and DATP methods for interpolation. The subsequent two mathematical and engineering examples were used for studying the performance of the proposed method for UQ and reliability analysis.

Mathematical Example I: Response Surface with Line Singularity

Consider a mathematical function

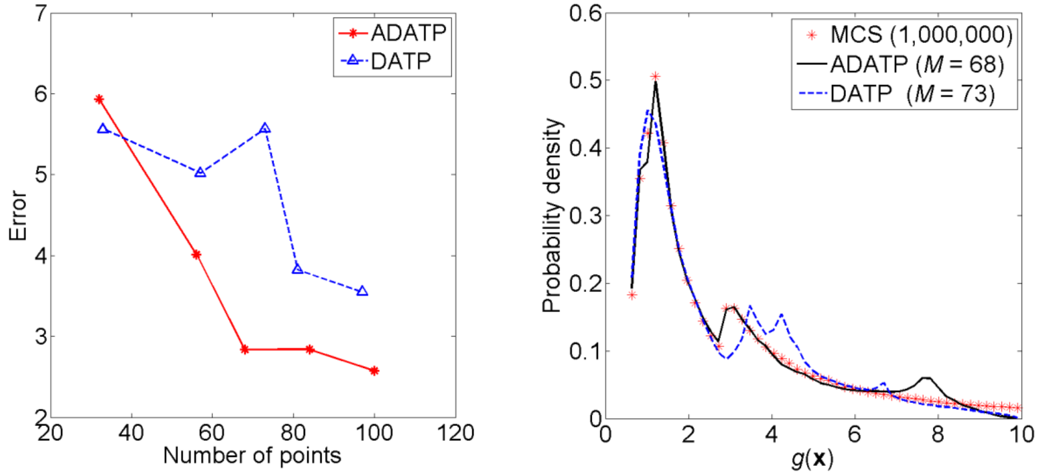
$$g(\mathbf{x}) = \frac{1}{\left|0.25 - x_1^2 - x_2^2\right| + 0.1} \tag{3.59}$$

where the two random variables were assumed to be statistically independent and uniformly distributed between 0 and 1. It is noted that, for notational convenience, we use the notation x_k instead of x^k to denote the k^{th} random variable for k = 1,…, N in this section. This modified function from [29] shows a line singularity in the third quadrant of the square $\Omega = [0, 1]^2$. We further defined the interpolation error ε_I as

$$\varepsilon_I = \max_{j=1,\ldots,ns} \left| g(\mathbf{x}_j) - \hat{g}(\mathbf{x}_j) \right| \tag{3.60}$$

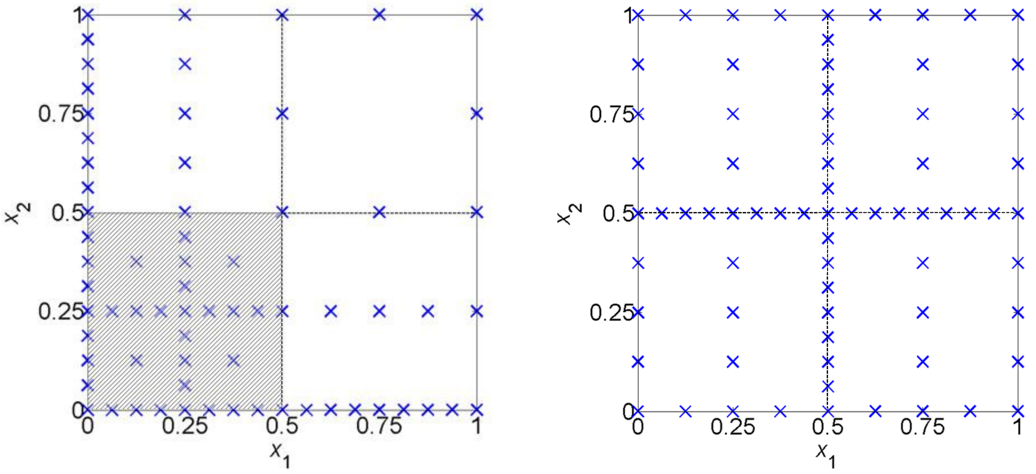
where ns denotes the number of Monte Carlo samples for interpolation and was set to 1,000,000 in this example. A relative error threshold $\varepsilon_C = 0.10$ was used in the ADATP method. Figure 3-11 illustrates the error decay and PDF approximations of the DATP and ADATP methods, both of which, for comparison purpose, employed the piecewise multi-linear basis functions e as the hierarchical basis functions and the Clenshaw-Curtis grid as the grid type. It should be noted that, since the ADATP and DATP methods employ different schemes for generating new collocation points, the numbers of collocation points achieved by both methods could be different. However, a meaningful comparison can still be carried out by observing a general trend of error decay. In Figure 3-11a, the ADATP method shows faster error decay and more accurate PDF approximation, compared to that of the DATP method. This is because the ADATP method identified high nonlinearity in the third quadrant and adaptively added collocation points to the quadrant region (see the shaded region in Figure 3-

12a) while the DATP method treated all quadrants as of equal importance and thus assigns points equally to all quadrants (see Figure 3-12b). This example verifies that, for a performance function with unequal degrees of nonlinearity in 2^N N-hyperoctants, the ADATP method is more efficient than the DATP method.



(a) (b)

Figure 3-11 Error decay (a) and PDF approximations (b) of the DATP and ADATP methods for example I



(a) (b)

Figure 3-12 Collocation points of the ADATP method ($M = 68$) (a) and the DATP method ($M = 73$) (b) for example I

Mathematical Example II: Trivariate Interaction

Consider a mathematical function

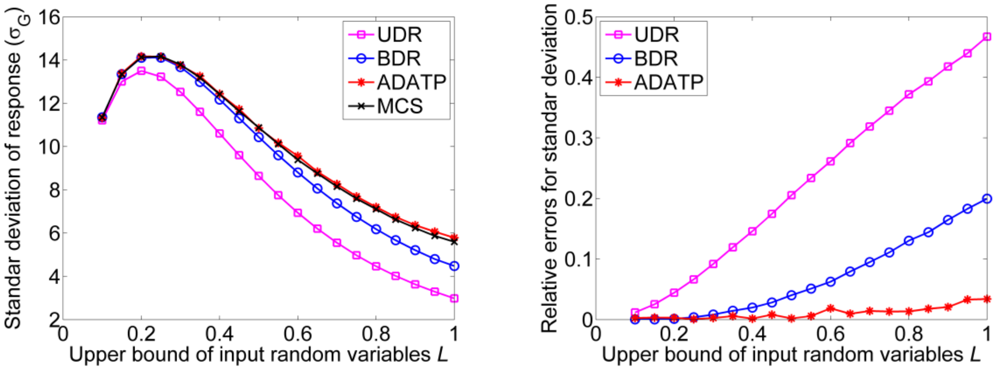
$$g(\mathbf{x}) = \exp\left[\sum_{k=1}^{N=5}(x_k - 1)^2 - \sum_{k=3}^{N=5} x_k^r x_{k-1}^r x_{k-2}^r\right] \tag{3.61}$$

where the five random variables were assumed to be statistically independent and uniformly distributed between 0 and L. The proposed ADATP method with $\varepsilon_C = 0.10$, $M_{\max} = 120$ and cubic Lagrange splines as the hierarchical basis functions was employed to compute the mean μ_G and standard deviation σ_g of $g(\mathbf{x})$. These two moments were calculated using the UDR and BDR integrations [9] based on a fully tensorized Gauss-Legendre quadrature technique [120] with the number of one-dimensional quadrature points $m_I = 5$. Two cases were considered: i) Case 1: increasing the trivariate order ($r = 0, 1, 2$, or 3; $L = 1.0$); ii) Case 2: increasing the uncertainty of input random variables (L increases from 0.1 to 1.0; $r = 2$). The results for Case 1 were summarized in Table 3-10. Both the BDR and ADATP methods provide good approximations of the mean μ_g, when compared with the results of MCS for a trivariate order up to 3. However, the UDR method can not accurately estimate μ_g for any trivariate order. Regarding the standard deviation σ_g, the ADATP method gives a consistently more accurate estimate, while both the UDR and BDR methods fail to give sufficiently accurate estimates. The results of σ_g for Case 2 are plotted in Figure 3-13. All three methods can give a good approximation when the uncertainty (controlled by L) of input random variables is small. However, as the uncertainty increases, the ADATP method becomes superior to the UDR and BDR methods.

Table 3-10 Moment estimations by the ADATP and dimension reduction methods (L = 1.0)

r	Mean (μ_g)				Standard deviation (σ_g)			
	UDR[a]	BDR[a]	ADATP[a]	MCS[b]	UDR	BDR	ADATP	MCS
0	0.2946	0.3283	0.3246	0.3336	0.1436	0.2119	0.2738	0.2691
1	4.7588	5.4681	5.3999	5.5650	2.9811	4.4782	5.7853	5.5953
2	5.6486	6.3037	6.2180	6.4028	2.9942	4.4296	5.6848	5.5264
3	5.8516	6.4914	6.4154	6.5855	2.9131	4.3343	5.6112	5.4657

[a] UDR and BDR required 21 and 181 function evaluations (FEs), respectively; ADATP required 121 FEs.
[b] MCS required 1,000,000 FEs at MC sample points.



(a) (b)

Figure 3-13 Estimates (a) and relative errors (b) of standard deviations for increasing values of L

This comparison with the UDR and BDR methods suggests that the ADATP method is better in terms of both accuracy and efficiency when the trivariate interaction is strong as in this problem. The ADATP method outperforms the UDR and BDR methods because of the following two reasons: (i) the UDR and BDR methods do not consider trivariate interactions; (ii) without an adaptive sampling scheme, the UDR and BDR methods may unnecessarily assign many uni- or bivariate sample points in regions with small nonlinearity. Even if we could resolve the first limitation by increasing S in the S-variate dimension reduction (DR) technique, the second limitation still remains unresolved. In fact, it is often difficult or impractical to predetermine *S* in the *S*-variate DR technique. In contrast, the ADATP method is capable of automatically detecting tri- and higher-variate interactions and generating corresponding collocation points to reproduce the interactions in the interpolation. Therefore, the ADATP method is distinctively advantageous from the *S*-variate dimension reduction (DR) technique.

Lower Control A-Arm: Nonlinear Fatigue Reliability

Vehicle suspension systems experience intense loading conditions throughout their service lives. Control arms act as the backbone of the suspension system, through which the majority of these loads are transmitted [118]. Therefore, it is crucial that the fatigue life of control arms be high enough to fulfill the design requirement. A HMMWV lower control-arm was used for fatigue reliability analysis using the ADATP method.

The lower control-arm was modeled with plane stress elements using 54,666 nodes, 53,589 elements, and 327,961 DOFs, where all welds were modeled using rigid beam elements. Hyper-Works 8.0 was used for finite element modeling and design parameterization. ANSYS 10.0 was used for stress analyses for 14 load cases at four joints for the A-arm: a ball joint, a spring-damper joint and front and rear pivot bushing joints, respectively. The stress contours for two loading cases are shown in Figure 3-14. The fe-safe 5.0 was employed for durability analysis based on the dynamic stress results from ANSYS. A preliminary durability analysis was executed in fe-safe to estimate the fatigue life of the HMMWV A-Arm and to predict the critical regions that experience a low fatigue life. For this preliminary durability analysis, the fatigue life for crack initiation was calculated using the equivalent von Mises stress-life approach at all surface nodes of the mechanical component (i.e., A-arm) in order to predict the critical regions. More accurate durability analysis was then carried using the strain-life method at the selected critical regions of the A-arm that experience short life spans.

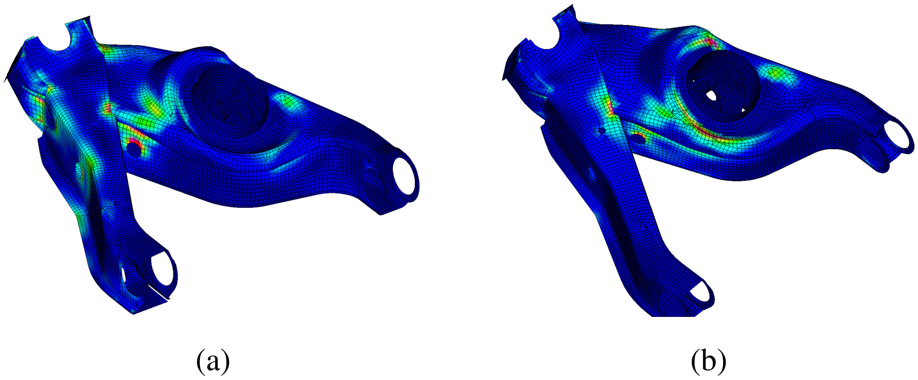


Figure 3-14 Stress contours for load case 2 (a) and load case 8 (b)

The random variables are the thicknesses of the eight major components of the control arm, as shown in Figure 3-15. The statistical information of these random variables is summarized in Table 3-11. From a worst-case scenario analysis, one hotspot with the smallest fatigue life was found at the rear pivot bushing joint and was selected for fatigue reliability analysis. In this case study, the fatigue reliability is defined as $R = \Pr(L > L^{t})$, where L^{t} denotes the target fatigue life.

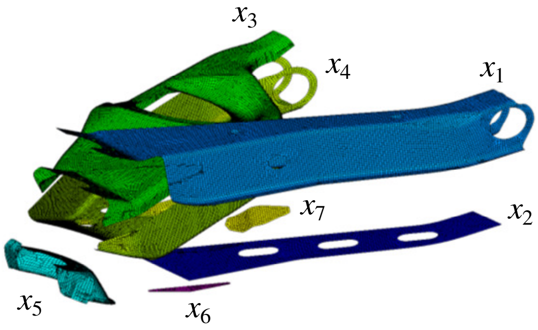


Figure 3-15 Seven thickness variables (x_8 not shown)

Table 3-11 Input random variables for the lower control A-arm example

Component	Distri. type	Mean (in)	Std. dev. (in)
x_1	Normal	0.157	0.006
x_2	Normal	0.183	0.006
x_3	Normal	0.178	0.009
x_4	Normal	0.200	0.007
x_5	Normal	0.312	0.013
x_6	Normal	0.250	0.009
x_7	Normal	0.200	0.007
x_8	Normal	0.201	0.009

The ADATP method with $\varepsilon_C = 0.10$, $M_{\max} = 80$ and cubic Lagrange splines was used to evaluate the fatigue reliability at the selected hotspot. The ADATP method allows for a stochastic response surface approximation from a small number of deterministic finite element and fatigue analyses, through constructing an explicit hierarchical interpolation formula with respect to the random inputs. Conducting the MCS on the explicit interpolation formula gives the full probability information (i.e., moments, reliability and PDF) of the fatigue life. A direct MCS with 1,000 samples was carried out as a reference. Figure 3-16 shows the PDF approximations by the ADATP method and MCS, where we can observe a good agreement between the two methods. Table 3-12 summarizes the uncertainty analysis results, where the ADATP method outperforms MCS in terms of efficiency while still maintaining good accuracy for moderate (between 0.70 and 0.80), high (between 0.90 and 0.95) and very high reliability levels (above 0.99). The 95% confidence intervals of the MCS reliability estimates include the corresponding ADATP estimates for all three reliability levels.

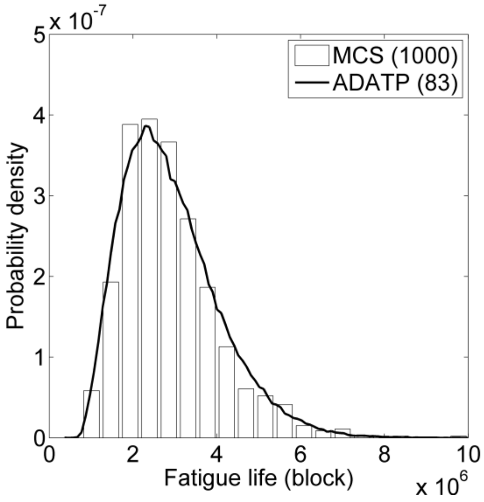


Figure 3-16 PDF approximations for the lower control A-arm example

Table 3-12 Uncertainty analysis results for the lower control A-arm example

	ADATP	MCS
Mean (blocks)	2.9014E+6	2.8866E+6
Std. dev. (blocks)	1.1501E+6	1.1612E+6
Skewness	8.9795E–1	1.2608E+0
Kurtosis	4.0504E+0	5.8083E+0
$R = \Pr(L > 2.0\times10^6)$	0.768	0.774 (±0.026[a])
$R = \Pr(L > 1.5\times10^6)$	0.920	0.930 (±0.016[a])
$R = \Pr(L > 1.0\times10^6)$	0.992	0.993 (±0.005[a])
No. FE	83	1,000

[a] Error bounds computed with a 95% confidence level

3.2.4 Summary

In the second part of this chapter, the asymmetric dimension-adaptive tensor-product (ADATP) method is proposed for efficient reliability analysis involving high nonlinearity. The ADATP method possesses three technical contributions: (i) an asymmetric dimension-adaptive sampling scheme considering both directional and dimensional importance, (ii) the concepts of the directional sparse grid (DSG) and directional index (DI) for the systematic generation of asymmetric collocation points, (iii) a hierarchical interpolation scheme using cubic Lagrange splines for eliminating the numerical inaccuracy of the high-order Lagrange interpolation.

It was found that the asymmetric dimension-adaptive sampling scheme and the hierarchical interpolation method showed better accuracy and efficiency than the DATP method in the case of unequal degrees of nonlinearity in 2^N N-hyperoctants. The better performance can be attributed to the fact that the ADATP method identifies the highly nonlinear hyperoctants and assigns more collocation points to these regions, while the DATP treats all the hyperoctants as of equal importance and thus assigns points equally.

A limited comparative study between the ADATP method and the widely used reliability analysis methods, including FORM and moment-based reliability (mostly DR) methods, was also conducted in this work. Our initial results suggest that the ADATP method achieves higher accuracy and comparable efficiency for problems with moderate dimensions. The higher accuracy can be attributed to the automatic detection and adaptive reproduction of significant variate interactions in structural system responses, including tri- and higher-variate interactions. We also expect that the ADATP method perform well for high dimensional engineering problems as exemplified in the lower control A-arm example discussed in this work. Relative to the DR and PCE methods, the ADATP method has the advantage of the complexity reduction in the algorithm controls, since the desired interpolation accuracy and resource constraints allow a user to easily define a relative error threshold and maximum number of collocation points. In contrast, it is often difficult or impractical to predetermine *S* in the *S*-variate DR technique, or the expansion order and the number of one-dimensional quadrature points in the PCE method. Furthermore, the proposed ADATP method can approximate a multi-modal PDF.

Chapter 4: System Reliability Analysis

Engineered systems generally consist of numerous failure mechanisms, and system reliability provides a statistical description of system success in the context of multiple failure mechanisms. As explained in the literature review, existing methods can only provide system reliability bounds and requires the probabilities of joint safety events that are often difficult to evaluate. Despite the technical advance by the CIM for series systems, one big challenge still remains as how to generalize this method so that it can be used for system reliability analysis and thus predictive resilience analysis regardless of system structures (series, parallel, and mixed system). To address this challenge, this research proposes the generalized CIM (GCIM) method which will be detailed in this chapter. First, a closed-form system reliability formula for a parallel system is derived through its transformation into a series system using the De Morgan's law. Second, a unified system reliability analysis framework is proposed for mixed systems by defining a new System Structure matrix (SS-matrix) and employing the Binary Decision Diagram (BDD) technique. The SS-matrix is used to present any system structure in a comprehensive matrix form. Then the BDD technique together with the SS-matrix automates the process to identify system's mutually exclusive path sets, of which each path set is a series system. As a result, system reliability with any system structure can be decomposed into the probabilities of the mutually exclusive path sets. The development of the GCIM enables system reliability analysis for series, parallel, and mixed systems in system resilience analysis and design.

4.1 System Reliability Analysis for Series Systems

Although the second-order system reliability bounds method or the linear programming (LP) bounds method can generally give fairly narrow system reliability bounds, they cannot provide a unique system reliability estimate. In contrast, the original CIM provides an explicit formula for series system reliability assessment.

Considering a structural serial system with m components, the probability of system failure can be expressed as

$$p_{fs} = P\left(\bigcup_{i=1}^{m} \bar{E}_i\right) \tag{4.1}$$

where p_{fs} represents the probability of system failure and $\bar{E}_i$ denotes the failure event of the i^{th} component. The simplest system reliability bounds are the so-called first-order bounds. Based on the well known Boolean bounds in Eq. (4.2), the first-order bounds of probability of system failure are given in Eq. (4.3).

$$\max_i \left(P\left(\bar{E}_i\right)\right) \le P\left(\bigcup_{i=1}^{m} \bar{E}_i\right) \le \sum_{i=1}^{m} P\left(\bar{E}_i\right) \tag{4.2}$$

$$\max\left[P\left(\bar{E}_i\right)\right] \le p_{fs} \le \min\left[\sum_{i=1}^{m} P\left(\bar{E}_i\right),\ 1\right] \tag{4.3}$$

The lower bound in Eq. (4.3) is obtained by assuming the component events are perfectly independent and the upper bound is derived by assuming the component events are mutually exclusive. Despite the simplicity (only component reliability analysis required), the first-order bounds method provides very wide bounds of system reliability that are not practically useful. Thus, the second-order bounds method was proposed by Ditlevsen [41] in Eq. (4.4) to give much narrower bounds of probability of system failure.

$$P\left(\bar{E}_1\right)+\sum_{i=2}^{m}\max\left\{\left[P\left(\bar{E}_i\right)-\sum_{j=1}^{i-1}P\left(\bar{E}_i\bar{E}_j\right)\right],\ 0\right\}\le p_{fs}$$
$$\le\min\left\{\left[\sum_{i=1}^{m}P\left(\bar{E}_i\right)-\sum_{i=2}^{m}\max_{j<i}P\left(\bar{E}_i\bar{E}_j\right)\right],\ 1\right\}\tag{4.4}$$

where E_1 is the event having the largest probability of failure.

Since the probabilities of all events are non-negative, the following inequalities must be satisfied as

$$\max_i\left(P\left(\bar{E}_i\right)\right)\le\sqrt{\sum_{i=1}^{m}\left[P\left(\bar{E}_i\right)\right]^2}\le\sum_{i=1}^{m}P\left(\bar{E}_i\right)\tag{4.5}$$

Based on Eqs. (4.4) and (4.5), the probability of system failure (p_{fs}) of a series system failure can be simplified to a unique explicit formula as

$$p_{fs}\cong P\left(\bar{E}_1\right)+\sum_{i=2}^{m}\left\langle P\left(\bar{E}_i\right)-\sqrt{\sum_{j=1}^{i-1}\left[P\left(\bar{E}_i\bar{E}_j\right)\right]^2}\right\rangle\tag{4.6}$$

It can be proved that this approximate probability lies in the second-order bounds in Eq. (4.4). Based on Eq. (4.6), serial system reliability can be assessed as (1 − the probability of system failure) and formulated as

$$R_{sys}=P(E_1E_2\cdots E_{m-1}E_m)\cong P\left(E_1\right)-\sum_{i=2}^{m}\left\langle P\left(\bar{E}_i\right)-\sqrt{\sum_{j=1}^{i-1}\left[P\left(\bar{E}_i\bar{E}_j\right)\right]^2}\right\rangle$$
$$\text{where }\left\langle A\right\rangle\equiv\begin{cases}A, & \text{if } A>0\\ 0, & \text{if } A\le 0\end{cases}\tag{4.7}$$

Note that the terms inside the bracket, ⟨•⟩, should be ignored if it is less than zero and R_{sys} should be set to zero if the approximated one given by Eq. (4.7) is less than zero. It is noted that Eq. (4.7) provides an explicit and unique formula for system reliability assessment based on the second-order reliability bounds shown in Eq. (4.4) and an inequality Eq. (4.5).

4.2 System Reliability Analysis for Parallel Systems

A parallel system reliability formula can be obtained based on the formula of series system reliability by using the De Morgan's law. According to the De Morgan's law, the probability of parallel system failure can be expressed as

$$P\left(\bigcap_{i=1}^{m}\bar{E}_i\right)=1-P\left(\overline{\bigcap_{i=1}^{m}\bar{E}_i}\right)=1-P\left(\bigcup_{i=1}^{m}E_i\right) \tag{4.8}$$

where $\bar{E}_i$ is the i^{th} component failure event. It is noted that the above equation relates the probability of parallel system failure with the probability of series system safety (reliability). If we treat E_{i} as the i^{th} component failure event in a series system, the right side of Eq. (4.8) is then the series system reliability. Based on this relationship, the probability of parallel system failure can be obtained from Eq. (4.7) by treating the safe events in the series system as the failure events in the parallel system as

$$p_{fs}\cong P(\bar{E}_1)-\sum_{i=2}^{m}\left\langle P(E_i)-\sqrt{\sum_{j=1}^{i-1}\left[P\left(E_iE_j\right)\right]^2}\right\rangle,\quad \langle A\rangle\equiv\begin{cases}A, & \text{if } A>0\\ 0, & \text{if } A\le 0\end{cases} \tag{4.9}$$

Finally, parallel system reliability can be obtained from Eq. (4.9) by one minus the probability of system failure as

$$R_{sys}\cong P(E_1)+\sum_{i=2}^{m}\left\langle P(E_i)-\sqrt{\sum_{j=1}^{i-1}\left[P\left(E_iE_j\right)\right]^2}\right\rangle,\quad \langle A\rangle\equiv\begin{cases}A, & \text{if } A>0\\ 0, & \text{if } A\le 0\end{cases} \tag{4.10}$$

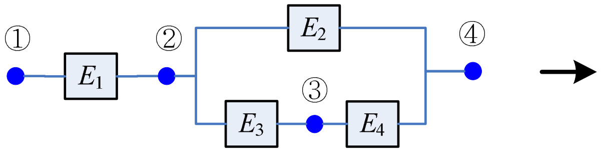
4.3 Mixed System Reliability Analysis

A mixed system may have various system structures. There is no unique system reliability formula available for a mixed system. This study develops a generic procedure for mixed system reliability analysis with an aim to produce a unique system reliability formula. The developed procedure is introduced below with an arbitrary mixed system structure. Considering a mixed system with *N* components, the following procedure can be proceeded to carry out system reliability analysis.

Step I: Constructing a System Structure Matrix

An SS-matrix, a 3-by-*M*, is proposed in this study to characterize any system structural configuration (components and their connections) in a matrix form. The SS-matrix contains the information about the constituting components and their connection. The first row of the matrix contains component numbers, while the second and third rows correspond to the starting and end nodes of the components. Generally, the total number of columns of a SS-matrix, *M*, is equal to the total number of system components, *N*. In the case of complicated system structures, one component may repeatedly appear in between different sets of nodes and, consequently, *M* could be larger than *N*, for example a 2-out-of-3 system.

Let us consider an example of a mixed system consisting of 4 components, as shown in Figure 4-1. The SS-matrix for the system can be constructed as a 3×4 matrix, as shown in Figure 4-1. The first column of the system structure matrix, $[1, 1, 2]^T$, indicates that the 1st component connects nodes 1 and 2.



$$\text{SS-matrix} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 2 & 3 \\ 2 & 4 & 3 & 4 \end{bmatrix}$$

Figure 4-1 Example to show the conversion of a system block diagram to SS-matrix

Step II: Finding Mutually Exclusive System Path Sets

Based on the SS-matrix, the Binary Decision Diagram (BDD) technique [121,122] can be employed to find the mutually exclusive system path sets, of which each path set is a series system. In probability theory, two events are said to be mutually exclusive if they cannot occur at the same time or, in other words, the occurrence of any one of them automatically implies the non-occurrence of the other. Here, system path sets are said to be mutually exclusive if any two of them are mutually exclusive. We note that, without the SS-matrix, it is not easy for the BDD technique to automate the process to identify the mutually exclusive path sets. The mixed system shown in Figure 4-1 can be decomposed into the two mutually exclusive path sets using the BDD, which is shown in Figure 4-2. Although the path sets $E_1\overline{E}_2E_3E_4$ and $E_1E_3E_4$ represent the same path that go through from the left terminal 1 to the right terminal 0 in Figure 4-1, the former belongs to the mutually exclusive path sets in Figure 4-2 while the latter does not. This is due to the fact that the path sets $E_1E_3E_4$ and E_1E_2 are not mutually exclusive. We also note however that we could still construct another group of mutually exclusively path sets, $\{E_1E_3E_4$, $E_1E_2\overline{E}_3\}$, which contains the path set $E_1E_3E_4$ as a member. This is due to the fact that a mixed system may have multiple BDDs with different configurations depending the ordering of nodes in BDDs and these BDDs results in several groups of mutually exclusive path sets, among which the one with the smallest number of path sets is desirable. Another point deserving of notice is that the mixed system considered here consists of only two mutually exclusive path sets. In cases of more than two mutually exclusive path sets, any two path sets are mutually exclusive. This suggests that the system path sets can equivalently be said to be pairwise mutually exclusive.

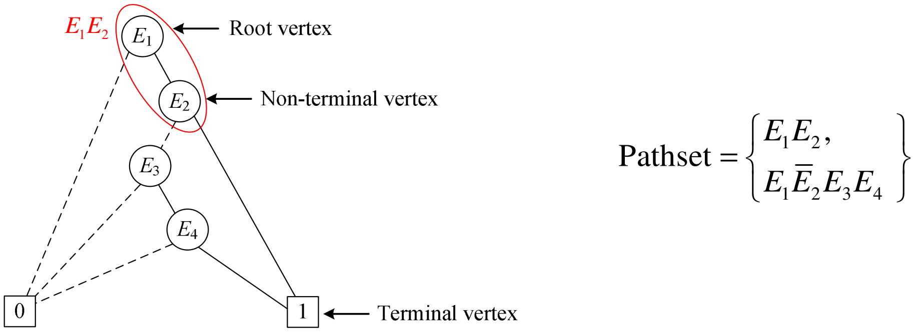


Figure 4-2 BDD diagram and the mutually exclusive path sets

Step III: Evaluating All Mutually Exclusive Path Sets and System Reliability

Due to the property of the mutual exclusiveness, the mixed system reliability, R_{sys}, is the sum of the probabilities of all paths as

$$R_{sys} = P\left(\bigcup_{i=1}^{N_p} \text{Path}_i \right) = \sum_{i=1}^{N_p} P\left(\text{Path}_i\right) \tag{4.11}$$

where Path_i is the i^{th} mutually exclusive path set obtained by the BDD and N_p is the total number of mutually exclusive path sets. For the system in Figure 4-1, the system reliability can be calculated as

$$\begin{aligned} R_{sys} &= P\left(\bigcup_{i=1}^{2} \text{Path}_i \right) = \sum_{i=1}^{2} P\left(\text{Path}_i\right) \\ &= P\left(E_1E_2\right) + P\left(E_1\bar{E}_2E_3E_4\right) \end{aligned} \tag{4.12}$$

where the probability of each individual path set can be calculated using the series system reliability formula given by Eq. (4.7).

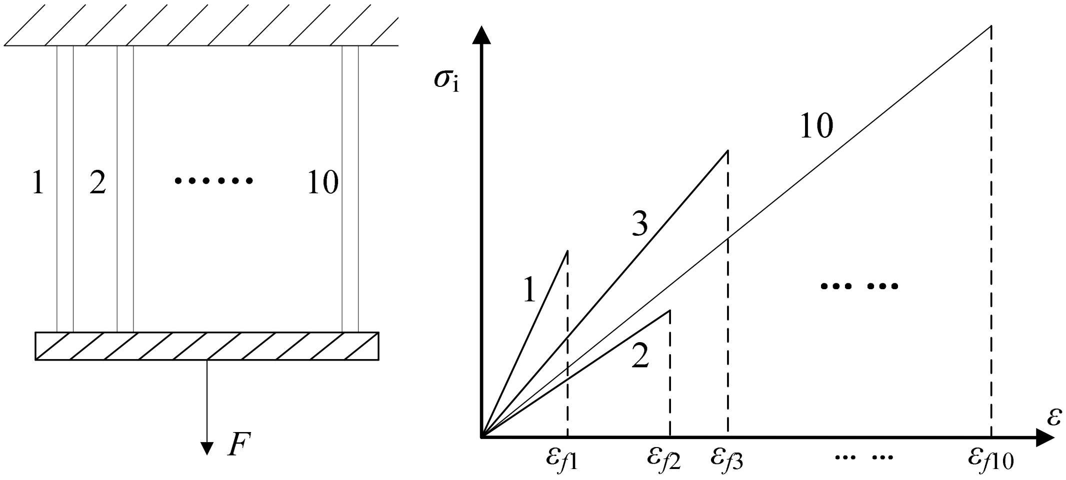
4.4 Case Studies for GCIM

This section presents two case studies for a parallel system and a mixed system, respectively, to demonstrate the effectiveness of the proposed GCIM for system reliability. For each case study, the generalized CIM framework is demonstrated in a wide range of system reliability levels and compared with MCS. For the parallel system, the results of the generalized CIM are also compared with the first-order bounds (FOB) and second-order bounds (SOB) methods. The main objective of the case studies is to demonstrate the theoretical accuracy of the proposed GCIM for system reliability analysis. So in the case studies we focus on a mathematical error produced by a system reliability formulae rather than a numerical error by a numerical method. In order to eliminate the numerical error in system reliability analysis, MCS with a large sample size was used to evaluate the probabilities of component and second-order joint safety events as shown in each case study.

4.4.1 Parallel System Example: A Ten Brittle Bar System

The following ten-bar system example is used to demonstrate the effectiveness of the GCIM framework for parallel systems. As shown in Figure 4-3, ten brittle bars are connected in parallel to sustain a load applied at one end. This case study is modified from the example employed in Ref. [123]. Ten bars are all brittle with different fracture strain limits ε_{fi}, $1 \le i \le 10$, which are sorted in an ascending order. If the exerted strain ε is between the $(i-1)^{\text{th}}$ and i^{th} fracture strain limits, i.e., $\varepsilon_{f(i-1)} \le \varepsilon < \varepsilon_{fi}$, , bar components with fracture strains below ε_{fi} will fail, and the allowable load is then the sum of the strength of components with fracture strains equal to or above ε_{fi}. Therefore, the strain level corresponding to the overall maximum allowable load is among the ten fracture strain limits. As the overall maximum allowable load, the system strength R_T can be formulated in Eq. (4.13).

$$\begin{aligned} R_T &= \max_{\varepsilon}\left\{\sum_{j=1}^{10} R_j(\varepsilon)\right\} = \max_{1\le i\le 10}\left\{\sum_{j=i}^{10} R_j(\varepsilon_{fi})\right\} \\ &= \max\left\{\sum_{j=1}^{10} R_j(\varepsilon_{f1}),\quad \sum_{j=2}^{10} R_j(\varepsilon_{f2}),\quad \cdots,\quad R_{10}(\varepsilon_{f10})\right\} \end{aligned} \tag{4.13}$$



(a) (b)

Figure 4-3 Ten brittle bar parallel system: (a) system structure model; (b) brittle material behavior in a parallel system

For example, if the exerted strain ε is equal to the fracture strain ε_{f2}, the 1st brittle bar fails due to the fracture and no longer contributes to the overall system strength. Thus, the system strength R_T at this fracture strain is the sum of strength of the other nine brittle bars. The brittle bar system fails to sustain the load F only if the system strength at any of the ten fracture strains is smaller than the load F. This is a parallel system with ten components, corresponding to the ten fracture strains. The component safety events can be expressed in terms of several random variables.

$$G_i = F - \sum_{j=i}^{10} R_j(\varepsilon_{fi}) = F - \sum_{j=i}^{10} (E_j A_j)\cdot \varepsilon_{fi}, \qquad 1 \le i \le 10 \tag{4.14}$$

where R_j represents the allowable load that can be sustained by the j^{th} brittle bar, A_j the cross section area of the j^{th} brittle bar, and E_j the Young's modulus of the j^{th} brittle bar. Random variables and their random properties are summarized in Table 4-1.

Table 4-1 Statistical information of input random variables for the ten bar system

Random variable	Mean	Standard deviation	Distribution type
E_1-E_{10} (GPa)	200.0	10.0	Gumbel
A_1 (mm^2)	100.0	5.0	Lognormal
A_2 (mm^2)	120.0	5.0	Lognormal
A_3 (mm^2)	140.0	5.0	Lognormal
A_4 (mm^2)	140.0	10.0	Lognormal
A_5 (mm^2)	140.0	10.0	Lognormal
A_6 (mm^2)	150.0	10.0	Lognormal
A_7 (mm^2)	150.0	15.0	Lognormal
A_8 (mm^2)	150.0	15.0	Lognormal
A_9 (mm^2)	200.0	15.0	Lognormal
A_{10} (mm^2)	300.0	25.0	Lognormal
ε_{f1}	0.0010	0.0002	Uniform
ε_{f2}	0.0012	0.0003	Uniform
ε_{f3}	0.0018	0.0004	Uniform
ε_{f4}	0.0025	0.0005	Uniform
ε_{f5}	0.0027	0.0006	Uniform
ε_{f6}	0.0030	0.0007	Uniform
ε_{f7}	0.0033	0.0008	Uniform
ε_{f8}	0.0036	0.0009	Uniform
ε_{f9}	0.0040	0.0010	Uniform
ε_{f10}	0.0050	0.0011	Uniform
F (kN)	---	30.0	Normal

Ten different system reliability levels are used for comparison with ten different loading conditions (F). These loading points are used to validate the GCIM method at different reliability levels. Table 4-2 summarizes the results of system reliability analyses which are illustrated in Figure 4-4. It can be seen that the first-order bounds are too wide to be of practical use. Whereas, the second-order bounds method gives tighter system reliability bounds compared with the first-order bounds method. It is expected based on the results that the GCIM method can produce accurate system reliability estimates at a wide variety of reliability levels and that this high accuracy can be maintained at high reliability levels, which are often encountered in engineering practices. Similar to the first case study, only the first- and second-order CI events were considered and the error for the GCIM comes from the effects of the third- or higher-order CI events. However, for a parallel system these effects tend to decrease as the system reliability decreases, thus the error at a low system reliability level is smaller than that at a higher system reliability level, as observed from Figure 4-4.

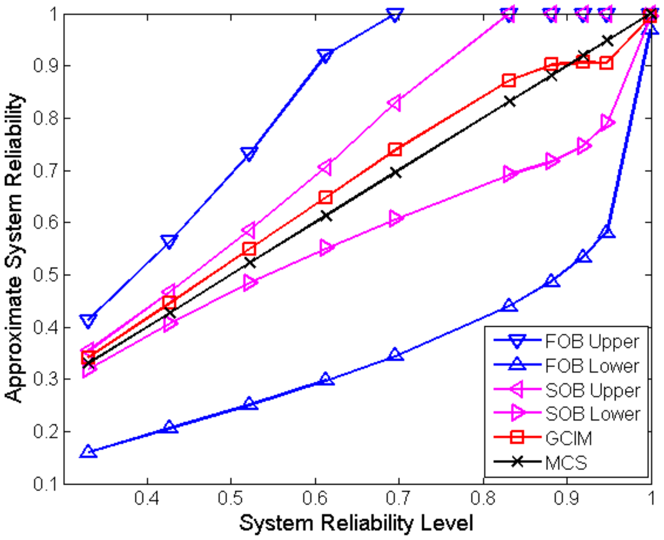


Figure 4-4 Results of system reliability analysis at ten different reliability levels

Table 4-2 Results of system reliability analysis with MCS, FOB using MCS, SOB using MCS, and GCIM using MCS (*ns* = 1,000,000)

Analysis Method		System Reliability Level at Each Design									
		1	2	3	4	5	6	7	8	9	10
FOB	Upper	0.4133	0.5639	0.7331	0.9216	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	Lower	0.1594	0.2054	0.2507	0.2974	0.3444	0.4395	0.4865	0.5334	0.5803	0.9705
SOB	Upper	0.3537	0.4670	0.5854	0.7065	0.8293	1.0000	1.0000	1.0000	1.0000	1.0000
	Lower	0.3192	0.4062	0.4849	0.5507	0.6068	0.6917	0.7161	0.7459	0.7897	0.9943
GCIM		0.3417	0.4456	0.5490	0.6482	0.7388	0.8714	0.9017	0.9069	0.9051	0.9943
MCS		0.3301	0.4272	0.5226	0.6131	0.6961	0.8314	0.8813	0.9192	0.9476	0.9998
GCIM Error		0.0116	0.0184	0.0264	0.0351	0.0427	0.0400	0.0204	0.0123	0.0425	0.0055

4.4.2 Mixed System Example: A Power Transformer Joint System

Power transformers are among the most expensive elements of high-voltage power systems [124]. The power transformer vibration induced by the magnetic field loading will cause the windings support joint loosening or the fatigue failures, which will gradually increase the vibration amplitude of the winding and eventually damage the core [125]. In this case study the proposed GCIM has been applied for the system reliability analysis of the power transformer winding support joints. We considered four failure modes, which are the fatigue failures at the four winding support joints. A power transformer simulation model was built using the finite element analysis tool ANSYS 10 (see Figure 4-5). Figure 4-6 shows the detail of the winding bolt joint, which assembles the windings of the power transformer with the bottom fixture. The transformer is fixed at the bottom and the vibration load is applied to the magnetic core with the frequency of 120 Hz. This case study employed ten random variables, as listed in Table 4-3, which include the geometric tolerances and material properties.

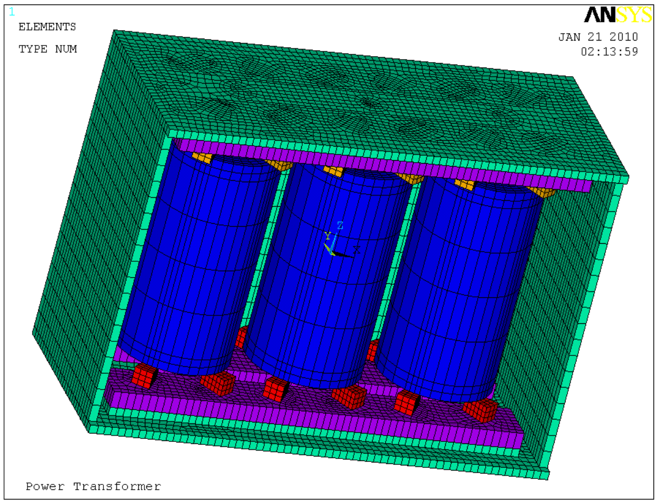
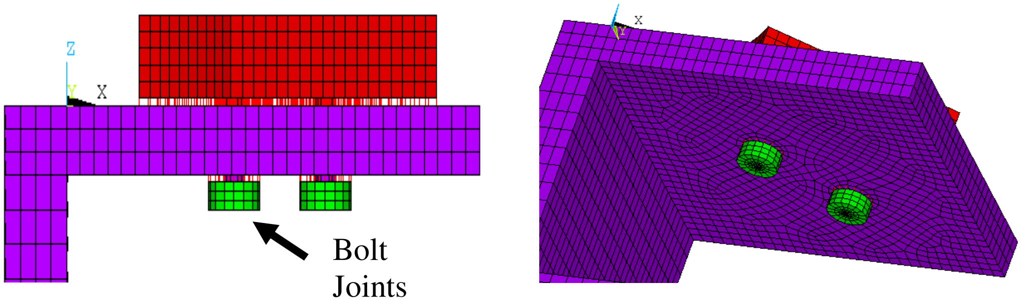


Figure 4-5 A power transformer FE model (without the covering wall)



(a) (b)

Figure 4-6 Winding support bolt joint: (a) side view, (b) bottom view

Table 4-3 Random property of input variables for the power transformer example

Random variable	Physical meaning	Mean	Standard deviation	Distribution type
X_1	Wall Thickness	3	0.06	Normal
X_2	Angular width of support joints	15	0.3	Normal
X_3	Height of support joints	6	0.12	Normal
X_4	Young's modulus of support joint	2e12	4e10	Normal
X_5	Young's modulus of loosening joints	2e10	4e8	Normal
X_6	Young's modulus of winding	1.28e12	3e10	Normal
X_7	Poisson ratio of joints	0.27	0.0054	Normal
X_8	Poisson ratio of winding	0.34	0.0068	Normal
X_9	Density of joints	7.85	0.157	Normal
X_{10}	Density of windings	8.96	0.179	Normal

This winding support system with the four joints was treated as a 3-out-of-4 system as shown in Figure 4-7, which means that the system becomes safe only if at least three out of the four support joints survive. The CI-matrix for this case study was evaluated using the MCS (with 1000 samples), as shown in Figure 4-8. Figure 4-9 shows the system reliability block diagram and Table 4-4 displays the SS-matrix for this transformer joint system. The mutually exclusive path sets can be determined using the BBD (see Figure 4-10) as

$$\text{Pathset} = \left\{ E_1E_2E_3,\ \bar{E}_1E_2E_3E_4,\ E_1\bar{E}_2E_3E_4,\ E_1E_2\bar{E}_3E_4 \right\}$$

These path sets are mutually exclusive with the series system structure. As shown in Table 4-5, the reliabilities for these mutually exclusive path sets can be obtained and the system reliability for this transformer joint system can be estimated using Eq. (4.11). Based on the results, the GCIM is expected to accurately assess system reliabilities of large-scale engineered systems. This case study demonstrates the feasibility and capability of the GCIM for system reliability analysis with any system structure.

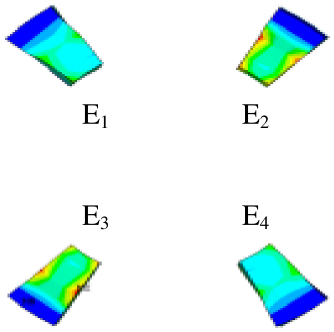


Figure 4-7 3 out of 4 system with support joints

$$\text{CI-matrix} = \begin{bmatrix} 0.999 & 0.000 & 0.238 & 0.242 \\ 0.000 & 0.999 & 0.238 & 0.242 \\ 0.000 & 0.000 & 0.761 & 0.008 \\ 0.000 & 0.000 & 0.000 & 0.757 \end{bmatrix}$$

Figure 4-8 CI-matrix for the power transformer example

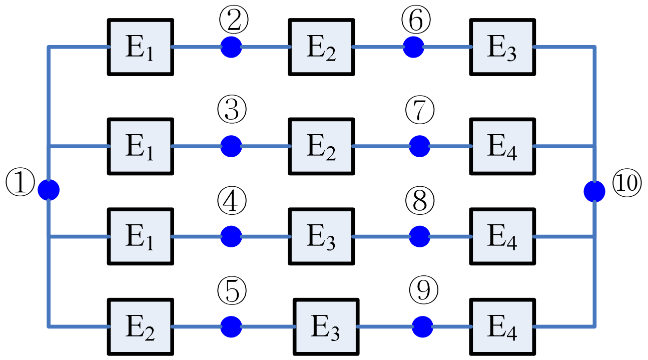


Figure 4-9 System reliability block diagram for the power transformer example

Table 4-4 System Structure matrix for the power transformer case study

Component No.	1	1	1	2	2	2	3	3	3	4	4
Starting node	1	1	1	1	2	3	4	5	6	7	8
End node	2	3	4	5	6	7	8	9	10	10	10

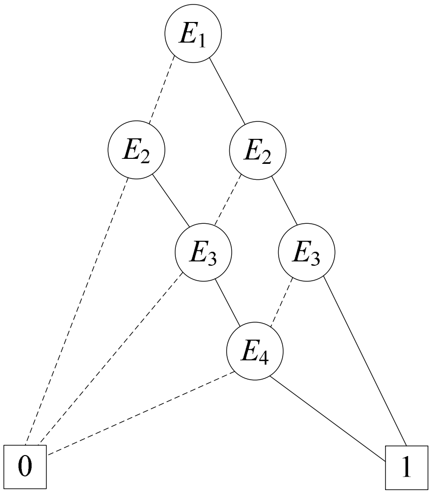


Figure 4-10 BDD diagram for the power transformer example

Table 4-5 Results of GCIM for power transformer case study comparing with MCS (ns = 10,000)

Analysis Method	Reliability of Path Set (Series System)				System Reliability
	$E_1E_2E_3$	$\bar{E}_1E_2E_3E_4$	$E_1\bar{E}_2E_3E_4$	$E_1E_2\bar{E}_3E_4$	
GCIM	0.761	0.000	0.000	0.002	0.763
MCS	**0.7611**	**0.0018**	**0.0000**	**0.0000**	**0.7629**

4.5 Summary

This chapter presents the generalized complementary intersection method (GCIM) for system reliability analysis. The GCIM generalize the original CIM so that it can be used for system reliability analysis regardless of system structures (series, parallel, and mixed system). This generalization leverages two ideas: (i) transforming a parallel system to the equivalent series system using the De Morgan's law to derive a closed-form system reliability formula; (ii) defining a new System Structure matrix (SS-matrix) and employing the Binary Decision Diagram (BDD) technique to develop a unified system reliability analysis framework for mixed systems. The two case studies (with one for parallel system, and the other for mixed system) were used to demonstrate that the proposed GCIM can assess system reliability regardless of the system structures. As observed through the case studies, the GCIM offers the generic system reliability analysis framework and thus shows a great potential to enhance our capability and understanding of system reliability analysis for system resilience analysis and design.

Chapter 5: System Prognostics and Health Management (PHM)

As mentioned in the literature review, tremendous research efforts have been devoted to the physics-based reliability analysis under uncertainties during the design stage of engineered systems. Recently, research on real-time diagnosis and prognosis which interprets data acquired by distributed sensor networks, and utilizes these data streams in making critical decisions provides significant advancements across a wide range of applications. Maintenance and life-cycle management is one of the beneficiary application areas because of the pervasive nature of design and maintenance activities throughout the manufacturing and service sectors. Maintenance and life-cycle management activities constitute a large portion of overall costs in many industries [48]. These costs are likely to increase due to the rising competition in today's global economy. For instance, in the manufacturing and service sectors, unexpected breakdowns can be prohibitively expensive since they immediately result in lost production, failed shipping schedules, no operational service, repair cost, and poor customer satisfaction. In order to reduce and possibly eliminate such problems, it is necessary to accurately assess current system health condition and precisely predict the remaining useful life (RUL) of operating components, subsystems, and systems. In general, a system' health condition can be predicted in (a) a single time-scale, or (b) multiple time-scales. In the former case, all the system health-relevant responses (e.g., health condition, measurable physical quantities) of interest tend to vary in the same time-scale. In the latter case, the system exhibits a time-scale separation. In other words, certain system health-relevant information (e.g., health condition) of interest tends to vary very slowly as opposed to other system health-relevant information (e.g., measurable physical quantities). This section presents an ensemble data-driven prognostic approach and a co-training data-driven prognostic approach to resolve the existing challenges in PHM for systems with a single time-scale, and introduces a multiscale framework with extended Kalman filter (EKF) for systems with multiple time-scales.

5.1 Ensemble of Data-Driven Prognostic Algorithms

Traditionally, a data-driven prognostic approach is to construct multiple candidate algorithms using a training data set, evaluate their respective performance using a testing data set, and select the one with the best performance while discarding all the others. This approach has three shortcomings: (i) the selected standalone algorithm may not be robust, i.e., it may be less accurate when the real data acquired after the deployment differs from the testing data; (ii) it wastes the resources for constructing the algorithms that are discarded in the deployment; (iii) it requires the testing data in addition to the training data, which increases the overall expenses for the algorithm selection. To overcome these drawbacks, this research proposes an ensemble data-driven prognostic approach which combines multiple member algorithms with a weighted-sum formulation. Three weighting schemes, namely the accuracy-based weighting, diversity-based weighting and optimization-based weighting, are proposed to determine the weights of member algorithms. The k-fold cross validation (CV) is employed to estimate the prediction error required by the weighting schemes. The results obtained from three case studies suggest that the ensemble approach with any weighting scheme gives more accurate RUL predictions compared to any sole algorithm when member algorithms producing diverse RUL predictions have comparable prediction accuracy and that the optimization-based weighting scheme gives the best overall performance among the three weighting schemes. This approach provides PHM efficiency measure as either the CV error or validation error for predictive resilience analysis and enables highly-confident health prognostics for resilient engineered systems with a single time-scale.

5.1.1 Weighted-Sum Formulation

A simple average of RUL predictions obtained using the member algorithms means assigning equal weights to the member algorithms used for prognostics. This is acceptable only when the member algorithms provide the same level of accuracy for a given problem. However, it is more likely that an algorithm tends to be more accurate than others. It is ideal to assign a greater weight to a member algorithm with higher prediction accuracy in order to enhance its prediction accuracy and robustness. Hence, member algorithms with different prediction performance should be multiplied by different weight factors.

Let $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\}$ be a data set consisting of multi-dimensional sensory signals (e.g., acceleration, strain, pressure) from N different run-to-failure units. An ensemble of prognostic member algorithms for RUL prediction can be expressed in a weighted-sum formulation as

$$\hat{L} = \sum_{j=1}^{M} w_j \hat{L}_j \left(\mathbf{y}_t, \mathbf{Y} \right) \tag{5.1}$$

where $\hat{L}$ denotes the ensemble predicted RUL for the testing data set $\mathbf{y}_t$; M denotes the number of algorithm members in the ensemble; w_j denotes the weight assigned to the j^{th} prognostic algorithm; $\hat{L}_j$ ($\mathbf{y}_t$, $\mathbf{Y}$) denotes the predicted RUL by the j^{th} prognostic member algorithm trained with the data set $\mathbf{Y}$. Let the weight vector $\mathbf{w} = [w_1,\ldots, w_M]^{\mathrm{T}}$ and the vector of predicted RULs by member algorithms $\hat{\mathbf{L}} = [\hat{L}_1,\ldots, \hat{L}_M]^{\mathrm{T}}$, the weighted-sum formulation in Eq. (5.1) can be expressed in a vector form as $\hat{L}\left(\mathbf{w},\hat{\mathbf{L}}\right) = \mathbf{w}^{\mathrm{T}}\hat{\mathbf{L}}$.

5.1.2 *K*-Fold Cross Validation

The *k*-fold cross validation is used in the offline process to evaluate the accuracy of a given ensemble. It randomly divides the original data set $\mathbf{Y}$ into k mutually exclusive subsets (or folds) $\mathbf{Y}_1$, $\mathbf{Y}_2,\ldots$, $\mathbf{Y}_k$ having an approximately equal size [109]. Of the k subsets, one is used as the test set and the other k–1 subsets are put together as a training set. The CV process is performed k times, with each of the k subsets used exactly once as the test set. Let $\mathbf{I}_m = \{i: \mathbf{y}_i \in \mathbf{Y}_m\}$, $m = 1, 2,\ldots, k$ denote the index set of the run-to-failure units whose sensory signals construct the subset $\mathbf{Y}_m$. Then the CV error is computed as the average error over all k trials and can be expressed as

$$\varepsilon_{CV} = \frac{1}{N}\sum_{m=1}^{k}\sum_{i\in\mathbf{I}_m} S\left(\hat{L}\left(\mathbf{w},\hat{\mathbf{L}}\left(\mathbf{y}_i,\mathbf{Y}\setminus\mathbf{Y}_m\right)\right),L_i^T\right) \tag{5.2}$$

where $S(\bullet)$ is a predefined evaluation metric that measures the accuracy of the ensemble-predicted RUL; N denotes the number of run-to-failure units for CV; L_i^T denotes the true RUL of the i^{th} unit. The above formula indicates that all units in the data set are used for both training and testing, and each unit is used for testing exactly once and for training k–1 times. Thus, the variance of the resulting estimate is likely to be reduced compared to the traditional holdout approach, resulting in superior performance when employing a small data set. It is important to note that the disadvantage of the *k*-fold CV against the holdout method is greater computational expense because the training process has to be executed *k* times. As a commonly used setting for CV, a 10-fold CV is employed in this study.

5.1.3 Weighting Schemes

This section will introduce three schemes to determine the weights of member algorithms: the accuracy-based weighting, diversity-based weighting and optimization-based weighting.

Accuracy-Based Weighting

The prediction accuracy of the j^{th} member algorithm is quantified by its CV error, expressed as

$$\varepsilon_{CV}^{j} = \frac{1}{N}\sum_{m=1}^{k}\sum_{i\in\mathbf{I}_m} S\left(\hat{L}_j\left(\mathbf{y}_i, \mathbf{Y}\setminus\mathbf{Y}_m\right), L_i^T\right) \tag{5.3}$$

The weight w_j of the j^{th} member algorithm can then be defined as the normalization of the corresponding inverse CV error, expressed as

$$w_j = \frac{\left(\varepsilon_{CV}^{j}\right)^{-1}}{\sum_{i=1}^{M}\left(\varepsilon_{CV}^{i}\right)^{-1}} \tag{5.4}$$

This definition indicates that a larger weight is assigned to a member algorithm with higher prediction accuracy. Thus, a member algorithm with better prediction accuracy has a larger influence on the ensemble prediction. This weighting scheme relies exclusively on the prediction accuracy to determine the weights of member algorithms.

Diversity-Based Weighting

The weight formulation in Eq. (5.4) relies exclusively on the prediction accuracy to determine the weights. However, the prediction accuracy of member algorithms is not the only factor that affects the ensemble performance. The prediction diversity, which measures the extent to which the predictions by a member algorithm are distinguishable from those by the others, also has a significant effect on the ensemble performance, especially on the robustness. More specifically, a larger weight should generally be assigned to a member algorithm with higher prediction diversity because of its larger potential to enhance the ensemble robustness.

We begin by formulating an N-dimensional error vector consisting of absolute RUL prediction errors by the j^{th} member algorithm as

$$\mathbf{e}_j = \left[\hat{L}_j\left(\mathbf{y}_1, \mathbf{Y} \setminus \mathbf{Y}_1\right) - L_1^T, \ldots, \hat{L}_j\left(\mathbf{y}_N, \mathbf{Y} \setminus \mathbf{Y}_m\right) - L_N^T \right]^{\mathrm{T}} \tag{5.5}$$

Repeatedly computing the error vectors for all M member algorithms gives M error vectors $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_M$. The prediction diversity of the j^{th} member algorithm can then be computed as the sum of Euclidean distances between the error vector $\mathbf{e}_j$ and all the other error vectors, given by

$$D_j = \sum_{i=1; i \neq j}^{M} \left\| \mathbf{e}_j - \mathbf{e}_i \right\| \tag{5.6}$$

The prediction diversity measures the extent to which the predictions by a member algorithm are distinguishable from those by any other. Based on the defined prediction diversity, the normalized weight w_j of the j^{th} member algorithm can then be calculated as

$$w_j = \frac{D_j}{\sum_{i=1}^{M} D_i} \tag{5.7}$$

This definition suggests that a member algorithm with higher prediction diversity will be given a larger weight and thus contributes more to the ensemble predicted RUL. For example, if, among all the member algorithms, one algorithm consistently gives early RUL predictions while any of the others late RUL predictions, the former will likely be given a larger weight than the latter. It is also noted that the weight formulation in Eq. (5.7) considers the prediction diversity as the only criterion for the weight determination.

Optimization-Based Weighting

Neither the accuracy-based nor diversity-based weighting scheme takes into account both the prediction accuracy and diversity in the weight calculation. Thus, the two schemes cannot produce an ensemble algorithm to achieve both high prediction accuracy and robustness. In what follows, an optimization-based weighting scheme is proposed to maximize the accuracy and robustness of data-driven prognostics by adaptively synthesizing the prediction accuracy and diversity of each member algorithm.

In the optimization-based weighting scheme, the weights in Eq. (5.1) can be obtained by solving an optimization problem of the following form

$$\begin{aligned} &\text{Minimize} \quad && \varepsilon_{CV} = \varepsilon_{CV}\left(\hat{L}\left(\mathbf{w}, \hat{\mathbf{L}}(\mathbf{y}_i)\right),\ L_i^T,\ i = 1, \ldots, N\right) \\ &\text{Subject to} \quad && \sum_{j=1}^{M} w_j = 1 \end{aligned} \tag{5.8}$$

After the prediction of RULs using the *M* member algorithms through the 10-fold CV, the above optimization problem can be readily solved with almost negligible computational effort since the weight optimization process does not require the execution of member algorithms. Thus, the overall computational cost mainly comes from the training and testing in the CV process. We expect that, by solving the optimization problem in Eq. (5.8), the resulting ensemble of algorithms will outperform any of the ensemble's individual member algorithms in terms of both accuracy and robustness. The capability of this weighting scheme to adaptively synthesize the prediction accuracy and diversity of each member algorithm will be demonstrated in the case study section.

5.1.4 Overall Procedure

Figure 5-1 shows the overall procedure of the proposed ensemble approach with the *k*-fold CV and three weighting schemes. This data-driven prognostic approach is composed of the offline and online processes. In the offline process, the offline training/testing process with the *k*-fold CV is employed to compute the CV error of an ensemble formulation; the weights of member algorithms are determined using the accuracy-based weighting, diversity-based weighting and optimization-based weighting. The online prediction process combines the RUL predictions from all member algorithms to form an ensemble RUL prediction using the weights obtained from the offline process. This process enables the continuous update of the health information and prognostic results in real-time with new sensory signals. Table 5-1 details the proposed ensemble prognostics approach with the five steps. STEPS 2-4 can be repeated to incorporate new training sensory signals from the online process and to update the RUL predictions.

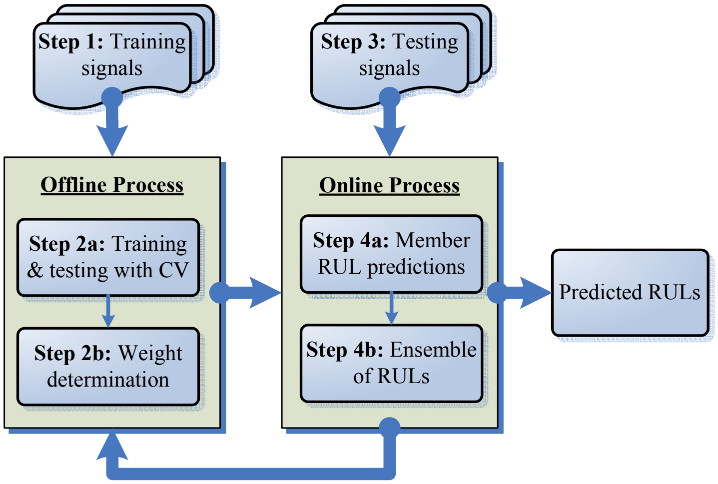


Figure 5-1 A flowchart of the ensemble approach

Table 5-1 Detailed procedure of the ensemble approach

STEP 1	Determine sensor configurations and acquire training sensory signals from offline system units;
STEP 2a	Perform the offline training & testing processes with the *k*-fold CV with the training sensory signals to compute the CV error;
STEP 2b	Determine the weights using the accuracy-based weighting, diversity-based weighting and optimization-based weighting schemes;
STEP 3	Acquire testing sensory signals from online system units;
STEP 4a	Predict RULs using the member algorithms through the online prediction process which employs the background health knowledge obtained from the offline training process;
STEP 4b	Predict the ensemble RULs with the optimum weights obtained from STEP 2b.

5.1.5 Case Studies for Ensemble Prognostics

In this section, the proposed ensemble of data-driven prognostic algorithms is demonstrated with three PHM case studies: (i) 2008 IEEE PHM challenge problem, (ii) power transformer problem, and (iii) electric cooling fan problem. In each case study, the ensemble approach combines RUL predictions from five popular data-driven prognostic algorithms, namely, a similarity-based interpolation (SBI) approach with RVM as the regression technique (RVM-SBI) [53,126], SBI with SVM (SVM-SBI) [53,127], SBI with the least-square exponential fitting (Exp-SBI) [53], a Bayesian linear regression with the least-square quadratic fitting (Quad-BLR) [55], and a recurrent neural network (RNN) approach (RNN) [56,127].

2008 IEEE PHM challenge problem

In an aerospace system (e.g., an airplane, a space shuttle), system safety plays an important role since failures can lead to dramatic consequences. In order to meet stringent safety requirements as well as minimize the maintenance cost, condition-based maintenance must be conducted throughout the system's lifetime, which can be enabled by system health prognostics. This case study aims at predicting the RULs of aircraft engine systems in an accurate and robust manner with massive and heterogeneous sensory data.

Description of Data Set

The data set provided by the 2008 IEEE PHM Challenge problem consists of multivariate time series signals that are collected from an engine dynamic simulation process. Each time series signal comes from a different degradation instance of the dynamic simulation of the same engine system [129]. 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 and different engine units start with different initial health conditions and manufacturing variations which are unknown. Three operational settings have a substantial effect on engine degradation behaviors and result in six different operation regimes as shown in Table 5-2. The 21 sensory 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. In the training data set, the damage growth in a unit was allowed until the occurrence of a system failure when one or more limits for safe operation have been reached. In the testing data set, the time series signals were pruned some time prior to the occurrence of a system failure. The objective of the problem is to predict the number of remaining operational cycles before failure in the testing data set.

Table 5-2 Six different operation regimes

Regime ID	Operating parameter 1	Operating parameter 2	Operating parameter 3
1	0	0	100
2	20	0.25	20
3	20	0.7	0
4	25	0.62	80
5	35	0.84	60
6	42	0.84	40

Implementation of Ensemble Approach

For the CV process, the training data set with 218 units were divided to 10 data subsets with a similar size. Each data subset was used for both training and testing and, more specifically, 9 times for training and once for testing. The training data subsets contain complete degradation information while the testing data subsets carry only partial degradation information. The latter were generated by truncating the original data subsets after pre-assigned RULs. The RUL pre-assigned to each unit in a testing data subset was randomly generated from a uniform distribution between its zero and half-remaining life. This range in the uniform distribution was selected based on the following two criteria: (i) the pre-assigned RULs should be small enough to allow the occurrence of substantial degradation; and (ii) the variation of the pre-assigned RULs should be large enough to test the robustness of algorithms.

Following the work in [53], this study selected 7 sensory signals (2, 3, 4, 7, 11, 12 and 15) among the 21 sensory signals for the use in the member algorithms: RVM-SBI, SVM-SBI, Exp-SBI and Quad-BLR. A monotonic lifetime trend can be observed from these 7 sensory signals of which the noise levels are relatively low. For the VHI construction, the system failure matrix $\mathbf{Q}_0$ was created with the sensory data in a system failure condition, $0 \leq L \leq 4$, while the system healthy matrix $\mathbf{Q}_1$ with those in a system healthy condition, $L > 300$. The RVM employed a linear spline kernel function with the initial most probable hyper-parameter vector for kernel weights $\boldsymbol{\alpha}_m = [1\times10^4,\ldots, 1\times10^4]$ and the initial most probable noise variance $\sigma_m{}^2 = 1\times10^{-4}$. In the SVM, a Gaussian kernel function is used with the parameter settings as: the regularization parameter $C = 10$ and the parameter of the ε-insensitive loss function $\varepsilon = 0.10$. In the RNN training, the 21 normalized sensory signals together with the regime ID at each cycle were used as the multi-dimensional inputs of the RNN and the RUL at the corresponding cycle was used as the output. The implementation details can be found in [56]. In the RNN architecture, the numbers of the input, recurrent and output units are $|I| = 22$, $|R| = 8$ and $|O| = 1$.

The evaluation metric considered for this example employed an asymmetric score function around the true RUL such that heavier penalties are placed on late predictions [129]. The score evaluation metric S can be expressed as

$$S\left(\hat{L}_i, L_i^T\right) = \begin{cases} \exp\left(-d_i/13\right) - 1, & d_i < 0 \\ \exp\left(d_i/10\right) - 1, & d_i \geq 0 \end{cases} \quad \text{where } d_i = \hat{L}_i - L_i^T \tag{5.9}$$

where $\hat{L}_i$ and L_i^T denote the predicted and true RUL of the i^{th} unit, respectively. This score function was used to compute the CV error ε_{CV} using Eq. (5.2) for the accuracy- and optimization-based weighting schemes. In this study the weight optimization problem in Eq. (5.8) was solved using a sequential quadratic optimization (SQP) method which is a gradient-based optimization technique.

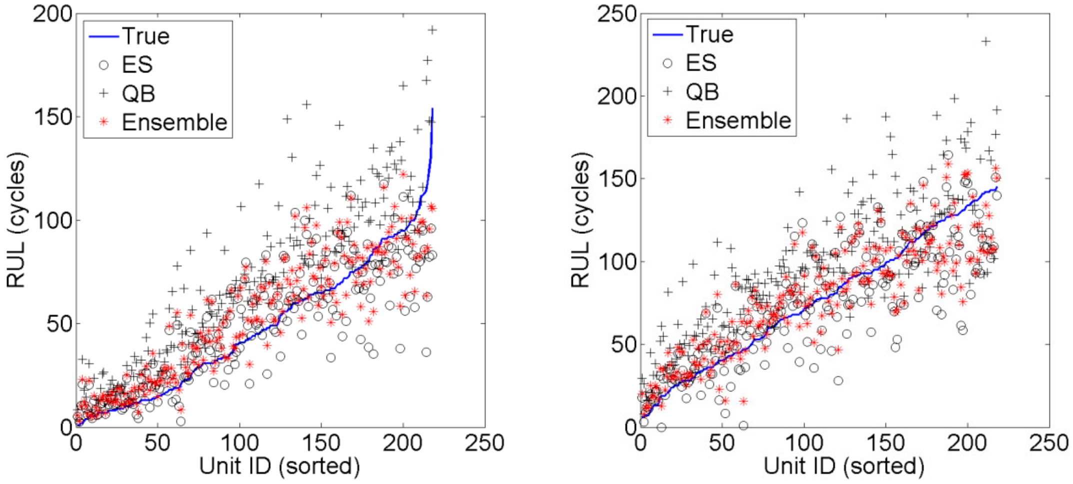
Results of Ensemble Approach

The five selected member algorithms are RVM-SBI (RS), SVM-SBI (SS), Exp-SBI (ES), Quad-BLR (QB) and RNN (RN). The three weighting schemes are the accuracy-based weighting (AW), diversity-based weighting (DW) and optimization-based weighting (OW). Table 5-3 summarizes the weighting results by the three weighting schemes as well as compares the CV and validation errors of the individual and ensemble approaches. It is observed that the ensemble approaches with all three weighting schemes outperforms any of the individual member algorithm in terms of the CV error and that the one with the optimization-based weighting achieves the smallest CV error of 4.8387 on the training data set, a 38.62% improvement over the best individual member algorithm, ES, whose CV error is 7.8834. As expected, the accuracy-based weighting scheme yields better prediction accuracy than the diversity-based weighting. This can be attributed to the fact that the former assigns larger weights to member algorithms with better prediction accuracy while the latter does not consider the prediction accuracy in the weight determination. To test the robustness of the ensemble approaches, the testing data set with 218 units were employed to compute the validation errors. Note that the testing data set is different from the training data set that was used to determine the weights in the ensemble approach. It is apparent that the ensemble approaches again outperform the individual member algorithms and that the one with the diversity-based weighting performs best, with a 34.7% improvement over the best individual member algorithm, SS. This suggests that the diversity-based weighting, compared to the accuracy-based weighting, provides a more robust ensemble of the member algorithms. It is noted that the optimization-based weighting scheme still achieves a comparable validation error to that of the diversity-based weighting scheme.

Table 5-3 Weighting results, CV and validation errors for 2008 PHM challenge problem

	RS	SS	ES	QB	RN	RS-SS-ES-QB-RN		
						AW	DW	OW
Weight by AW	0.3063	0.3029	0.3137	0.0151	0.0620	---	---	---
Weight by DW	0.1478	0.1488	0.1488	0.3354	0.2191	---	---	---
Weight by OW	0.0000	0.0470	0.7462	0.2068	0.0000	---	---	---
CV error	8.0743	8.1646	7.8834	163.3376	39.8583	6.9159	7.0852	**4.8387**
Validation error	10.2393	9.3907	10.4710	247.0079	20.1499	8.5544	**6.1280**	6.1955

Under the optimization-based weighting scheme, the RUL predictions by two individual algorithms, ES and QB, with the largest weights and the ensemble approach are plotted for 218 training and testing units in Figure 5-2. The units are sorted by the RULs in an ascending order. It is seen that ES tends to give consistently early RUL predictions while QB tends to provide consistently late RUL predictions. In contrast, the ensemble approach gives RUL predictions closer to the true values while eliminating many outliers produced by the two individual algorithms. The optimization-based weighting scheme provides better performance since the scheme employs an optimum ensemble formulation.



(a) (b)

Figure 5-2 RUL predictions of training units (a) and testing units (b) for 2008 PHM challenge problem (optimization-based weighting)

Comparison of Different Combinations of Member Algorithms

Out of the five member algorithms, 31 different combinations can be chosen to formulate an ensemble approach. It would be interesting to study how a choice of combination affects the performance of an ensemble approach. Table 5-4 summarizes the CV errors for ensemble approaches with all possible combinations of the member algorithms under the optimization-based weighting scheme. Three important remarks can be derived from the results. First of all, it is observed that the ES, as the individual member algorithm with the best performance, always serves as a member algorithm of the best ensemble approach. We also observe that the ES, when involved in the ensemble approach, always had a larger weight than any other. It indicates that the best member algorithm exhibits good cooperative performance which can be identified by the optimization-based weighting scheme. Secondly, the QB, which gives the worst individual performance, was surprisingly selected as an important member of the best ensemble approach. These results, though counterintuitive, suggest that the ensemble approach can adaptively synthesize the prediction ability and diversity of each individual algorithm to enhance the accuracy and robustness of RUL predictions. Indeed, the QB is prone to give late RUL predictions as shown in Figure 5-2 and thus possesses higher prediction diversity. Thirdly, both the mean and standard deviation of CV errors decrease as the number of member algorithms increases. The mean and standard deviation of CV errors of ensemble approaches with a single member algorithm are 45.4636 and 67.3188, respectively, and they monotonically decrease to 5.1896 and 0.7440, respectively, by the ensemble approach with four member algorithms. Thus it would be beneficial to have more member algorithms to enhance the prediction accuracy and reduce the uncertainty of this accuracy.

Table 5-4 Comparison of CV errors of different combinations of member algorithms for 2008 PHM challenge problem (optimization-based weighting)

Combination	CV error	Combination	CV error	Combination	CV error
RS	8.0743	RS-SS	8.0769	RS-SS-ES	7.8834
SS	8.1646	RS-ES	7.8834	RS-SS-QB	4.9123
ES	**7.8834**	RS-QB	4.9162	RS-SS-RN	6.7983
QB	163.3376	RS-RN	6.8002	RS-ES-QB	4.8391
RN	39.8583	SS-ES	7.8834	RS-ES-RN	6.5194
Mean	*45.4636*				
Std[a]	*67.3188*				
RS-SS-ES-QB	**4.8387**	SS-QB	4.9362	RS-QB-RN	4.9162
RS-SS-ES-RN	6.5194	SS-RN	6.8376	**SS-ES-QB**	**4.8387**
RS-SS-QB-RN	4.9123	**ES-QB**	**4.8391**	SS-ES-RN	6.5194
RS-ES-QB-RN	4.8391	ES-RN	6.5194	SS-QB-RN	4.9362
SS-ES-QB-RN	**4.8387**	QB-RN	17.5868	ES-QB-RN	4.8391
Mean	*5.1896*	*Mean*	*7.6279*	*Mean*	*5.7002*
Std	*0.7440*	*Std*	*3.7182*	*Std*	*1.1234*
RS-SS-ES-QB-RN	4.8387				

[a] *Standard deviation*

Power Transformer Problem

The power transformer is a critical power element in nuclear power plants, since an unexpected breakdown of the transformer causes plant shut-down and substantial societal expense. So it is very important to ensure high reliability and safety of the transformer during its operation. Investigations of the failures causes have revealed that mechanical breakdowns constitute a large portion of unexpected breakdowns of transformers in nuclear power plants [130]. Therefore, health monitoring and prognostics of the transformer with respect to mechanical failures is of significant importance to preventing unexpected breakdowns and minimizing interruptions to reliable customer service. This case study conducts transformer health prognostics with sensory signals obtained from a transformer finite element (FE) model.

Model Description

The FE model of a power transformer was created in ANSYS 10 as shown in Figure 5-3, where one exterior wall is concealed to make the interior structure visible. The transformer is fixed at the bottom surface and a vibration load with the frequency of 120 Hz is applied to the magnetic core. The three windings have a total number of twelve support joints, with each having four support joints. The random parameters considered in this study are listed in Table 5-5, which includes the material properties of support joints and windings as well the geometries of the transformer. The uncertainties in vibration responses propagated from these uncertain parameters will be accounted for when generating prognostic data.

Since it is very difficult, if not impossible, to obtain direct measurements of the health condition of transformers, indirect measurements are most often used to diagnose the health condition and predict the RULs of transformers [131]. In particular, the vibrations of the magnetic core and of the windings could characterize transitory overloads and permanent failures before any irreparable damage occurs [132,133]. Thus, this case study employs the vibration signals of the magnetic core and of the windings of a power transformer to predict the RULs of transformers.

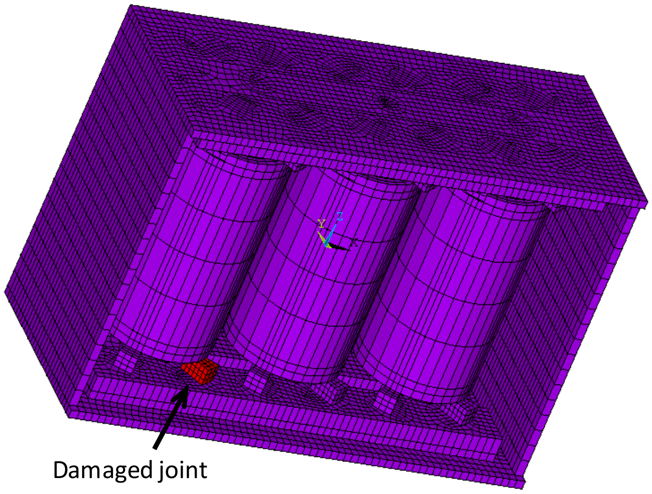


Figure 5-3 A power transformer FE model (without the covering wall)

Table 5-5 Random geometries and material properties for power transformer problem

Component	Physical meaning	Distri. type	Mean	Std
x_1	Wall Thickness	Normal	3	0.015
x_2	Angular width of support joints	Normal	15	0.075
x_3	Height of support joints	Normal	6	0.03
x_4	Young's modulus of support joint	Normal	2E+12	1E+10
x_5	Young's modulus of winding	Normal	1.28E+12	6E+8
x_6	Poisson ratio of joints	Normal	0.27	0.0027
x_7	Poisson ratio of winding	Normal	0.34	0.0034
x_8	Density of joints	Normal	7.85	0.000785
x_9	Density of windings	Normal	8.96	0.0896

Prognostic Data Generation

The failure mode considered in this study is the loosening of a winding support joint (see Figure 5-3) induced by the magnetic core vibration. The joint loosening was realized by reducing the stiffness of the joint. The failure criterion is defined as a 99%

stiffness reduction of the joint. To model the trajectory of change in stiffness over time, this study uses a damage propagation model with an exponential form as [129]

$$E(t) = E_0 + b_E \left(1 - \exp\left(a_E t\right)\right) \tag{5.10}$$

where E_0 is the initial Young's modulus of the joint; a_E and b_E are the model parameters; t is the cycle time. The initial Young's modulus E_0 follows the same normal distribution with x_4 (see Table 5-5). The model parameters a_E and b_E are independent and normally distributed with means 0.002 and 4E+12, each of which has a 10% coefficient of variation.

Since data-driven prognostic approaches require a large amount of prognostic data, it is computationally intolerable, if not impossible, to simply run the simulation to generate every data point. To overcome this difficulty, this study employed the univariate decomposition method that only uses a certain number of univariate sample points to construct the response surface for a general multivariate response function while achieving good accuracy [8]. This study selected 5 strain gauges (see Figure 5-4) from the optimally designed sensor network consisting of 9 strain gauges and thus requires the construction of 5 responses surfaces. The data generation process involves four sequentially executed procedures: (i) four univariate sample points were obtained from the harmonic analysis to construct response surfaces, along the damage propagation path, that approximate the strain components at five sensor locations as functions of random variables detailed in Table 5-5; (ii) 400 randomly generated samples of E_0, a_E and b_E were used in conjunction with Eq. (5.10) to produce 400 damage propagation paths, of which 200 paths were assigned to the training units and the rest to the testing units; (iii) the constructed response surfaces were used to interpolate the strain components at five sensor locations for a given set of randomly generated geometries and material properties and damage propagation paths, and repeatedly executing this process for 400 times gave the training data set with 200 training units and the testing data set with 200 testing units; (iv) measurement noise following a zero mean normal distribution was added to both the training and testing data sets to finalize the data generation. The cubic spline was used as the numerical scheme for the response surface construction and interpolation. Simulated measurements by sensors 1 and 5 are plotted against the adjusted cycle index, defined as the subtraction of the cycle-to-failure from the actual operational cycle, in Figure 5-5 for all 200 training units in the training data set.

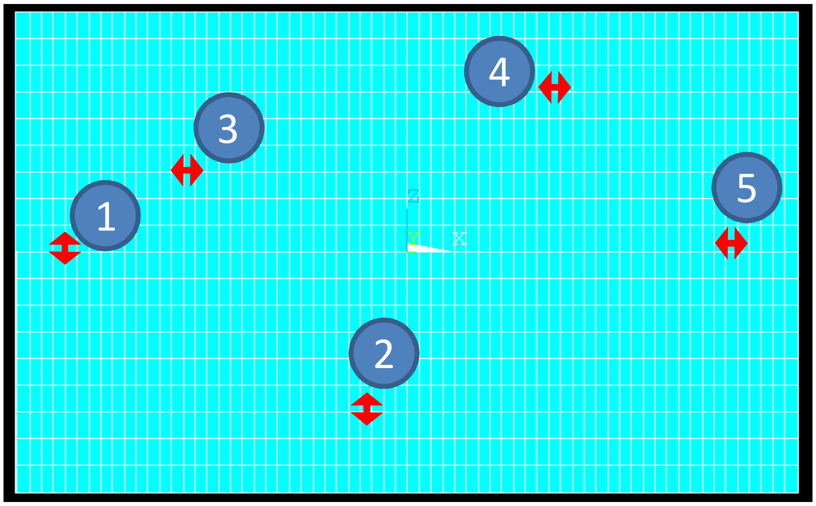
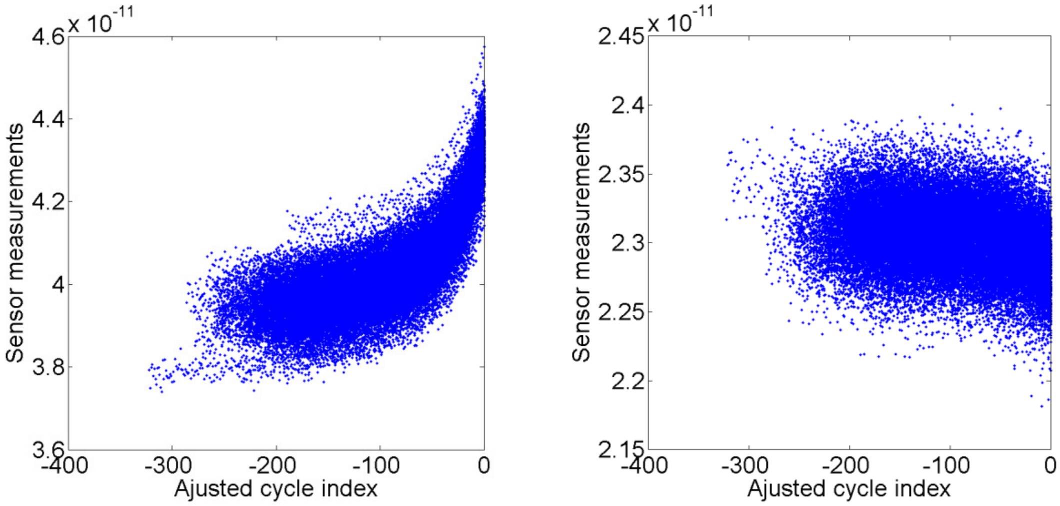


Figure 5-4 5 strain gauges located on the side wall of power transformer



(a) (b)

Figure 5-5 Simulated measurements by sensors 1 (a) and 5 (b) for power transformer problem

Implementation of Ensemble Approach

The training data set with 200 units were equally and randomly divided to 10 subsets. Similar to the first example, when used for the testing in CV, each unit in a subset was assigned with a randomly generated RUL from a uniform distribution between its zero and half-remaining life. All the five member algorithms used the same parameter settings with those detailed in the first case study. The score function in Eq. (5.9) was again used to compute the CV error ε_{CV} for the accuracy- and optimization-based weighting schemes.

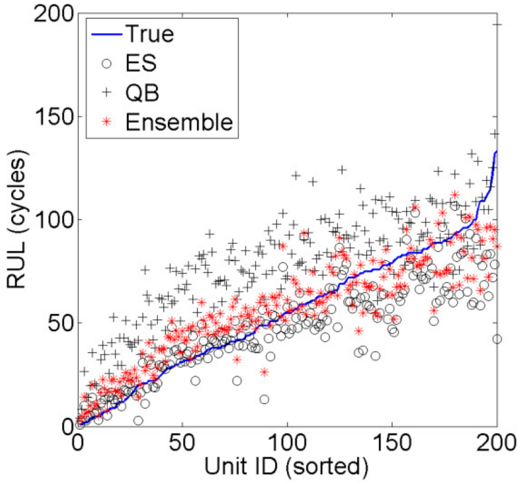
Results of Ensemble Approach

Table 5-6 summarizes the weighting results by the three weighting schemes as well as compares the CV and validation errors of the individual and ensemble approaches. Compared to the first example, similar results can be observed: (i) the ensemble approaches with all three weighting schemes yield smaller CV error than any of the individual member algorithm and the one with the optimization-based weighting gives the smallest CV error of 2.7258 on the training data set, a 66.48% improvement over the best individual member algorithm, RN, whose CV error is 8.1323; (ii) the accuracy-based weighting scheme yields a comparable CV error to that of the diversity-based weighting; (iii) the optimization-based weighting scheme achieves a validation error of 5.6138, which is comparable to the smallest validation error of 5.6119 by the diversity-based weighting scheme.

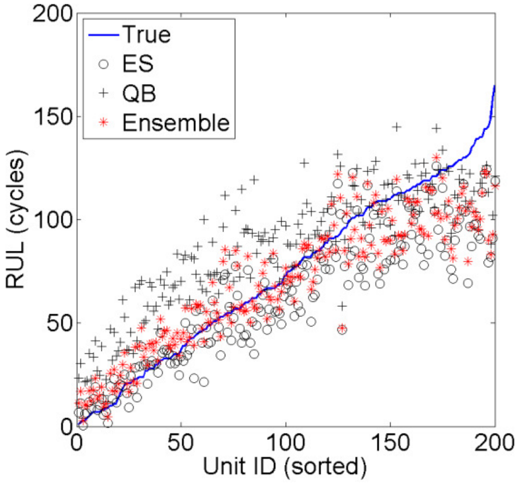
Under the optimization-based weighting scheme, the RUL predictions by two individual algorithms, ES and QB, with the largest weights and the ensemble approach are plotted for 218 training and testing units in Figure 5-6. It can be observed that ES and QB are prone to produce early and late RUL predictions, respectively, while the ensemble approach gives RUL predictions closer to the true values with a much smaller number of outliers.

Table 5-6 Weighting results, CV and validation errors for power transformer problem

	RS	SS	ES	QB	RN	RS-SS-ES-QB-RN		
						AW	DW	OW
Weight by AW	0.2128	0.2265	0.2343	0.0677	0.2588	---	---	---
Weight by DW	0.1488	0.1486	0.1688	0.3290	0.2048	---	---	---
Weight by OW	0.0000	0.0000	0.6303	0.2336	0.1361	---	---	---
CV error	9.8922	9.2945	8.9849	31.0891	8.1323	3.4874	3.4124	**2.7258**
Validation error	6.5737	6.8847	7.8251	20.0356	15.2265	5.7825	**5.6119**	5.6138



(a)



(b)

Figure 5-6 RUL predictions of training units (a) and testing units (b) for power transformer problem (optimization-based weighting)

Comparison of Different Combinations of Member Algorithms

A comparison study of different combinations of member algorithms was again carried out using the optimization-based weighting scheme for the power transformer problem. Table 5-7 summarizes the comparison results from which several important remarks similar to those in the first example can be derived. First of all, the member algorithms ES and QB can always be observed in the best ensemble approach with more than one member algorithms. We also observe that the combination ES and QB, when involved in the ensemble approach, always had a larger weight than any other. This result is different from what we observe in the first example, where the largest weight was assigned to the best individual member algorithm. This suggests that the optimization-based weighting scheme makes less use of or even discarded the best member algorithm that does not exhibit good cooperative performance with other members. Secondly, the QB, which gives the worst individual performance and is prone to give later RUL predictions (see Figure 5-6), was selected as an important member of the best ensemble approach. This again suggests that the prediction diversity plays an important role in the weight determination. Thirdly, as is the case in the first example, both the mean and standard deviation of CV errors decrease as the number of member algorithms increases. Thus the addition of member algorithms tends to enhance the prediction accuracy and reduce the uncertainty of this accuracy.

Table 5-7 Comparison of CV errors of different combinations of member algorithms for power transformer problem (optimization-based weighting)

Combination	CV error	Combination	CV error	Combination	CV error
RS	9.8922	RS-SS	9.2945	RS-SS-ES	8.9561
SS	9.2945	RS-ES	8.9849	RS-SS-QB	3.1688
ES	8.9849	RS-QB	3.1764	RS-SS-RN	3.9651
QB	31.0891	RS-RN	3.9744	RS-ES-QB	2.7894
RN	**8.1323**	SS-ES	8.9561	RS-ES-RN	3.4557
Mean	*13.4786*				
Std	*9.8650*				
RS-SS-ES-QB	2.7894	SS-QB	3.1815	RS-QB-RN	3.1470
RS-SS-ES-RN	3.4557	SS-RN	3.9671	SS-ES-QB	2.7894
RS-SS-QB-RN	3.1433	**ES-QB**	**2.7894**	SS-ES-RN	3.4557
RS-ES-QB-RN	**2.7258**	ES-RN	3.4557	SS-QB-RN	3.1559
SS-ES-QB-RN	**2.7258**	QB-RN	6.9724	**ES-QB-RN**	**2.7258**
Mean	*2.9680*	*Mean*	*5.4752*	*Mean*	*3.7609*
Std	*0.3232*	*Std*	*2.7412*	*Std*	*1.8640*
RS-SS-ES-QB-RN	2.7258				

Electric Cooling Fan Problem

In addition to the numerical studies, we also conducted experimental testing to verify the effectiveness of the ensemble approach. In this case study, we applied the ensemble approach to the health prognostics of electronic cooling fan units. This study aims to demonstrate the proposed ensemble prognostics with 32 electronic cooling fans.

Experimental Setup

In this experimental study, thermocouples and accelerometers were used to measure temperature and vibration signals. To make time-to-failure testing affordable, the accelerated testing condition for the DC fan units was sought with inclusion of a small amount of tiny metal particles into ball bearings and an unbalanced weight on one of the fan units. The experiment block diagram of DC fan accelerated degradation test is shown in Figure 5-7. As shown in the diagram, the DC fan units were tested with 12V regulated power supply and three different signals were measured and stored in a PC through a data acquisition system. Figure 5-8(a) shows the test fixture with 4 screws at each corner for the DC fan units. As shown in Figure 5-8(b), an unbalanced weight was used and mounted on one blade for each fan. Sensors were installed at different parts of the fan, as shown in Figure 5-9. In this study, three different signals were measured: the fan vibration signal by the accelerometer, the Printed Circuit Board (PCB) block voltage by the voltmeter, and the temperature measured by the thermocouple. An accelerometer was mounted to the bottom of the fan with superglue, as shown in Figure 5-9(a). Two wires were connected to the PCB block of the fan to measure the voltage between two fixed points, as shown in Figure 5-9(b). As shown in Figure 5-9(c), a thermocouple was attached to the bottom of the fan and measures the temperature signal of the fan. Vibration, voltage, and temperature signals were acquired by the data acquisition system and stored in PC. The data acquisition system from National Instruments Corp. (NI USB 6009) and the signal conditioner from PCB Group, Inc. (PCB 482A18) were used for the data acquisition system. In total, 32 DC fan units were tested at the same condition and all fan units run till failure.

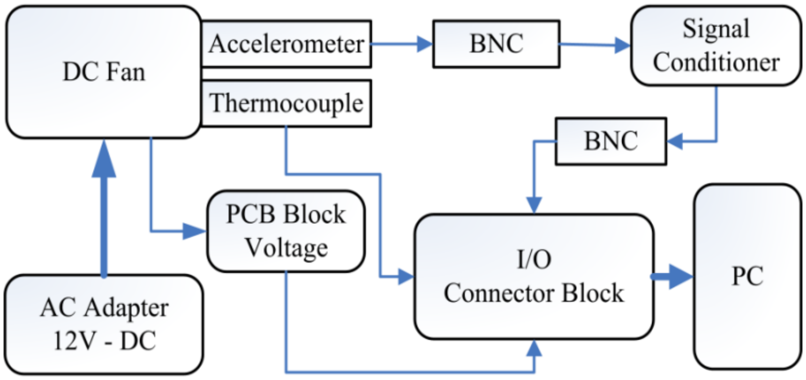


Figure 5-7 DC fan degradation test block diagram


(a)

(b)

Figure 5-8 DC fan test fixture (a) and the unbalance weight installation (b)

(a) (b) (c)

Figure 5-9 Sensor installations for DC fan test: (a) accelerometer, (b) voltmeter and (c) thermocouples

Implementation of Ensemble Approach

The sensory signal screening found that the fan PCB block voltage and the fan temperature did not show clear degradation trend, whereas the vibration signal showed health degradation behavior. This study involved the root mean squares (RMS) of the vibration spectral responses at the first five resonance frequencies and defined the RMS of the spectral responses as the PHI for the DC fan prognostics. Figure 5-10 shows the RMS signals of three fan units to demonstrate the health degradation behavior. The RMS signal gradually increased as the bearing in the fan degraded over time. It was found that the PHI is highly random and non-monotonic because of metal particles, sensory signal noise, and input voltage noise.

Among 32 fan units, the first 20 fan units were used to construct the training data set for the CV, while the rest were used to build the testing data set for the validation. Due to the small amount of training data, this case study employed the 5-fold CV where the training data set with 20 units was equally and randomly divided to 5 subsets. Similar to the previous examples, when used for the testing in CV, each unit in a subset was assigned with a randomly generated RUL from a uniform distribution between its zero and half-remaining life. To expand the number of testing units, each testing fan unit was assigned with two randomly generated RUL from a uniform distribution between its zero and half-remaining life, resulting in totally 24 testing units. The parameter settings detailed in the first case study was again used for the five member algorithms. With one cycle defined as every ten minutes, the score function in Eq. (5.9) was again used to compute the CV error ε_{CV} for the accuracy- and optimization-based weighting schemes.

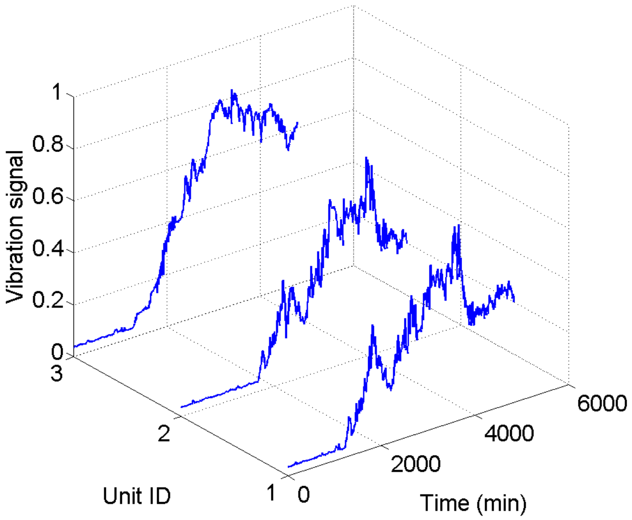


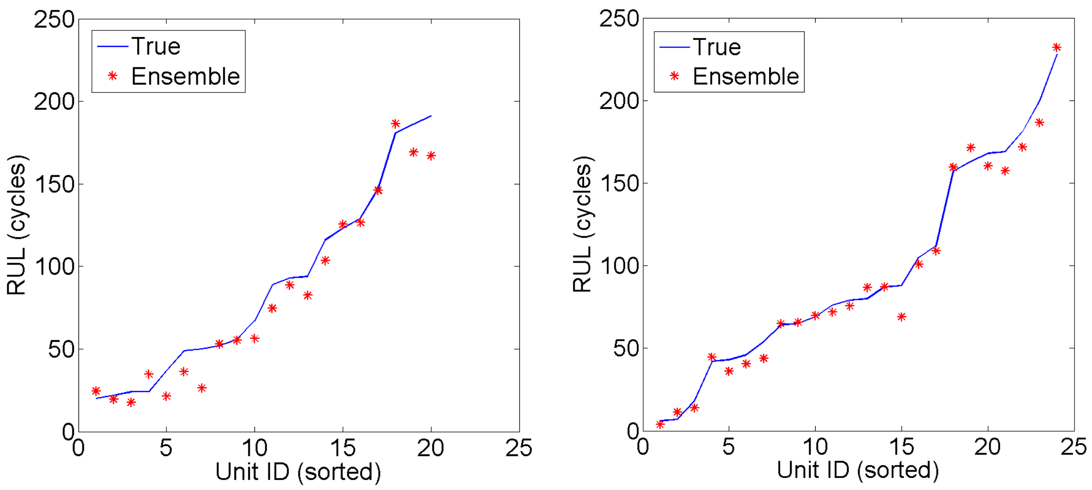
Figure 5-10 Sample degradation signals from DC fan testing

Results of Ensemble Approach

The weighting results by the three weighting schemes and the CV and validation errors of the individual and ensemble approaches are summarized in Table 5-8. Compared to the previous examples, we observed quite different results from which three important remarks can be derived. First of all, the ensemble approach with the diversity-based weighting scheme gives considerably larger CV and validation errors than the best individual member algorithms, RS and ES. This result is due to the fact that the diversity-based weighting, which relies exclusively on the prediction diversity for the weight determination, assigned larger weights to the member algorithms, QB and RN, which produced very low prediction accuracy due to the random and non-monotonic nature of the PHI (see Figure 5-10). Secondly, compared to the best individual member algorithms, RS and SS, the ensemble approach with the optimization-based weighting gave smaller CV and validation errors. However, the improvement is insignificant. Since non-zero weights are only assigned to the two member algorithms, RS and ES, with superb prediction capability, the performance of the resulting ensemble is totally determined by these two algorithms. However, RS and ES gave similar RUL predictions and the resulting ensemble, which is indeed a combination of two algorithms with similar prediction behavior, cannot achieve significant improvement in the prediction performance. Therefore, we expect that the ensemble approach achieves significant improvement in the prediction performance only in cases where member algorithms with comparable prediction accuracy produce diverse RUL predictions. Thirdly, although the member algorithms, QB and RN, have larger prediction diversity, their prediction accuracy is not comparable with that of the best member algorithms, RS and SS. As a result, these two algorithms were discarded from the algorithm pool by the optimization-based weighting. Under the optimization-based weighting scheme, the RUL predictions the ensemble approach are plotted for the training and testing units in Figure 5-11 where we observed very accurate RUL predictions by the ensemble approach.

Table 5-8 Weighting results, CV and validation errors for electric cooling fan problem

	RS	SS	ES	QB	RN	RS-SS-ES-QB-RN		
						AW	DW	OW
Weight by AW	0.3646	0.3767	0.2552	0.0008	0.0027	---	---	---
Weight by DW	0.1423	0.1427	0.1496	0.3285	0.2369	---	---	---
Weight by OW	0.1155	0.8845	0.0000	0.0000	0.0000	---	---	---
CV error	1.4770	1.4298	2.1100	717.8430	199.0067	1.5188	11.8520	**1.4292**
Validation error	0.7027	0.9223	0.7037	461.5064	84.3975	0.7185	11.0177	**0.6984**



(a) (b)

Figure 5-11 RUL predictions of training units (a) and testing units (b) for electric cooling fan problem (optimization-based weighting)

5.1.6 Conclusion

This section proposed a novel ensemble approach for the data-driven prognostics of high-risk engineered systems. By combining the predictions of all member algorithms, the ensemble approach achieves better accuracy in RUL predictions compared to any sole member algorithm. Furthermore, the ensemble approach has an inherent flexibility to incorporate any advanced prognostic algorithm that will be newly developed. To the best of our knowledge, this is the first study of an ensemble approach with three weighting scheme for the data-driven prognostics. Since the computationally expensive training process is done offline and the online prediction process requires a small amount of computational effort, the ensemble approach raises little concerns in the computational feasibility. Three engineering case studies (2008 PHM challenge problem, power transformer problem and electric cooling fan problem) demonstrated the superb performance of the proposed ensemble approach for the data-driven prognostics. Among the three weighting scheme, the optimization-based weighting scheme showed the capability of adaptively synthesizing the prediction accuracy and diversity of each member algorithm to enhance the accuracy of RUL predictions. Considering the enhanced accuracy and robustness in RUL predictions, the proposed ensemble approach is a promising data-driven prognostic method for resilient engineered systems with a single time-scale.

5.2 Co-Training Prognostics

The data-driven prognostics often requires a large amount of failure data for the offline training in order to achieve good accuracy for the online prediction. However, in many engineered systems, failure data are fairly expensive and time-consuming to obtain while suspension data are readily available. In such cases, it becomes essentially critical to utilize suspension data, which may carry rich information regarding the degradation trend and help achieve more accurate remaining useful life (RUL) prediction. To this end, this research proposes a co-training data-driven prognostic algorithm, denoted by COPROG, which uses two individual data-driven algorithms with each predicting RULs of suspension units for the other. The confidence of an individual data-driven algorithm in predicting the RUL of a suspension unit is quantified by the extent to which the inclusion of that unit in the training data set reduces the sum square error (SSE) in RUL prediction on the failure units. After a suspension unit is chosen and its RUL is predicted by an individual algorithm, it becomes a virtual failure unit that is added to the training data set. The co-training algorithm addresses the difficulty in achieving highly-confident health prognostics with the lack of failure data for resilient engineered systems with a single time-scale.

5.2.1 Description of Prognostic Algorithms

An artificial neural network (ANN) can be treated as a non-linear model that establishes a set of interconnected functional relationships between input patterns and desired outputs where a training process is employed to adjust the parameters (mainly network weights) of the functional relationships to achieve optimal performance. In recent years, neural networks have been extensively applied to predict the remaining useful lives (RULs) in various contexts such as machinery prognostics [57,60], flight control prognostics [134] and battery prognostics [135]. This section briefly introduces two selected neural network approaches for data-driven prognostics: a feed-forward neural network (FFNN) approach and a radial basis network (RBN) approach. A validation mechanism with multiple trials is used to train both the FFNN and RBN with an aim to minimize the overfitting as well as improve the generalization.

Feed-Forward Neural Network

Network Structure

The feed-forward neural network (FFNN), also known as the multi-layer perceptron, can fit any finite input-output mapping problem with a sufficient number of neurons in the hidden layer [136]. The network is composed of three layers (see Figure 5-12), namely, the input layer I, hidden layer H, and output layer O. Units of the input layer and the hidden layer are fully connected through the weights $\mathbf{W}^{HI}$ while units of the hidden layer and output layer are fully connected through the weights $\mathbf{W}^{OH}$. Let $\mathbf{I}^{(t)} = (I_1{}^{(t)},\ldots, I_i{}^{(t)},\ldots, I_{|I|}{}^{(t)})$, $\mathbf{H}^{(t)} = (H_1{}^{(t)},\ldots, H_j{}^{(t)},\ldots, H_{|H|}{}^{(t)})$ and $\mathbf{O}^{(t)} = (O_1{}^{(t)},\ldots, O_k{}^{(t)},\ldots, O_{|O|}{}^{(t)})$ be the input patterns, hidden activities and output activities at the time step t, respectively, where $|I|$, $|R|$ and $|O|$ denote the numbers of the input, hidden and output units, respectively, and let $\mathbf{b}^H$ and b^O be the bias terms added to the net inputs of hidden units and that of the output unit, respectively. The net input of the i^{th} hidden unit can then be computed as

$$\tilde{H}_j^{(t)} = \sum_i W_{ji}^{HI} I_i^{(t)} + b_{ji}^H \tag{5.11}$$

Given the hyperbolic tangent sigmoid transfer function as the activation function f_H, the output activity of the j^{th} hidden unit can then be computed as

$$H_j^{(t)} = f_H\left(\tilde{H}_j^{(t)}\right) = \frac{2}{1+\exp\left(-2\tilde{H}_j^{(t)}\right)} - 1 \tag{5.12}$$

Given the linear transfer function as the activation function f_R, the net input and output activity of the i^{th} output unit can be computed, respectively, as

$$\tilde{O}_k^{(t)} = \sum_j W_{kj}^{OH} H_j^{(t)} \tag{5.13}$$

and

$$O_k^{(t)} = f_O\left(\tilde{O}_k^{(t)}\right) = \tilde{O}_k^{(t)} \tag{5.14}$$

We note that, in order to use the FFNN for RUL prediction, both the network weights and biases need to be determined through the network training which will be detailed in the subsequent section.

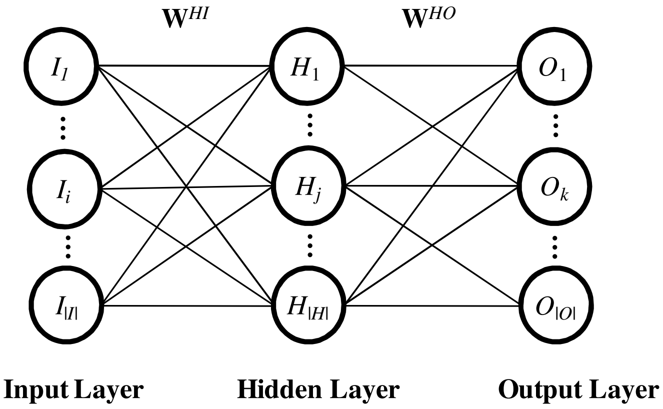


Figure 5-12 Structure of a FFNN with one hidden layer.

For data-driven prognostics, the inputs to the FFNN are the normalized current cycle value, normalized sensory measurements at the current and previous cycles. If we have N_s sensory measurements as the condition monitoring data at each cycle, the vector of network input patterns $\mathbf{I}^{(t)}$ is denoted by an input vector $\mathbf{x} = (x_1, x_2, \ldots, x_{2N_s+1})$ with x_1 being the current cycle value, x_{2i} and x_{2i+1} being the $(i-1)^{\text{th}}$ sensory measurement at the current and previous cycles, respectively, for $1 \leq i \leq N_s$. The output is the normalized predicted RUL associated with the current sensory measurement, denoted by L^P. As pointed out in previous works [60,137], the combined use of two consecutive data points provides valuable information regarding the rate of change of sensory measurements and thus the rate of system health degradation. We do not intend to use more than two data points due to the following reasons: (i) more out-of-date information regarding the "trend" of sensory measurements is carried by earlier data points, the addition of which may lead to the distortion of the most up-to-date information obtained from the two most recent data points; and (ii) an increase in the number of input patterns causes an increase in the network weights to be trained, which results in a higher chance of over fitting and deteriorates the generalization performance.

Training Process

The training of FFNN refers to the adjustment of network parameters (weights and biases) by exposing the network to a set of training input instances, observing the network outputs, and readjusting the parameters to minimize a training error. With the improvement of generalization being the main focus of FFNN training, we employ a validation mechanism based on the so-called holdout approach. In this mechanism, the holdout approach divides the original training data set into two mutually exclusive subsets called a training set and a validation set (or a holdout set). The training set is used to compute the gradient and update the network weights in order to minimize a performance function. The sum-square error (SSE) on the validation set is treated as the performance function or validation error, expressed as

$$SSE = \sum_{k=1}^{N} e_k^2 = \sum_{k=1}^{N} \left(L_k^P - L_k^T \right)^2 \tag{5.15}$$

where N is the number of training input and output instances, e_k is the prediction error for the k^{th} training instance, and L_k^P and L_k^T are the predicted and true normalized

RULs for the k^{th} instance. During the initial phase of training, the training error as well as the validation error typically decreases since the network is learning to find a good mapping between the training inputs and outputs. However, when the network begins to fit the noise, not just the signal, the overfitting occurs, leading to an increase in the validation error in spite of an uninterrupted decrease in the training error. The training is stopped when the increase in the validation error lasts for a specified number of training iterations, and the network weights and biases at the minimum of the validation error will be used to construct the FFNN model for RUL prediction. In this work, we used 60% of the original data set as the training data set and the rest as the validation set. We observed that this setting resulted in a FFNN model with good modeling and generalization performance. The backpropagation training with an adaptive learning rate [136] is employed to obtain the optimal weights and biases of the FFNN. Since the training algorithm is random, resulting in slightly different SSE values produced by different training executions, we train the FFNN 10 times to obtain 10 trained FFNNs among which the one with the lowest SSE is saved for future use.

Radial Basis Network

Network Structure

Another neural network approach we employ for data-driven prognostics is the radial basis network (RBN) which was reported to have important universal approximation properties [138], and whose structure bears a striking resemblance to that of FFNN shown in Figure 5-12. In an RBN, each unit in the hidden layer is a radial basis function ϕ with its own center, and for each input pattern $\mathbf{x} = (x_1, x_2, \ldots,$

x_{2N_s+1}), it computes the Euclidean distance between **x** and its center and then applies a polyharmonic basis function, expressed as

$$\phi\left(\mathbf{x},\mathbf{c}_j\right)=\begin{cases}\left\|\mathbf{x}-\mathbf{c}_j\right\|^{k_j}, & k_j=1,3,5,\ldots\\ \left\|\mathbf{x}-\mathbf{c}_j\right\|^{k_j}\ln\left(\left\|\mathbf{x}-\mathbf{c}_j\right\|\right), & k_j=2,4,6,\ldots\end{cases}\tag{5.16}$$

where $\mathbf{c}_j$ and k_j are the center and function order of the j^{th} unit in the hidden layer. We can observe from the above expression that each hidden unit in the RBN computes an output that depends on a radially symmetric function and, when the input is at the center of the unit, the strongest output can be obtained. The network output is the normalized predicted RUL L^P, expressed as a weighted summation of the outputs of hidden units

$$L^P=\sum_{j=1}^{M}W_{kj}^{OH}\phi\left(\mathbf{x},\mathbf{c}_j\right)\tag{5.17}$$

We note that, although the RBN and FFNN (or MLP) share a similar network structure, there are mainly three differences between these two networks:

(1) The activation function of the hidden layer in an RBN is a radially symmetric function (or a radial basis function) which computes the Euclidean distance between the input pattern and its center, whereas the activation function of a FFNN computes the inner product between the input pattern and the input weight vector.

(2) The output layer of an RBN is always in a linear form, whereas in a FFNN it can be in a linear or nonlinear form.

(3) An RBN typically has a single hidden layer, whereas a FFNN can have multiple hidden layers.

In order to use the RBN for RUL prediction, both the centers of hidden units and network weights need to be determined through the network training which will be detailed in the subsequent section.

Training Process

The training of an RBN can be viewed as a curve-fitting problem in a multidimensional space from the following two perspectives: (i) the objective of the training is to find an optimal response surface in a multidimensional space that provides the best fit to the training instances; and (ii) the testing (i.e., output of the network to input data not seen before) is equivalent to the use of this multidimensional surface to interpolate the test data. In this study, a two-phase learning scheme [139] is used to train the RBN with the multivariate polyharmonic basis function as the activation function. This training process is detailed as follows:

Phase 1: Initialize the centers **C** of radial basis functions (RBFs) with training input instances randomly selected from the original training data set, i.e., $\mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_M]$ with $\mathbf{c}_j$ being the j^{th} RBF center. The width σ of any RBF neuron is set to be one.

Phase 2: Determine the output layer weights $\mathbf{W}^{OH}$ which best approximate the training instances by a matrix pseudo-inverse technique, expressed as

$$\mathbf{W}^{OH} = \left(\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}\right)^{-1}\mathbf{\Phi}^{\mathrm{T}}\mathbf{L}^{T} \tag{5.18}$$

where the target output vector $\mathbf{L}^T = [L_1^T, \ldots, L_N^T]^{\mathrm{T}}$, and $\mathbf{\Phi}$ is an $N\times(M+1)$ design matrix constructed based on the training instances and RBF centers with $\mathbf{\Phi}_{ij} = \phi(\mathbf{x}_i, c_j)$. It is noted that the gradient-descent error backpropagation learning method is not used in this study since, compared to the matrix pseudo-inverse technique, it requires much higher computational effort. With an aim to improve the generalization performance of the RBN, we divide the original training data set into the mutually exclusive training set (60% of the original set) and validation set (40% of the original set), train the RBN with randomly selected RBF centers and evaluate the validation error 10 times, and choose the trained RBN with the lowest validation error for future use.

5.2.2 Co-Training Prognostics

This section presents the proposed co-training algorithm (COPROG) for data-driven prognostics. Firstly, the overall procedure of this algorithm is described. Next, details are provided on the measure to quantify the confidence of an individual data-driven algorithm in predicting the RUL of a suspension unit**.** This is followed by an introduction of the weight optimization scheme for combining RUL estimates from two algorithms for online prediction. Finally, remarks on how COPROG can help improve the prognostic performance are given.

Overall Procedure

Under the context of machine learning, the two data-driven prognostic algorithms (FFNN and RBN) detailed earlier can be treated as two regressors whose focus is to model the relationship between the RUL (dependant variable) and the current cycle value and sensory measurements (independent variables). Furthermore, failure data can be treated as labeled data since each input combination (current cycle value and sensory measurements) has its corresponding label (RUL), while suspension data can be named as unlabeled data since the label (RUL) of each input combination is unknown.

Let $\mathcal{L} = \{(\mathbf{x}_1, L_1^T), \ldots, (\mathbf{x}_{|\mathcal{L}|}, L_{|\mathcal{L}|}^T)\}$ and $\mathcal{U}$ represent the failure (labeled) and suspension (unlabeled) training data sets, respectively, where $\mathbf{x}_i$ is the i^{th} input instance composed of $2N_s + 1$ elements, L_i^T is its normalized RUL, i.e., its label, $|\mathcal{L}|$ is the number of labeled instances, and the RULs (labels) of instances in $\mathcal{U}$ are unknown. The pseudo-code of the COPROG algorithm is shown in Table 5-11, where the function TrainFun($\mathcal{L}$,j) returns the j^{th} trained algorithm ($j = 1$ for FFNN and $j = 2$ for RBN) based on the labeled data $\mathcal{L}$. In the training process (see Figure 5-13), COPROG works as follows: initially two trained algorithms h_1 and h_2 are generated from $\mathcal{L}$., and, for a predefined number T of iterations, the refinement of each algorithm is executed with the help of unlabeled instances labeled by the latest version of the other algorithm. For each iteration, a set $\mathcal{U}'$ of u suspension units is randomly sampled from $\mathcal{U}$. Each algorithm h_j predicts the labels of input instances of each unit in $\mathcal{U}'$ and selects the unit $\mathbf{X}_j'$ with the highest labeling confidence. Then the other algorithm is refined with the labeled unlabeled instances π_j added to its training data set $\mathcal{L}_j$. Note that a failure or suspension unit contains multiple input instances and, to distinguish a failure or suspension unit from an input instance, we use the notation $\mathbf{X}$ to denote the former and the notation $\mathbf{x}$ to denote the latter. In the testing process, the RUL estimate for a given testing instance is the weighted sum of the outputs of two algorithms built after the last COPROG iteration.

Table 5-9 Pseudo-code of the COPROG algorithm

```
ALGORITHM: COPROG

INPUT: $\mathcal{L}$ – failure data set, $\mathcal{U}$ – suspension data set, $T$ – maximum number
    of co-training iterations, $u$ – suspension pool size
TRAINING PROCESS:
    $\mathcal{L}_1 = \mathcal{L}$; $\mathcal{L}_2 = \mathcal{L}$
    $h_1 = TrainFun(\mathcal{L}_1, 1)$;  $h_2 = TrainFun(\mathcal{L}_2, 2)$;
    Repeat for $T$ times
      Create a pool $\mathcal{U}'$ of $u$ suspension units by random sampling from $\mathcal{U}$
      for $j = 1$ to 2
        for each $\mathbf{X}_u \subset \mathcal{U}'$
          $\mathbf{L}_u^P = h_j(\mathbf{X}_u)$;
          $h_j' = TrainFun(\mathcal{L}_j \cup \{\mathbf{X}_u, \mathbf{L}_u^P\}, j)$;
          $\Delta_{j,\mathbf{X}u} = \sum(L_i^T - h_j(\mathbf{x}_i))^2 - \sum(L_i^T - h_j'(\mathbf{x}_i))^2$
        end
        if there exists an $\Delta_{j,\mathbf{X}u} > 0$
          $\mathbf{X}_j' = \mathrm{argmax}_{\mathbf{X}u \subset \mathcal{U}'} \Delta_{j,\mathbf{X}u}$;  $\mathbf{L}_j' = h_j(\mathbf{X}_j')$;
          $\pi_j = \{(\mathbf{X}_j', \mathbf{L}_j')\}$; $\mathcal{U}' = \mathcal{U}' \setminus \pi_j$;
        else
          $\pi_j = \emptyset$;
        end
      end
      if $\pi_j == \emptyset$ && $\pi_2 == \emptyset$  exit
      else $\mathcal{L}_1 = \mathcal{L}_1 \cup \pi_2$; $\mathcal{L}_2 = \mathcal{L}_2 \cup \pi_1$;
          $h_1 = TrainFun(\mathcal{L}_1, 1)$;  $h_2 = TrainFun(\mathcal{L}_2, 2)$;
    end
TESTING PROCESS:
    $L^P = w_1 h_1(\mathbf{x}) + w_2 h_2(\mathbf{x})$ for any test data $\mathbf{x}$
```

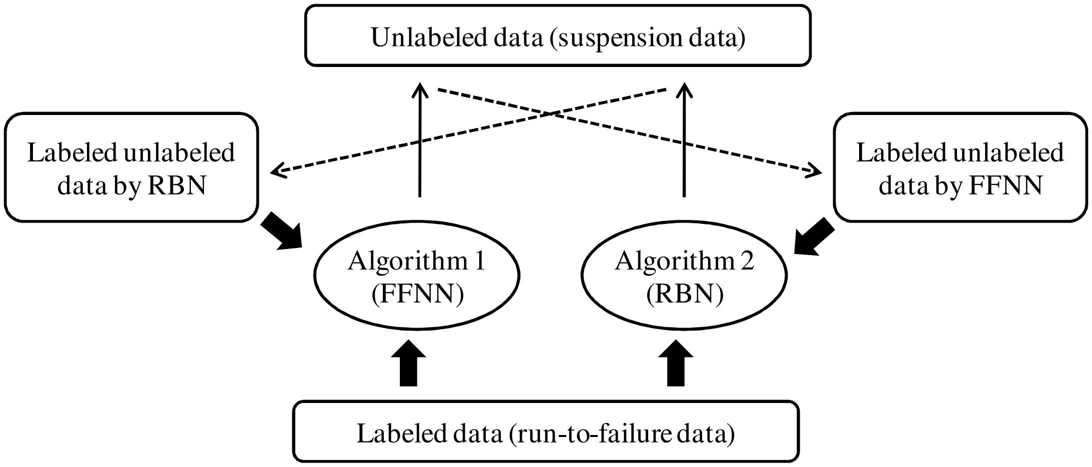


Figure 5-13 Flowchart of training process in COPROG

Confidence Measure

One critical issue in the co-training prognostics is how to select an appropriate suspension unit to utilize. An inappropriate selection may lead to a mislabeled suspension unit (or with an incorrect RUL estimate) which, if added to the training data set, may negatively affect the performance of an algorithm. We believe that the most confidently labeled suspension unit by a data-driven prognostic algorithm should help decrease the error of that algorithm on the labeled data set to the greatest extent. Therefore, we quantify the confidence in labeling a suspension unit by the extent to which the inclusion of that unit in the training data set reduces the sum square error (SSE) in RUL prediction on the failure units. Mathematically, the confidence measure of the j^{th} algorithm on a suspension unit $\mathbf{X}_u$ can be expressed as

$$\begin{aligned} \Delta_{j,\mathbf{X}_u} &= \sum_{\mathbf{x}_i \in \mathcal{L}_j} \left(\left(L_i^T - L_j^P\left(\mathbf{x}_i, \mathcal{L}_j\right) \right)^2 - \left(L_i^T - L_j^P\left(\mathbf{x}_i, \mathcal{L}_j \cup \left\{\mathbf{X}_u, \mathbf{L}_u^P\right\}\right) \right)^2 \right) \\ &= \sum_{\mathbf{x}_i \in \mathcal{L}_j} \left(\left(L_i^T - h_j\left(\mathbf{x}_i\right) \right)^2 - \left(L_i^T - h_j'\left(\mathbf{x}_i\right) \right)^2 \right) \end{aligned} \tag{5.19}$$

where L_i^T denotes the true RUL of the input instance $\mathbf{x}_i$ contained in the labeled data set $\mathcal{L}_j$, $L_j^P(\mathbf{x}_i, \mathcal{L}_j)$ denotes the predicted RUL by the j^{th} prognostic algorithm trained with the labeled data set $\mathcal{L}_j$, $\mathbf{L}_u^P$ denotes the predicted RULs of input instances contained in the suspension unit $\mathbf{X}_u$, h_j denotes the original algorithm and h_j' denotes the one refined with the suspension information $\{\mathbf{X}_u, \mathbf{L}_u^P\}$. The above confidence measure reflects the fact that the most confidently selected suspension unit is the one which keeps the prognostic algorithm most consistent with its existing training data set.

Weight Optimization

After using two data-driven prognostic algorithms to select and label the unlabeled suspension units during the offline training, we then obtain two augmented labeled training data sets $\mathcal{L}_1$ and $\mathcal{L}_2$, each of which contributes a trained algorithm for the online prediction. Then, the RUL predictions of these two algorithms are combined in a weighted-sum formulation as the final prediction. The simplest way is to average the two predictions, which is acceptable only when the prognostic algorithms provide the same level of accuracy. However, it is more likely that an algorithm tends to be more accurate than the other. In such cases, it would be ideal to assign a greater weight to a member algorithm with higher prediction accuracy in order to enhance its prediction accuracy. Hence, two individual algorithms with different prediction performance should be multiplied by different weight factors. In what follows, we propose a weight optimization scheme to maximize the accuracy in RUL prediction by adaptively synthesizing the prediction accuracy of each individual algorithm. In this scheme, the weights can be obtained by solving an optimization problem of the following form

$$\begin{aligned} &\text{Minimize} && SSE = \sum_{\mathbf{x}_i \in \mathcal{L}} \left(L_i^T - \left(w_1 h_1\left(\mathbf{x}_i\right) + w_2 h_1\left(\mathbf{x}_i\right) \right) \right)^2 \\ &\text{Subject to} && w_1 + w_2 = 1 \end{aligned} \tag{5.20}$$

where $\mathcal{L}$ denotes original training data set. After the prediction of RULs using the two prognostic algorithms, the above optimization problem can be readily solved with almost negligible computational effort since the weight optimization process does not require the re-execution of these algorithms. We expect that, by solving the optimization problem in Eq. (5.8), the resulting ensemble of the two algorithms will outperform its counterpart with equal weights in terms of prediction accuracy.

Remarks on Co-Training Prognostics

In what follows, we intend to elaborate on how the proposed COPROG algorithm can utilize the suspension data to improve the prognostic performance from two perspectives: (i) how an individual prognostic algorithm can benefit from the utilization of suspension data; and (ii) how the use of two algorithms can enhance the prognostic accuracy as compared to an individual algorithm.

Regarding the utilization of suspension data, Figure 5-14 illustrates that using one prognostic algorithm (FFNN or RBN) to label the unlabeled instances help improve the prediction accuracy on the test data in a prognostic sample space $\mathcal{P}$. Here, $\mathcal{P}$ consists of all possible prognostic samples obtained under different testing situations (e.g., manufacturing condition, health condition and degradation rate). We have sparse labeled data (or failure data) but plenty of unlabeled data (or suspension data).

For test data in the close vicinity of labeled data, we believe the training algorithm used to build the prognostic algorithm with only labeled data can generalize sufficiently well to make reasonable predictions. This is not to say all predictions made in such cases are highly accurate: at points that are sparsely populated by labeled data, relatively large errors are expected (as is the case in Figure 5-14), but the predictions will still be meaningful. For test data that fall significantly away from labeled data, we expect that FFNN or RBN outputs could contain intolerably large errors. If the unlabeled data can be properly labeled and added to the labeled data set, the algorithm can provide more accurate RUL predictions for test data that are close neighbors of these labeled unlabeled data. We note that the *proper labeling* is realized by selecting appropriate unlabeled data according to the maximization of the confidence measure in Eq. (5.19).

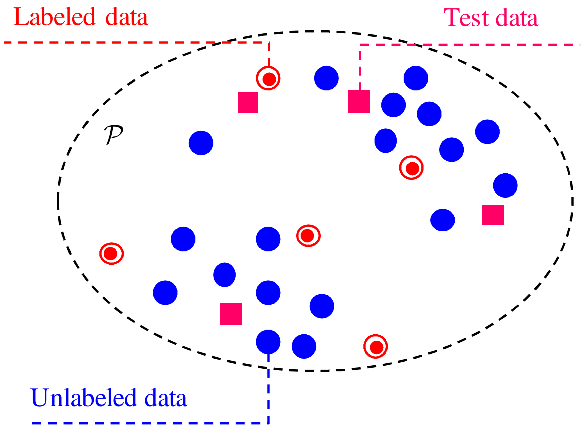


Figure 5-14 Prognostic space with labeled, unlabeled and test data

Regarding the use of two prognostic algorithms, we note that this strategy can produce the following two desirable effects:

Creating diversity: The two algorithms with different network structures and training procedures lead to the diversity in RUL prediction, based on which the ensemble obtains better predictive performance than could be obtained from any individual algorithm. Since, during each iteration, the suspension unit chosen by h_1 will not be chosen by h_2, the suspension units two algorithms label for each other are different, which can be treated as another mechanism for encouraging the diversity.

Reducing overfitting: If we consider that the labeled training data set contains noise, the use of two prognostic algorithms can be helpful to reduce overfitting. Let $\mathcal{N}$ denote the set of noisy data in $\mathcal{L}$. For a suspension (unlabeled) unit $\mathbf{X}_u$, either of the algorithms h_1 and h_2 will rely on a set of neighboring labeled data to label $\mathbf{X}_u$. Assume this set is Ω and $\mathbf{X}_u$ is labeled by h_1. Then, $\{\mathbf{X}_u, h_1(\mathbf{X}_u)\}$ is added to $\mathcal{L}_1$, where the labels $h_1(\mathbf{X}_u)$ suffers from the noisy data in $\Omega \cap \mathcal{N}$. For another unlabeled unit $\mathbf{X}_v$, which we assume is very close to $\mathbf{X}_u$, the neighboring labeled data for labeling $\mathbf{X}_v$ will be approximately $\Omega \cup \{\mathbf{X}_u, h_1(\mathbf{X}_u)\}$. Thus, $h_1(\mathbf{X}_v)$ will be roughly affected by $(\Omega \cap \mathcal{N}) \cup \{\mathbf{X}_u, h_1(\mathbf{X}_u)\}$. Note that $\{\mathbf{X}_u, h_1(\mathbf{X}_u)\}$ has already suffered from the noisy data in $\Omega \cap \mathcal{N}$. Thus, $h_1(\mathbf{X}_v)$ will be affected by $\Omega \cap \mathcal{N}$ more seriously than $h_1(\mathbf{X}_u)$ does. As we label more suspension units, the effect of noise continues to propagate and becomes more severe. Whereas if the unit $\mathbf{X}_u$ is labeled by h_2 and $\{\mathbf{X}_u, h_2(\mathbf{X}_u)\}$ is put into $\mathcal{L}_1$, then $h_1(\mathbf{X}_v)$ will suffer from $\Omega \cap \mathcal{N}$ only once, thereby preventing the effect of noise from propagating.

5.2.3 Case Study: Rolling-Element Bearing Prognostics

In this section, the proposed COPROG algorithm for data-driven prognostics is demonstrated with a PHM case study on rolling-element bearing health prognostics (simulation). The rolling-element bearing is a critical component in rotational machines, since an unexpected failure of the bearing leads to machine shut-down and catastrophic damage. Thus, it is very important to ensure high reliability and safety of the bearing during its operation. This case study conducts bearing health prognostics with sensory signals obtained from a vibration model of the rolling-element bearing.

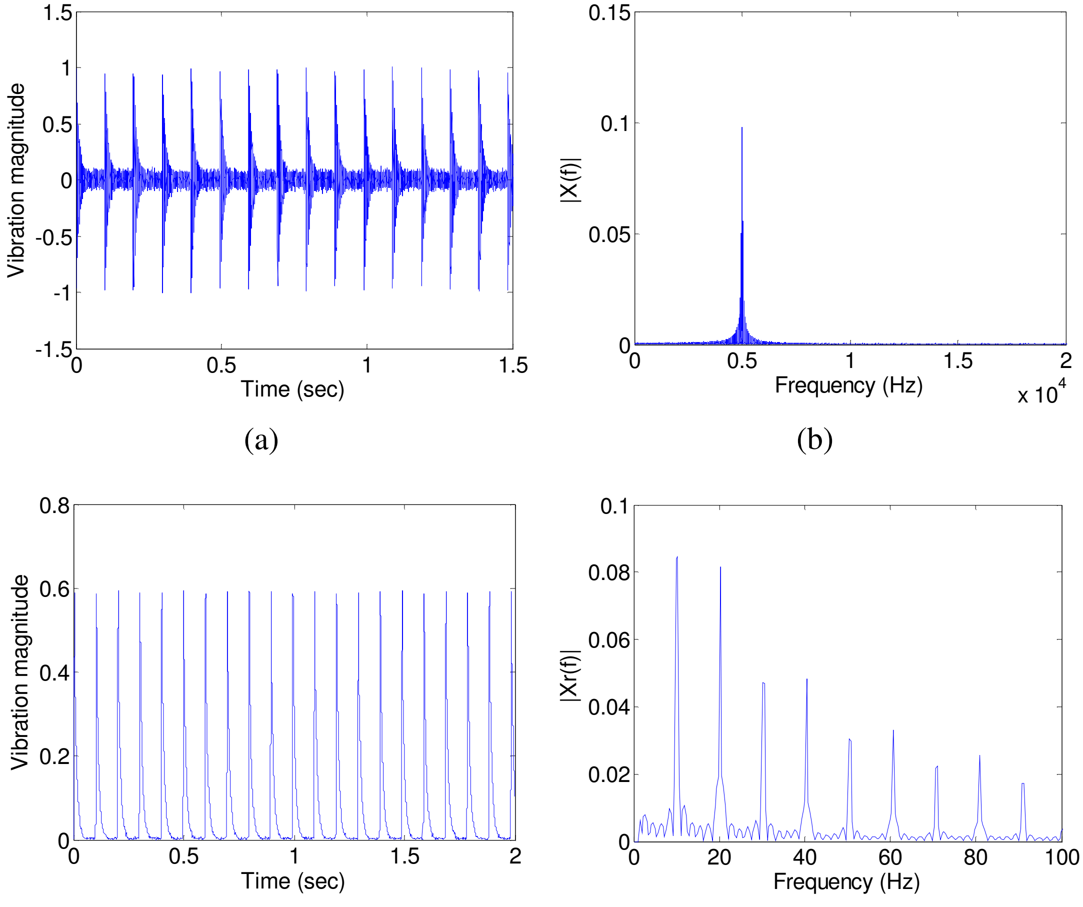
To study how the exploitation of suspension data affects the prognostic performance, we compared the co-training approach and the FFNN and RBN without the use of suspension data in terms of prognostic accuracy and robustness.

Bearing Defect Simulation

We employed an existing vibration model [140,141] to simulate the vibration signal produced by a single point defect on the inner race of a rolling-element bearing under constant radial load. The model takes into account the effects of the single point defect, shaft speed, bearing load distribution, and the exponential decay of vibration. The simulation assumes the following bearing parameters: pitch angle θ = 0°, shaft rotational speed v_r = 100rpm corresponding to shaft rotational frequency $f_r \approx$ 1.67Hz, bearing-induced resonant frequency f_s = 5000Hz, pitch diameter d_p = 23mm, roller diameter d_r = 8mm and number of rollers n_r = 9. Then the characteristic defective frequency corresponding to an inner race fault can be computed as

$$f_{IRF} = \frac{n_r f_r}{2}\left(1+\frac{d_r}{d_p}\cos\left(\theta\right)\right) \approx 10.11\text{Hz} \tag{5.21}$$

Figure 5-15(a) plots the simulated vibration signal of a bearing with an inner race fault in the time domain. Using the fast Fourier transform (FFT), we converted this signal to the frequency domain and obtained its frequency spectrum in Figure 5-15(b) where the spectrum is dominated by high-frequency resonant signals. Through band-pass filtering and rectifying the raw vibration signal, we excluded these resonant signals by other parts of the rotational machine and derived a demodulated signal as shown in Figure 5-15(c). The frequency domain plot of the demodulated signal in Figure 5-15(d) indicates the presence of a defect with the characteristic frequency of 10.13Hz which exhibits good consistency with the calculated inner race fault frequency in Eq. (5.21).



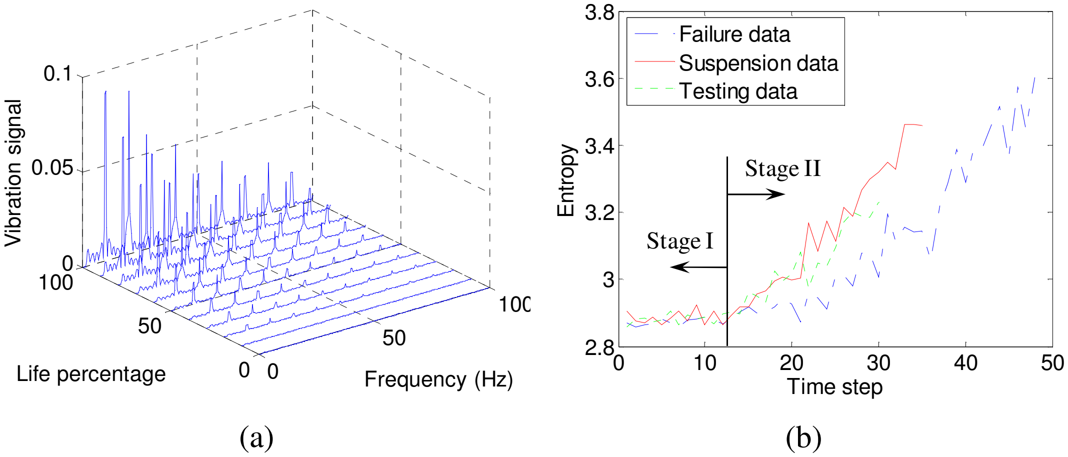
(a) (b)

(c) (d)

Figure 5-15 Simulated signal of outer-race defect: (a) time domain plot and (b) frequency spectrum of raw signal; (c) time domain plot and (d) frequency spectrum of demodulated signal.

We then repeatedly generated the vibration signals with exponentially increasing defect magnitudes corrupted by random fluctuations. A set of initial values and increasing rates of defect amplitudes were randomly generated to produce a set of bearing units. The lifecycle evolution of vibration spectra of an example bearing unit is plotted in Figure 5-16(a) where it can observed that, as degradation progresses over time, the defect magnitudes at harmonic defective frequencies (positive integer multiples of the characteristic defective frequency) begin to appear and increase exponentially. The feature we employed for data-driven prognostics is the entropy as shown in Figure 5-16(b). We can observe from both figures that the degradation undergoes two distinct stages. The first stage is referred to as normal operation period characterized by a relatively flat region. In this stage, no obvious defect can be found in the bearing. In the second stage, the degradation of the bearing begins and the signal is characterized by exponentially increasing defect magnitudes with random fluctuations. This two-stage degradation behavior is consistent with previous works on bearing prognostics [50,51,142].

For the training process, we generated a training data set consisting of 100 failure (labeled) units and 100 suspension (unlabeled) units. As shown in Figure 5-16(b), the failure data contain complete degradation information while the suspension data carry only partial degradation information. The latter were generated by truncating the original failure data after pre-assigned suspension times. The suspension time pre-assigned to each suspension unit was randomly generated from a uniform distribution between 90 and 100 percentile lives. This range in the uniform distribution was selected based on the assumption that the suspension unit is taken out of service when it approaches its end of life. For the testing process, we generated a testing data set consisting of 100 testing units by truncating the original failure data after pre-assigned RULs. The RUL pre-assigned to each testing unit was randomly generated from a uniform distribution between its zero and half-remaining life. The lifecycle evolution of entropy of a testing unit is plotted in Figure 5-16(b), where we can observe a smaller portion of health degradation pathway compared to a suspension unit.



(a) (b)

Figure 5-16 Lifecycle evolution of vibration spectra (a) and entropy (b) with an inner race defect.

Implementation of COPROG Algorithm

To investigate the effect of the amount of failure data on the performance improvement by COPROG, we evaluated algorithms under two different settings: *Setting 1* (lack of failure data) – 3 failure units and 10 suspension units (i.e., 3L-10U) and *Setting 2* (plenty of failure data) – 10 failure units and 10 suspension units (i.e., 10L-10U). For each setting, the failure and suspension data were randomly selected from the training data set consisting of 100 failure units (labeled) and 100 suspension units (unlabeled). To comprehensively test the performance of algorithms under various sets of failure and suspension data as well as account for the randomness in the training of FFNN and RBN, we repeatedly executed the evaluation process 50 times, each with a different set of failure and suspension units, and computed the mean (accuracy) and standard deviation (robustness) of root mean square errors (RMSEs) on the testing data. Mathematically, the mean RMSE can be expressed as

$$\begin{aligned}\mu_{RMSE} &= \frac{1}{50}\sum_{1\le k\le 50} RMSE_k \\ &= \frac{1}{50}\sum_{1\le k\le 50}\sqrt{\frac{\sum_{\mathbf{x}\in\mathcal{T}}\left(L^T(\mathbf{x})-L_k^P(\mathbf{x})\right)^2}{N_t}}\end{aligned} \tag{5.22}$$

where $L^T(\mathbf{x})$ denotes the true RUL of the input instance $\mathbf{x}$, $L^P(\mathbf{x})$ denotes the predicted RUL by an algorithm, N_t denotes the number of input instances in the testing data set $\mathcal{T}$. Since the health degradation at a very early stage is almost negligible and thus the occurrence of a failure is almost impossible, we extracted the testing input instances from the time step 6 of each testing trajectory. Since the RUL prediction at a late stage exerts a larger influence on maintenance decision-making than that at an early stage, we intended to separately investigate the prognostic accuracy when a bearing approaches its end of life. For this purpose, we extracted the testing input instances at the last 5 time steps of each testing trajectory and computed a critical-time RMSE using Eq. (5.22). In the COPROG algorithm, both the maximum number of co-training iterations T and the suspension pool size u were set to 5. Regarding the FFNN training, we employed 8 hidden units in the hidden layer and set the maximum training epochs to 100. Regarding the RBN training, we employed 20 RBF centers with first-order polyharmonic functions.

Results of COPROG Algorithm

Table 5-10 summarizes the RMSE results of supervised (FFNN and RBN) and semi-supervised (COPROG) learning. Here, FFNN and RBN refer to initial algorithms before utilizing any suspension data. In what follows, we intend to interpret the results from the following two perspectives:

Prognostic accuracy: It can be observed from Table 5-10 that the COPROG algorithm under any setting always outperforms any of the initial algorithms in terms of the life- and critical-time mean RMSEs, which verifies that COPROG is capable of exploiting the suspension data to improve the prognostic accuracy. Under the setting with the lack of failure data (i.e., 3L-10U), COPROG achieves the life- and critical-time mean RMSEs of 5.2674 and 4.5505 on the testing data set, 16.26% and 15.19% improvements over the best initial algorithm, RBN, whose mean RMSEs are 6.2905 and 5.3654, respectively. The accuracy improvement can be attributed to the effective utilization of valuable information that is only carried by the suspension data. As expected, the accuracy improvement becomes less significant when we have more failure data (i.e., 10L-10U). This is due to the fact that a larger amount of failure data captures more information regarding the degradation trend and leads to a reduced amount of information gained by utilizing suspension data.

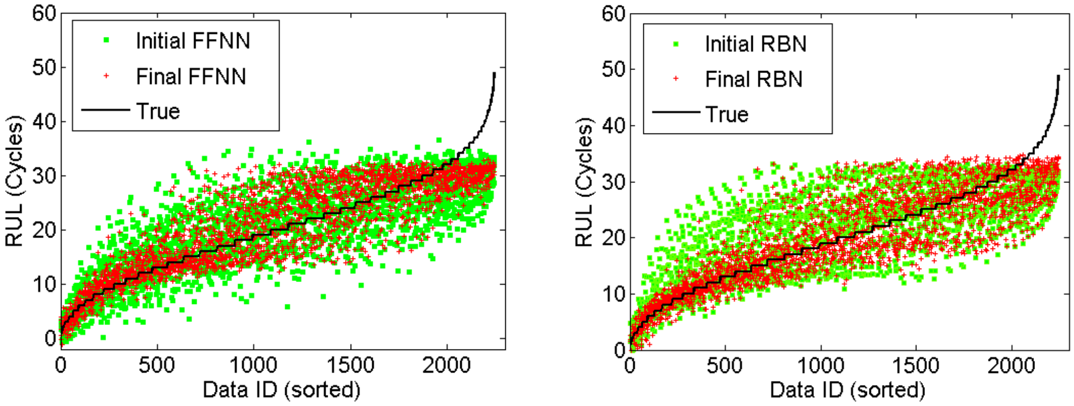
Prognostic robustness: In addition to the prognostic accuracy, we also evaluated the algorithms in terms of the prognostic robustness, that is, the extent to which the performance of an algorithm is insensitive to the variation in the training data. Here, the prognostic robustness was quantified using the standard deviation of RMSEs obtained from 10 random sets of training data. As shown in Table 5-10, COPROG always performs significantly better than the initial algorithms, which suggests that the exploitation of suspension data by COPROG helps improve the prediction robustness. The superior performance of COPROG in robustness can be attributed to the enrichment of degradation information by utilizing the suspension data and the combined use of two algorithms.

To illustrate the accuracy improvements obtained by exploiting suspension data, the RUL predictions by the initial algorithms (that is, FFNN and RBN trained without the utilization of any suspension data) and final algorithms (that is, FFNN and RBN after the co-training process) under the setting of 3L-10U are plotted for 200 training and testing units in Figure 5-17. The units are sorted by the RULs in an ascending order. The relative large scatter of RUL predictions around the true curve can be attributed to the lack of failure units (only 3) as well as the large noise in the entropy feature data (see Figure 5-16(b)). It can be seen that, compared to the two initial algorithms, the final algorithms yield RUL predictions that are closer to the true values while eliminating many outliers produced by the initial algorithms.

Table 5-10 RMSE results of supervised (FFNN and RBN) and semi-supervised (COPROG) learning for rolling-element bearing problem

Training data	Statistics	Life-time RMSE (cycles)			Critical-time RMSE (cycles)		
		FFNN	RBN	COPROG	FFNN	RBN	COPROG
3L-10U	Mean	6.3119	6.2905	**5.2674**	5.5487	5.3654	**4.5505**
	Std[a]	1.2980	1.2593	**0.4851**	1.5794	1.3378	**0.7659**
10L-10U	Mean	5.2051	5.0116	**4.7928**	4.5234	4.2165	**4.0406**
	Std	0.3501	0.4143	**0.2637**	0.6504	0.6291	**0.5108**

[a] Standard deviation



(a) (b)

Figure 5-17 RUL predictions by initial and final FFNNs (a) and RBNs (b) for rolling-element bearing problem (3L-10U)

5.2.4 Conclusion

This section presents a co-training prognostics (COPROG) algorithm, which, to the best of our knowledge, is one of the earliest efforts on semi-supervised learning for data-driven prognostics. By utilizing the suspension data, the COPROG algorithm achieves better accuracy and robustness in RUL predictions compared to any individual algorithm without utilizing the suspension data. Results from an engineering case study (rolling element bearing problem) suggested that COPROG is capable of effectively exploiting the suspension data to improve the prognostic performance and that the improvement becomes more pronounced when we have lack of failure data for the offline training.

5.3 A Multiscale Framework with Extended Kalman Filter (EKF)

In general, system's health condition is a slow time-varying system state, which can be highly correlated with a fast time-varying system state. For instance, the health condition (or capacity) of a battery is highly correlated with a fast time-varying system state, such as the state-of-charge (SOC). Should the existing techniques be used for such a system, they generally demand tremendous computational efforts and provide noisy estimates of the slow time-varying system state (or health condition). To overcome these shortcomings, this research proposes a multiscale framework with an extended Kalman filter (EKF) and applied this framework to Li-ion battery SOC and capacity estimation. When applied for battery prognostics, the proposed framework comprises two ideas: (i) a multiscale framework to estimate SOC and capacity that exhibit time-scale separation and (ii) a state projection scheme for accurate and stable capacity estimation. As a hybrid of coulomb counting and adaptive filtering techniques, the framework achieves higher accuracy and efficiency than joint/dual EKF. This multiscale framework enables highly-confident health prognostics for resilient engineered systems with multiple time-scales.

5.3.1 System Description

To make the discussion more concrete, we will use discrete-time state-space models with multiple time-scales. Without loss of generality, we assume the system has two time-scales: the macro and micro time-scales. System quantities on the macro time-scale tend to vary slowly over time while system quantities on the micro time-scale exhibit fast variation over time. The former are referred to as the model parameters of the system while the latter are called the states of the system. We then begin by defining the nonlinear state-space model considered in this work as

$$\begin{aligned}&\text{Transition:} \quad \mathbf{x}_{k,l+1} = \mathbf{F}\left(\mathbf{x}_{k,l}, \mathbf{u}_{k,l}, \boldsymbol{\theta}_k\right) + \mathbf{w}_{k,l}, \quad \boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \mathbf{r}_k, \\ &\text{Measurement:} \ \mathbf{y}_{k,l} = \mathbf{G}\left(\mathbf{x}_{k,l}, \mathbf{u}_{k,l}, \boldsymbol{\theta}_k\right) + \mathbf{v}_{k,l}\end{aligned} \tag{5.23}$$

where $\mathbf{x}_{k,l}$ is the vector of system states at the time $t_{k,l} = t_{k,0} + l \cdot T$, for $1 \le l \le L$, with T

being a fixed time step between two adjacent measurement points, and k and l being the indices of macro and micro time-scales, respectively; $\boldsymbol{\theta}_k$ is the vector of system model parameters at the time $t_{k,0}$; $\mathbf{u}_{k,l}$ is the vector of observed exogenous inputs; $\mathbf{y}_{k,l}$ is the vector of system observations (or measurements); $\mathbf{w}_{k,l}$ and $\mathbf{r}_k$ are the vectors of process noise for states and model parameters, respectively; $\mathbf{v}_{k,l}$ is the vectors of measurement noise; $\mathbf{F}(\bullet,\bullet,\bullet)$ and $\mathbf{G}(\bullet,\bullet,\bullet)$ are the state transition and measurement functions, respectively. Note that L represents the level of time-scale separation and that $\mathbf{x}_{k,0} = \mathbf{x}_{k-1,L}$. With the system defined, we aim at estimating both the system states $\mathbf{x}$ and model parameters $\boldsymbol{\theta}$ from the noisy observations $\mathbf{y}$.

Let us take the battery system as an example. In the battery system, the system state x refers to the SOC, which changes very rapidly and may transverse the entire range 100%-0% within minutes. Here we use an italic, non-bold letter x to indicate that the system state in the battery system is a scalar rather than a vector, and the same notational rule applies to all other functions and variables. The system model parameter θ represents the cell capacity which tends to vary very slowly and typically decreases 1.0% or less in a month with regular use. The state transition equation $F(\bullet,\bullet,\bullet)$ models the variation of SOC over time while the cell dynamic model $G(\bullet,\bullet,\bullet)$ relates the measured cell terminal voltage y with the unmeasured state (SOC) and model parameter (capacity) and the measured exogenous input u. Given the system's state-space model in Eq. (5.23) and knowledge of the measured input/output signals (cell current/cell terminal voltage), we are interested in estimating the unmeasured state (SOC) and model parameter (capacity) in real-time and in a dynamic environment.

5.3.2 Review of Dual Extended Kalman Filter Method

The dual extended Kalman filter (EKF) method is a commonly used technique to simultaneously estimate the states and model parameters [143]. The essence of the dual EKF method is to combine the state and weight EKFs with the state EKF estimating the system states and the weight EKF estimating the system model parameters. In the algorithm, two EKFs are run concurrently and, at every time step when observations are available, the state EKF estimates the states using the current model parameter estimates from the weight EKF while the weight EKF estimates the model parameters using the current state estimates from the state EKF. This section gives a brief review of the dual EKF method. The first part of this section presents the numerical formulation and the second part describes the numerical implementation.

Numerical Formulation: Dual Estimation

The algorithm of the dual EKF for the system described in Eq. (5.23) is summarized in Table 5-11. Since the dual EKF does not take into account the time-scale separation, $\boldsymbol{\theta}_k$ is estimated on the micro time-scale. To reflect this, we use the notations $\boldsymbol{\theta}_{k,l}$ and $\mathbf{r}_{k,l}$ to replace $\boldsymbol{\theta}_k$ and $\mathbf{r}_k$, respectively. Also note that, to be consistent with the system description in Eq. (5.23), we use two time indices k and l to present the dual EKF algorithm and this presentation is equivalent to a simpler version in [144] with only one time index l. The algorithm is initialized by setting the model parameters $\boldsymbol{\theta}$ and states $\mathbf{x}$ to the best guesses based on the prior information. The covariance matrices $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$ and $\boldsymbol{\Sigma}_{\mathbf{x}}$ of estimation errors are also initialized based on the prior information. At each measurement time step, the time- and measurement-updates are performed in the following two EKFs: weight EKF and state EKF.

Table 5-11 Algorithm of dual extended Kalman filter [144]

Initialization

$$\hat{\boldsymbol{\theta}}_{0,0} = E\left[\hat{\boldsymbol{\theta}}_{0,0}\right], \boldsymbol{\Sigma}_{\boldsymbol{\theta}_{k,l}} = E\left[\left(\boldsymbol{\theta}_{0,0} - \hat{\boldsymbol{\theta}}_{0,0}\right)\left(\boldsymbol{\theta}_{0,0} - \hat{\boldsymbol{\theta}}_{0,0}\right)^{\mathrm{T}}\right],$$
$$\hat{\mathbf{x}}_{0,0} = E\left[\mathbf{x}_{0,0}\right], \boldsymbol{\Sigma}_{\mathbf{x}_{k,l}} = E\left[\left(\mathbf{x}_{0,0} - \hat{\mathbf{x}}_{0,0}\right)\left(\mathbf{x}_{0,0} - \hat{\mathbf{x}}_{0,0}\right)^{\mathrm{T}}\right]. \tag{5.24}$$

For $k \in \{1,\ldots,\infty\}$, $l \in \{1,\ldots,L\}$, compute

Time-update equations for the weight filter

$$\hat{\boldsymbol{\theta}}_{k,l}^{-} = \hat{\boldsymbol{\theta}}_{k,l-1}, \quad \boldsymbol{\Sigma}_{\boldsymbol{\theta}_{k,l}}^{-} = \boldsymbol{\Sigma}_{\boldsymbol{\theta}_{k,l-1}} + \boldsymbol{\Sigma}_{\mathbf{r}_{k,l-1}}. \tag{5.25}$$

Time-update equations for the state filter

$$\hat{\mathbf{x}}_{k,l}^{-} = \mathbf{F}\left(\hat{\mathbf{x}}_{k,l-1}, \mathbf{u}_{k,l-1}, \hat{\boldsymbol{\theta}}_{k,l}^{-}\right), \qquad \boldsymbol{\Sigma}_{\mathbf{x}_{k,l}}^{-} = \mathbf{A}_{k,l-1}\boldsymbol{\Sigma}_{\mathbf{x}_{k,l-1}}\mathbf{A}_{k,l-1}^{\mathrm{T}} + \boldsymbol{\Sigma}_{\mathbf{w}_{k,l-1}}. \tag{5.26}$$

Measurement-update equations for the state filter

$$\mathbf{K}_{k,l}^{\mathbf{x}} = \boldsymbol{\Sigma}_{\mathbf{x}_{k,l}}^{-}\left(\mathbf{C}_{k,l}^{\mathbf{x}}\right)^{\mathrm{T}}\left[\mathbf{C}_{k,l}^{\mathbf{x}}\boldsymbol{\Sigma}_{\mathbf{x}_{k,l}}^{-}\left(\mathbf{C}_{k,l}^{\mathbf{x}}\right)^{\mathrm{T}} + \boldsymbol{\Sigma}_{\mathbf{v}_{k,l}}\right]^{-1}. \tag{5.27}$$

$$\hat{\mathbf{x}}_{k,l} = \hat{\mathbf{x}}_{k,l}^{-} + \mathbf{K}_{k,l}^{\mathbf{x}}\left[\mathbf{y}_{k,l} - \mathbf{G}\left(\hat{\mathbf{x}}_{k,l}^{-}, \mathbf{u}_{k,l}, \hat{\boldsymbol{\theta}}_{k,l}^{-}\right)\right], \qquad \boldsymbol{\Sigma}_{\mathbf{x}_{k,l}} = \left(\mathbf{I} - \mathbf{K}_{k,l}^{\mathbf{x}}\mathbf{C}_{k,l}^{\mathbf{x}}\right)\boldsymbol{\Sigma}_{\mathbf{x}_{k,l}}^{-}. \tag{5.28}$$

Measurement-update equations for the weight filter

$$\mathbf{K}_{k,l}^{\boldsymbol{\theta}} = \boldsymbol{\Sigma}_{\boldsymbol{\theta}_{k,l}}^{-}\left(\mathbf{C}_{k,l}^{\boldsymbol{\theta}}\right)^{\mathrm{T}}\left[\mathbf{C}_{k,l}^{\boldsymbol{\theta}}\boldsymbol{\Sigma}_{\boldsymbol{\theta}_{k,l}}^{-}\left(\mathbf{C}_{k,l}^{\boldsymbol{\theta}}\right)^{\mathrm{T}} + \boldsymbol{\Sigma}_{\mathbf{n}_{k,l}}\right]^{-1}. \tag{5.29}$$

$$\hat{\boldsymbol{\theta}}_{k,l} = \hat{\boldsymbol{\theta}}_{k,l}^{-} + \mathbf{K}_{k,l}^{\boldsymbol{\theta}}\left[\mathbf{y}_{k,l} - \mathbf{G}\left(\hat{\mathbf{x}}_{k,l}^{-}, \mathbf{u}_{k,l}, \hat{\boldsymbol{\theta}}_{k,l}^{-}\right)\right], \qquad \boldsymbol{\Sigma}_{\boldsymbol{\theta}_{k,l}} = \left(\mathbf{I} - \mathbf{K}_{k,l}^{\boldsymbol{\theta}}\mathbf{C}_{k,l}^{\boldsymbol{\theta}}\right)\boldsymbol{\Sigma}_{\boldsymbol{\theta}_{k,l}}^{-}. \tag{5.30}$$

where

$$\mathbf{A}_{k,l-1} = \left.\frac{\partial \mathbf{F}\left(\mathbf{x}, \mathbf{u}_{k,l-1}, \hat{\boldsymbol{\theta}}_{k,l}^{-}\right)}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_{k,l-1}}, \quad \mathbf{C}_{k,l}^{\mathbf{x}} = \left.\frac{\partial \mathbf{G}\left(\mathbf{x}, \mathbf{u}_{k,l}, \hat{\boldsymbol{\theta}}_{k,l}^{-}\right)}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_{k,l}^{-}},$$
$$\mathbf{C}_{k,l}^{\boldsymbol{\theta}} = \left.\frac{d\mathbf{G}\left(\hat{\mathbf{x}}_{k,l}^{-}, \mathbf{u}_{k,l}, \boldsymbol{\theta}\right)}{d\boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}_{k,l}^{-}}. \tag{5.31}$$

Weight EKF (Parameter Estimation)

The weight EKF first executes the time-update, where prior parameter estimates $\hat{\boldsymbol{\theta}}_{k,l}^{-}$ and their error covariance $\boldsymbol{\Sigma}_{\boldsymbol{\theta}_{k,l}}^{-}$ are computed with Eq. (5.25). Due to the addition of unpredictable process noise $\mathbf{r}_{k,l}$ in Eq. (5.23), the uncertainties $\boldsymbol{\Sigma}_{\boldsymbol{\theta}_{k,l}}^{-}$ in the parameter estimates always increase. Following the time-update step, the estimated measurements are then computed by

$$\hat{\mathbf{y}}_{k,l} = \mathbf{G}\left(\hat{\mathbf{x}}_{k,l}^{-}, \mathbf{u}_{k,l}, \hat{\boldsymbol{\theta}}_{k,l}^{-}\right) \tag{5.32}$$

The above predicted measurements are compared with the real measurements $\mathbf{y}_{k,l}$ to obtain prediction errors which state the novelty or the new information that the measurements $\mathbf{y}_{k,l}$ brought to the filter relative to the parameters $\boldsymbol{\theta}_{k,l}$. The prediction errors are used to adapt the current parameter estimates and obtain posteriori parameter estimates $\hat{\boldsymbol{\theta}}_{k,l}$ using Eq. (5.30). Due to the addition of one set of measurements, the error uncertainties are reduced as can be seen in Eq. (5.30). This process is referred to as the measurement-update.

In the battery system, the measured terminal quantities are the cell terminal voltage y and current u. Since the capacity affects the SOC transition which further affects the cell terminal voltage, the cell terminal voltage measurement y can be used to adapt the capacity by following the steps detailed above.

State EKF (State Estimation)

The state EKF essentially follows the same manner as the weight EKF. One difference lies in the fact that the time-update in the weight EKF employs the state transition function $\mathbf{F}(\bullet,\bullet,\bullet)$ as shown in Eq. (5.26). Similar to the weight EKF, the measurement-update in the state EKF also uses the differences between the predicted measurements in Eq. (5.32) and the real measurements to adapt the states $\mathbf{x}_{k,l}$. As shown in Eq. (5.28), the posteriori state estimates are obtained by correcting the priori state estimates with the prediction errors multiplied by gain factors.

When applied to the battery system, the state EKF aims at estimating the SOC x based on the measured cell terminal voltage y and current u. Since the SOC directly affects the cell terminal voltage through the cell dynamic model $G(\bullet,\bullet,\bullet)$, the cell terminal voltage measurement, as the model output, can be used to back-estimate the SOC, as one model input, by following the steps detailed above.

Numerical Implementation: Recurrent Derivative Computation

The dual EKF method, which adapts the states and parameters using two concurrently running EKFs, has a recurrent architecture associated with the computation of $\mathbf{C}_{k,l}^{\boldsymbol{\theta}}$ in the weight filter. The computation of $\mathbf{C}_{k,l}^{\boldsymbol{\theta}}$ involves a total derivative of the measurement function with respect to the parameters $\boldsymbol{\theta}$ as

$$\mathbf{C}_{k,l}^{\boldsymbol{\theta}} = \left.\frac{d\mathbf{G}\left(\hat{\mathbf{x}}_{k,l}^{-},\mathbf{u}_{k,l},\boldsymbol{\theta}\right)}{d\boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}_{k,l}^{-}}. \tag{5.33}$$

This computation requires a recurrent routine similar to a real-time recurrent learning [145]. Decomposing the total derivative into partial derivatives and propagating the states back in time results in the following recursive equations

$$\frac{d\mathbf{G}\left(\hat{\mathbf{x}}_{k,l}^{-},\mathbf{u}_{k,l},\boldsymbol{\theta}\right)}{d\boldsymbol{\theta}} = \frac{\partial\mathbf{G}\left(\hat{\mathbf{x}}_{k,l}^{-},\mathbf{u}_{k,l},\boldsymbol{\theta}\right)}{\partial\boldsymbol{\theta}} + \frac{\partial\mathbf{G}\left(\hat{\mathbf{x}}_{k,l}^{-},\mathbf{u}_{k,l},\boldsymbol{\theta}\right)}{\partial\hat{\mathbf{x}}_{k,l}^{-}}\frac{d\hat{\mathbf{x}}_{k,l}^{-}}{d\boldsymbol{\theta}}, \tag{5.34}$$

$$\frac{d\hat{\mathbf{x}}_{k,l}^{-}}{d\boldsymbol{\theta}} = \frac{\partial\mathbf{F}\left(\hat{\mathbf{x}}_{k,l-1},\mathbf{u}_{k,l-1},\boldsymbol{\theta}\right)}{\partial\boldsymbol{\theta}} + \frac{\partial\mathbf{F}\left(\hat{\mathbf{x}}_{k,l-1},\mathbf{u}_{k,l-1},\boldsymbol{\theta}\right)}{\partial\hat{\mathbf{x}}_{k,l-1}}\frac{d\hat{\mathbf{x}}_{k,l-1}}{d\boldsymbol{\theta}}, \tag{5.35}$$

$$\begin{aligned}\frac{d\hat{\mathbf{x}}_{k,l-1}}{d\boldsymbol{\theta}} &= \frac{d\hat{\mathbf{x}}_{k,l-1}^{-}}{d\boldsymbol{\theta}} - \mathbf{K}_{k,l-1}^{\mathbf{x}}\frac{d\mathbf{G}\left(\hat{\mathbf{x}}_{k,l-1}^{-},\mathbf{u}_{k,l-1},\boldsymbol{\theta}\right)}{d\boldsymbol{\theta}} \\ &\quad + \frac{\partial\mathbf{K}_{k,l-1}^{\mathbf{x}}}{\partial\boldsymbol{\theta}}\left[\mathbf{y}_{k,l-1} - \mathbf{G}\left(\hat{\mathbf{x}}_{k,l-1}^{-},\mathbf{u}_{k,l-1},\boldsymbol{\theta}\right)\right].\end{aligned} \tag{5.36}$$

The last term in Eq. (5.36) can be set to zero with the assumption that $\mathbf{K}_{k,l}^{\mathbf{x}}$ is not dependant on **θ**. Indeed, since $\mathbf{K}_{k,l}^{\mathbf{x}}$ is often very weakly dependant on **θ,** the extra computational effort to consider this dependence is not worth the improvement in performance. Therefore, we drop the last term in Eq. (5.36) in this study. Then the three total derivatives can be computed in a recursive manner with initial values set as zeros. It noted that the partial derivatives of the transition and measurement functions with respect to the states **x** and parameters **θ** can be easily computed with the explicitly given function forms.

5.3.3 A Multiscale Framework with Extended Kalman Filter

As discussed in Section 5.3.2, the dual EKF method estimates both the states and parameters on the same time-scale. However, for systems that exhibit the time-scale separation, it is natural and desirable to adapt the slowly time-varying parameters on the macro time-scale while keeping the estimation of the fast time-varying states on the micro time-scale. This multiscale framework is expected to reduce the computational effort and provide more stable estimates of model parameters. This section is dedicated to the discussion of this framework and is organized in a similar manner as Section 5.3.2: the first part of this section presents the numerical formulation of the multiscale framework with EKF and the numerical implementation of the recurrent derivative computation in the multiscale framework is described in the subsequent part.

Numerical Formulation: Multiscale Estimation

As opposed to the dual estimation, we intend to derive a multiscale estimation which allows for a time-scale separation in the state and parameter estimation. More specifically, we aim at estimating the slowly time-varying model parameters on the macro time-scale and, at the same time, intend to keep the estimation of fast time-varying states on the micro time-scale to utilize all the measurements. The algorithm of the multiscale framework for the system described in Eq. (5.23) is summarized in Table 5-12. Note that, in contrast to the dual EKF algorithms in Table 5-11, we only use the macro time-scale index k to present the macro EKF since the parameter estimation is performed only every macro time step.

The algorithm is initialized by setting the model parameters $\boldsymbol{\theta}$ and states $\mathbf{x}$ to the best guesses based on the prior information. The covariance matrices $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$ and $\boldsymbol{\Sigma}_{\mathbf{x}}$ of estimation errors are also initialized based on the prior information. The main algorithm essentially consists of the so-called micro and macro EKFs running on the micro and macro time-scales, respectively. Note that, the micro time-scale here refers to the time-scale on which system states exhibit fast variation while the macro time-scale refers to the one on which system model parameters tend to vary slowly. For example, in the battery system, the SOC, as a system state, changes every second, which suggests the micro time-scale to be approximately one second. In contrast, the cell capacity, as a system model parameter, typically decreases 1.0% or less in a month with regular use, resulting in the macro time-scale being approximately one day or so. The time- and measurement-updates performed in the macro EKF and micro EKF are detailed as follows.

Table 5-12 Algorithm of a multiscale framework with extended Kalman filter

---

Initialization

$$\begin{aligned}&\hat{\boldsymbol{\theta}}_0 = E\left[\hat{\boldsymbol{\theta}}_0\right],\ \boldsymbol{\Sigma}_{\boldsymbol{\theta}_{k,l}} = E\left[\left(\boldsymbol{\theta}_0 - \hat{\boldsymbol{\theta}}_0\right)\left(\boldsymbol{\theta}_0 - \hat{\boldsymbol{\theta}}_0\right)^{\mathrm{T}}\right],\\ &\hat{\mathbf{x}}_{0,0} = E\left[\mathbf{x}_{0,0}\right], \boldsymbol{\Sigma}_{\mathbf{x}_{k,l}} = E\left[\left(\mathbf{x}_{0,0} - \hat{\mathbf{x}}_{0,0}\right)\left(\mathbf{x}_{0,0} - \hat{\mathbf{x}}_{0,0}\right)^{\mathrm{T}}\right].\end{aligned} \tag{5.37}$$

For $k \in \{1, \ldots, \infty\}$, compute

Time-update equations for the macro EKF

$$\hat{\boldsymbol{\theta}}_k^- = \hat{\boldsymbol{\theta}}_{k-1}, \qquad \boldsymbol{\Sigma}_{\boldsymbol{\theta}_k}^- = \boldsymbol{\Sigma}_{\boldsymbol{\theta}_{k-1}} + \boldsymbol{\Sigma}_{\mathbf{r}_{k-1}}. \tag{5.38}$$

State projection equation for the macro EKF

$$\tilde{\mathbf{x}}_{k-1,L} = \mathbf{F}_{0\to L}\left(\hat{\mathbf{x}}_{k-1,0}, \mathbf{u}_{k-1,0:L-1}, \hat{\boldsymbol{\theta}}_k^-\right). \tag{5.39}$$

Measurement-update equations for the macro EKF

$$\mathbf{K}_k^{\boldsymbol{\theta}} = \boldsymbol{\Sigma}_{\boldsymbol{\theta}_k}^- \left(\mathbf{C}_k^{\boldsymbol{\theta}}\right)^{\mathrm{T}} \left[\mathbf{C}_k^{\boldsymbol{\theta}} \boldsymbol{\Sigma}_{\boldsymbol{\theta}_k}^- \left(\mathbf{C}_k^{\boldsymbol{\theta}}\right)^{\mathrm{T}} + \boldsymbol{\Sigma}_{\mathbf{n}_k}\right]^{-1}. \tag{5.40}$$

$$\hat{\boldsymbol{\theta}}_k = \hat{\boldsymbol{\theta}}_k^- + \mathbf{K}_k^{\boldsymbol{\theta}}\left[\hat{\mathbf{x}}_{k-1,L} - \tilde{\mathbf{x}}_{k-1,L}\right],\ \boldsymbol{\Sigma}_{\boldsymbol{\theta}_k} = \left(\mathbf{I} - \mathbf{K}_k^{\boldsymbol{\theta}}\mathbf{C}_k^{\boldsymbol{\theta}}\right)\boldsymbol{\Sigma}_{\boldsymbol{\theta}_k}^-. \tag{5.41}$$

For $l \in \{1, \ldots, L\}$, compute

Time-update equations for the micro EKF

$$\hat{\mathbf{x}}_{k,l}^- = \mathbf{F}\left(\hat{\mathbf{x}}_{k,l-1}, \mathbf{u}_{k,l-1}, \hat{\boldsymbol{\theta}}_{k-1}\right), \qquad \boldsymbol{\Sigma}_{\mathbf{x}_{k,l}}^- = \mathbf{A}_{k,l-1}\boldsymbol{\Sigma}_{\mathbf{x}_{k,l-1}}\mathbf{A}_{k,l-1}^{\mathrm{T}} + \boldsymbol{\Sigma}_{\mathbf{w}_{k,l-1}}. \tag{5.42}$$

Measurement-update equations for the micro EKF

$$\mathbf{K}_{k,l}^{\mathbf{x}} = \boldsymbol{\Sigma}_{\mathbf{x}_{k,l}}^- \left(\mathbf{C}_{k,l}^{\mathbf{x}}\right)^{\mathrm{T}} \left[\mathbf{C}_{k,l}^{\mathbf{x}} \boldsymbol{\Sigma}_{\mathbf{x}_{k,l}}^- \left(\mathbf{C}_{k,l}^{\mathbf{x}}\right)^{\mathrm{T}} + \boldsymbol{\Sigma}_{\mathbf{v}_{k,l}}\right]^{-1}. \tag{5.43}$$

$$\hat{\mathbf{x}}_{k,l} = \hat{\mathbf{x}}_{k,l}^- + \mathbf{K}_{k,l}^{\mathbf{x}}\left[\mathbf{y}_{k,l} - \mathbf{G}\left(\hat{\mathbf{x}}_{k,l}^-, \mathbf{u}_{k,l}, \hat{\boldsymbol{\theta}}_{k-1}\right)\right], \qquad \boldsymbol{\Sigma}_{\mathbf{x}_{k,l}} = \left(\mathbf{I} - \mathbf{K}_{k,l}^{\mathbf{x}}\mathbf{C}_{k,l}^{\mathbf{x}}\right)\boldsymbol{\Sigma}_{\mathbf{x}_{k,l}}^-. \tag{5.44}$$

where

$$\begin{aligned}&\mathbf{A}_{k,l-1} = \left.\frac{\partial \mathbf{F}\left(\mathbf{x}, \mathbf{u}_{k,l-1}, \hat{\boldsymbol{\theta}}_{k-1}\right)}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_{k,l-1}},\ \mathbf{C}_{k,l}^{\mathbf{x}} = \left.\frac{\partial \mathbf{G}\left(\mathbf{x}, \mathbf{u}_{k,l}, \hat{\boldsymbol{\theta}}_{k-1}\right)}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_{k,l}^-},\\ &\mathbf{C}_k^{\boldsymbol{\theta}} = \left.\frac{d\mathbf{F}_{0\to L}\left(\hat{\mathbf{x}}_{k-1,0}, \mathbf{u}_{k-1,0:L-1}, \boldsymbol{\theta}\right)}{d\boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}_k^-}.\end{aligned} \tag{5.45}$$

Macro EKF (Parameter Estimation)

At every macro time step, the macro EKF executes the time-update where prior parameter estimates $\hat{\boldsymbol{\theta}}_k^-$ and their error covariance $\boldsymbol{\Sigma}_{\boldsymbol{\theta}_k}^-$ are computed with Eq. (5.38). The addition of unpredictable process noise $\mathbf{r}_k$ increases the uncertainties $\boldsymbol{\Sigma}_{\boldsymbol{\theta}_k}^-$ in the parameter estimates. After the time-update step, the state projection is conducted to project the state estimates from the micro EKF through the macro time step, expressed as the state projection function $\mathbf{F}_{0\to L}(\bullet,\bullet,\bullet)$ in Eq. (5.39). We note that $\mathbf{F}_{0\to L}(\bullet,\bullet,\bullet)$ can be expressed as a nested form of the state transition function $\mathbf{F}(\bullet,\bullet,\bullet)$ in Eq. (5.23) and that the computational effort involved in computing $\mathbf{F}_{0\to L}(\bullet,\bullet,\bullet)$ is negligible compared to the time- and measurement-updates conducted in L micro time steps. In the measurement-update step, the macro EKF computes the difference between the projected states and the estimated states from the micro EKF and uses the difference to obtain posterior parameter estimates, which is detailed in Eq. (5.41).

Compared with the weight EKF, the macro EKF possesses two distinctive characteristics: (i) the time- and measurement-updates are performed over the macro time-scale ($L\cdot T$) instead of the micro time-scale (T), leading to the possibility to greatly reduce the computational complexity; and (ii) the macro EKF uses the state estimates from the micro EKF for the measurement-update, enabled by the state projection in Eq. (5.39), and the resulting parameter estimation becomes decoupled with the state estimation where the real measurements in Eq. (5.23) are used for the measurement-update. The first characteristic could magnify the effect of the parameters on the states, i.e., that the parameters could affect the states projected on the macro time-scale ($L\cdot T$) more significantly than those projected on the micro time-scale (T). The second characteristic helps distinguish the effects of the two unknowns (states and parameters) on the only measurements. In the subsequent section, these characteristics will be further explained and verified when we apply the algorithm to the battery system.

Micro EKF (State Estimation)

The micro EKF bears a strong resemblance to the state EKF in the dual EKF. The only difference is that, for the state transition, the micro EKF uses the capacity estimate from the previous macro time step (see Eq. (5.42)) while the state EKF employs that the previous micro time step (see Eq. (5.26)). It is important to note that, at the start of every macro time step, i.e., at the time $t_{k-1,0}$, the micro EKF sends the state estimate to the macro EKF which then projects it through the macro time step according to the state projection equation in Eq. (5.39). Upon the completion of the state projection at the end of every macro time step, i.e., at the time $t_{k-1,L}$, the micro EKF sends another state estimate to the macro EKF which then compares it with the projected estimate and uses the difference to adapt the parameter estimate in the measurement-update step detailed in Eq. (5.41).

Numerical Implementation: Recurrent Derivative Computation

In the multiscale framework, the computation of $\mathbf{C}_k^{\boldsymbol{\theta}}$ in the macro EKF involves a total derivative of the state projection function with respect to the parameters $\boldsymbol{\theta}$ as

$$\mathbf{C}_k^{\boldsymbol{\theta}} = \left. \frac{d\mathbf{F}_{0\to L}\left(\hat{\mathbf{x}}_{k-1,0}, \mathbf{u}_{k-1,0:L-1}, \boldsymbol{\theta}\right)}{d\boldsymbol{\theta}} \right|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}_k^-} . \tag{5.46}$$

Similar to the total derivative in Eq. (5.33), this computation also requires a recurrent routine. Decomposing the total derivative into partial derivatives, we then obtain the following equation

$$\begin{aligned}\frac{d\mathbf{F}_{0\to L}\left(\hat{\mathbf{x}}_{k-1,0},\mathbf{u}_{k-1,0:L-1},\boldsymbol{\theta}\right)}{d\boldsymbol{\theta}}=&\frac{\partial\mathbf{F}_{0\to L}\left(\hat{\mathbf{x}}_{k-1,0},\mathbf{u}_{k-1,0:L-1},\boldsymbol{\theta}\right)}{\partial\boldsymbol{\theta}}\\&+\frac{\partial\mathbf{F}_{0\to L}\left(\hat{\mathbf{x}}_{k-1,0},\mathbf{u}_{k-1,0:L-1},\boldsymbol{\theta}\right)}{\partial\hat{\mathbf{x}}_{k-1,0}}\frac{d\hat{\mathbf{x}}_{k-1,0}}{d\boldsymbol{\theta}}.\end{aligned}\tag{5.47}$$

The total derivative in the last term can be obtained by using the recursive equations Eqs. (5.34)-(5.36). The partial derivatives of the state projection function with respect to the states $\mathbf{x}$ and parameters $\boldsymbol{\theta}$ can be easily computed with the explicitly given function forms.

5.3.4 Application to Li-Ion Battery System

In this section, we use the proposed framework to estimate the SOC and capacity in a Li-ion battery system. When applied to the battery system, the multiscale framework can be treated as a hybrid of coulomb counting and adaptive filtering techniques and comprises two new ideas: (i) a multiscale framework to estimate SOC and capacity that exhibit time-scale separation and (ii) a state projection scheme for accurate and stable capacity estimation. The first part of this section presents the discrete-time cell dynamic model used in this study. The second part presents the multiscale estimation of SOC and capacity in the battery system.

Discrete-Time Cell Dynamic Model

In order to execute the time-update in the micro and macro EKFs, we need a state transition model that propagate the SOC forward in time. In order to execute the measurement-update in the micro-EKF, we need a "discrete-time cell dynamic model" that relates the SOC to the cell voltage. Here we employ the enhanced self-correcting (ESC) model which considers the effects of open circuit voltage (OCV), internal resistance, voltage time constants and hysteresis [75]. The effects of voltage time constants and hysteresis in the ESC model can be expressed as [75]

$$\begin{aligned} \begin{bmatrix} f_{k,l+1} \\ h_{k,l+1} \end{bmatrix} &= \begin{bmatrix} \mathrm{diag}(\boldsymbol{\alpha}) & 0 \\ 0 & \varphi(i_{k,l+1}) \end{bmatrix} \begin{bmatrix} f_{k,l} \\ h_{k,l} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1-\varphi(i_{k,l+1}) \end{bmatrix} \begin{bmatrix} i_{k,l} \\ M(x,\dot{x}) \end{bmatrix}, \\ \varphi(i_{k,l+1}) &= \exp\left(-\left| \frac{\eta_i \cdot i_{k,l} \cdot \gamma \cdot T}{C_k} \right| \right). \end{aligned} \tag{5.48}$$

where x is the SOC, f the filter state, h the hysteresis voltage, $\boldsymbol{\alpha}$ the vector of filter pole locations, γ the hysteresis rate constant, i the current, $M(\cdot,\ \cdot)$ maximum hysteresis, η_i the coulombic efficiency, T the length of measurement interval, C the nominal capacity. The coulombic efficiency of a battery cell is defined as the ratio of the amount of charge that is stored in the cell during charging compared to the amount that can be extracted from the cell during discharging. We then obtain the state transition and measurement equations as

$$\begin{aligned} x_{k,l+1} &= F\left(x_{k,l}, i_{k,l}, C_k\right) = x_{k,l} - \frac{\eta_i \cdot T \cdot i_{k,l}}{C_k}, \\ y_{k,l+1} &= G\left(x_{k,l}, i_{k,l}, C_k\right) = \mathrm{OCV}\left(z_k\right) - i_{k,l} \cdot R + h_{k,l+1} + S \cdot f_{k,l+1}. \end{aligned} \tag{5.49}$$

where OCV is the open circuit voltage, y_k the predicted cell terminal voltage, R the cell resistance, S a vector of constants that blend the time constant states together in the output.

Multiscale Estimation of SOC and Capacity

We then begin to introduce the multiscale framework with EKF for the Li-ion battery system by drawing a flowchart in Figure 5-18, where T is a fixed time step between two adjacent measurement points, $x_{k,l}$ is the SOC estimate at the time $t_{k,l} = t_{k,0} + l \cdot T$., for $1 \le l \le L$ (k and l are the indices of macro and micro time-scales, respectively), y and i are the cell voltage measurement and the cell current measurement (equivalent to u used before), respectively, and C is the cell capacity estimate (equivalent to θ used before).

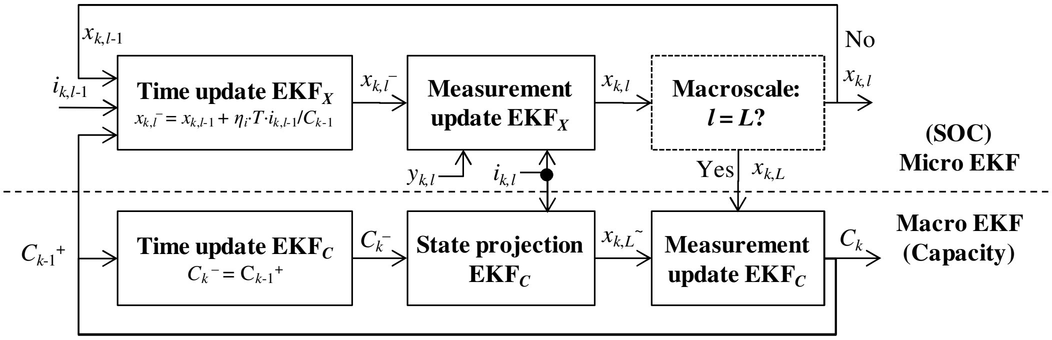


Figure 5-18 Flowchart of a multiscale framework with EKF for battery SOC and capacity estimation

The framework consists of two EKFs running in parallel: the top one (micro EKF) adapting the SOC in the micro time-scale and the bottom one (macro EKF) adapting the capacity in the macro time-scale. The micro EKF sends the SOC estimate to the macro EKF and receives the capacity estimate from the macro EKF. In what follows, we intend to elaborate on the macro EKF, the key technical component of the multiscale framework, which consists of the following recursively executed procedures (see Figure 5-19):

Step 1: At the macro time step k, the capacity transition step, also referred to as the time-update step, computes the expected capacity and its variance based on the updated estimates at the time step $k - 1$, expressed as

$$C_k^- = C_{k-1}^+, \quad \Sigma_{C_k}^- = \Sigma_{C_{k-1}}^+ + \Sigma_{r_{k-1}}. \tag{5.50}$$

For a stable system, the capacity variance term $\Sigma_{C_{k-1}}^{+}$ tends to decrease over time with the measurement-update (see *Step 3* in Figure 5-19). However, the process noise term $\Sigma_{r_{k-1}}$ always increases the uncertainty of the capacity estimate. To clearly illustrate the proposed idea, we intend to classify the capacity estimates into three cases (see Figure 5-19): a larger estimate $C_{k-1}^{(L)}$, an accurate estimate $C_{k-1}^{(N)}$, and a smaller estimate $C_{k-1}^{(S)}$.

Step 2: Based on the capacity estimate C_k^{-}, the state projection scheme projects the SOC through the macro time step, expressed as a state projection equation derived from Eqs. (5.39) and (5.49)

$$x_{k,L} = x_{k,0} + \frac{\eta \cdot T}{C_k^{-}} \cdot \sum_{j=0}^{L-1} i_{k,j}. \tag{5.51}$$

As can be seen in Figure 5-19, the projected SOCs exhibit large deviations from their true values (obtained from the micro EKF), which suggests a magnified effect of the capacity on the SOC.

Step 3: Following the state projection step, the difference between the projected SOC and the estimated SOC by the micro EKF is used to update the capacity estimate, known as the measurement-update. It is noted that the measurement-update requires accurate SOC estimates which can be obtained from the micro EKF. The updated capacity estimate equals the predicted capacity estimate in *Step 1* plus a correction factor, expressed as

$$C_k^{+} = C_k^{-} + K_k^{C}\left[\hat{\mathbf{x}}_{k,L} - \tilde{\mathbf{x}}_{k,L}\right], \quad \Sigma_{C_k}^{+} = \left(1 - K_k^{C} C_k^{C}\right)\Sigma_{C_k}^{-}. \tag{5.52}$$

where the Kalman gain K_k^{C} and the total derivative C_k^{C} can be estimated using Eqs. (5.40) and (5.45), respectively.

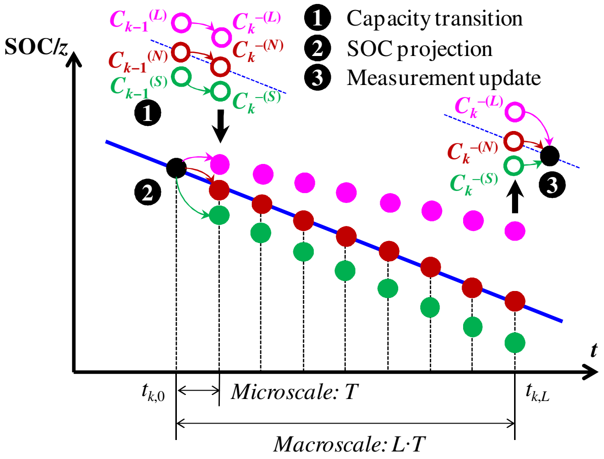


Figure 5-19 Procedures of capacity estimation in macro EKF

Remarks on Mutiscale Framework

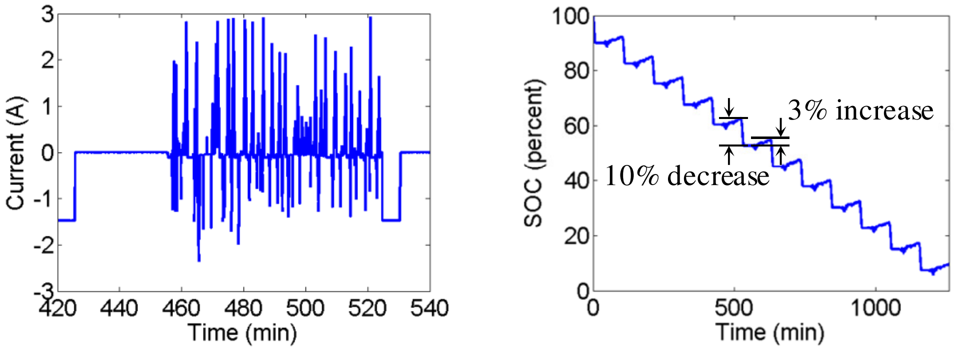
We note that the proposed framework decouples the SOC and capacity estimation in terms of both the measurement and time-scale, with an aim to avoid the concurrent SOC and capacity estimation relying on the only measurement (cell terminal voltage) in the dual EKF [76]. In fact, the very motivation of this work lies in the fact that the coupled estimation in the dual EKF falls short in the way of achieving stable capacity estimation, precisely because it is difficult to distinguish the effects of two states (SOC and capacity) on the only measurement (cell terminal voltage), especially in the case of the micro time-scale where the capacity only has a very small influence on the SOC. Regarding the measurement decoupling, the multiscale framework uses the cell terminal voltage exclusively as the measurement to adapt the SOC (micro EKF) which in turn serves as the measurement to adapt the capacity (macro EKF). Regarding the time-scale decoupling, the state projection using the coulomb counting in Eq. (5.51) significantly magnifies the effect of the capacity on the SOC, i.e., that the capacity affects the SOC projected on the macro time-scale ($L \cdot T$) more significantly than that projected on the micro time-scale (T). The larger influence of the capacity on the SOC leads to the possibility of more stable capacity estimation, and that is precisely the main technical characteristic that distinguishes our approach from the dual EKF.

5.3.5 Experimental Results

The verification of the proposed multiscale framework was accomplished by conducting an extensive urban dynamometer drive schedule (UDDS) test. This section reports the UDDS test results of five Li-ion prismatic cells.

Description of Test Procedure

In addition to the numerical study using synthetic data, we also conducted the UDDS cycle test to verify the effectiveness of the multiscale framework. The cells used in the test are Li-ion prismatic cells with a nominal capacity of 1.5Ah. Since the cell cannot withstand the high current pulse on a typical HEV cell, the UDDS profile was scaled down to within the rate range of ±2C. The scaled UDDS cycle was replicated two times to obtain the final UDDS cycle used in this test (see Figure 5-20a). It is noted that, in a battery system, we often use C or C-rate to measure the rate at which a cell is charged or discharged relative to its full capacity. For the Li-ion prismatic cell with the capacity of 1.5Ah, a discharge current with a 1C rate (1.5A) will discharge the full cell capacity in 1 hour. Here, 2C or a 2C rate (3.0A) defines the upper and lower bounds (±3.0A) of the scaled UDDS profile (see Figure 5-20a). The cycle test is composed of 12 UDDS cycles, separated by 1C constant current discharge for 6min and 30min rest. This test profile resulted in the spread of SOC over the 100%-4% range. The SOC profile for 12 UDDS cycles is plotted in Figure 5-20b, where the cell experiences an SOC increase by about 3% during each UDDS cycle, and an SOC decrease by about 10% due to the 1C discharge between cycles. The discharge setting (1C for 6min) was designed in order to excite the entire SOC range (100%-4%) for the UDDS cycle test as well as to practice the UDDS cycle test at many different SOC levels separated by a small gap (about 7%).



(a) (b)

Figure 5-20 SOC profile and one cycle rate profile for UDDS cycle test. Figure (a) plots the rate profile for one UDDS cycle and (b) plots the SOC profile for 12 UDDS cycles.

We set up a UDDS test system (see Figure 5-21) which comprises of an Arbin BT2000 cycle tester with a data acquisition device, an Espec SH-241 temperature chamber at 25°C and a test jig as a connector holder for prismatic cells. Five prismatic cells were placed in the temperature chamber and held by the test jig throughout the test.

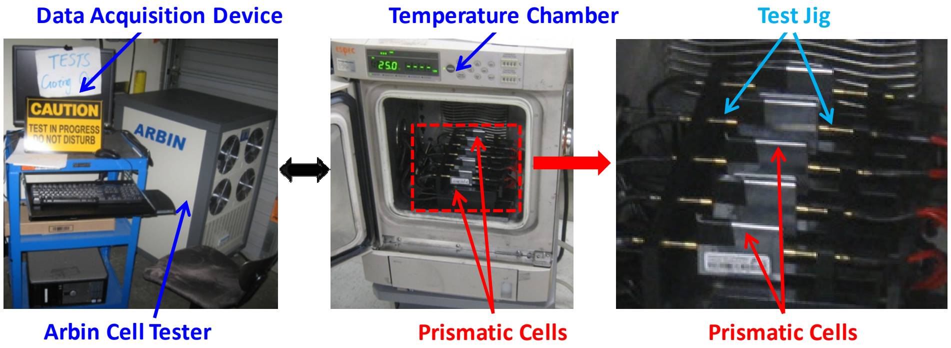


Figure 5-21 Experiment setup – UDDS test system.

Training of ESC Cell Model

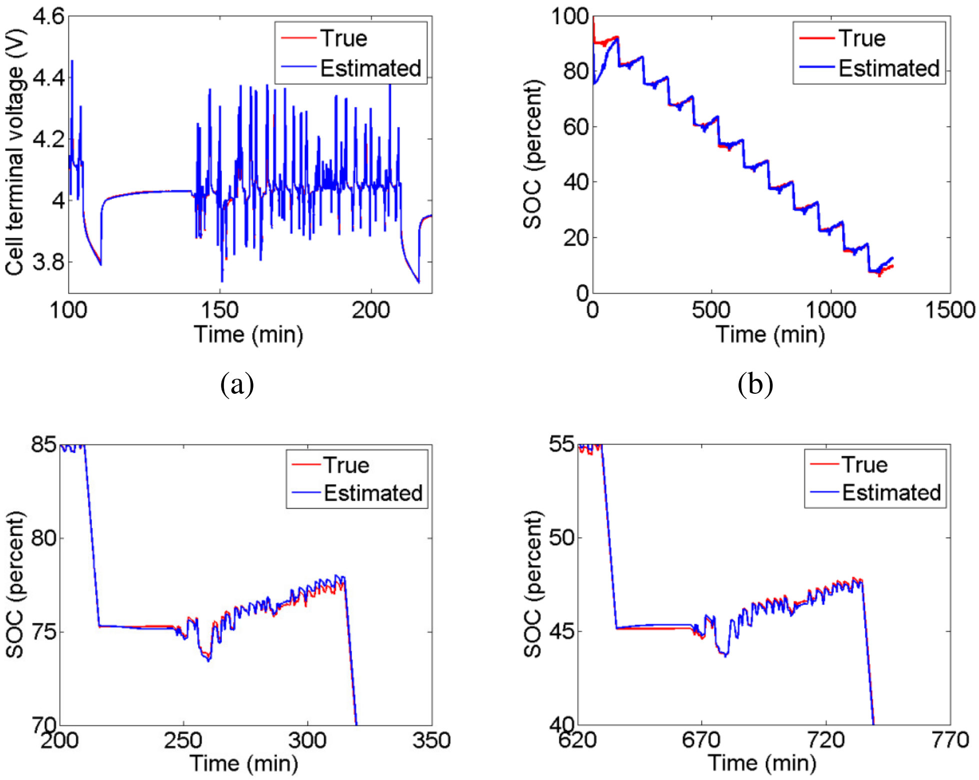
The current and voltage measurements of one cell (cell 1) were used to train the ESC model [75] while the other four cells (cells 2-5) were treated as the testing cells. We followed the procedures described in [146] to obtain the open circuit voltage (OCV) curve. Through numerical optimization, optimum ESC model parameters were obtained which minimize the root mean squared (RMS) error of the cell terminal voltage. The numerical optimization was performed using with a sequential quadratic programming (SQP) method. In this study, we employed a nominal capacity of 1.5Ah, a measurement interval of $T \approx$ 1s with "≈" indicating small measurement-to-measurement fluctuation, and four filter states $n_f = 4$. The voltage modeling results for one UDDS cycle are shown in Figure 5-22a, where a good agreement can be observed between the modeled and measured cell terminal voltage. The RMS error of voltage modeling for 12 UDDS cycles was 13.3mV.

SOC and Capacity Estimation Results

The SOC estimation results for the training cell for all 12 UDDS cycles, the 3rd UDDS cycle and the 7th UDDS cycle are shown in Figure 5-22b, Figure 5-22c and Figure 5-22d, respectively, where the initial SOC is set to be smaller (90%) than the true SOC (100%) and the multiscale framework (L = 1200) still produced converged SOC estimate. Table 5-13 summarizes the SOC estimation errors under four different settings of the initial SOC and capacity. Here, the RMS and maximum errors take into account the initial offset in the case of an incorrect initial SOC and are formulated as

$$\varepsilon_{RMS} = \sqrt{\frac{1}{nm}\sum_{k,j}\left(\hat{x}_{k,l} - x_{k,l}\right)^2}, \quad \varepsilon_{Max} = \max_{k,j}\left|\hat{x}_{k,l} - x_{k,l}\right|. \qquad (5.53)$$

where *nm* is the number of measurements and reads 74,484 (about 1250mins) in this study; and $x_{k,l}$ is the true SOC at the time $t_{k,l}$. In this study, we computed the true SOCs based on the coulomb counting technique. We attempted to achieve an accurate approximation to the true SOCs through the following ways: (i) we first fully charge a battery cell with a constant-current (0.1A constant current up to 4.2V) constant-voltage (4.2V constant voltage down to 0.01A) strategy to ensure an accurate initial SOC (100%); (ii) we measure the cell current with the Arbin current sensors whose high measurement accuracy leads to a sufficiently small error in the current accumulation over a relatively short test duration (around 20hours).



(a)

(b)

(c)

(d)

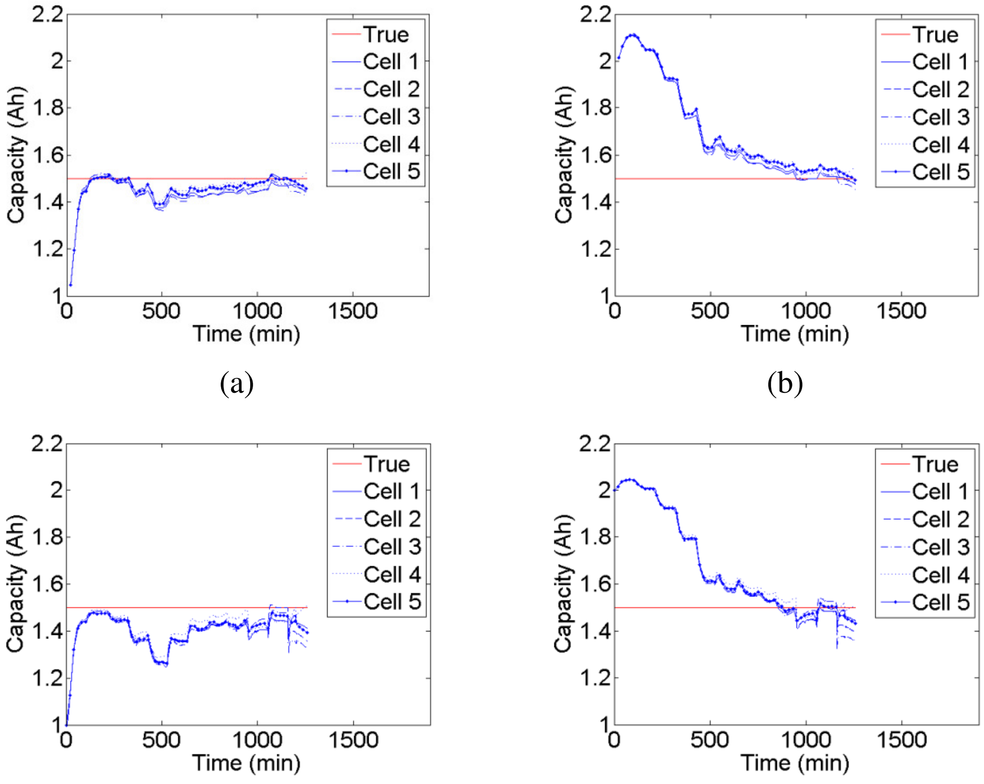
Figure 5-22 Voltage modeling results and SOC estimation results for UDDS cycle test. Figure (a) plots modeled and measured cell terminal voltage for one UDDS cycle; (b), (c) and (d) plot the estimated and true SOCs for all 12 UDDS cycles, the 3rd UDDS cycle and the 7th UDDS cycle, respectively.

Table 5-13 SOC estimation results under different settings of initial SOC and capacity

Initial SOC	Initial capacity	SOC errors	Cell 1	Cell 2	Cell 3	Cell 4	Cell 5
Correct (100%)	Correct (1.5Ah)	RMS (%)	1.02	1.34	0.81	1.05	0.75
		Max (%)	2.19	2.37	1.82	2.79	1.91
	Incorrect (1.0Ah)	RMS (%)	1.31	1.59	1.10	1.39	1.14
		Max (%)	4.84	4.84	4.84	4.85	4.84
Incorrect (90%)	Correct (1.5Ah)	RMS (%)	1.91	2.07	1.84	2.03	1.8
		Max (%)	10.00	10.00	10.00	10.00	10.00
	Incorrect (1.0Ah)	RMS (%)	2.65	2.77	2.59	2.76	2.58
		Max (%)	14.74	14.75	14.74	14.76	14.75

Three important observations can be made from the results. First of all, it is observed that the RMS SOC estimation errors produced by the multiscale framework are less than 3.00%, regardless of initial values of the SOC and capacity. Secondly, under both initial SOC settings, the SOC estimation errors with incorrect initial capacity (1.0Ah) are consistently larger than those with correct initial capacity (1.5Ahs). These results suggest that the SOC is strongly dependant on the capacity and that the lack of accuracy in the capacity estimation may reduce the accuracy in the SOC estimation. It is thus important to produce accurate capacity estimation not only to provide insights into the cell SOH but also to enable accurate SOC estimation. Thirdly, under both initial capacity settings, the SOC estimation with incorrect initial SOC (90%) consistently shows larger errors than that with correct initial SOC (100%). Clearly, the larger SOC errors under incorrect initial SOC (90%) can be attributed to the larger errors before the convergence of SOC estimation (at the initial stage). After the convergence, the SOC errors under different initial SOCs become almost the same. We note that the RMS SOC estimation errors with incorrect initial SOC (90%) are still less than 3.00% since the multiscale framework produced converged SOC estimate for both cases.

Regarding the capacity estimation, both methods with initial values smaller than the true value (see Figure 5-23a and Figure 5-23c) and larger than the real value (see Figure 5-23b and Figure 5-23d) for all the five cells achieves convergence to the true capacity within an error range of around 5%. Compared with the capacity estimation (see Figure 5-23a and Figure 5-23b) by the multiscale framework, the capacity estimation (see Figure 5-23c and Figure 5-23d) by the dual EKF contains larger noise. The poorer accuracy produced by the dual EKF (consisting of an SOC EKF and a capacity EKF) can be attributed to the measurement and time-scale coupling in the SOC and capacity estimation. Regarding the measurement coupling, the dual EKF uses the cell terminal voltage as the measurement to adapt both the SOC and capacity. When the voltage modeling contains relatively large errors, the capacity estimation can be largely compromised by the measurement update (in the capacity EKF) which only aims at minimizing the difference between the modeled and measured voltages. In other words, the measurement update may give an incorrect capacity estimate to counteract the voltage modeling error. In this experimental study, the relatively large voltage modeling errors directly affect the accuracy in the capacity estimation by the dual EKF. Regarding the time-scale coupling, the dual EKF estimates both the SOC and capacity on the micro time-scale, which makes the capacity estimation vulnerable to the local voltage modeling error (on the micro time-scale). This vulnerability further leads to noisy capacity estimation. In contrast, the multiscale framework decouples the SOC and capacity estimation in terms of both the measurement and time-scale and avoids the concurrent SOC and capacity estimation relying on the only measurement (cell terminal voltage). The decoupling enables accurate capacity estimation in spite of SOC estimation error.



(a)

(b)

(c)

(d)

Figure 5-23 Capacity estimation results for UDDS cycle test. Figures (a) and (b) plot capacity estimation results by the multiscale framework with the initial values smaller than and larger than the true value, respectively; (c) and (d) plots capacity estimation results by the dual EKF with the initial values smaller than and larger than the true value, respectively.

To quantify the accuracy of both methods, we computed average RMS errors after convergence (at t = 200mins and 1000mins for smaller and larger initials, respectively). For the smaller initial, the average RMS errors produced by the dual

EKF and the multiscale framework are 0.108Ah (relative error 7.227%) and 0.063Ah (relative error 4.200%), respectively. For the larger initial, the errors are 0.049Ah (relative error 3.233%) and 0.023Ah (relative error 1.533%). Finally, we note that, since the 12 UDDS cycle test on one cell can be treated as one unique cycle test, we do not expect large deviation from the current results (based on the small difference between capacity estimation results on different cells in Figure 5-23) if we conduct another 12 UDDS cycle test.

To investigate the long-term behavior of capacity estimation after convergence, we set the initial capacity value as the real value and executed the two methods over the 12 UDDS cycles for all the five cells. It is noted that this is virtually equivalent to adding another 12 UDDS cycles (after convergence of capacity estimates) for the cases of smaller and larger initial capacities (see Figure 5-23). As can be seen in Figure 5-24a and Figure 5-24b, both methods produced capacity estimates around the real value. Again, the capacity estimation (see Figure 5-24a) by the multiscale framework contains smaller noise than that (see Figure 5-24b) by the dual EKF. To quantify the accuracy of both methods, we computed average RMS errors over the entire time domain. The errors of the dual EKF and the multiscale framework read 0.099Ah (relative error 6.573%) and 0.059Ah (relative error 3.931%), respectively.

In addition to the accuracy comparison, we also compared the two methods in terms of computational efficiency. To minimize the effect of randomness, we executed both methods ten times with the test data obtained from each of the five cells. The mean computational time is summarized in Table 5-14, where we observe that the multiscale framework consumed less computational time than the dual EKF.

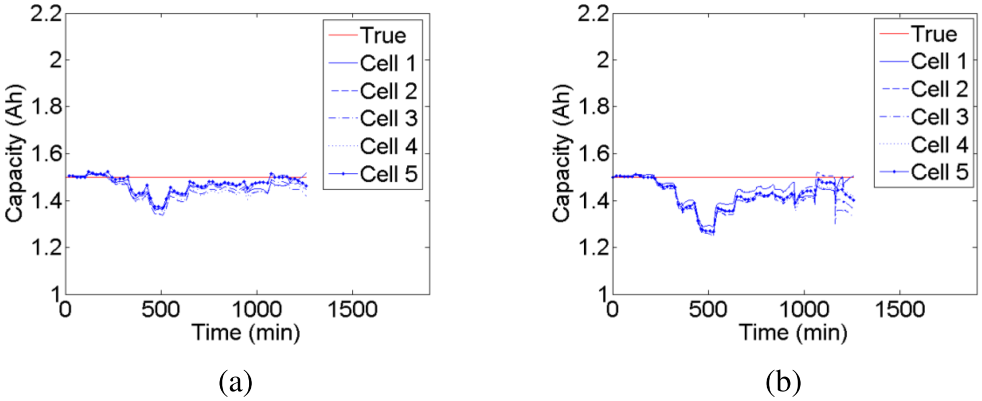


Figure 5-24 Capacity estimation results after convergence (by setting initial capacity as real value). Figures (a) and (b) plot capacity estimation results by the multiscale framework and by the dual EKF, respectively.

Table 5-14 Comparison results of computation efficiency with UDDS test data

Method	Computational time (s)	Improvement (%)
Dual EKF	5.813	---
Mutiscale Framework with EKF	3.711	36.163

5.3.6 Conclusion

The multiscale framework with EKF is proposed in this chapter for efficient and accurate state and parameter estimation for engineered systems that exhibit time-scale separation. The proposed framework was applied to the Li-ion battery system for SOC and capacity estimation. When applied to the battery system, the multiscale framework can be treated as a hybrid of coulomb counting and adaptive filtering techniques. Our contribution to battery SOC and capacity estimation lies in the construction of a multiscale computational scheme that decouples the SOC and capacity estimation from two perspectives, namely the measurement and time-scale. The resulting decoupled estimation greatly reduces the computational time involved in obtaining the SOC and capacity estimates, while enhancing the accuracy in the capacity estimation. It is noted that the higher efficiency makes the proposed methodology more suitable for onboard estimation devices that require computationally efficient estimation techniques. Experiments with the synthetic data and UDDS cycle test verify that the proposed framework achieves more accurate and efficient capacity estimation than the dual EKF, suggesting that the proposed framework is a promising methodology for the battery prognostics.

Chapter 6: System Resilience Analysis and Design

This chapter is dedicated to integrating system RBDO and PHM introduced in the previous chapters with an aim to develop a theoretical basis and design framework of engineering resilience in a mathematical and statistical context. A conceptual definition of a resilient engineered system is firstly instituted by introducing engineering resilience into a conventional engineered system, followed by the development of a mathematical definition of engineering resilience with a generic formula in terms of reliability and restoration. The mathematical definition of engineering resilience gives rise to a resilience-driven system design framework composed of three hierarchical design tasks, namely the resilience allocation problem (RAP) as a top-level design problem, and the system RBDO and PHM design as two bottom-level design problems. The proposal design framework is demonstrated with a simplified aircraft control actuator design problem.

6.1 Definition of Resilient Engineered Systems

Conventionally, an engineered system is composed of hardware, software, and human elements in a physical domain, which interact through a functional decomposition in a functional domain. This conventional system could fail catastrophically in the presence of adverse events (e.g., extreme weather, hardware fault, human error) because the system can neither respond nor adapt to the adverse events. There is thus a desperate need to build resilient engineered systems by introducing a pioneering feature, engineering resilience, into conventional engineered systems (see Figure 6-1).

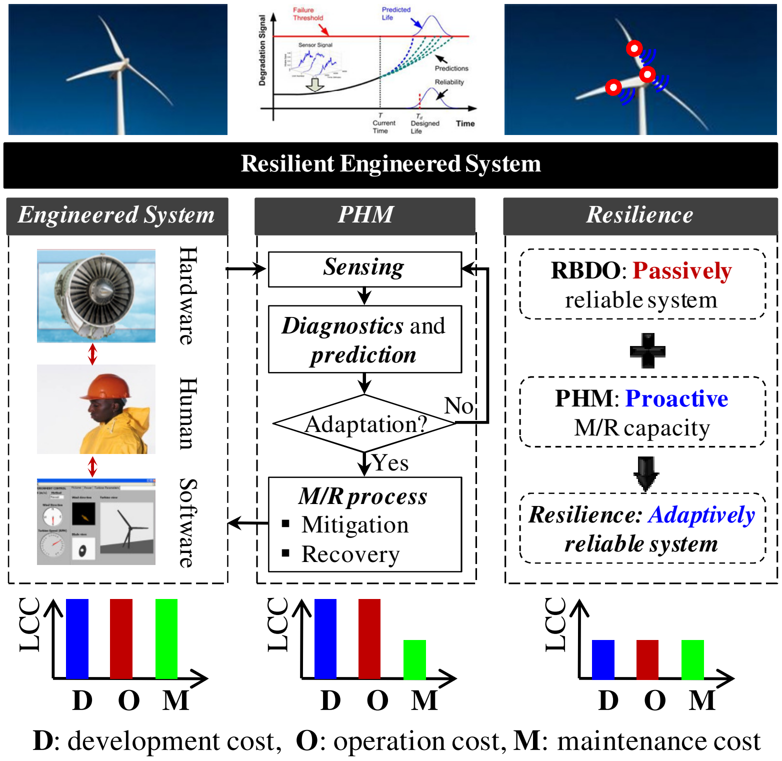


Figure 6-1 Resilience practice in a resilient engineered system.

We then investigate a conceptual definition of a complex engineered system having engineering resilience, characterized with three key functions including:

- *Sensing function*: It senses the effect of adverse events on engineered systems. The sensing function can be realized by employing an optimally designed sensor network.
- *Reasoning (diagnostics and prediction) function*: It extracts system health-relevant information in real-time with feature extraction techniques, classifies system health condition with health classification techniques, and predicts the time remaining before an engineered system no longer performs the required function(s) or the remaining useful life (RUL) in real-time with advanced machine learning techniques. The system health condition and RUL provide

valuable information for field engineers to make proactive mitigation/replacement (M/R) actions to prevent catastrophic system failure.

- *Mitigation or recovery (M/R) action process*: This process enables engineered systems to respond to and quickly recover from catastrophic system failures. It employs two types of actions, namely, mitigation and recovery. In general, the mitigation can be categorized as an M/R action for a short-term resilience while the replacement contributes to a long-term resilience.

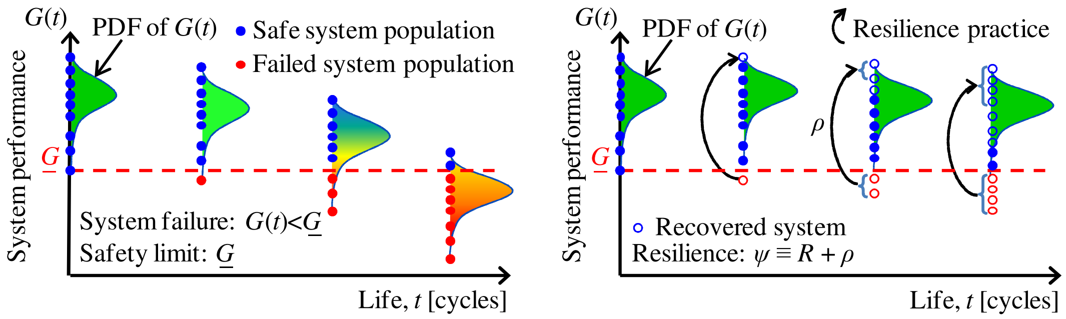
In what follows, the focus is to seek for a mathematically definition of engineering resilience, which then gives rise to a design framework of engineering resilience, namely resilience-driven system design.

6.2 Definition of Engineering Resilience

This section aims at proposing a conceptual definition of engineering resilience, which will facilitate the derivation of its generic formula in terms of reliability and other key PHM attributes. Non-resilient system designs encounter gradual degradation of system capacity and performance due to adverse events (see Figure 6-2a). In contrast, resilient system designs will be able to recover from their critical health states by restoring the system capacity (see Figure 6-2b). PHM will support logical decisions about when and how to restore the system capacity. The capacity restoration (ρ) can be defined as the degree of reliability recovery. It can be found that the restoration is a joint probability of a system failure event (E_{sf}), a correct diagnosis event (E_{cd}), a correct prognosis event (E_{cp}), and an M/R action success event (E_{mr}), expressed as

$$\begin{aligned}\rho\left(R,\Lambda_P,\Lambda_D,\kappa\right) &= \Pr\left(\mathrm{E}_{sf}\mathrm{E}_{cp}\mathrm{E}_{mr}\right) \\ &= \Pr\left(\mathrm{E}_{mr}|\mathrm{E}_{cp}\mathrm{E}_{cd}\mathrm{E}_{sf}\right)\cdot\Pr\left(\mathrm{E}_{cp}|\mathrm{E}_{cd}\mathrm{E}_{sf}\right)\cdot\Pr\left(\mathrm{E}_{cd}|\mathrm{E}_{sf}\right)\cdot\Pr\left(\mathrm{E}_{sf}\right) \\ &= \kappa\cdot\Lambda_P\cdot\Lambda_D\cdot\left(1-R\right)\end{aligned} \tag{6.1}$$

where κ, Λ_P and Λ_D are the conditional probabilities of the M/R action success, correct prognosis and diagnosis, and $(1-R)$ is the probability of system failure. In this study, the value of κ is held constant here by assuming that M/R maintenance actions are consistently performed. However, there is no restriction on the form of κ. In particular, κ can be a nonlinear function of the system reliability R, indicating that the performance of an M/R action is affected by the health condition of the engineered system. Since both Λ_P and Λ_D measure the efficiency for PHM (diagnostics and prognostics), we combine these terms as a PHM efficiency measure as: $\Lambda = \Lambda_P \cdot \Lambda_D$.



(a) (b)

Figure 6-2 System performance changes over lifetime without (a) and with the resilience practice (b).

The conceptual definition of engineering resilience is the degree of a passive survival rate (or reliability) plus a proactive survival rate (or restoration). Mathematically, the resilience measure can be defined as the addition of reliability and restoration as (see Figure 6-2b)

$$\begin{aligned}&\text{Resilience}\left(\Psi\right)=\text{Reliability}\left(R\right)+\text{Restoration}\left(\rho\right)\\&\rightarrow\Psi=R+\rho\left(R,\Lambda_P,\Lambda_D,\kappa\right)\end{aligned}\tag{6.2}$$

It is noted that the above definition turns engineering resilience to a quantifiable property, making it possible to analyze the resilience potential of an engineered system. In what follows, we intend to further elaborate on the two hallmarks of engineering resilience: reliability and restoration.

- *Reliability:* Reliability quantifies the ability of an engineered system to maintain its capacity and performance above a safety limit during a given period of time under stated conditions. Accordingly, resilience is characterized by preserving an acceptable level of capacity and performance despite adverse events. We note that reliability should be treated as an important system characteristic that contributes to engineering resilience from the perspective of system self-preservation. Conventional reliability-based design optimization (RBDO) practice endeavors to pursue high reliability with low cost through cost minimization under stringent reliability constraints [101-104].
- *Restoration*: Restoration measures the ability of an engineered system to restoring its capacity and performance by detecting, predicting and mitigating or recovering from the system-wide effects of adverse events. It can be viewed as the adaptability of an engineered system to its changing performance and capacity due to adverse events. This adaptability enables an adaptive reliability throughout the system's lifetime.

From the perspective of conventional reliability engineering, the failure of an engineered system typically refers to a breakdown or malfunctioning of the system and/or its components. This view assumes that success or failure is only an observable consequence that does not necessarily reflect the system's characteristic of adaptation. From the perspective of engineering resilience, however, success is due to the ability of an engineering system to make right adjustments in a timely manner, in particular to anticipate failures before they occur. Failure is due to the absence of that ability — either temporarily when the system performance falls below the safety limit or permanently. Reliability and restoration works in a highly cooperative manner to build the system's resilience. System design must encompass enhancing this cooperative strength, rather than just reducing the probabilities of failures. For this purpose, one should be able to enhance the reliability and restoration (or PHM efficiency levels) for given engineered system designs. This necessitates the developments of system RBDO (including reliability analysis) and PHM methodologies which have been discussed in details in the previous chapters. The framework of system design integrating system RBDO with PHM will be discussed in details in the next section.

6.3 Framework of Resilience-Driven System Design (RDSD)

We begin with an overview of the resilience-driven system design (RDSD) framework. This framework is composed of three hierarchical design tasks (see Figure 6-3), namely the resilience allocation problem (RAP) as a top-level design problem to define a resilience measure as a function of reliability and PHM efficiency in an engineering context, the system RBDO as a bottom-level design problem for the detailed design of components, the system PHM design as a bottom-level design problem for the detailed design of PHM units.

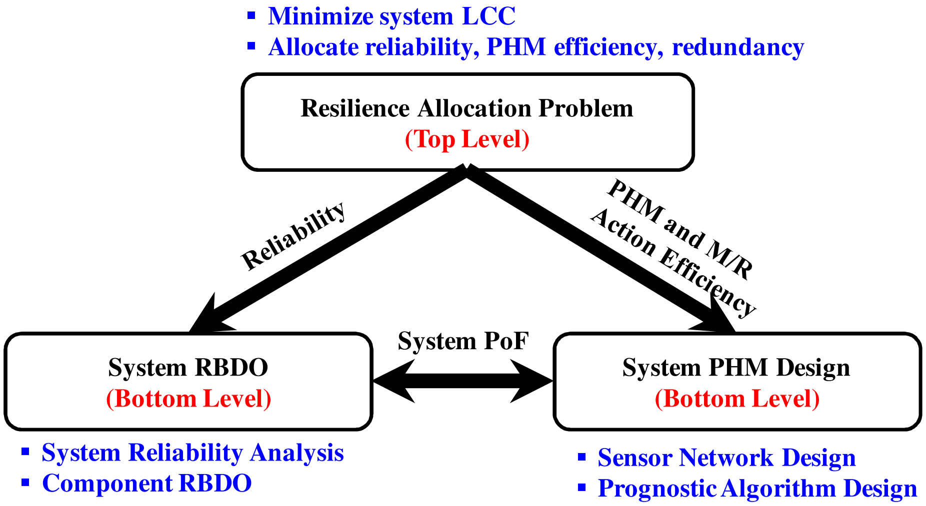


Figure 6-3 A hierarchical resilience-driven system design framework.

It is noted that, since optimal design of PHM units depends on failure mechanisms and layouts of components, PHM units must be designed upon the completion of component design. Therefore, we should first solve the system RBDO to determine an optimal component design while ensuring the optimally allocated reliability of the component and then conduct the system PHM design to determine an optimal PHM unit design for the optimal component design while meeting the optimally allocated PHM efficiency. It is natural that with reliability and PHM efficiency being the focuses of the system RBDO and system PHM design, respectively, one should be able to predict the reliability and PHM efficiency levels for given components and PHM unit designs. This necessitates the developments of system reliability analysis and PHM analysis for these two bottom-level design problems which then completes the framework.

6.4 Aircraft Control Actuator Case Study

This section presents a case study for the design of a simplified aircraft control actuator. The aircraft control actuator considered is the electro-hydrostatic actuator (EHA). In this case study, we aim at demonstrating the RDSD framework by designing a highly resilient EHA with optimized reliability, PHM efficiency and redundancy. Hypothetical data will be used for demonstration purposes.

The EHA (see Figure 6-4), as a closed-loop, hydrostatic control system, mainly consists of an electronic control unit (ECU), a variable-speed electric motor (EM), a fixed-displacement hydraulic pump and a hydraulic piston actuator [148]. In the EHA, a variable-speed electric motor (typically DC) is used to drive a fixed-displacement hydraulic pump, which in turn, powers a hydraulic piston actuator. Compared to a conventional hydraulic actuator, the EHA can achieve higher energy efficiency (with on-demand usage) and positional accuracy with enhanced compactness. These advantages have led to the wide use of the EHA for flight surface actuation in today's commercial and military aircrafts. Failures of the EHAs in these safety critical applications can be catastrophic, resulting in great loss of lives. Therefore, the EHA must be designed to achieve a sufficiently high reliability level. To this end, a common practice is to introduce a great deal of redundancy into the EHA (e.g., a triplex-redundant flight control system [149]). While a high redundancy level improves reliability, it results in a strikingly high life-cycle cost (LCC) to be incurred in development, operation, and maintenance processes. To reduce the LCC while still maintaining an equivalent reliability level, we apply the proposed RDSD framework to the EHA with an aim to compensate the redundancy reduction with the

PHM technology. It is worth noting that the RDSD framework leads to the possibility to implement this compensation in an optimum manner.

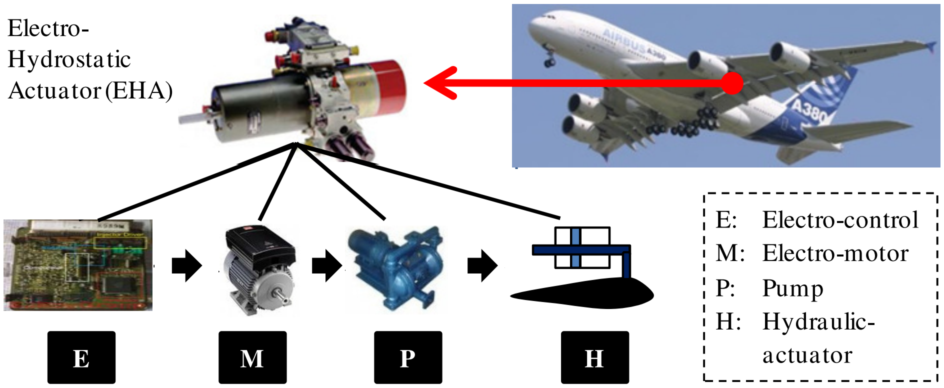


Figure 6-4 An airplane control actuator with series-connected subsystems

6.4.1 Top-Level Resilience Allocation Problem (RAP)

This subsection aims at demonstrating the top-level RAP in the RDSD framework by allocating a target system resilience into the target component-reliabilities, component-PHM efficiencies and component-redundancies of the four subsystems.

RAP Formulation

Solving the top-level RAP will allocate a target system resilience level into the target resilience levels of the four subsystems. Assumptions under which this design problem is solved are listed as follows:

(1) The failure times all components considered in the example are exponentially distributed, leading to constant failure rates.

(2) PHM will detect critical system health states and predict system RUL through health diagnostics and prognostics

(3) The redundancy level of each subsystem should be no more than nine due to subsystem weight and volume constraints.

(4) All the components and PHM units fail independently. An observed failure is due to the loss of resilience, i.e., the failures of both a component and its associated PHM unit.

The top-level RAP can be formulated as follows:

$$\begin{aligned}&\text{find } \mathbf{r}^t = \left(r_1^t, r_2^t, r_3^t, r_4^t\right),\ \boldsymbol{\lambda}^t = \left(\lambda_1^t, \lambda_2^t, \lambda_3^t, \lambda_4^t\right),\ \mathbf{m} = \left(m_1, m_2, m_3, m_4\right)\\ &\text{to minimize} \quad LCC(\mathbf{r}^t, \boldsymbol{\lambda}^t, \mathbf{m}) \\ &\text{subject to } \Psi = \prod_{j=1}^{4}\left[1-\left(1-r_j^t\right)^{m_j}\left(1-\lambda_j^t\right)^{m_j}\right] \geq \Psi^t,\ \mathbf{0} \leq \mathbf{r}^t, \boldsymbol{\lambda}^t \leq \mathbf{1},\ \mathbf{1} \leq \mathbf{m} \leq \mathbf{9}\end{aligned} \tag{6.3}$$

where *LCC* is the system life-cycle cost (LCC), Ψ and Ψ^t are system resilience and its target value, the lower and upper bounds for any target component-reliability or target component-PHM efficiency are 0 and 1, respectively, and the lower and upper bounds for any target component-redundancy are 1 and 9, respectively.

Life-Cycle Cost (LCC) Analysis with PHM

In this study, we derive a LCC model by modifying and adding PHM relevant cost elements to an existing LCC model for deteriorating structural systems [**150**]. The LCC model consists of four cost elements: the expected initial development cost of components, the expected cost of preventive maintenance, the expected cost of corrective maintenance, and the expected development cost of PHM. Given the target component-reliability vector $\mathbf{r}^t$, the target component-PHM efficiency vector $\boldsymbol{\lambda}^t$, and the target component- redundancy vector **m**, this LCC model can be expressed as

$$LCC\left(\mathbf{r}, \mathbf{m}, \boldsymbol{\lambda}\right) = C^I + C^{PM} + C^{CM} + C^{PHM} \tag{6.4}$$

where C^I denotes the initial development cost of components, C^{PM} denotes the cost of preventive maintenance, C^{CC} denotes the cost of corrective maintenance, and C^{PHM}

denotes the cost of PHM units. In the following subsections, the four cost elements will be discussed in details.

System Development Cost C^I

In the binary-state reliability-redundancy allocation problem, it is often assumed that there is an inverse power relationship between component cost and component failure rate [151,152]. Under the assumption of a constant failure rate, the initial development cost of the j^{th} subsystem with m_j parallel components can be expressed as [151,152]

$$C_j^I = c_j^I\left(r_j^t\right)\cdot\left[m_j + \exp\left(\frac{m_j}{4}\right)\right],\ \text{where}\ c_j^I\left(r_j^t\right) = \alpha_j^C\left(-\frac{T}{\ln\left(r_j^t\right)}\right)^{\beta_j^C} \tag{6.5}$$

where $c_j^I(r_j^t)$ is the cost function of a component in the j^{th} subsystem, $c_j^I(r_j^t)\cdot m_j$ is the cost of components in the j^{th} subsystem, an additional cost $c_j^I(r_j^t)\cdot\exp(m_j/4)$ accounts for the cost for interconnecting parallel components, T is the required system mission time, α_j^C and β_j^C denote constants representing the physical characteristics of each component in the j^{th} subsystem and can be determined based on the collected data of component cost and reliability. Therefore, the system initial development cost can computed as

$$C^I = \sum_{j=1}^{N}\alpha_j^C\left(-\frac{T}{\ln\left(r_j^t\right)}\right)^{\beta_j^C}\cdot\left[m_j + \exp\left(\frac{m_j}{4}\right)\right] \tag{6.6}$$

Preventive Maintenance Cost C^{PM}

That preventive maintenance occurs if PHM successfully detects critical system health states and accurately predicts the system RUL. As a function of the component reliability, subsystem redundancy and PHM efficiency, the preventive maintenance cost can be expressed as

$$C^{PM} = \sum_{j=1}^{N} m_j \lambda_j^t \left(1 - r_j^t\right) C_j^{PM} \tag{6.7}$$

where C_j^{PM} denote the preventive maintenance cost of each component in the j^{th} subsystem. The assumption here is that a preventive maintenance occurs when any component approaches its end of life predicted by the PHM and that all the components and PHM systems fail independently.

Corrective Maintenance Cost C^{CM}

The corrective maintenance occurs if PHM fails in detecting critical system health states and making an accurate prediction of the system RUL. As a function of the component reliability, subsystem redundancy and PHM efficiency, the corrective maintenance cost can be expressed as

$$C^{CM} = \sum_{j=1}^{N} m_j \left(1 - \lambda_j^t\right)\left(1 - r_j^t\right) C_j^{CM} \tag{6.8}$$

where C_j^{CM} denote the corrective maintenance cost of each component in the j^{th} subsystem. The assumption here is that a corrective maintenance occurs upon the failure of any component and that all the components and PHM systems fail independently.

PHM Unit Cost C^{PHM}

The PHM unit cost is specifically the costs associated with developing PHM units to be integrated with components. In this study, the PHM unit cost will be formulated as a parametric model with the subsystem redundancy and component PHM efficiency as inputs. Inspired by the component cost function for reliability-redundancy allocation, shown in (3), we define the PHM unit cost as

$$C^{PHM} = \sum_{j=1}^{N} \alpha_j^{PHM} \left(-\frac{T}{\ln\left(\lambda_j\right)} \right)^{\beta_j^{PHM}} \cdot m_j \tag{6.9}$$

where α_j^{PHM} and β_j^{PHM} denote constants representing the physical characteristics of each PHM unit in the j^{th} subsystem. Prior to solving the optimization problem in Eq. (6.3), these constants can be determined based on the collected data of the PHM unit cost and efficiency. It is noted that, in general, there is no interconnections between parallel PHM units. Therefore, unlike the component cost, the additional cost for interconnecting parallel elements is not considered in the PHM unit cost.

The parameters for the cost models are listed in Table 6-1 and the system mission time T = 1000. The RAP problem is a mixed-integer non-linear programming problem. To determine an optimum solution of the RAP problem, we employed a genetic algorithm of which the details will be presented in the subsequent subsection.

Table 6-1 Model parameters for the EHA case study

Subsystem	α_j^C ($\times 10^{-5}$)	β_j^C	C_j^{PM}	C_j^{CM}	α_j^{PHM}($\times 10^{-6}$)	β_j^{PHM}
1	0.5	1.5	2.5	7.5	3.3	1.5
1	0.8	1.5	5.0	15.0	5.3	1.5
2	1.0	1.5	6.5	19.5	6.7	1.5
3	0.7	1.5	12.5	37.5	4.7	1.5

Genetic Algorithm as the Optimization Solution Method

This problem is a mixed-integer nonlinear programming (MINLP) problem. As far as we know, the most widely used algorithm to solve this type of problem is the so-called genetic algorithm (GA) [153] due to the following advantages: (i) the encoding scheme (binary or decimal encoding) in the GA leads to the flexibility to represent both continuous and discrete design variables; and (ii) the search in the solution space for optimal solutions can be very efficient due to the use of fitness evaluation and genetic operator functions. We note that the RAP in Eq. (6.3) can be readily solved using the GA with almost negligible computational effort since the system LCC and resilience are computed through the evaluation of analytic functions.

In the GA, each candidate solution is called a chromosome and a set of candidate solutions is called a population. The GA for solving the RAP in this case study employed the decimal encoding. The solution procedures are presented as follows [153]:

Step 1 (Initialization): Set the population size and maximum number of iterations as 500 and 100, respectively. Since one decimal digit represents one design variable in the RAP shown in Eq. (6.3), the length L of a chromosome reads: $L = 3N$. Set the upper and lower bounds for both component-reliability and component efficiency to 0 and 1, respectively. Set the upper and lower bounds for component-redundancy to 1 and 9, assuming the redundancy level should not be too high. Set the generation index $k_g = 1$ and randomly generate an initial population $\mathbf{\Gamma}(1)$.

Step 2 (Evaluation): Evaluate the fitness function *ftn* for each chromosome in the current population $\mathbf{\Gamma}(k_g)$. The fitness function is a composite of both the objective value (i.e., system LCC) and the penalty arising from the violation of the constraint (i.e., system resilience). Mathematically, the fitness function *ftn* can be expressed as

$$ftn = \begin{cases} LCC(\mathbf{r}^t, \boldsymbol{\lambda}^t, \mathbf{m}), & \text{if } \Psi \geq \Psi^t \\ \text{inf}, & \text{otherwise} \end{cases} \tag{6.10}$$

Step 3 (Parent Selection): Select chromosomes from the current population based on their fitness values to form a new generation $\mathbf{\Gamma}(k_g + 1)$. Here the roulette-wheel selection scheme is used. These chromosomes are called parent and will be used in the next step to generate new chromosomes in the new generation.

Step 4 (Crossover & Mutation): Implement the two-point crossover operator with a crossover rate of 0.85 and the uniform mutation operator with a mutation rate of 0.10 to generate new chromosomes in the new population.

Step 5 (Termination Check): If the generation index k_g exceeds the maximal number of iterations, terminate the iteration and report the solution. Otherwise, increase the generation index: $k_g = k_g + 1$, and go back to Step 2.

Results and Discussion

We would like to investigate scenarios with different target system resilience levels. First let us look at the scenario in which the target system resilience Ψ^t is set as 0.90. The optimum solution is shown in Table 6-2. It can be seen that the incorporation of PHM by the proposed RDSD reduces the system redundancy from $\mathbf{m} = (3, 2, 3, 2)$ to $\mathbf{m} = (2, 2, 2, 1)$. As a consequence, the system LCC decreases from 73.6301 under the traditional design (without PHM) to 38.3416 under the RDSD (with PHM). It is noted that, even though the target component-reliabilities are relatively low for both traditional design (below 0.8500) and RDSD (below 0.7500), the incorporation of redundant components (traditional design and RDSD) and PHM (RDSD) still leads to high subsystem reliabilities (above 0.90). Finally, the system resilience levels under both optimum designs read 0.9000, which just satisfies the system resilience requirement.

Table 6-2 Optimum results of traditional design and RDSD with $\Psi^t = 0.90$

Subsystem	Traditional design (without PHM)					RDSD (with PHM)				
	r_j^t	m_j	λ_j^t	LCC	Ψ	r_j^t	m_j	λ_j^t	LCC	Ψ
1	0.7371	3	0	73.6301	0.9000	0.6291	2	0.6721	38.3416	0.9000
2	0.8088	2	0			0.6412	2	0.6682		
3	0.7287	3	0			0.6519	2	0.6732		
4	0.8292	2	0			0.7363	1	0.7679		

Raising the target system resilience to 0.95 and 0.99, respectively, we then obtained two sets of optimal designs, which are listed in Table 6-3 and Table 6-4, respectively. We observe that, in order to meet higher target system resilience level, more components are used with higher component-reliabilities and PHM efficiencies. Compared with the traditional design, the RDSD still yields optimum designs with much lower LCCs by considering PHM in the early design stage. The target component-reliabilities and component-PHM efficiencies allocated in this RAP can serve as design specifications for bottom-level system RBDO and PHM design that will be detailed in the subsequent subsections.

Table 6-3 Optimum results of traditional design and RDSD with $\Psi^t = 0.95$

Subsystem	Traditional design (without PHM)					RDSD (with PHM)				
	r_j^t	m_j	λ_j^t	LCC	Ψ	r_j^t	m_j	λ_j^t	LCC	Ψ
1	0.7901	3	0	82.2774	0.9500	0.6152	2	0.6448	45.9357	0.9500
2	0.7731	3	0			0.6437	2	0.6644		
3	0.7872	3	0			0.6486	2	0.6677		
4	0.8574	2	0			0.7539	2	0.7423		

Table 6-4 Optimum results of traditional design and RDSD with $\Psi^t = 0.99$

Subsystem	Traditional design (without PHM)					RDSD (with PHM)				
	r_j^t	m_j	λ_j^t	LCC	Ψ	r_j^t	m_j	λ_j^t	LCC	Ψ
1	0.8102	4	0	111.6017	0.9900	0.6488	3	0.6772	55.0199	0.9900
2	0.7745	4	0			0.6483	3	0.7049		
3	0.7850	4	0			0.6567	2	0.8014		
4	0.8411	3	0			0.7720	2	0.7678		

6.4.2 Bottom-Level System RBDO

This subsection aims at demonstrating the bottom-level system RBDO in the RDSD framework. We intend to determine the optimal design of the hydraulic actuator satisfying the target reliability obtained from the RAP with the target system resilience Ψ^t being 0.99. The success event of the actuator is considered as a series system success event consisting of four component success events.

Description of EHA Model

In order to investigate the performance of different actuator designs, we employed an EHA model built in a 1D multi-domain simulation platform LMS Imagine.Lab AMESim [154]. A simplified schematic of the EHA model is shown in Figure 6-5, where each submodel (e.g., motor, pump and actuator) is composed of a set of algebraic and differential equations accounting for linear and nonlinear effects such as friction and leakage. Here, a variable-speed DC motor drives a fixed-displacement hydraulic pump, which supplies oil to the actuator. A proportional controller controls the flow rate by varying the speed of the electric motor. An accumulator is used to prevent cavitation and compensate leakage loss with re-feeding valves. The pressure relief valves aim at preventing excessive pressure build-up in the hydraulic lines. As the flow encounters the actuator, the fluid pressure increases. The pressure difference between the two actuator chambers results in an actuation force and thus a linear piston motion. A mechanical arm (not shown in Figure 6-5) then transforms the piston motion to an equivalent aileron angle while taking into account the aileron inertia.

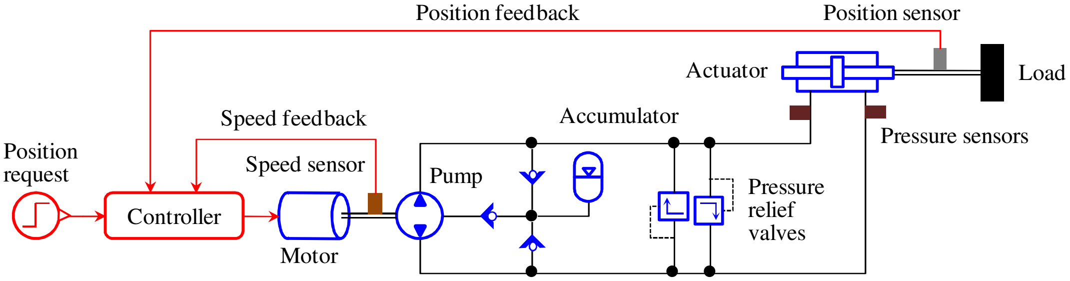


Figure 6-5 Schematic of an electro-hydrostatic actuator (EHA) model

System RBDO Formulation

For the system RBDO, we used the weight of the actuator to build the objective function and the control performance to formulate the reliability constraints. Regarding the control performance, we intended to take into account two aspects: timeliness and robustness. We applied a step request (1cm) on the piston position at the time t = 0.5s and a resistive torque (2000N·m) at the time t = 1.0s to test the control timeliness and robustness, respectively. The piston position response is shown in Figure 6-6, where the reliability constraints G_1 and G_2 are treated as timeliness-relevant constraints and G_3 and G_4 robustness-relevant. Specifically, this system RBDO problem can be formulated as follows:

$$\begin{aligned}
&\underset{\mathbf{x}}{\text{Minimize}} && C(\mathbf{x}) = \omega \cdot V_s(d_p, l_s) + (1-\omega) \cdot V_r(d_r, l_s) \\
&\text{where} && \omega = 0.098,\ V_s = l_s \cdot \pi (d_p / 2)^2,\ V_r = l_s \cdot \pi (d_r / 2)^2 \\
&\text{Subject to} && r^{sys} = \Pr\left(E^{sys}\right) = \Pr\left(\bigcap_{i=1}^{4} G_i(\mathbf{x}) \le 0\right) \ge r^t \\
& && G_1 = \int_0^2 \left| Y(t) - Y_{ref}(t) \right| dt - e_{nc} \qquad \text{(normal control error)} \\
& && G_2 = \underset{0.5 \le t \le 2}{\arg\min} \left\{ \left| Y(t) - Y_{ref}(t) \right| \le \varepsilon_{tol,s} \right\} - t_{c,s} \quad \text{(stabilization time)} \\
& && G_3 = \int_2^4 \left| Y(t) - Y_{ref}(t) \right| dt - e_{pc} \qquad \text{(disturbed control error)} \\
& && G_4 = \min_{2 \le t \le 4} \left\{ \left| Y(t) - Y_{ref}(t) \right| \right\} - \varepsilon_{tol,d} \qquad \text{(disturbed steady-state error)} \\
& && G_5 = \eta - d_r / d_p \qquad \text{(rod-to-piston diameter ratio)}
\end{aligned} \tag{6.11}$$

where the critical normal control error e_{nc} = 0.20cm·s, the stabilization error tolerance $\varepsilon_{tol,s}$ = 0.03cm and is used to determine whether a stable state is achieved, the critical stabilization time $t_{c,s}$ = 0.90s, the critical perturbed control error e_{np} = 0.05cm·s, the steady-state error tolerance under a disturbance $\varepsilon_{tol,d}$ = 0.04cm and the rod-to-piston diameter ratio η = 1/3. To avoid having a weak rod relative to a piston, we also add the fifth constraint G_5 in Eq. (6.11) which ensures that the rod-to-piston ratio exceed a certain level. The following two design variables are considered: the piston diameter d_p, the rod diameter d_r. These design variables are assumed to follow normal distributions with their standard deviations, initial values and lower and upper bounds detailed in 0. Three model parameters (i.e., the leakage coefficient β, viscous friction coefficient υ and the stroke length l_s) are considered as random noise variables with their statistical information summarized in Table 6-6. In summary, five random design and noise variables are considered in this study.

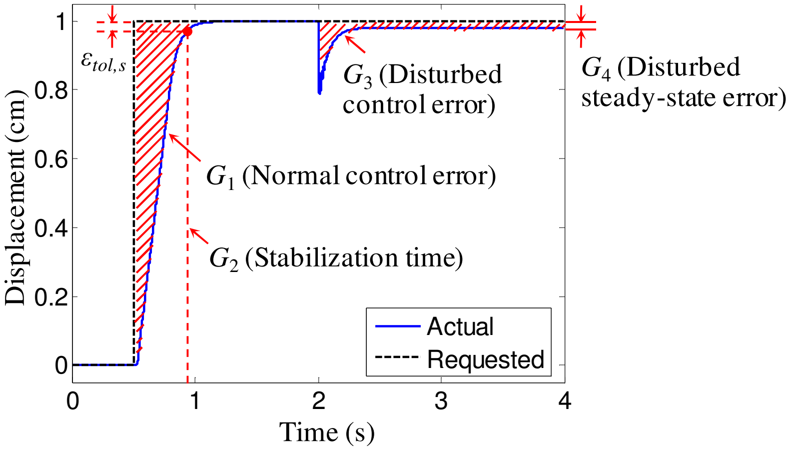


Figure 6-6 Piston position response under a step request and resistive torque

Table 6-5 Random design variables for the hydraulic actuator model

Design variable	Distri. type	Lower bound	Initial des.	Upper bound	Std. dev.
d_p (mm)	Normal	55.0	62.0	75.0	3.5
d_r (mm)	Normal	10.0	22.0	30.0	1.0

Table 6-6 Random noise variables for the hydraulic actuator model

Random variable	Distri. type	Mean	Std. dev.
β (L/min/Bar)	Normal	1.2E-3	6.0E-5
υ (N·s/m)	Normal	5.0E3	2.5E2
l_s (mm)	Normal	50.0	2.5

Results of System RBDO

The adaptive-sparse polynomial chaos expansion (PCE) method with $4nv + 1$ (= 21) univariate samples was carried out to evaluate the objective function, system reliability and their sensitivities at any design iteration, without considering the bivariate polynomial basis functions. The sensitivities of the objective function and system reliability with respect to the two design variable were computed using a finite difference method (FDM). The perturbed values of the objective function and system reliability were estimated based on approximate stochastic response surfaces (PCE) with perturbed design variables, without requiring gradients of the original weight or displacement functions. A perturbation size of 0.1% is employed in this study.

The design optimization problem was solved using a gradient-based optimization technique (e.g., sequential quadratic optimization). The histories of the design parameters, objective function, component reliabilities and system reliability are shown in Table 6-7. At the initial design, the system reliability r^{sys} severely violated the reliability requirement due to the relatively low component reliabilities r_3 and r_4. After six design iterations, the optimum design was found where the system reliability requirement was satisfied.

Table 6-7 Design history of the hydraulic actuator model

Iter.	Design variables		r_1	r_2	r_3	r_4	r_5	r^{sys}	Obj.
	d_p	d_r							
0	62.0000	22.0000	0.9985	0.9978	0.4955	0.5350	0.8053	0.3093	3.2015E4
1	65.8472	24.5219	0.9819	0.9767	0.7695	0.8162	0.9522	0.7136	3.8062E4
2	66.7479	24.7483	0.9694	0.9617	0.8309	0.8699	0.9471	0.7620	3.8916E4
3	66.9844	24.7694	0.9640	0.9551	0.8459	0.8826	0.9429	0.7698	3.9075E4
4	67.1807	24.6941	0.9586	0.9479	0.8607	0.8947	0.9314	0.7707	3.9044E4
5	67.1847	24.7177	0.9587	0.9479	0.8604	0.8944	0.9331	0.7717	3.9088E4
6	67.1617	24.7268	0.9594	0.9488	0.8587	0.8929	0.9346	0.7720	3.9092E4
Opt	67.1617	24.7268	0.9594	0.9488	0.8587	0.8929	0.9346	0.7720	3.9092E4

Overall, the adaptive-sparse PCE method required 126 simulations for system RBDO. After the optimization, the direct MCS with 10,000 random samples was employed to verify the reliability results at the optimum design. The component reliabilities r_1, r_2, r_3, r_4 and r_5 and system reliability r^{sys} were estimated by the MCS as 95.36%, 94.40%, 85.99%, 89.74%, 93.65% and 77.24%, respectively.

6.4.3 Bottom-Level System PHM Design

This subsection is dedicated to demonstrating the bottom-level system PHM design in the RDSD framework. We intend to design a data-driven prognostic algorithm for the actuator leakage prognostics by identifying the most appropriate algorithm from an algorithm pool. We assume that the PHM unit can successfully identify the incipient leakage degradation of the actuator among various possible failure modes ($\Lambda_D = 1$) and that, upon a correct prognosis event, the M/R maintenance actions can fully restore the reliability of the actuator ($\kappa = 1$).

Prognostic Data Generation

The failure mode considered in this study is the actuator cross-line leakage which is relatively common in practice [155,156]. The wear of the piston seal causes an increase of internal cross-port leakage and thus an increase in the leakage coefficient of the actuator. Thus, the cross line leakage was realized by increasing the leakage coefficient of the actuator. The end of life is defined as the time when the actuator leakage reaches ten times its initial value. We note that, for demonstration purposes, this study only considers a single failure mode, but the same idea can be readily extended to cases with multiple failure modes.

Since it is very difficult, if not impossible, to obtain direct measurements of the leakage coefficients of actuators, indirect measurements are most often used to diagnose the health condition and predict the RULs of actuators [155,156]. In particular, the stabilized piston displacement after disturbance (measured by a position sensor) and the stabilized rotary speed of the motor shaft after disturbance (measured by a rotary speed sensor) could characterize the actuator leakage degradation before any irreparable damage occurs [155]. Thus, this case study employs these two sensory signals of an EHA to predict the RULs of actuators. To model the trajectory of increase in leakage over time, this study uses a damage propagation model with an exponential form as [129]

$$\beta(t) = \beta_0 + b_E\left(\exp\left(a_E t\right) - 1\right) \tag{6.12}$$

where β_0 is the initial leakage coefficient; a_E and b_E are the model parameters; t is the cycle time. The initial Young's modulus E_0 follows the same normal distribution with β (see Table 6-6). The model parameters a_E and b_E are independent and normally distributed with means 0.01 and 1.2E-3, each of which has a 10% coefficient of variation. The random parameters considered in this study are listed in 0 and Table 6-6, which include the material properties as well the geometries of the actuator. The uncertainties in the two sensory signals propagated from these uncertain parameters will be accounted for when generating prognostic data. The prognostic data generation was conducted under the optimal actuator design obtained in the bottom-level system design (see Subsection 6.4.2).

Since data-driven prognostic approaches require a large amount of prognostic data, it is computationally expensive, if not impossible, to simply run the simulation to generate every data point. To overcome this difficulty, this study employed the univariate decomposition method that only uses a certain number of univariate sample points to construct the response surface for a general multivariate response function while achieving good accuracy [157]. Since this study employed two sensory signals, namely the stabilized piston displacement after disturbance and the stabilized rotary speed of the motor shaft after disturbance, the data generation requires the construction of two responses surfaces. Specifically, the data generation process involves four sequentially executed procedures:

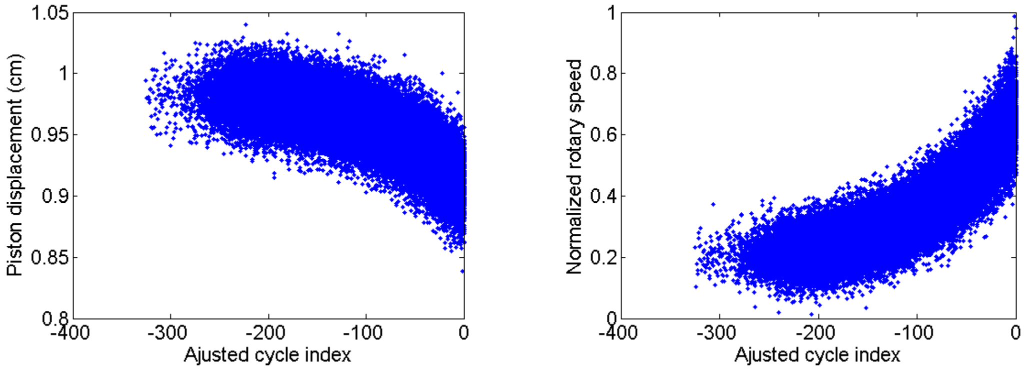
Step 1: Obtain univariate sample points from the dynamic simulation in LMS Imagine.Lab AMESim to construct response surfaces, along the damage propagation path, that approximate the two sensory measurements as functions of random variables detailed in 0 and Table 6-6. We used four univariate sample points for each random variable. The piecewise linear spline was used as the numerical scheme for the response surface construction.

Step 2: Generate 400 random samples of β_0, a_E and b_E and use these samples in conjunction with Eq. (6.12) to produce 400 damage propagation paths, of which 200 paths were assigned to the training units and the rest to the testing units.

Step 3: Interpolate, based on the constructed response surfaces, the two sensory measurements for a given set of randomly generated geometries and material properties and damage propagation paths and repeatedly execute this process for 400 times to obtain the training and testing data sets with each having 200 units.

Step 4: Add measurement noise following a zero mean normal distribution to both the training and testing data sets to finalize the data generation.

The two simulated measurements are plotted against the adjusted cycle index, defined as the subtraction of the cycle-to-failure from the actual operational cycle, for all 200 training units in Figure 6-7, where we can clearly observe monotonic lifetime trends.



(a) (b)

Figure 6-7 Simulated measurements by piston displacement sensor (a) and rotary speed sensor (b) for the hydraulic actuator model

<u>Description of Prognostic Algorithms</u>

This subsection provides a brief overview of the five selected data-driven prognostic algorithms: Method 1 - a similarity-based interpolation (SBI) approach with RVM as the regression technique (RVM-SBI) [53,126], SBI with SVM (SVM-SBI) [53,127], SBI with the least-square exponential fitting (Exp-SBI) [53], a Bayesian linear regression with the least-square quadratic fitting (Quad-BLR) [55], and a recurrent neural network (RNN) approach (RNN) [56,127]. A data processing scheme with a generic health index system is used for the first four algorithms while a data processing scheme with a simple normalization scheme for the last algorithm. These five algorithms represent the current state-of-art in data-driven prognostics and cover a wide range of techniques that include the interpolation (Methods 1-3), extrapolation (Method 4) and machine learning (Method 5).

For the construction of the virtual health index in Methods 1-3, the system failure matrix $\mathbf{Q}_0$ was created with the sensory data in a system failure condition, $0 \leq L \leq 4$, while the system healthy matrix $\mathbf{Q}_1$ with those in a system healthy condition, $L > 250$. The RVM employed a linear spline kernel function with the initial most probable hyper-parameter vector for kernel weights $\boldsymbol{\alpha}_m = [1\times10^4, \ldots, 1\times10^4]$ and the initial most probable noise variance $\sigma_m^2 = 1\times10^{-4}$. In the SVM, a Gaussian kernel function is used with the parameter settings as: the regularization parameter $C = 10$ and the parameter of the ε-insensitive loss function $\varepsilon = 0.10$. In the RNN training, the two normalized sensory signals were used as the multi-dimensional inputs of the RNN and the RUL at the corresponding cycle was used as the output. The implementation details can be found in [56]. In the RNN architecture, the numbers of the input, recurrent and output units are $|I| = 2$, $|R| = 4$ and $|O| = 1$.

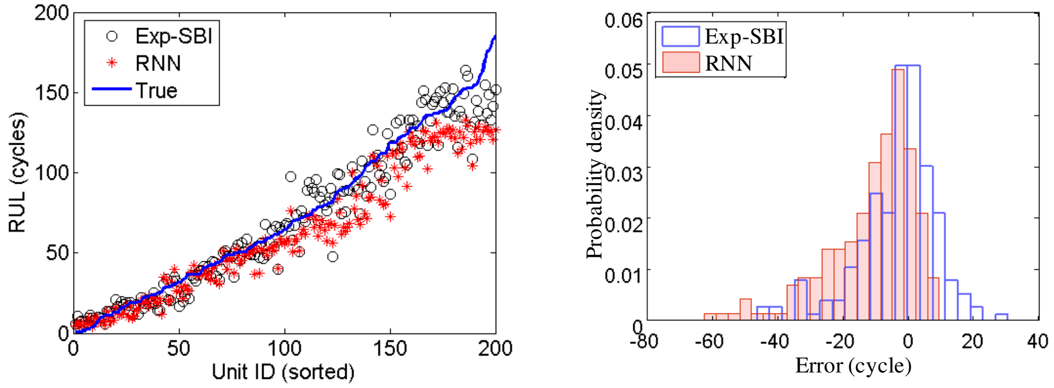
Results of System PHM Design

Table 6-8 summarizes the prognostic accuracy of the five candidate algorithms as well as the detailed information regarding the empirical error PDFs. The lower bound e_C of the error tolerance zone was set as -35cycles. Among the five candidate algorithms, RNN yields the highest prognostic accuracy of 0.790 on the testing data set, a 43.6% improvement over the second best algorithm, Exp-SBI, whose prognostic accuracy reads 0.550. To further investigate this accuracy gap, we plotted the RUL predictions by the two algorithms for 200 testing units in Figure 6-8(a) and their error PDFs in Figure 6-8(b). It can be observed from both plots that RNN consistently gives early RUL predictions while Exp-SBI is prone to produce RUL predictions being randomly distributed around zero. Therefore, RNN provides higher accuracy in spite of a larger variance and, with failure prevention being the main goal of PHM, one would select RNN among the candidate algorithms to have moderately early RUL predictions. Indeed, it is noted that only RNN satisfies the target component-PHM efficiency 0.7678 obtained in Subsection 6.4.1 with the assumption of perfect diagnostics and M/R actions. Finally, we note that, by capitalizing on PHM, the EHA is capable of detecting, predicting and mitigating or recovering from the actuator leakage, and thus achieves the resilience characteristics intended by the underlying idea of RDSD, namely the optimal restoration of system performance.

Table 6-8 Prognostic accuracy and empirical error PDF results for the hydraulic actuator model

Algorithm	Prognostic accuracy	Distri. type	Mean (cycle)	Std. dev. (cycle)	Parameters for non-normal distributions
RVM-SBI	0.480	Weibull	-2.66	12.14	$\alpha_1 = 49.22,\ \beta_1 = 4.15^a$
SVM-SBI	0.430	Weibull	-3.40	14.91	$\alpha_2 = 60.10,\ \beta_2 = 4.12^a$
Exp-SBI	0.550	Normal	-3.17	12.03	-
Quad-BLR	0.125	Weibull	10.62	12.76	$\alpha_4 = 63.82,\ \beta_4 = 5.31^a$
RNN	0.790	Weibull	-12.10	13.07	$\alpha_5 = 55.17,\ \beta_5 = 4.35^a$

[a] $e_1 \geq -47.36$; [b] $e_2 \geq -57.96$; [c] $e_4 \geq -48.17$; [d] $e_5 \geq -62.35$



(a) (b)

Figure 6-8 RUL prediction results (a) and error PDFs (b) for the hydraulic actuator model

6.5 Conclusion

This chapter presents a novel design framework, namely resilience-driven system design (RDSD), to incorporate resilience characteristics into engineered systems. This design framework consists of three hierarchical tasks, namely the top-level RAP, the bottom-level system RBDO and the bottom-level system PHM design. The top-level RAP incorporates the trade-off decisions regarding the component-reliability, component-PHM efficiency and component-redundancy into the system-level design for an optimum integration of PHM with minimum LCC. The bottom-level system RBDO determines an optimal component design while ensuring the optimally allocated target component-reliability from the top-level RAP. Following the system RBDO, the bottom-level system PHM design derives an optimal PHM unit design for the optimal component design while meeting the optimally allocated PHM efficiency from the top-level RAP.

The proposed RDSD framework is demonstrated with a simplified aircraft control actuator design problem, in which the incorporation of PHM significantly reduces the system LCC and the detailed component and PHM unit designs respectively satisfy the target component-reliability and component-PHM efficiency.

It is noted that, despite the difficulty in simultaneously and comprehensively exploring system RBDO and PHM in an integrated case study, this chapter provides a strategic guidance for carrying out this exploration through an engineering case study on the design of a safety-critical aerospace system. The proposed RDSD framework, featured with a rigorous theoretical basis and design strategy of engineering resilience, is expected to ensure highly resilient system designs under various loading/environmental conditions and system-wide effects of adverse events while considerably reducing systems' LCC.

Chapter 7: Contributions and Future Works

7.1 Contributions and Impacts

The proposed research in this dissertation aims at establishing a solid theoretical basis and systematic design framework of engineering resilience to facilitate the development of formal methodologies for resilient engineered system design and to advance conventional engineered systems to resilient engineered systems. This research is composed of three research thrusts: (i) system RBDO, (ii) system PHM, and (iii) resilience-driven system design. It is expected that the proposed research offers the following potential contributions and broader impacts in various engineering fields:

- Contribution 1: An adaptive-sparse polynomial chaos expansion (PCE) method for component reliability analysis involving bivariate interactions

  This dissertation proposes an adaptive-sparse PCE method for performing engineering reliability analysis and design. The proposed method consists of three technical contributions: (i) an adaptive-sparse scheme to build sparse PCE with the minimum number of bivariate basis functions, (ii) a new projection method using dimension reduction techniques to effectively compute the expansion coefficients of system responses, and (iii) an integration of copula to handle nonlinear correlation of input random variables. To the best of our knowledge, the proposed method can be treated as the first attempt to optimize the number of the bivariate terms to be considered for reliability analysis. The optimization process takes advantage of the PCE as the projection basis and proceeds by adaptively adding the most significant bivariate terms to the PCE model until a convergence criterion is achieved. The inherent characteristics of orthogonal polynomials make the adaptive process computationally efficient and convergent. Therefore, the adaptive-sparse PCE achieves an optimal compromise between the univariate dimensional reduction (UDR) and bivariate dimension reduction (BDR) methods (more accurate than the UDR and more efficient than the BDR).

- Contribution 2: An adaptive stochastic collocation method for component reliability analysis involving tri- and higher-variate interactions

  This research presents an asymmetric dimension-adaptive tensor-product (ADATP) method as an adaptive stochastic collocation method for component reliability analysis involving tri- and higher-variate interactions. The proposed method leverages three ideas: (i) an asymmetric dimension-adaptive scheme to efficiently build the tensor-product interpolation considering both directional and dimensional importance, (ii) a hierarchical interpolation scheme using either piecewise multi-linear basis functions or cubic Lagrange splines, (iii) a hierarchical surplus as an error indicator to automatically detect the highly nonlinear regions in a random space and adaptively refine the collocation points in these regions. The adaptive hierarchical interpolation scheme possesses both directional and dimensional adaptivity. The proposed sensitivity-free method achieves faster error decay than its non-asymmetric counterpart and greatly alleviates the curse of dimensionality.

- Contribution 3: A unified framework for system reliability analysis

The unified framework proposed in this dissertation automatically decomposes a mixed system (represented by a system block diagram) into multiple disjoint series systems (not independent but mutually exclusive), which allows one to apply the original CIM to these series systems and obtain a unique estimate of system reliability, and that's precisely the main contribution of the proposed framework, brought by way of a BDD-based algorithm for computing mutually exclusive path sets. Indeed, the basic idea behind this generalization is to add another computational layer in the original CIM structure and to reformulate the problem in a way that allows for the use of the original CIM. Such a reformulation is an extension of the original work on CIM, with the advantage that it greatly expands the application domain and achieves a unique solution of system reliability regardless of system structures (series, parallel, and mixed systems).

- Contribution 4: An ensemble data-driven prognostic approach for PHM in a single time-scale

  This research proposes an ensemble approach that employs the *k*-fold cross validation (CV) to estimate the accuracy of a given weighted-sum ensemble and proposes three weighting schemes, namely, namely, the accuracy-based weighting, diversity-based weighting and optimization-based weighting, to determine the weight values. By combining the predictions of all member algorithms, the ensemble approach achieves better accuracy and robustness in RUL predictions compared to any sole member algorithm. The proposed ensemble approach enables highly accurate and robust life predictions for resilient engineered systems with a single time-scale.

- Contribution 5: A co-training data-driven prognostic approach for PHM in a single time-scale

  The proposed co-training-based data-driven prognostic algorithm, denoted by COPROG, uses two individual data-driven algorithms with each predicting RULs of suspension units for the other. The confidence of an individual data-driven algorithm in predicting the RUL of a suspension unit is quantified by the extent to which the inclusion of that unit in the training data set reduces the sum square error (SSE) in RUL prediction on the failure units. After a suspension unit is chosen and its RUL is predicted by an individual algorithm, it becomes a virtual failure unit that is added to the training data set. COPROG gives more accurate RUL predictions compared to any individual algorithm without the consideration of suspension data and that COPROG can effectively exploit suspension data to improve the accuracy in data-driven prognostics.

- Contribution 6: A multiscale filtering technique for PHM in multiple time-scales

  The proposed multiscale filtering technique achieves efficient and accurate state and parameter estimation for engineered systems that exhibit time-scale separation. When applied to the battery system, the multiscale framework can be treated as a hybrid of coulomb counting and adaptive filtering techniques. The contribution of this framework lies in the fact that the construction of a multiscale computational scheme decouples the SOC and capacity estimation from two perspectives, namely the measurement and time-scale. The resulting decoupled estimation greatly reduces the computational time involved in obtaining the SOC and capacity estimates, while enhancing the accuracy in the capacity estimation.

- Contribution 7: Enhancement of the understanding of resilient engineered systems and support for system design paradigm shift

  This research develops a rigorous definition of engineering resilience and constitutes solid theoretical basis of resilient engineered systems. A systematic decomposition of resilience process in resilient engineered systems enables thorough understanding of resilience behavior and cohesively integrates this behavior into future system design. The proposed system resilience analysis and design framework enables a transformative shift in engineered system design from reliability-based to resilience-driven system design. The theoretical basis and design framework of engineering resilience gained from this research will facilitate the development of formal methodologies for resilient engineered system design.

7.2 Suggestions for Future Research

Although the technical advances presented in this dissertation successfully address critical challenges in both system RBDO and PHM as well as make the first attempt to derive a systematic framework for resilient engineered system design, there are still several research areas where further investigations and developments are required to truly bring resilience-driven system design into reality. Specific suggestions for the continuation of the study on resilience-driven system design are listed as follows:

- Advancement of model-based prognostics techniques

This research develops two advanced data-driven prognostics techniques, namely ensemble prognostics and co-training prognostics, to design a highly accurate and robust PHM system for resilient engineered system design. However, no efforts have been devoted to the advancement of model-based prognostics techniques in this research. When we have ample understanding of system physics-of-failure and underlying system degradation models but possess very limited run-to-failure and suspension data, the model-based prognostics techniques become more desirable than their data-driven counterparts. Thus, to expand the application domain of PHM technologies, advanced prognostics techniques based on system physics-of-failure and underlying system degradation models must be developed and applied to resilient engineered system design.

- Development of PHM techniques for Li-ion battery SOL prediction

  This research proposes a multiscale framework with EKF for real-time estimation of SOC and capacity by tracking readily available measurements (e.g., voltage, current and temperature) for Li-ion battery systems. However, the online prediction of SOL with these measurements was untouched in this research. Future research should be focused on the development of data-driven, model-based or hybrid PHM techniques for online prediction of SOL. To verify the effectiveness of the techniques to be developed, we can conduct an extensive accelerated life testing (ALT) on a number of battery cells of the same specifications.

- Consolidation and verification of the proposed RDSD framework

The proposed RDSD framework provides the first strategic guidance for simultaneously and comprehensively exploring system RBDO and PHM to enhance the resilience of an engineered system in its early design stage. However, further research works are still needed to consolidate and verify the proposed design framework. Specifically, future investigations should be devoted to the following tasks: (i) the definition of engineering resilience and the design framework of system PHM must be generalized to consider multiple failure mechanisms; (ii) the diagnostics design (SN design) should be coherently integrated with the prognostics design (prognostic algorithm design) for the bottom-level system PHM design; and (iii) the verification of the proposed framework needs to be conducted using testing data from a real engineered system, as opposed to the use of a simulation model with hypothetical data in this dissertation.

- Decision-making on maintenance/restoration (M/R) action

The conditional probability κ of the M/R action success quantifies the extent to which the system is able to restore the system's original performance. This research has assumed κ as a constant. However, in engineering practice, the restoration (M/R action) requires a decision-making process that optimizes the allocation of available resources based on the system health information from the diagnostics and prognostics processes, making κ non-constant. Thus, in addition to the three tasks mentioned earlier, research efforts should also be devoted to systematically carrying out the decision-making process and determining the optimum κ for a specific restoration context. One possible way is to define key metrics that are indicative of system capacity and conduct a trade-off analysis between the system capacity restoration and resource consumption. In this way, the reasoning process can identify the most effective M/R actions as well as allocate the available resources to these M/R actions in such a way that the system capacity restoration can be maximized.

Appendices

Appendix A: Derivation of the Error Decomposition

Error Source I: Truncation

$$
\begin{aligned}
\varepsilon_P^2 &= \int_\Gamma \left[g - w_{UB}^p \right]^2 f(\mathbf{x})\, d\mathbf{x} \\
&= \int_\Gamma \left[\begin{array}{l}
\displaystyle\sum_k \sum_{j>p} s_j^k \cdot \Phi_j(\zeta_k) + \sum_{[x_k, x_l]\notin \mathbf{B}} \sum_{j_1+j_2\le p} s_{j_1,j_2}^{k,l} \cdot \Phi_{j_1,j_2}(\zeta_k,\zeta_l) \\
\displaystyle + \sum_{k<l} \sum_{j_1+j_2>p} s_{j_1,j_2}^{k,l} \cdot \Phi_{j_1,j_2}(\zeta_k,\zeta_l) \\
\displaystyle + \sum_{k<l<m} \sum_{j_1,j_2,j_3\ge 1} s_{j_1,j_2,j_3}^{k,l,m} \cdot \Phi_{j_1,j_2,j_3}(\zeta_k,\zeta_l,\zeta_m) + \cdots
\end{array} \right]^2 f(\mathbf{x})\, d\mathbf{x}
\end{aligned}
$$

Error source II: Bivariate decomposition

$$
\begin{aligned}
\varepsilon_B^2 &= \int_\Gamma \left[w_{UB}^p - w_U^p \right]^2 f(\mathbf{x})\, d\mathbf{x} \\
&= \int_\Gamma \left[\sum_{[x_k, x_l]\in \mathbf{B}} \sum_{j_1+j_2\le p} \left(s_{j_1,j_2}^{k,l} - \hat{s}_{j_1,j_2}^{k,l} \right) \Phi_{j_1,j_2}(\zeta_k,\zeta_l) \right]^2 f(\mathbf{x})\, d\mathbf{x} \\
&= \int_\Gamma \left[\sum_{[x_k, x_l]\in \mathbf{B}} \sum_{j_1+j_2\le p} \frac{E\left[\begin{array}{l} \left(g(\mathbf{x}) - g\left(x_k, x_l, \boldsymbol{\mu}^{k,l}\right) \right) \\ \times \Phi_{j_1,j_2}(\zeta_k,\zeta_l) \end{array} \right]}{E\left[\Phi_{j_1,j_2}^2(\zeta_k,\zeta_l) \right]} \cdot \Phi_{j_1,j_2}(\zeta_k,\zeta_l) \right]^2 f(\mathbf{x})\, d\mathbf{x}
\end{aligned}
$$

where

$$
\begin{aligned}
& E\left[\left(g(\mathbf{x}) - g\left(x_k, x_l, \boldsymbol{\mu}^{k,l}\right) \right) \Phi_{j_1,j_2}(\zeta_k,\zeta_l) \right] \\
&= \frac{1}{2!1!1!} \sum_{i\ne k,l} \frac{\partial^4 g}{\partial x_i^2 \partial x_k \partial x_l}(\boldsymbol{\mu}) \cdot E\left[(x_i-\mu_i)^2 (x_k-\mu_k)(x_l-\mu_l) \Phi_{j_1,j_2}(\zeta_k,\zeta_l) \right] + \cdots
\end{aligned}
$$

Error source III: Univariate decomposition

$$\begin{aligned}
\varepsilon_U^2 &= \int_\Gamma \left[w_U^p - w_I^p \right]^2 f\left(\mathbf{x}\right) d\mathbf{x} \\
&= \int_\Gamma \left[\sum_{k=1}^{N} \sum_{j=1}^{p} \left(s_j^k - \hat{s}_j^k \right) \Phi_j\left(\zeta_k\right) \right]^2 f\left(\mathbf{x}\right) d\mathbf{x} \\
&= \int_\Gamma \left[\sum_{k=1}^{N} \sum_{j=1}^{p} \frac{E\left[\left(g\left(\mathbf{x}\right) - g\left(x_k, \boldsymbol{\mu}^k\right) \right) \Phi_j\left(\zeta_k\right) \right]}{E\left[\Phi_j^2\left(\zeta_k\right) \right]} \cdot \Phi_j\left(\zeta_k\right) \right]^2 f\left(\mathbf{x}\right) d\mathbf{x}
\end{aligned}$$

where

$$\begin{aligned}
&E\left[\left(g\left(\mathbf{x}\right) - g\left(x_k, \boldsymbol{\mu}^k\right) \right) \Phi_j\left(\zeta_k\right) \right] \\
&= \frac{1}{2!1!} \sum_{i \neq k} \frac{\partial^3 g}{\partial x_i^2 \partial x_k} (\boldsymbol{\mu}) E\left[\left(x_i - \mu_i\right)^2 \left(x_k - \mu_k\right) \Phi_j\left(\zeta_k\right) \right] + \cdots
\end{aligned}$$

Error source IV: Aliasing error

$$\begin{aligned}
\varepsilon_I^2 &= \int_\Gamma \left[w_I^p - w^p \right]^2 f\left(\mathbf{x}\right) d\mathbf{x} \\
&= \int_\Gamma \left[\sum_{k=1}^{N} \sum_{j=1}^{p} \left(\hat{s}_j^k - \hat{\hat{s}}_j^k \right) \Phi_j\left(\zeta_k\right) + \sum_{[x_k, x_l] \in \mathbf{B}} \sum_{j_1 + j_2 \leq p} \left(s_{j_1, j_2}^{k,l} - \hat{\hat{s}}_{j_1, j_2}^{k,l} \right) \Phi_{j_1, j_2}\left(\zeta_k, \zeta_l\right) \right]^2 f\left(\mathbf{x}\right) d\mathbf{x} \\
&= \int_\Gamma \left[\begin{array}{l} \displaystyle \sum_{k=1}^{N} \sum_{j=1}^{p} \frac{\left(E - \hat{E}\right)\left[g\left(x_k, \boldsymbol{\mu}^k\right) \Phi_j\left(\zeta_k\right) \right]}{E\left[\Phi_j^2\left(\zeta_k\right) \right]} \cdot \Phi_j\left(\zeta_k\right) \\ \displaystyle + \sum_{[x_k, x_l] \in \mathbf{B}} \sum_{j_1 + j_2 \leq p} \frac{\left(E - \hat{E}\right)\left[g\left(x_k, x_l, \boldsymbol{\mu}^{k,l}\right) \cdot \Phi_{j_1, j_2}\left(\zeta_k, \zeta_l\right) \right]}{E\left[\Phi_{j_1, j_2}^2\left(\zeta_k, \zeta_l\right) \right]} \cdot \Phi_{j_1, j_2}\left(\zeta_k, \zeta_l\right) \end{array} \right]^2 f\left(\mathbf{x}\right) d\mathbf{x}
\end{aligned}$$

where $\hat{E}\left(\cdot\right)$ denotes the approximate expectation by using the SMLS and Gaussian quadrature integration.

Appendix B: Pseudo Code of ADATP Algorithm

Pseudo code :

```
$\mathbf{i} = (1,\ldots,1)$
$\mathbf{I}_\mathrm{O} = \varnothing,\ \mathbf{I}_\mathrm{A} = \{\mathbf{i}\}$
$l = 0,\ \varepsilon_r = 1,\ M = 1$
while$\left(\max\limits_{\mathbf{I}_\mathrm{A}}(\boldsymbol{\varepsilon}_r) > \varepsilon_c\ \&\&M < M_{\max}\right)$
  select $\mathbf{I}_\mathrm{T} \subset \mathbf{I}_\mathrm{A}$ with $\boldsymbol{\varepsilon}_r(\mathbf{I}_\mathrm{T}) > \varepsilon_c$
  $\mathbf{I}_\mathrm{O} = \mathbf{I}_\mathrm{O} \bigcup \mathbf{I}_\mathrm{A},\ \mathbf{I}_\mathrm{A} = \varnothing$
  while$(\mathbf{I}_\mathrm{T} \neq \varnothing\ \&\&M < M_{\max})$
    select $\mathbf{i}_\mathrm{t} \subset \mathbf{I}_\mathrm{T}$ with $\varepsilon_r(\mathbf{i}_\mathrm{t}) \geq \varepsilon_r(\mathbf{i}_\mathrm{r}),\ \forall \mathbf{i}_\mathrm{r} \in \mathbf{I}_\mathrm{T}$
    $\mathbf{I}_\mathrm{T} = \mathbf{I}_\mathrm{T} \setminus \mathbf{i}_\mathrm{t},\ \mathbf{I}_\mathrm{F}(\mathbf{i}_\mathrm{t}) = \left\{\mathbf{i}_\mathrm{t} + e_k^{+/-},\ 1 \leq k \leq N\right\}$
    $\mathbf{I}_\mathrm{A} = \mathbf{I}_\mathrm{A} \bigcup \mathbf{I}_\mathrm{F}(\mathbf{i}_\mathrm{t})$
    $\mathbf{w}^{\mathbf{i}_\mathrm{f}} = g\left(\mathbf{X}^{\mathbf{i}_\mathrm{f}}\right) - \mathrm{adatpstep}\left(\mathbf{I}_\mathrm{O}, \left\{\mathbf{w}^{\mathbf{I}_\mathrm{O}}\right\}, \left\{\mathbf{X}^{\mathbf{i}_\mathrm{f}}\right\}\right)$
    $M = M + M_{\mathbf{i}_\mathrm{f}}$
    $\varepsilon_r(\mathbf{i}_\mathrm{f}) = \frac{1}{(g_{\max} - g_{\min}) M_{\mathbf{i}_\mathrm{f}}} \sum_{\mathbf{j}} \left|w_{\mathbf{j}}^{\mathbf{i}_\mathrm{f}}\right|$
  endwhile
  $l = l + 1$
endwhile
```

Symbols:

$\mathbf{I}_{\mathrm{O}}$	old index set
$\mathbf{I}_{\mathrm{A}}$	active index set
$\mathbf{I}_{\mathrm{T}}$	trial index set
$\mathbf{I}_{\mathrm{F}}\left(\mathbf{i}_{\mathrm{t}}\right)$	forward neighborhood of trial index $\mathbf{i}_{\mathrm{t}}$
ε_r	relative error indicator
ε_c	relative error threshold
$g_{\max}$	maximum function value
$g_{\min}$	minimum function value
$e_k^{+/-}$	k^{th} directional unit vector
adatpstep	step hierarchical interpolation function

Glossary

ADATP:	Asymmetric Dimension-Adaptive Tensor-Product
CDF:	Cumulative Distribution Function
CIM:	Complementary Intersection Method
CV:	Cross Validation
DI:	Directional Index
EKF:	Extended Kalman Filter
FOB:	First-Order Bounds
FORM:	First-Order Reliability Method
PCE:	Polynomial Chaos Expansion
PDF:	Probability Density Function
PHM:	Prognostics and Health Management
RBDO:	Reliability-Based Design Optimization
RUL:	Remaining Useful Life
RVM:	Relevance Vector Machine
SBI:	Similarity-Based Interpolation
SOB:	Second-Order Bounds
SOC:	State of Charge
SOH:	State of Health
SOL:	State of Life
SORM:	Second-Order Reliability Method
SVM:	Support Vector Machine

Bibliography

1. Hasofer A.M., and Lind N.C., 1974, "Exact and invariant second-moment code format," *Journal of Engineering Mechanics. Div. ASCE*, v100, p111–121.

2. Breitung K., 1984, "Asymptotic approximations for multinormal integrals," *Journal of Engineering Mechanics, ASCE*, v110, n3, p357–366.

3. Tvedt L., 1984, "Two second-order approximations to the failure probability," Section on structural reliability, A/S vertas Research, Hovik, Norway.

4. Rubinstein R.Y., 1981, "*Simulation and the Monte Carlo method*," Wiley: New York, NY.

5. Fu G., and Moses F., 1988, "Importance sampling in structural system reliability," *Probabilistic Methods in Civil Engineering*, p340–343.

6. Au S.K., and Beck J.L., 2001, "Estimation of small failure probabilities in high dimensions by subset simulation," *Probabilistic Engineering Mechanics*, v16, n4, p263–277.

7. Hurtado J.E., 2007, "Filtered importance sampling with support vector margin: a powerful method for structural reliability analysis," *Structural Safety*, v29, n1, p2–15.

8. Rahman S., and Xu H., 2004, "A univariate dimension-reduction method for multi-dimensional integration in stochastic mechanics," *Probabilistic Engineering Mechanics*, v19, p393–408.

9. Xu H., and Rahman S., 2004, "A generalized dimension-reduction method for multi-dimensional integration in stochastic mechanics," *International Journal of Numerical Methods in Engineering*, v61, p1992–2019.

10. Youn B.D., Zhimin X., and Wang P., 2007, "Eigenvector dimension reduction (EDR) method for sensitivity-free probability analysis," *Structural and Multidisciplinary Optimization*, v37, p13–28.

11. Ghanem R.G., and Spanos P.D., 1991, "*Stochastic Finite Elements: A Spectral Approach*," Springer: New York, NY.

12. Wiener N., 1938, "The homogeneous chaos," *American Journal of Mathematics*, v60, p897–936.

13. Xiu D., and Karniadakis G.E., 2003, "The Wiener–Askey polynomial chaos for stochastic differential equations," *SIAM Journal on Scientific Computing*, v187, p137–167.

14. Le Maitre O., Najm H., Ghanem R., and Knio O., 2004, "Multi-resolution analysis of Wiener-type uncertainty propagation schemes," *Journal of Computational Physics*, v197, p502–531.

15. Wan X., and Karniadakis G.E., 2006, "Multi-element generalized polynomial chaos for arbitrary probability measures," *SIAM Journal of Scientific Computing*, v28, p901–928.

16. Paffrath M., and Wever U., 2007, "Adapted polynomial chaos expansion for

failure detection," *Journal of Computational Physics*, v226, p263–281.

17. Sudret B., and A. Der Kiureghian, 2002, "Comparison of finite element reliability methods," *Probabilistic Engineering Mechanics*, v17, p337–348.
18. Choi S., Grandhi R.V., and Canfield, R.A., 2004, "Structural reliability under non-Gaussian stochastic behavior," *Computers & Structures*, v82, p1113–1121.
19. Choi S., Grandhi R.V., Canfield, R.A., and Pettit, C.L., 2004, "Polynomial Chaos Expansion with Latin Hypercube Sampling for Estimating Response Variability," *AIAA Journal*, v42, n6, p1191–1198.
20. Wan X., and Karniadakis G.E., 2005, "An adaptive multi-element generalized polynomial chaos method for stochastic differential equations," *Journal of Computational Physics*, v209, n2, p617–642.
21. Foo J., Wan X., and Karniadakis G.E., 2008, "The multi-element probabilistic collocation method (ME-PCM): error analysis and applications," *Journal of Computational Physics*, v227, p9572–9595.
22. Foo J., and Karniadakis G.E., 2010, "Multi-element probabilistic collocation method in high dimensions," *Journal of Computational Physics*, v229, p1536–1557.
23. Todor R.A., and Schwab C., 2007, "Convergence rates for sparse chaos approximations of elliptic problems with stochastic coefficients," *IMA Journal of Numerical Analysis*, v27, p232–261.

24. Bieri M., and Schwab C., 2009, "Sparse high order FEM for elliptic sPDEs," *Computer Methods in Applied Mechanics and Engineering*, v198, p1149–1170.

25. Lee S.H., and Chen W., 2007, "A comparative study of uncertainty propagation methods for black-box-type problems," *Structural and Multidisciplinary Optimization*, v37, n3, p239–53.

26. Smolyak S., "Quadrature and Interpolation Formulas for Tensor Product of Certain Classes of Functions," *Soviet Mathematics - Doklady*, Vol. 4, 1963, pp. 240–243.

27. Gerstner T., and Griebel M., "Numerical Integration Using Sparse Grids," *Numerical Algorithms*, Vol. 18, No. 3–4, 1998, pp. 209–232.

28. Barthelmann V., Novak E., and Ritter K., "High Dimensional Polynomial Interpolation on Sparse Grids," *Advances in Computational Mathematics*, Vol. 12, No. 4, 2000, pp. 273–288.

29. Griebel M., "Adaptive Sparse Grid Multilevel Methods for Elliptic PDEs Based on Finite Differences," *Computing*, Vol. 61, No. 2, 1998, pp. 151–179.

30. Xiu D., and Hesthaven J. S., "High Order Collocation Methods for the Differential Equation with Random Inputs," *SIAM Journal on Scientific Computing*, Vol. 27, No. 3, 2005, pp. 1118–1139.

31. Xiu D., "Efficient Collocational Approach for Parametric Uncertainty Analysis," *Communications in Computational Physics*, Vol. 2, No. 2, 2007, pp. 293–309.

32. Nobile F., Tempone R., and Webster C., “A Sparse Grid Collocation Method for Elliptic Partial Differential Equations with Random Input Data,” *SIAM Journal on Numerical Analysis*, Vol. 46, No. 5, 2008, pp. 2309–2345.

33. Grestner T., and Griebel M., “Dimension-Adaptive Tensor-Product Quadrature,” *Computing*, Vol. 71, No. 1, pp. 65–87.

34. Klimke A., “Uncertainty Modeling Using Fuzzy Arithmetic and Sparse Grids,” Ph.D. Dissertation, Universität Stuttgart, Shaker Verlag, Aachen, 2006.

35. Ganapathysubramanian B., and Zabaras N., “Sparse Grid Collocation Schemes for Stochastic Natural Convection Problems,” *Journal of Computational Physics*, Vol. 225, No. 1, 2007, pp. 652–685.

36. Ma X., and Zabaras N., “An Adaptive Hierarchical Sparse Grid Collocation Algorithm for the Solution of Stochastic Differential Equations,” *Journal of Computational Physics*, Vol. 228, No. 8, 2009, pp. 3084–3113.

37. Nobile F., Tempone R., and Webster C., “An Anisotropic Sparse Grid Stochastic Collocation Method for Partial Differential Equations with Random Input Data,” *SIAM Journal on Numerical Analysis*, Vol. 46, No. 5, 2008, pp. 2411–2442.

38. Eldred M.S., and Burkardt J., “Comparison of Non-Intrusive Polynomial Chaos and Stochastic Collocation Methods for Uncertainty Quantification,” *Proceedings of the 47th AIAA Aerospace Sciences Meeting*, No. AIAA-2009-0976, Orlando, Florida, January 5–8, 2009.

39. Eldred M.S., Webster C.G., and Constantine P., “Design Under Uncertainty Employing Stochastic Expansion Methods,” *Proceedings of the 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, No. AIAA-2008-6001, Victoria, British Columbia, Canada, September 10–12, 2008.

40. Haldar A., and Mahadevan S., 2000, “*Probability, Reliability, and Statistical Methods in Engineering Design*,” Wiley, New York, p238–248.

41. Ditlevsen O., and Bjerager P., 1984, “Narrow Reliability Bounds for Structural Systems,” *Journal of Engineering Mechanics*, v110, n5, p671–693.

42. Thoft-Christensen P., and Murotsu Y., 1986, “*Application of Structural Reliability Theory*,” Springer Verlag, Berlin.

43. Karamchandani A., 1987, “*Structural System Reliability Analysis Methods*,” Report No. 83, John A. Blume Earthquake Engineering Center, Stanford University, Stanford, CA.

44. Xiao Q., and Mahadevan S., 1998, “Second –Order Upper Bounds on Probability of Intersection of Failure Events.” *Journal of Engineering Mechanics*, ASCE, v120, n3, p49–57.

45. Ramachandran K., 2004, "System reliability bounds: a new look with improvements," *Civil* Engineering and Environmental Systems, v21, n4, p265–278.

46. Song J., and Der Kiureghian A., 2003, “Bounds on system reliability by linear

programming," *Journal of Engineering Mechanics*, v129, n6, p627–636.

47. Youn B.D., and Wang P., 2009, "Complementary intersection method for system reliability analysis," *Journal of Mechanical Design*, v131, n4, 041004(15).

48. Dekker R., 1996, "Applications of maintenance optimization models: a review and analysis," *Reliability Eng. Syst. Safety*, v51, n3, p229–240.

49. Luo J., Pattipati K.R., Qiao L., and Chigusa S., 2008, "Model-based prognostic techniques applied to a suspension system," *IEEE Transactions on Systems, Man and Cybernetics, Part A,* v38, n5, p1156–1168.

50. Gebraeel N., and Pan J., 2008, "Prognostic degradation models for computing and updating residual life distributions in a time-varying environment," *IEEE Transactions on Reliability*, v57, n4, p539–550.

51. Gebraeel N., Elwany A., and Pan J., 2009, "Residual life predictions in the absence of prior degradation knowledge," *IEEE Transactions on Reliability*, v58, n1, p106–117.

52. Schwabacher M., 2005, "A survey of data-driven prognostics", *Proceedings of AIAA Infotech@Aerospace Conference*, Arlington, VA.

53. Wang T., Yu J., Siegel D., and Lee J., 2008, "A Similarity-Based Prognostics Approach for Remaining Useful Life Estimation of Engineered Systems," *International Conference on Prognostics and Health Management*, Denver, CO, Oct 6-9.

54. Zio E., and Di Maio F., 2010, “A data-driven fuzzy approach for predicting the remaining useful life in dynamic failure scenarios of a nuclear power plant,” *Reliability Engineering and System Safety*, v95, n1, p49–57.

55. Coble J.B., and Hines J.W., 2008, “Prognostic algorithm categorization with PHM challenge application,” *IEEE, International Conference on Prognostics and Health Management*, Denver, CO, Oct 6-9.

56. Heimes, F.O., “Recurrent neural networks for remaining useful life estimation,” *IEEE, International Conference on Prognostics and Health Management*, Denver, CO, Oct 6-9.

57. Heng A., Tan A.C.C., Mathewa J., Montgomery N., Banjevic D., and Jardine A.K.S., 2009, “Intelligent condition-based prediction of machinery reliability,” *Mechanical Systems and Signal Processing*, v23, n5, p1600–1614.

58. Caesarendra W., Widodo A., and Yang B-S., 2010, “Application of relevance vector machine and logistic regression for machine degradation assessment,” *Mechanical Systems and Signal Processing*, v24, n4, p1161–1171.

59. Widodo A., and Yang B-S., 2011, “Application of relevance vector machine and survival probability to machine degradation assessment,” *Expert Systems with Applications*, v38, n3, p2592–2599.

60. Tian Z., Wong L., and Safaei N., 2010, “A neural network approach for remaining useful life prediction utilizing both failure and suspension histories,” *Mechanical Systems and Signal Processing*, v24, n5, p1542–1555.

61. Kozlowski J.D., Watson M.J., Byington C.S., Garga A.K., and Hay T.A., 2001, "Electrochemical cell diagnostics using online impedance measurement, state estimation and data fusion techniques," *Proceedings of 36th Intersociety Energy Conversion Engineering Conference,* Savannah, Georgia.

62. Goebel K., Eklund N., and Bonanni P., 2006, "Fusing Competing Prediction Algorithms for Prognostics," *Proceedings of 2006 IEEE Aerospace Conference*, New York.

63. Saha B., Goebel K., Poll S., and Christophersen J., 2009, "Prognostics methods for battery health monitoring using a Bayesian framework," *IEEE Transaction on Instrumentation and Measurement*, v58, n2, p291–296.

64. Plett G.L., 2004, "Extended Kalman Filtering for Battery Management Systems of LiPb-Based HEV Battery Packs Part 1. Background," *Journal of Power Sources*, v134, n2, p252–261.

65. Finger E.P., and Marwell E.M., 1977, Battery control system for battery operated vehicles, Patent No. 538466.

66. Çadırcı Y., and Özkazanç Y., 2004, "Microcontroller-based on-line state-of-charge estimator for sealed lead-acid batteries," *Journal of Power Sources*, v129, n2, p330–342.

67. Huet F., 1998, "A review of impedance measurements for determination of the state-of-charge or state-of-health of secondary batteries," *Journal of Power Sources*, v70, n1, p59–69.

68. Mauracher P., and Karden E., 1997, "Dynamic modelling of lead/acid batteries using impedance spectroscopy for parameter identification," *Journal of Power Sources*, v67, n1–2: p69–84.

69. Hammouche A., Karden E., and De Doncker R.W., 2004, "Monitoring state-of-charge of Ni-MH and Ni-Cd batteries using impedance spectroscopy," *Journal of Power Sources*, v127, n1–2, p105–111.

70. Blanke H., et al., 2005, "Impedance measurements on lead-acid batteries for state-of-charge, state-of-health and cranking capability prognosis in electric and hybrid electric vehicles," *Journal of Power Sources*, v144, n2, p418–425.

71. Christianson C.C. and Bourke R.F., 1976, Battery state of charge gauge, Patent No. 3946299.

72. Tanjo, Y., et al., 2000, State of charge indicator, Patent No. 6127806.

73. Pop V., Bergveld H.J., Notten P.H.L., and Regtien P.P.L., 2005, "Smart and accurate State-of-Charge indication in portable applications," *IEEE Power Electronics and Drive Systems* v1, p262–267.

74. Pop V., Bergveld H.J., Notten P.H.L., Op het Veld J.H.G., and Regtien P.P.L., 2009, "Accuracy analysis of the State-of-Charge and remaining run-time determination for lithium-ion batteries Measurement," *Measurement*, v42, n8, p1131–1138.

75. Plett G.L., 2004, "Extended Kalman filtering for battery management systems of

LiPB-based HEV battery packs Part 2. Modeling and identification," *Journal of Power Sources*, v134, n2, p. 262–276.

76. 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*, v134, n2, p277–292.

77. Plett G.L., 2006, "Sigma-point Kalman filtering for battery management systems of LiPB-based HEV battery packs Part 1: Introduction and state estimation," *Journal of Power Sources*, v161, n2, p1356–1368.

78. 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*, v161, n2, p1369–1384.

79. Santhanagopalan S., and White R.E., 2006, "Online estimation of the state of charge of a lithium ion cell," *Journal of Power Sources*, v161, n2, p1346–1355.

80. Jeong K.S., Lee W.Y., and Kim C.S., 2005, "Energy management strategies of a fuel cell/battery hybrid system using fuzzy logics," *Journal of Power Sources*, v145, n2, p319–326.

81. Salkind A.J., et al., 1999, "Determination of state-of-charge and state-of-health of batteries by fuzzy logic methodology," *Journal of Power Sources*, v80, n1–2, p293–300.

82. Gerard O., Patillon J.N., and d'Alche-Buc F., 1997, "Neural network adaptive

modelling of battery discharge behavior," Lecture Notes in Computer Science, v1327, p1095–1100.

83. Grewal S., and Grant D.A., 2001, "A novel technique for modelling the state of charge of lithium ion batteries using artificial neural networks," in Proceedings of International Telecommunications Energy Conference (IEEE), Oct 14-18, p174–179.

84. Piller S., Perrin M., and Jossen A., 2001, "Methods for state-of-charge determination and their applications," *Journal of Power Sources*, v96, n1, p113–120.

85. Ramadass P., Gomadam P.M., White R.E., and Popov B.N., 2004, "Development of first principles capacity fade model for Li-ion cells," *Journal of the Electrochemical Society*, v151, A196–A203.

86. Pop V., Bergveld H.J., Danilov D., Regtien P.P.L., and Notten P.H.L., 2008, "*Battery Management Systems: Accurate State-Of-Charge Indication for Battery-Powered Applications*," Springer Verlag, London.

87. Lee S., Kim J., Lee J., and Cho B.H., 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*, v185, n2, p1367–1373.

88. Zhang Q., and White R.E., 2008, "Capacity fade analysis of a lithium ion cell," *Journal of Power Sources*, v179, n2, p793–798.

89. Zhang Q., and White R.E., 2008, “Calendar life study of Li-ion pouch cells Part 2: Simulation,” *Journal of Power Sources*, v179, n2, p785–792.

90. Saha B., and Goebel K., 2009, “Modeling Li-ion battery capacity depletion in a particle filtering framework,” *In Proceedings of Annual Conference of the PHM Society*, San Diego, CA, Sep. 27-Oct. 1.

91. Webb C.T., 2007, “What Is The Role Of Ecology in Understanding Ecosystem Resilience?” *BioScience*, v57, n6, p470–471.

92. Hartvigsen G., Kinzig A., and Peterson G., 1998, “Complex Adaptive Systems: Use and Analysis of Complex Adaptive Systems in Ecosystem Science: Overview of Special Section,” *Ecosystems*, v1, n5, p427–430.

93. Kerkhoff A.J., and Enquist B.J., 2007, “The Implications of Scaling Approaches for Understanding Resilience and Reorganization in Ecosystems,” *Bioscience*, v57, n6, p489–499.

94. Luthar S.S., Cicchetti D., and Becker B., 2000, “The Construct of Resilience: a Critical Evaluation and Guidelines for Future Work,” *Child Development*, v71, n3, p543–562.

95. Bonanno G.A., Moskowitz J.T., Papa A., and Folkman S., 2005, “Resilience to Loss in Bereaved Spouses, Bereaved Parents, and Bereaved Gay Men,” *Journal of Personality and Social Psychology*, v88, n5, p827–843.

96. Bonanno G.A., and Galea S., 2007, “What Predicts Psychological Resilience

After Disaster? The Role of Demographics, Resources, and Life Stress," *Journal of Consulting and Clinical Psychology*, v75, n5, p671–682.

97. Luthar S.S., 1999, *Poverty and Children's Adjustment.* Newbury Park, CA: Sage.

98. Sheffi Y., 2005, *The Resilient Enterprise: Overcoming Vulnerability for Competitive Enterprise*. MIT Press.

99. Hollnagel E., Woods D.D., and Leveson N. (Eds.), 2006, *Resilience Engineering: Concepts and Precepts*, Ashgate.

100. Hollnagel E., 2006, "Achieving System Safety by Resilience Engineering," *In: The 1st IET International Conference on System Safety*, June 6-8, London, UK, p184–195.

101. Du X., and Chen W., 2004, "Sequential Optimization and Reliability Assessment Method for Efficient Probabilistic Design," *ASME Journal of Mechanical Design*, v126, n2, p225–233.

102. Youn B.D., Choi K.K., and Du L., 2005, "Enriched Performance Measure Approach (PMA+) for Reliability-Based Design Optimization," *AIAA Journal*, v43, n4, pp. 874–884.

103. McDonald M., and Mahadevan S., 2008, "Reliability-Based Optimization with Discrete and Continuous Decision and Random Variables," *ASME Journal of Mechanical Design*, v130, n6, p. 061401.

104. Kim C., and Choi K. K., 2008, "Reliability-Based Design Optimization Using

Response Surface Method with Prediction Interval Estimation," *ASME Journal of Mechanical Design*, v130, n12, p. 121401.

105. Cameron R.H., and Martin W.T., 1947, "The orthogonal development of nonlinear functionals in series of Fourier–Hermite functionals," *Annals of Mathematics*, v48, p385–392.

106. 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*, v173, p481–511.

107. 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*, v181, p9–44.

108. Field R.V., 2002, "Numerical methods to estimate the coefficients of the polynomial chaos expansion," *Proceedings of the 15th ASCE engineering mechanics conference*.

109. Kohavi R., 1995, "A study of cross-validation and bootstrap for accuracy estimation and model selection," *Proceedings of the International Joint Conference on Artificial Intelligence - IJCAI'95'*.

110. Noh Y., Choi K.K., and Du L., "Selection of copula to generate input joint CDF for RBDO," *ASME International Design Engineering Technical Conference & Computers and Information in Engineering Conference*, IDETC2008-49494, Brooklyn, New York, United States, Aug. 3-6 2008.

111. Roser B.N., 1999, "*An Introduction to Copulas*," Springer, New York.

112. Rosenblatt M. Remarks on a multivariate transformation. Annals of Mathematical Statistics 1952; 23:470–472.

113. Rahman S., 2009, "Stochastic sensitivity analysis by dimensional decomposition and score functions," *Probabilistic Engineering Mechanics*, v24, p278–287.

114. Rahman S., 2006, "A dimensional decomposition method for stochastic fracture mechanics," *Engineering fracture mechanics*, v73, n15, p2093–2109.

115. Lee I., Choi K.K., Du L., and Gorsich D., 2008, "Dimension reduction method for reliability-based robust design optimization," *Computers and Structures*, v86 n13-14, p1550–1562.

116. Noh Y., Choi K.K., and Du L., 2009, "Reliability-based design optimization of problems with correlated input variables using a Gaussian Copula," *Structural and Multidisciplinary Optimization*, v38, n1, p1–16.

117. Youn B.D., Choi K.K., Gu L., and Yang R.J., 2004, "Reliability-based design optimization for crashworthiness of side impact," *Structural and Multidisciplinary Optimization*, v26, i3–4, p272–283.

118. Youn B.D., Zhimin X., and Wang P., 2007, "Reliability-based robust design optimization using the eigenvector dimension reduction (EDR) method," *Structural and Multidisciplinary Optimization*, v37, p475–492.

119. Kvasov B.I., 2000, "*Methods of Shape-Preserving Spline Approximation*,"

Singapore: World Scientific Publ. Co. Inc..

120. Abramowitz M., and Stegun I.A., 1972, "*Handbook of Mathematical Functions*," 9th ed. New York: Dover Publications, Inc..

121. Lee C.Y., 1959, "Representation of switching circuits by binary-decision programs," *Bell Labs Technical Journal*, v38, p985–999.

122. Akers B., 1978, "Binary decision diagrams," *IEEE Transactions on Computers*, v27, n6, p509–516.

123. Mahadevan S., Zhang R.X., and Smith N., 2001, "Bayesian networks for system reliability reassessment," *Structural Safety*, v23, n3, p231–251.

124. Rivera H.L., Garcia-Souto J.A., and Sanz J., 2000, "Measurement of mechanical vibrations at magnetic cores of power transformers with fiber-optic interferometric intrinsic sensor," *IEEE Journal on Selected Topics in Quantum Electronics*, v6, n5.

125. Kim Y.D., Shim J.M., Park W.Y., Kim S.J., Hyun D.S., and Lee D.D., 2009, "Structure vibration analysis of a power transformer (154kV/60MVA/single phase)," *International Journal of Electrical Power and Energy Systems Engineering* v2, n4.

126. Tipping M.E., 2001, "Sparse Bayesian learning and the relevance vector machine," *Journal of Machine Learning Research*, v1, p211–244.

127. Smola A.J., and Schölkopf B., 2004, "A tutorial on support vector regression,"

Statistics and Computing, v14, n3, p199–222.

128. Cernansky M., Makula M., and Cernansky L., 2007, "Organization of the state space of a simple recurrent network before and after training on recursive linguistic structures," *Neural Networks*, v20, n2, p236–244.

129. Saxena A., and Goebel K., "Damage propagation modeling for aircraft engine run-to-failure simulation," *IEEE, International Conference on Prognostics and Health Management*, Denver, CO, Oct 6-9.

130. Lee W.R., Jung S.W., Yang K.H. and Lee J.S., 2005, "A study on the determination of subjective vibration velocity ratings of main transformers under operation in nuclear power plants," *12th International Congress on Sound and Vibration*, Paper 1017, Portugal, July 11-14.

131. Wang M., Vandermaar A. J., and Srivastava K. D., 2002, "Review of condition assessment of power transformers in service," *IEEE Electrical Insulation Magazine*, v18, n6, p12–25.

132. Ji S., Luo Y., and Li Y., 2006, "Research on Extraction Technique of Transformer Core Fundamental Frequency Vibration Based on OLCM", *IEEE Transactions on Power Delivery*, v21, n4, p1981–1988.

133. García B., Member, Burgos J.C., and Alonso Á.M., 2006, "Transformer Tank Vibration Modeling as a Method of Detecting Winding Deformations—Part II: Experimental Verification," *IEEE Transactions on Power Delivery*, v21, n1, p164–169.

134. Byington C.S., Watson M., and Edwards D., 2004, “Data-driven neural network methodology to remaining life predictions for aircraft actuator components,” *Proceedings of IEEE Aerospace Conference*, March 6-13, v6, p3581–3589, DOI: 10.1109/AERO.2004.1368175.

135. Liu J., Saxena A., Goebel K., Saha B., and Wang W., 2010, “An Adaptive Recurrent Neural Network for Remaining Useful Life Prediction of Lithium-ion Batteries,” Proceedings of Annual Conference of the PHM Society, October 10-16, Portland, Oregon.

136. M. Hagan, H. Demuth, and M. Beale. *Neural Network Design*. PWS Publishing, Boston, MA, 1996.

137. Wu S.J., Gebraeel N., Lawley M.A., and Yih Y., 2007, “A neural network integrated decision support system for condition-based optimal predictive maintenance policy,” *IEEE Transactions on Systems Man and Cybernetics Part A – Systems and Humans*, v37, n2, p226–236.

138. Park, J., and Sandberg, I.W., 1993, “Approximation and radial-basis function networks,” *Neural computation*, v5, p305–316.

139. Schwenker, F., Kestler, H., and Palm, G., 2001, “Three learning phases for radial basis function networks,” *Neural Networks*, v14, n4–5, p439–458.

140. McFadden, P.D., and Smith, J.D., 1984, “Model for the vibration produced by a single point defect in a rolling element bearing,” *Journal of Sound and Vibration*, v96, n1, p69–82.

141. Wang, Y.F., and Kootsookos, P.J., 1998, “Modeling of low shaft speed bearing faults for condition monitoring,” *Mechanical Systems and Signal Processing*, v12, n3, p415–426.

142. Shao, Y., and Nezu, K., 2000, “Prognosis of remaining bearing life using neural networks,” *Proceedings of the Institute of Mechanical Engineer, Part I, Journal of Systems and Control Engineering*, v214, n3, p217–230.

143. Haykin S., 2001, “*Kalman Filtering and Neural Networks*,” Wiley/Inter-Science, New York.

144. Wan E., Nelson A., 2001, “*Dual extended Kalman filter methods*,” in: Haykin S. (Ed.), “*Kalman Filtering and Neural Networks*,” Wiley/Inter-Science, New York, p123–174.

145. Williams R.J., and Zipser D., 1989, “A learning algorithm for continually running fully recurrent neural networks,” *Neural Computation*, v1, n2, p270–280.

146. Plett G., 2005, “Results of temperature-dependent LiPB cell modeling for HEV SOC estimation,” *In Proceedings of the 21st Electric Vehicle Symposium (EVS21)*, Monaco, April 2-6.

147. Frischermeier S., 1997 “Electrohydrostatic Actuators for Aircraft Primary Flight Control - Types, Modelling and Evaluation”, *In Proceedings of the Fifth Scandinavian International Conference on Fluid Power*, May 28-30, Linköping, Sweden.

148. Botten S., Whitley C., and King A., 2000, “Flight Control Actuation Technology for Next-Generation All-Electric Aircraft,” *Technology Review Journal - Millenium Issue*, Fall/Winter.

149. Osder S., 1999, “Practical View of Redundancy Management Application and Theory,” *AIAA Journal of Guidance, Control, and Dynamics*, v22, n1, Jan-Feb, p12–21.

150. Frangopol D.M., Lin K-Y., Estes A., 1997, “Life-Cycle Cost Design of Deteriorating Structures,” *Journal of Structural Engineering,* v123, n10, p1390–1401.

151. Tillman F.A., Hwang C.L., and Kuo W., 1977, “Determining Component Reliability and Redundancy for Optimum System Reliability,” *IEEE Transactions on Reliability*, v26, n3, p162–165.

152. Dhingra A.K., 1992, “Optimal Apportionment of Reliability and Redundancy in Series Systems under Multiple Objectives,” *IEEE Transactions on Reliability*, v41, n4, p576–582.

153. Gen M., and Cheng R., 2000, *Genetic Algorithms and Engineering Optimization*. New York: John Wiley & Sons.

154. The LMS Imagine.Lab AMESim Suite, 2010. http://www.lmsintl.com/imagine-amesim-1-d-multi-domain-system-simulation.

155. Crowther W.J., Edge K.A., Burrows C.R., Atkinson R.M., and Wollons D.J.,

1998, “Fault Diagnosis of a Hydraulic Actuator Circuit using Neural Networks an Output Vector Space Classification approach,” *Proceedings of the Institution of Mechanical Engineers - Part I: Journal of Systems & Control Engineering*, v212, n11, p57–68.

156. An L., and Sepehri N., 2006, “Hydraulic Actuator Leakage Quantification Scheme Using Extended Kalman Filter and Sequential Test Method,” *In Proceedings of American Control Conference*, June 14-16, p6.

157. Xu H., and Rahman S., 2005, “Decomposition Methods for Structural Reliability Analysis,” *Probabilistic Engineering Mechanics*, v20, n3, p239–250.