PREDICTIVE RELIABILITY ANALYSIS AND MAINTENANCE PLANNING OF COMPLEX SYSTEMS WORKING UNDER DYNAMIC OPERATING CONDITIONS

A Dissertation
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By
Ameneh Forouzandeh Shahraki

In Partial Fulfillment of the Requirements
for the Degree of
DOCTOR OF PHILOSOPHY

Major Department:
Industrial and Manufacturing Engineering

November 2022

Fargo, North Dakota

North Dakota State University
Graduate School

Title

PREDICTIVE RELIABILITY ANALYSIS AND MAINTENANCE PLANNING OF COMPLEX SYSTEMS WORKING UNDER DYNAMIC OPERATING CONDITIONS

By

Ameneh Forouzandeh Shahraki

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

DOCTOR OF PHILOSOPHY

SUPERVISORY COMMITTEE:

Dr. Om Prakash Yadav
Chair

Dr. Simone Ludwig

Dr. Nita Yodo

Dr. Chrysafis Vogiatzis

Approved:

11/09/2022	Dr. David Grewell
Date	Department Chair

ABSTRACT

Predictive analytics has multiple facets that range from failure predictability and optimal asset management to high-level managerial insights. Predicting the failure time of assets and estimating their reliability through efficient prognostics and reliability assessment framework allow for appropriate maintenance actions to avoid catastrophic failures and reduce maintenance costs. Most of the systems used in the manufacturing and service sectors are composed of multiple interdependent components. Moreover, these systems experience dynamic operating conditions during their life. The dynamic operating conditions and the system complexity pose three challenging questions: how to perform the prognostic and reliability assessment of a complex multi-component system, how to perform the prognostic and reliability assessment of a system functioning under dynamic operating conditions, and how to use the condition based and reliability assessment data to find the optimal maintenance strategy for complex systems.

This dissertation investigates five tasks to address these challenges. (1) To capture the stochastic dependency between interdependent components of a system through a continuous time Markov process with the transition rate depending on the state of all components of the system. This technique helps get an accurate estimation of system reliability. (2) To propose a framework based on instance-based learning to predict the remaining useful life (RUL) of a complex system. This technique can be used for highly complex systems with no need of having prior expertise on the system behavior. (3) To incorporate time-varying operating conditions in the prognostics framework through a proportional hazards model with external covariates dependent on the operating condition and internal covariates dependent on the degradation state of the system. (4) To propose a prognostic framework based on deep learning to predict the RUL of a system working in dynamic operating conditions. This framework has two main steps: first identifying the

degrading point and developing the Long Short-Term Memory model to predict the RUL. (5) To propose an efficient algorithm for reliability analysis of a phased-mission system, its behavior changes at different phases during the mission. This technique accounts for imperfect fault coverage for the components to get accurate reliability analysis.

ACKNOWLEDGMENTS

It would not have been possible to write this doctoral thesis without the support of the amazing people around me. It is a great pleasure to express my sincere gratitude to all of them. A special thanks to Dr.Yadav for his help of reflecting, reading, encouraging, and most of all patience throughout the entire process. I would like to express my special thanks to other three committee members, Dr.Ludwig , Dr. Yodo, and Dr. Vogiatzis for their time, support, and agreeing to serve on my committee.

I would like to acknowledge and thank my school division for allowing me to conduct my research and providing any assistance requested.

DEDICATION

To my beloved family for their love, endless support, and encouragement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

LIST OF ABBREVIATIONS

AFTM	Accelerated Failure Time Model
AI	Accuracy Index
ANN	Artificial Neural Network
BC	Boundary Condition
BSS	Binary State System
CBM	Condition-Based Maintenance
CEM	Cumulative Exposure Model
CTMC	Continuous Time Markov Chain
CV	Constraint Violation
DP	Degrading Point
ESR	Equivalent Series Resistance
EWMA	Exponential Weighted Moving Average
FSB	Fuzzy Similarity-Based
GA	Genetic Algorithm
GMSknS	Generalized Multi-State *K*-Out-Of-*N* System
GRU	Gated Recurrent Unit
HMM	Hidden Markov Model
HSMM	Hidden Semi-Markov Model
IFC	Imperfect Fault Coverage
IG	Inverse Gaussian
KNN	K-Nearest-Neighbor
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MCS	K-Nearest-Neighbor

MPE	Monte Carlo Simulation
MSS	Multi-State System
NN	Neural Networks
NSGAII	Non-Dominated Sorting Genetic Algorithm II
PH model	Proportional Hazards Model
PHM	Prognostics and Health Management
PMS	Phased Mission System
RMSE	Root Mean Square Error
RMSprop	Resilient Mean Square Backpropagation
RNN	Recurrent Neural Networks
RUL	Remaining Useful Life
S-dependent	Stochastic Dependence
SEA	Simple and Efficient Algorithm
SVM	Support Vector Machine
WMAE	Weighted Mean Absolute Error

1. INTRODUCTION

1.1. Overview

Reliability and safety of critical engineering systems are key aspects to be considered to ensure proper operation, prevent undesirable situations, minimize risks, and reduce the life cycle costs of the systems. The engineering systems are subjected to a gradual degradation process as the result of usage and age, which considerably reduces their efficiency and eventually causes the systems to be unable to perform the required functions for the intended period of use [1]. Failure to do so may lead to system failures and severe damage to the environment and society, such as loss of lives, environmental contamination, and substantial financial costs. The following examples demonstrate the damage caused by systems' failures.

On September 2008, the malfunction of a turbine generator at the D.C. Cook Nuclear Power Plant resulted in a fire, which led to a manual plant shutdown and a massive loss of revenues for the one-year outage [1]. As another example, in July 2013, the Lac-Mégantic rail disaster occurred in Canada that resulted in the damage of 63 tank cars and the release of about 6 million liters of crude oil [2]. The train derailment and explosion caused 47 deaths, destroyed 30 buildings, and contaminated much of the downtown core. Most recently, the Texas blackout of 2021 caused a widespread power outage throughout the state of Texas [3]. More than 4.5 million homes and businesses were left without power for several days. The power outage had a broad range of impacts on the communication network, emergency services, water treatment, supply and distribution, food distribution, banking services, traffic services, and government services. The blackout caused at least 111 deaths and cost more than $195 billion, making it the costliest disaster in Texas history. The failure of such systems was due to several reasons ranging from faulty design and poor equipment quality to not considering the effects of environmental conditions on system performance.

One of the best ways to keep a system in service and prevent such failures is to perform proper maintenance activities that assure a satisfactory level of reliability throughout the life of the system. Traditional maintenance approaches such as corrective maintenance and scheduled maintenance take place either at breakdowns or at periodic intervals regardless of the health condition of the system and its components [4]. These approaches would have still exposed the system to failures and unnecessary maintenance actions. On the other hand, condition-based data related to the health condition of the system can be used to identify the degradation state and predict the remaining useful life (RUL), i.e., the amount of time the system will continue to perform its functions according to design specifications without catastrophic failure. Knowledge of the degradation state and RUL of a system can be used to plan efficient maintenance strategies. The advancement in technology brings sophistication that poses many challenging and interesting questions about condition monitoring and predictive analytics of complex engineering systems, including predicting the RUL and maintenance planning. Solving these questions has recently gained much attention to improve the accuracy of predictions to achieve goals ranging from decreasing the system life cycle costs to protecting human life. We address three main questions in our research.

The first question is how to perform the prognostic and reliability assessment of a complex system composed of multiple components that may be interdependent. In complex systems such as power grids, the failure or degradation of a component may lead to the failure of other interdependent components or accelerate their degradation processes. Existing reliability assessment approaches usually study simple one-unit systems or assume independence between system components that leads to inefficient maintenance plans for complex systems [5]. The second question is how to perform the prognostic and reliability assessment of a system working under dynamic or time-varying operating conditions. The engineering systems in industrial applications usually work under dynamic operating conditions caused by environmental conditions such as temperature and humidity or operating profiles. For example, the lithium-ion batteries of electric vehicles are often influenced by time-varying ambient temperature and other factors like the discharge-charge rate. The temperature changes dynamically between day and night, between four seasons and different places, and discharge-charge rates vary from one user to another. Therefore, it is necessary to capture the effect of dynamic operating conditions on the degradation and failure processes to get a more accurate estimation of the system reliability and RUL. The third question is how to use the condition-based and reliability assessment data to find the optimal maintenance strategies of complex systems. Condition-based maintenance (CBM) based on condition-based data can effectively improve system reliability at reduced costs. Research on CBM has been rapidly growing due to the advancement of condition monitoring technologies. Because of the complexity of real-world systems, however, the application of CBM in practice is lagging behind.

The mentioned challenging questions have enticed us to focus on developing models for prognostics, reliability assessment, and maintenance planning that can capture the dependency between interdependent components and incorporate the effect of dynamic operating conditions. The techniques and approaches developed in this research can be adopted and applied in numerous industrial applications. In 1981, the maintenance costs in the United States were estimated at 600 billion dollars, a figure that at least doubled in the subsequent 40 years, that nearly one-half of the costs were because of ineffective maintenance [6]. Certainly, this research has a significant potential to reduce maintenance costs and save billions of dollars.

1.2. Research Challenges

Prognostic, reliability assessment, and maintenance planning of complex systems that are functioning in dynamic operating conditions are important yet challenging research topics. In this section, we discuss their common challenges.

Engineering systems often consist of many components, where different inter-component dependencies such as stochastic, structural, economic, and source dependence affect the availability of the system [7]–[10]. Stochastic dependence means the state of one component influences the lifetime distributions of other components. Structural dependence applies if components structurally form a part, so maintenance of a failed component also implies maintenance of other components. In economic dependence, maintenance costs can be reduced by performing maintenance of multiple components jointly instead of separately. Source dependence means multiple components are connected through, e.g., shared spares, tools, or maintenance workers. This research focuses on stochastic dependence (S-dependence) between components. We assume that either the failure or degradation process of a component affects the performance of other components in the system by accelerating their degradation process. Due to the presence of S-dependence, prognostics, reliability assessment, and maintenance planning of the system have challenging issues.

Limited literature considers the S-dependence between components of a complex system [5][8]–[10]. However, some simple assumptions make them inappropriate for real-world problems. First, it is assumed that the interaction between system components is constant during the time and independent of the age and state of the system and its components. For many systems, the interaction effect between components will change over time. For example, the failure of a component has higher impact on a degraded component compared to a new one. Further, the interaction effect is stochastic in nature, and may be influenced by other factors such as system environmental/operational conditions, including the state of non-critical components in the system. Second, most of the existing approaches for reliability assessment and RUL prediction characterize the S-dependence between components by a specific model. They assume that the influence of each component on other components is observable and can be easily estimated. For a complex system, however, it is hard or even impossible to establish such specific models that can capture true interactions between components. This can be attributed to lack of knowledge about the system's dynamics and the existence of many unknown factors that complicate the interaction among components. Therefore, it is necessary to develop methods that can face the challenges brought up by the complexity of real-world systems.

Beside the system complexity, the dynamic operating conditions experienced by many practical systems throughout their life poses some challenges for prognostic, reliability assessment, and maintenance planning. The factors of the operating conditions include environmental conditions such as ambient temperature, humidity, pressure, vibration, shocks, and any other stresses or operational profiles such as speed, mission load, use rate, and so on. Most of the conventional predictive analytics rely on simple assumptions that the operating conditions are temporarily constant or irrelevant to the system health condition [11]. However, in practice, the system usually operates under complex and uncertain conditions that affect its useful lifetime. For example, increasing the load and speed of a rotating machine may accelerate the degradation of its components and causes its earlier failure. Therefore, capturing the effects of the operating conditions is an important issue, especially for the safety-critical systems, to get a reasonable assessment of the reliability and find the optimal maintenance strategy. The limited literature that considers the effects of the operating conditions has key issues that we address three of them.

First, the prognostics, reliability assessment, and maintenance optimization are mostly performed assuming the operating condition in the future is known and deterministic. This restrictive assumption leads to overestimation or underestimation of system reliability for the cases that the future operating condition evolves stochastically. Thus, it is necessary to model the evolution of dynamic operating conditions and capture its effects on the failure or degradation process of the system to get a reasonable estimation of system reliability.

Second, some studies have made simple assumptions related to the statistical models of the degradation process, dynamic operating conditions, and the influence of operating conditions on the degradation and failure processes. It is challenging to obtain accurate and sufficient time-to-failure data to build and validate these statistical models, especially for highly reliable systems. To model the changes of the dynamic operating conditions in future and incorporate them in an appropriate way, it is required to have a good understanding of their real changes in the future. Many studies use the continuous time Markov chain (CTMC) to model the progression of dynamic operating conditions. The estimation of the CTMC parameters needs sufficient historical data, and it is not appropriate for describing some practical conditions. Moreover, it is hard to model the effect of the operating condition on the system's degradation or failure process for the cases where the effect may change during the life of the system. For example, a degraded battery compared to a new battery is more vulnerable to harsh temperatures. Therefore, modeling the influence of operating conditions without considering the current state of the system makes them inappropriate for real-world applications.

Third, the exact reliability evaluation of a phased mission system (PMS) is a time-consuming and complicated task. A PMS is a system involving multiple, consecutive, and non-overlapping phases of tasks during its mission, which abounds in complex technological systems such as aerospace systems, nuclear power plants, and high-performance computing systems. Due to the different operating conditions of each phase, the variation in system structure between phases, and the dependencies across different phases for each component of the system, the reliability evaluation of a PMS is a challenging problem. Although many approaches have been proposed for the reliability assessment of PMS, there is still the need for a robust and efficient algorithm relaxing the limitations of previous studies.

1.3. Proposed Research

This dissertation focuses on solving the forementioned challenges. Our research contributes to data-driven predictive analytics for effective prognostics, reliability assessment, and maintenance planning of complex engineering systems that are operating in the field. This dissertation is organized into eight chapters. Figure 1.1 provides an overview of two main research topics and the related chapters.

CHAPTER 2: Degradation modeling literature review

Degradation modeling is an effective approach for reliability assessment, remaining useful life prediction, maintenance planning, and prognostics and health management. In this chapter, we present a comprehensive review of existing data-driven degradation modeling approaches commonly used in engineering applications.

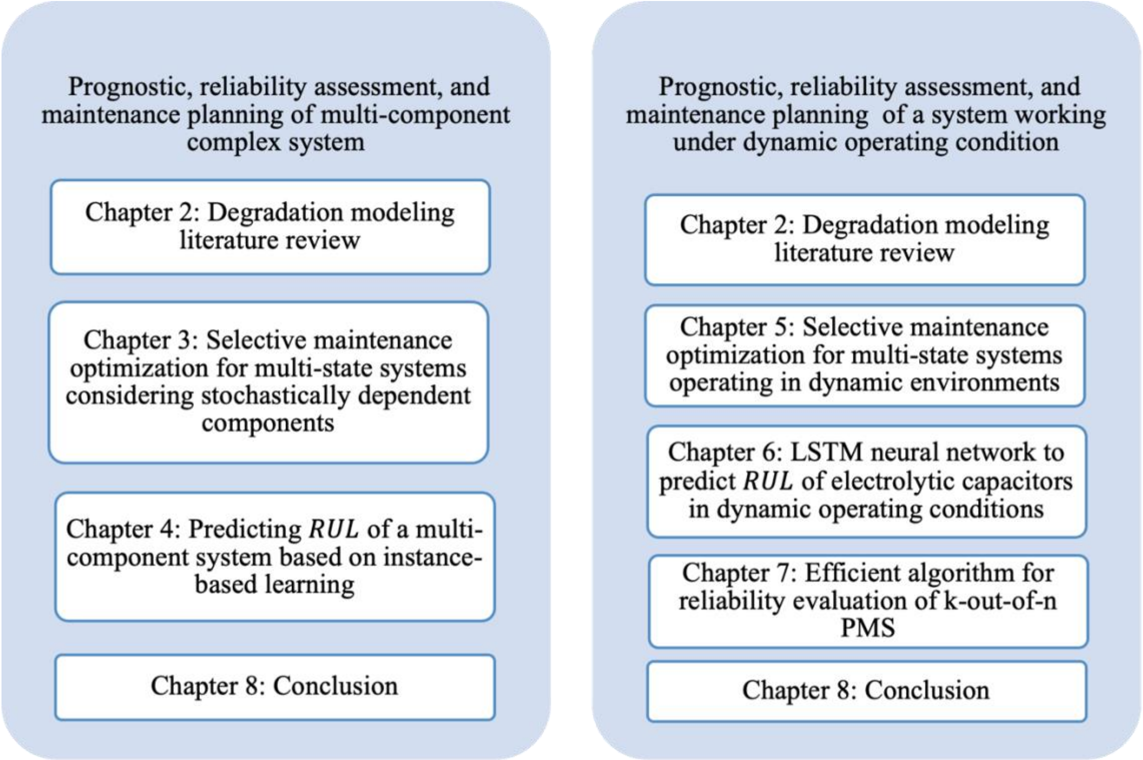

Prognostic, reliability assessment, and maintenance planning of multi-component complex system
Chapter 2: Degradation modeling literature review
Chapter 3: Selective maintenance optimization for multi-state systems considering stochastically dependent components
Chapter 4: Predicting RUL of a multi-component system based on instance-based learning
Chapter 8: Conclusion
Prognostic, reliability assessment, and maintenance planning of a system working under dynamic operating condition
Chapter 2: Degradation modeling literature review
Chapter 5: Selective maintenance optimization for multi-state systems operating in dynamic environments
Chapter 6: LSTM neural network to predict RUL of electrolytic capacitors in dynamic operating conditions
Chapter 7: Efficient algorithm for reliability evaluation of k-out-of-n PMS
Chapter 8: Conclusion

Figure 1.1: Overview of the research

CHAPTER 3: Selective maintenance optimization for multi-state systems considering stochastically dependent components and stochastic imperfect maintenance actions

In this chapter, we capture the S-dependency between interdependent components of a complex system through a continuous time Markov process with the transition rate depending on the state of system components. The key advantage of the proposed technique is that it considers stochastic interaction between components by capturing the effects of environmental/operational conditions and the state of non-critical components in the system. This technique helps get an accurate estimation of system reliability at the end of its mission. Then, we find the optimal maintenance strategy for complex systems considering stochastic imperfect maintenance actions along with the do-nothing and perfect maintenance actions. We formulate a selective maintenance optimization model with two objective functions to find robust solutions with higher reliability and reduced uncertainty.

CHAPTER 4: Predicting remaining useful life of a multi-component system based on instance-based learning

We propose a framework based on instance-based learning to predict RUL of a complex system that establishing a specific model to capture true interactions between its components is hard or even impossible. To predict the RUL of the operating system, we use the similarity between the degradation process of the operating system and the degradation process of the systems in the training set. The key advantage of this technique is that it can be used for highly complex systems with no need of having prior expertise on the system behavior and the S-dependency between components.

CHAPTER 5: Selective maintenance optimization for multi-state systems operating in dynamic environments

In this chapter, we incorporate time-varying operating conditions in the prognostics framework through a proportional hazards (PH) model with external covariates dependent on the operating condition and internal covariates dependent on the degradation state of the system. This technique allows incorporating the effect of time-varying operating conditions on the degradation rate of a system as well as the age and degradation state of the system to get reasonable estimation of the system reliability. We consider the fact that the influence of the operation condition on the degradation process depends on the current degradation state of the system. The proposed model is used in the selective maintenance optimization model to determine the best maintenance strategy of a multi-state series system.

CHAPTER 6: Using LSTM neural network to predict remaining useful life of electrolytic capacitors in dynamic operating conditions

To overcome the challenge of predicting the RUL of a complex system working in dynamic operating conditions, we propose a prognostic framework based on deep learning. This framework has two main steps, first identifying the degrading point and then developing the Long Short-Term Memory (LSTM) model to predict the RUL. This general framework can be used for many complex systems without the necessity of assuming any particular type of degradation process and, therefore, avoiding the requirement of establishing a specific link between model parameters and operating conditions. The effectiveness of the proposed framework is demonstrated by utilizing the degradation and temperature time series data of aluminum electrolytic capacitors.

CHAPTER 7: Efficient algorithm for reliability evaluation of k-out-of-n PMS

In this chapter, we propose an efficient algorithm for the reliability analysis of a PMS. The proposed recursive technique considers the dynamic of the system in each phase and the statistical dependence of components states across the phases, and also accounts for imperfect fault coverage for the components to get accurate reliability analysis.

CHAPTER 8: Conclusion and future work

This chapter summarizes the whole work and provides the future research direction.

1.4. Statement of Authorship

Most of the content of the chapters presented in this dissertation is based on the following journal and conference papers, which are published, accepted, or submitted for publication:

- A. F. Shahraki, O. P. Yadav, and H. Liao, “A review on degradation modelling and its engineering applications,” published in *International Journal of Performability Engineering*, 2017. [Used in Chapter 2]
- A. F. Shahraki, O. P. Yadav, and C. Vogiatzis, “Selective maintenance optimization for multi-state systems considering stochastically dependent components and

stochastic imperfect maintenance actions," published in *Reliability Engineering and System Safety Journal*, 2020. [Used in Chapter 3]

- A. F. Shahraki, A. Roy, O. P. Yadav, and A. P. S. Rathore, "Predicting remaining useful life based on instance-based learning," published in 2019 *Annual Symposium on Reliability and Maintainability (RAMS)*. [Used in Chapter 4]
- A. F. Shahraki and O. P. Yadav, "Selective Maintenance Optimization for Multi-State Systems Operating in Dynamic Environments," published in 2018 *Annual Symposium on Reliability and Maintainability (RAMS)*. [Used in chapter 5]
- A. F. Shahraki, S. Al-Dahidi, A. R. Taleqani, and O. P. Yadav, "Using LSTM neural network to predict remaining useful life of electrolytic capacitors in dynamic operating conditions,"published in *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability.* [Used in Chapter 6]
- A. F. Shahraki, S. Amari, and O. P. Yadav, "Efficient algorithm for reliability evaluation of *k*-out-f-*n* phased-mission systems considering the imperfect fault coverage", 2022 (submitted) [Used in Chapter 7]

2. LITERATURE REVIEW[1]

Degradation modeling is an effective approach for reliability assessment, remaining useful life prediction, maintenance planning, and prognostics and health management. Degradation models are usually developed based on degradation data and/or prior understandings of the physics behind degradation processes of products or systems. Further, the effects of environmental or operational conditions on degradation processes and the knowledge about the dependency between degradation processes help improve the explanatory capabilities of degradation models. This chapter presents a comprehensive review of existing data-driven degradation modeling approaches commonly used in engineering applications.

2.1. Introduction

Degradation of a system is an inherent process influenced by internal and external factors including environmental and operating conditions. Degradation, such as damage accumulation over time, is usually an irreversible process leading to failure when the accumulated damage exceeds a natural or predetermined threshold level. For a critical system, such failures may cause severe losses. Therefore, it is imperative to understand and model the system's degradation behavior for prediction and prevention of potential failures so that subsequent losses can be effectively avoided. For the past few decades, extensive research has been conducted in degradation modelling for reliability analysis and other applications.

[1] The present chapter is based on the following paper:
A. F. Shahraki, O. P. Yadav, and H. Liao, "A review on degradation modelling and its engineering applications," published in International Journal of Performability Engineering, 2017.
Contribution of Ameneh Forouzandeh Shahraki: review all research papers and drafting the paper. Contribution of Om Yadav and Haitao Liao: verification of the results and proofreading the draft paper.

Compared to failure time data, degradation data provide valuable information on product failure behavior for making quick reliability assessment and other logistic decisions. The characteristics of degradation data and the methods used for data collection play a significant role in selecting appropriate degradation models. Generally, such data come from laboratory tests, field applications and/or real-time condition monitoring data collected either at normal operating or at accelerated stress conditions. These data can be the direct measurements of degradation processes (e.g., crack growth, decrease in light intensity of light emitting diodes) or the measurements of other characteristics that are closely related to the product's degradation process (e.g., vibration, change in output voltage, temperature).

Sometimes complex systems may experience multiple degradation processes affected by randomly changing covariates, such as temperature, humidity, and voltage. So far, some research effort has been focused on capturing the interaction and dependency of degradation processes along with their influences on failure propagation, and on the impact of shocks that accelerate these failure processes. In a variety of engineering applications, more and many other complex situations must be dealt with. For a practitioner, it is critical to construct a proper degradation model that can capture the true degradation behavior of a product in the field.

In recent years, degradation-based reliability analysis has gained momentum for getting some insights into product behavior and reducing the overall product development time. An increasing emphasis on accelerated degradation testing has created a need for further research on degradation modelling. Si et al. [12] focused their review on remaining useful life (*RUL*) estimation for a product. Their effort confined to statistical data-driven approaches that mostly rely on historical data. Statistical data-driven models are classified into stochastic process models, general path models, and others with a focus on stochastic models. Later, Zhang et al. [13] provided a review on degradation model-based *RUL* estimation approaches for systems with heterogeneity. Because of the importance of degradation-based reliability analysis, Shahraki et al. [14] conducted a comprehensive review on the state-of-the-art of degradation modelling. The existing degradation modelling methods were classified into two broad categories: the data-driven and physics-of-failure based models. Data-driven models are established using statistical fitting to the observed degradation data without considering degradation mechanisms. On the other hand, physics-based models capture the failure mechanisms or physical phenomena to build a mathematical description of the degradation process. This chapter covers data-driven approaches for modeling single degradation process and multiple degradation processes.

2.2. Data-Driven Models for Single Degradation Process

Data-driven models are becoming increasingly popular when it is difficult to capture and understand the physics behind a degradation process. These models can be classified into two subcategories: statistical models and artificial intelligence models. In the first subcategory, the general path and stochastic process models have been widely used. The artificial intelligence approaches, such as neural networks, have also been used in reliability estimation using degradation data. The following sub-sections provide more detailed discussion and recent advances in each of these subcategories.

2.2.1. General Path Model

In general statistical models for continuous degradation data, also called general path model, the degradation process is described as a function of time, possibly with a set of fixed-effects parameters and a set of random-effects parameters. Lu and Meeker [14] presented a general nonlinear regression model to characterize the degradation path of a random population of units. The model can be represented as:

$$y_{ij} = D_{ij} = D\left(t_{ij}; \varphi, \theta_i\right) + \varepsilon_{ij} \tag{2.1}$$

where y_{ij} represents the observed degradation, and $D(.)$ represents the actual degradation of the i^{th} unit at time (t_{ij}), φ is the vector of fixed effect regression coefficients (common for all units), θ_i is the vector of random-effect parameters representing characteristics of the i^{th} unit, and ε_{ij} is associated random error of the i^{th} unit at time t_{ij} which is assumed to be normally distributed with ε~$N(0, \sigma^2)$.

Several extensions of the general path model have been made by considering different types of statistical modelling approaches for different applications. For example, Freitas et al. [15] used a linear degradation path model to estimate the lifetime distribution of train wheels. They considered a single random parameter with lognormal, Weibull, and normal distributions. Yu [16] assumed a linear degradation path with a reciprocal Weibull-distributed degradation rate to determine the optimal design of an accelerated degradation test. Although the simplicity of linear random-effects model is an advantage, it might not be a good representative for the actual degradation path compared to a nonlinear random-effects model. Usually, nonlinear models may capture the degradation behavior better, and thus can provide better model fitting to the actual data. Bae and Kvam [17] modelled the degradation path of highly reliable light display components as a nonlinear random- coefficient model, which allows for a non-monotonic degradation path to capture the burn-in characteristic of the component. Bae et al. [18] discussed additive and multiplicative degradation models to derive the lifetime distribution of degraded components. In the additive model, the random effects are added to the mean path of degradation, while in the multiplicative model, the random effects are multiplicative to the mean degradation.

The general path models are widely used due to their simplicity, capability to model continuous processes and allow different variance-covariance structures of the response vector.

Nevertheless, sometimes these models might not well describe the actual degradation process because of oversimplification of reality. Further, general path models consider the inherent degradation to be deterministic and thus have difficulty in capturing the time-varying behavior of a product. Their inability to capture the temporal variability and the uncertainty inherent in the progression of deterioration over time limits their engineering applications [14].

2.2.2. Wiener Processes

Wiener process is also called Gaussian process or Brownian motion with drift. In general, a Wiener process can be expressed as [19]:

$$\mathrm{W(t) = \nu\Lambda(t) + \sigma B(\Lambda(t))} \tag{2.2}$$

where ν is the drift parameter showing the rate of degradation, σ is the volatility parameter, $B(.)$ is the standard Brownian motion, and $\mathit{\Lambda}(t)$ is a monotone increasing function representing a general time scale. The process $W(t)$ is often used to represent system degradation and has the following properties:

- $W(0) = 0$ almost certainly.
- $W(t)$ is a continuous process having stationary and independent increments, i.e., $\mathit{\Delta} W(t) = W(t + \mathit{\Delta} t) - W(t)$ is s-independent of $W(t)$, and $\Delta W(t) \sim N(\beta\Lambda(t + \Delta t) - \beta\Lambda(t), \sigma^2\Lambda(t + \Delta t) - \sigma^2\Lambda(t))$.

Due to its useful mathematical properties and physical interpretations, the Wiener process has been extensively used for modeling degradation processes. When the mean of degradation is linearly increasing, the Wiener process with linear drift is used. Further, the distribution of the first passage time, i.e., the time when the degradation process first reaches the critical failure threshold, follows an inverse Gaussian distribution. For some nonlinear degradation processes, the mean degradation path can be linearized using appropriate transformation such as time-scale transformation and log-transformation [20]. However, not all nonlinear degradation processes can be properly linearized. Therefore, nonlinear structures have been proposed to capture the dynamics of nonlinear degradation processes. Wang et al. [21] proposed a general degradation-modelling framework for hybrid deteriorating systems, which have both linear and nonlinear degradation components. Although Wiener processes have been used to model many degradation phenomena, they are not suitable for modelling monotonic degradation processes, such as wear or cumulative damage processes [14].

2.2.3. Gamma Processes

Abdel-Hameed [22] first suggested Gamma process as a useful model for degradation processes. It is appropriate to use Gamma process when the gradual damage is monotonically increasing or decreasing over time, such as fatigue, corrosion, crack growth, and corrosion of steel coatings. The basic Gamma process $\{Y(t);\ t \geq 0\}$ with shape function $\eta(t) > 0$ and scale parameter $\mu > 0$ is a continuous-time stochastic process with the following properties:

- $Y(0) = 0$ with certainty
- $Y(t)$ has independent non-negative increments, i.e., $Y(t + \mathrm{u}) - \mathrm{Y(u)}$ and $\mathrm{Y(s + v) - Y(v)}$ are independent for $\mathrm{t + u \ > \ u \ \geq \ s + v \ > \ v}$;
- $Y(t + u) - Y(u) \sim Gamma(\eta(t + u) - \eta(u), \mu)$, where $\eta(t)$ is a monotone increasing function with $\eta(0) = 0$.

Van Noortwijk [23] reviewed the application of Gamma process in maintenance, its statistical properties, and parameter estimation methods. Gamma process-based models can easily manage temporal variability in degradation process, while the unit-to-unit variability can be modelled by introducing random effects into the basic model.

The use of Gamma process for degradation modelling is getting popular as it has a physical interpretation, and its mathematical representations are straightforward. Moreover, the model considers the temporal variability of a degradation process. On the other hand, its Markov property, and the fact that it is strictly applicable to monotonic processes may restrict its application for some degradation processes [14].

2.2.4. Inverse Gamma Processes

The inverse Gaussian (IG) process has been used to model monotone degradation data when other processes do not fit the data well such as GaAs laser degradation data [24] and energy pipeline corrosion data [25]. The basic IG process $\{Y(t);\ t\ \geq\ 0\}$ has the following properties:

- $Y(0)\ =\ 0$ with certainty.
- $Y(t)$ has independent increments, i.e., $Y(t_2) - Y(t_1)$ is independent of $Y(t_4) - Y(\mathrm{t}_3)$ for $t_4 > t_3 \geq\ t_2 >\ t_1 \geq 0$.
- Each increment follows an IG distribution, that is, $Y(t)\ -\ Y(s) \sim IG(\mu(\Lambda(t)\ -\ \Lambda(s)), \lambda(\Lambda(t)\ -\ \Lambda(s))^2)$ for $t > s \geq 0$, where $\Lambda\ (t)$ is nonnegative and monotone increasing function of time ($\Lambda\ (0)\ = 0$).

The IG probability density function $IG(\mu\Lambda(\mathrm{t}), \lambda\Lambda(\mathrm{t})^2)$ is defined by:

$$f_{IG}(y;\mu,\lambda) = \sqrt{\frac{\lambda\Lambda(t)^2}{2\Pi y^3}}\, exp\left\{\frac{-\lambda}{2y}\left(\frac{y}{\mu} - \Lambda(t)\right)^2\right\}\ y,\lambda > 0\ ,\mu \in R \qquad (2.3)$$

where the process parameters, μ and λ, represent the degradation rate and degradation volatility, respectively.

Recently, IG process has received more attention in modelling degradation data because of its mathematical properties and flexibility in dealing with random effects and covariates. Wang and Xu [24] proposed IG process to model the degradation data of GaAs lasers by incorporating both the unit-to-unit variability and covariate information into the model. Later, Ye and Chen [26] developed two other random-effects models to make IG process more flexible than Gamma process in considering unit-to-unit variability. They also attempted to explain the physical meaning of IG process by presenting the relationship between the IG process and the compound Poisson process.

There is an inverse relationship between Wiener and IG processes that allows many of Wiener process properties to be extended to the IG process. Moreover, the IG process is flexible for incorporating random effects and covariates in degradation data analysis. It is easier to determine the probability density function and cumulative distribution function of RUL analytically in an IG process model [14].

2.2.5. Finite-State Degradation Models

Unlike the Wiener process, Gamma process and Inverse Gaussian process models, a finite-state Markov process is a stochastic process that evolves through a finite number of states. Due to the Markov property, the future state of the process is independent of past states given the current state. In Markovian-based degradation models, the transition probabilities (or rates) depend only on the states involved in the transition. In real-world applications, however, the transition probabilities (or rates) may also depend on other factors, such as the actual level of degradation, the time when the product reached the current state, the sojourn time, the total age of the system, and some covariates. Semi-Markovian models extend the application of Markovian-based models by incorporating the effects of these factors. Moghaddass and Zuo [27] and Moghaddass et al. [28] classified semi-Markov processes into four categories based on the type of transitions as shown in Figure 2.1.

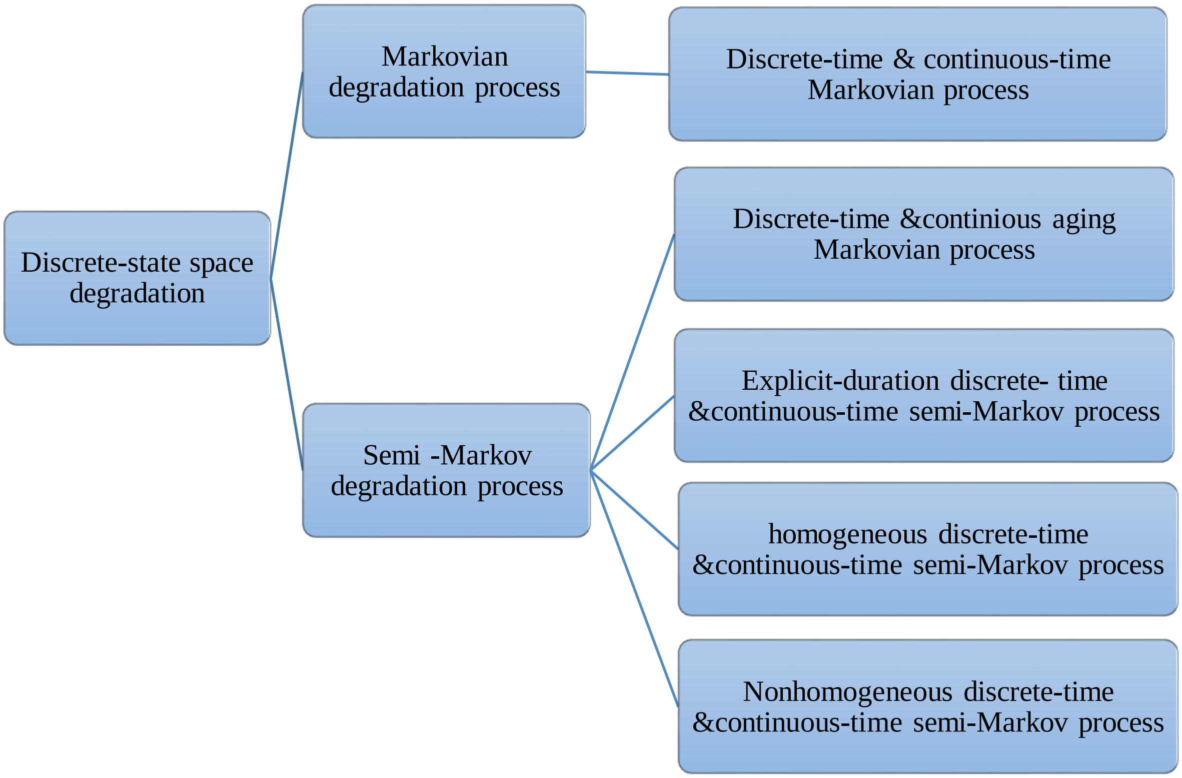

Discrete-state space degradation
Markovian degradation process
Discrete-time & continuous-time Markovian process
Semi -Markov degradation process
Discrete-time &continious aging Markovian process
Explicit-duration discrete- time &continuous-time semi-Markov process
homogeneous discrete-time &continuous-time semi-Markov process
Nonhomogeneous discrete-time &continuous-time semi-Markov process

Figure 2.1: Different categories of discrete-state space degradation models

Discrete-time and continuous-time semi-Markov processes represent two classes of semi-Markov processes. In homogeneous discrete-time and continuous-time semi-Markov processes, the time spent at the current state can affect the time-dependent transition rate (or the transition probability for a discrete-time process) between states [29]. In these models, however, the transition rates (or the transition probability for a discrete-time process) are independent of the age of the system. Therefore, these models are not applicable when the degradation process depends on both the sojourn time at each state and the total age of a system. On the other hand, nonhomogeneous discrete-time and continuous-time semi-Markov processes provide more general models that cover many other structures [28]. For example, the degradation transitions between states in a single system can follow non-identical structures and the transitions between states can depend on any combination of influential factors.

All the models depicted in Figure 2.1 are based on the assumption that the degradation process is directly observable. However, in many cases, the degradation level is not directly accessible due to the complexity of degradation process or nature of the product type. To deal with such cases, hidden Markov models (HMM) and hidden semi-Markov models (HSMM) have been developed. The HMM deals with two different stochastic processes: an unobservable degradation process and a measurable characteristic that is dependent on the actual degradation process. In HHMs, finding a stochastic relationship between the degradation process and the observation process is essential for condition monitoring and reliability analysis. HMMs have been used successfully in a wide range of applications.

Despite the extensive use of Markovian-based models for degradation modelling, a few limitations restrict their wider applications. From a practical application point of view, defining the discrete degrading states and estimating corresponding transition probabilities or rates has been one of the biggest challenges [14].

2.2.6. Other Models

Artificial intelligence methods, such as neural networks and fuzzy logic, belong to another set of data-driven models for degradation modelling and prediction. A neural network model learns from given inputs in a way to produce the desired outcome. After learning from historical data, the network model can be used to predict system status. Gebraeel and Lawley [30] developed a neural network-based degradation model for computing the remaining life distributions of partially degraded components and updated those using in-situ condition monitoring signals. Malhi et al. [31] applied recurrent neural networks for long-term prediction of defect progression in machines. For the first time, Fink et al. [32] applied multilayer feed-forward neural networks with multi-valued neurons to degradation modelling and reliability prediction. The main benefit of using artificial neural networks is its ability to model complex multidimensional degradation signals, for which analytic models are difficult to obtain. However, the requirement for large amounts of training data limits the application of neural networks. Kan et al. [33] reviewed the application of artificial neural networks and discussed thoroughly their advantages and limitations.

Another set of data-driven models uses fuzzy logic that maps an input data vector to a scalar output to handle non-linear, non-stationary, and complex system modelling. Zio and Di Maio [34] proposed fuzzy similarity analysis for estimating the RUL of a system. Neuro-fuzzy systems, the combination of fuzzy logic and neural networks, have been used in many studies to take the advantage of both fuzzy logic and neural networks [35].

2.3. Modeling Multiple Degradation Processes

Failures of a complex engineering system may be caused by multiple degradation processes. As a result, it is important to consider multiple degradation processes simultaneously for system reliability assessment. Some of the earlier work on multiple degradation modelling assumed that the degradation measures are either independent or dependent with a multivariate normal distribution [36][37]. For example, Crk [36] assumed that system failure is governed by several independent mechanisms and presented an effective way to estimate the system's reliability by monitoring each degradation measure. In practice, assuming independent degradation processes eases the modeling of multiple degradation processes, but it may not be realistic and can lead to poor or ineffective modelling of component and system reliability. To address this concern, Wang and Coit [37] proposed a multivariate degradation model based on a multivariate normal distribution. Recently, the use of copula in modelling dependent degradation measures has gained more attention. Pan and Balakrishnan [38] adopted a bivariate Birnbaum-Saunders distribution and its marginal distributions to approximate the reliability function of a system with two Gamma degradation processes.

Some researchers have attempted to model degradation processes of systems comprising of stochastically dependent components. The limited literature that considers stochastic dependency can be divided into two groups. In the first group, an external event or the failure of one component triggers the stochastic dependency between components. The triggering event can influence those affected components in three possible ways: leads to an immediate failure with a certain probability, increases degradation rate, or changes the degradation levels of remaining affected components [7].

Another group of models focuses on scenarios where the degradation behavior of components is influenced by one another. To model this phenomenon, Bian and Gebraeel [10] considered that the degradation level of one component affects the degradation rate of other components in the system. They assumed that when the degradation signal of one component reaches a certain value, it triggers a change in the degradation rate of other interdependent components. Later, Bian and Gebraeel [9] extended their previous work by considering the fact that component interactions can occur on a continuous basis. They modelled multivariate component degradation signals using a linear system of stochastic differential equations. Similarly, Eryilmaz [39] considered two-component systems where degradation processes of the two multi-state components follow a Markov process and modelled the dependence of the components using a copula. Recently, Dao and Zuo [40] considered a multistate series system with two types of stochastic dependency among components: the failure of a component can instantly cause complete failure of other components, and a component's degradation affects the state and the degradation rate of components.

2.4. Summary

This chapter presented a comprehensive review of data-driven degradation modelling methods. Different models for a single degradation process and the related work were provided for each of these categories. In addition, this review also included the methods for modelling multiple degradation processes. The modelling techniques developed so far need to be improved further to deal with more complex and hybrid systems. It is believed that there is a need to address the following research concerns.

- Most of degradation models are developed based on several assumptions that simplify model formulation and reduce the computational complexity. Most of the current literature deals with a single degradation process. However, new hybrid technologies consisting of mechanical, electronic, software and several other elements result in more complex physical systems. These elements, sometimes miniaturized elements, of complex physical systems exhibit multiple degradation processes. To deal with multiple degradation processes, which could be dependent, in a complex system, more advanced stochastic models and statistical methods need to be developed with the help of the increasing capability of modern computational technology.
- Engineering systems usually operate in time-varying operating conditions. The variation in operating conditions causes degradation processes to evolve in a complex manner. It is, therefore, important to develop loading-dependent degradation models that can capture the degradation behavior of the systems in response to the field operating conditions.

- The advancement in sensor technology has made it possible to simultaneously collect real-time data on multiple degradation processes. Such data could be collected from regions with different operating environments. The huge amount of data may be in different forms, such as time series or images. Those multi-dimensional data provide an excellent opportunity for future research on building more realistic degradation models for reliability prediction. This will require more refined analytical techniques to handle multi-dimensional data, and advanced data fusion techniques, such as multi-dimensional covariate analysis, noise reduction and Bayesian analysis.
- New emerging research issues that need to be addressed are related to decision making in logistics. The analysis of real-time degradation data of a system in the field can facilitate RUL prediction, real-time maintenance planning, and effective spare parts inventory management. To this end, more research is required to integrate degradation modelling into logistic decision-making strategies.

3. SELECTIVE MAINTENANCE OPTIMIZATION FOR MULTI-STATE SYSTEMS CONSIDERING STOCHASTICALLY DEPENDENT COMPONENTS AND STOCHASTIC IMPERFECT MAINTENANCE ACTIONS [2]

This chapter presents a selective maintenance optimization problem for complex systems composed of stochastically dependent components [41]. The components of a complex system degrade during mission time, and their degradation states vary from perfect functioning to complete failure states. The degradation rate of each component depends not only on its intrinsic degradation but also on the state of other dependent components of the system. The proposed approach captures the two-way interactions between components through system performance rates and uses Monte Carlo simulation to compute the reliability of the system in the next operational mission. Different maintenance actions such as do-nothing, perfect, and stochastic imperfect maintenance are considered during the maintenance break to improve the reliability of the system. The selective maintenance bi-optimization problem is modeled considering both the expected value and variance of the system reliability as objective functions. Time and budget are considered as constraints for finding the optimal maintenance strategy. Two illustrative examples are provided for a better understanding of the proposed approach and for demonstrating its effectiveness. The notations used in this chapter are given in the Table 3.1.

[2] The present chapter is based on the following paper:
A.F. Shahraki, O. P. Yadav, and C. Vogiatzis, “Selective maintenance optimization for multi-state systems considering stochastically dependent components and stochastic imperfect maintenance actions,”, in Reliability Engineering and System Safety Journal, 2020. Contribution of Ameneh Forouzandeh Shahraki: developing the mathematical models, analysis of the case study, discussion of the results, and drafting the paper. Contribution of Om Yadav and Chrysafis Vogiatzis,: verification of the results and proofreading the draft paper.

Table 3.1: Notations

$a_i^{(l_i)}$	Binary decision variable that is 1 if maintenance action with level l_i performed on component i
$c_i(y_i, y_i + l_i)$	The cost of performing maintenance action with level l_i on component i
C_0	Available maintenance budget
C_M	Total maintenance cost
D	Demand level
$g_i(t)$	Performance rate of component i at time t
$g_{i,k}$	Performance rate of component i in state k
G_K	System performance rate in its perfect state
$G_s(t)$	System performance rate at time t
I	Index of the components, $i = 1,2, \dots, N$
j	Index of the possible system states, $j = 1,2, \dots., 2^M$
k	Index of state of a component or system $k = K, K-1, \dots, 0$
$K+1$	Number of states for each component (or system)
l_i	Maintenance level of component i
$\lambda_{i,k}$	Intrinsic transition rate (degradation rate) of component i from state k to state $k-1$
$\lambda_{i,k}^m(t)$	Modified transition rate (degradation rate) of component i from state k to state $k-1$
M	Number of components with imperfect maintenance action
$n_I(t)$	Number of influencing components up to time t
N	Number of components in the system
N_U	Number of different δ values considered associated with the S-dependence
N_{MCS}	Number of MCS runs
P_i^{IM}	Probability of success for imperfect maintenance action performed on component i
P_i^{PM}	Probability of success for perfect maintenance action performed on component i
$P_k(t)$	Probability that the multi-state system is in state k at time t
$P_{s_1,s_2,\dots,s_N}(t)$	Probability that the state of each component i at time t is s_i
$R_s(t, D)$	System reliability at time t with demand level D
$s_i(t)$	State of component i at time $t, s_i(t) \in \{0,1, \dots, K\}$
$S_s(t)$	State of the system at time t
$t_i(y_i, y_i + l_i)$	Time of performing maintenance action with level l_i on component i
T_0	Available maintenance time
T_M	Total maintenance time

X	State of the system at the beginning of the next mission
x_i	State of component i at the beginning of the next mission
Y	State of the system at the beginning of the maintenance break
y_i	State of component i at the beginning of the maintenance break
z_i	Number of imperfect maintenance actions performed on component i
τ	Operational mission duration
$\bar{R}$	Mean of system reliability
σ_R^2	Variance of system reliability

3.1. Introduction

Recent technological advancements have resulted in the development of more complex, hybrid systems that can serve multiple functionalities. These complex systems consist of several sub-systems and components, and each of these components may be subject to one or more degradation processes. Assessing the reliability and planning the maintenance of such convoluted systems require a more realistic evaluation of the degradation states of its components and of the propagation of degradation processes in the future [14]. The existing literature on maintenance optimization presents several approaches that have considered such complex systems for reliability analysis [42]. Most of the approaches presented assume that system components are independent for two reasons. Primarily, this renders the problem easier to model and enables us to solve it; secondly, dependency is not a critical factor for some applications of interest [43][44]. However, in a complex and hybrid system, components can be viewed as inter-related and hence, it is important to consider component dependency when modeling the system degradation and plan for maintenance actions.

Nicolai and Dekker [7] classify the dependencies among components as economic, structural, and stochastic dependence (S-dependence). In economic dependence, the maintenance cost can be reduced by performing maintenance on multiple components jointly instead of separately. Structural dependence assumes that certain components structurally form a unit, and hence maintenance of a component in a unit implies maintenance of other components as well. Finally, S-dependence refers to system interaction where the degradation or failure process of one component is influenced by the degradation state of one or more inter-related and/or neighboring components of the system. Several researchers have considered S-dependence between components with either two possible degradation states (binary states) [45] or infinite degradation states (continuous degradation) [10][46][47] However, there has not been substantial progress to investigate S-dependence among multi-state components considering more than two but finite number of possible degradation states of each component [40].

Many complex systems consist of multiple components structured in series, such as aircraft engines, wind turbines, and power generation systems, to perform consecutive missions. During the time interval between two consecutive missions, maintenance actions can be carried out to improve the probability of successfully executing the next mission. Since there are limited resources for completing maintenance activities, it is important to decide on the maintenance strategy (e.g., the subset of components to maintain and the level of maintenance actions) considering the system requirements. This type of maintenance, referred to as selective maintenance, has received increasing attention in the literature over the years [40][48][48][49]. Most of the earlier selective maintenance studies have proposed formulating it as an optimization problem under the assumption that all components are stochastically independent. However, most of the components in many complex engineering systems, especially in mechanical and electronic systems, do influence the degradation behavior of other components. For example, the degradation of a component in an electronics system may cause a significant increase in the temperature that in turn might increase the degradation rates of other temperature-sensitive components [50][51], which may or may not be directly connected. Similarly in power grids systems, the degradation of generators or transformers in a subnetwork may increase the demand on other elements of the network causing increase in their loading profiles and in turn accelerating their degradation processes [10].

The performance of a multi-state system (MSS) depends on the state of its components, which are gradually degrading with usage time. When a component degrades, its state can be anywhere between perfect functioning and complete failure states. We can perform two different maintenance actions on each component during the maintenance break besides the do-nothing action. A perfect maintenance action entirely improves the state of a component, restoring its state to the perfect state. On the other hand, imperfect maintenance actions generally restore the state of components to any state between "as good as new" and "as bad as current". Since imperfect maintenance actions do not necessarily achieve a perfect state, the maintenance cost will be lower as compared to a perfect action. However, the consideration of imperfect maintenance actions adds to the complexity of the selective maintenance optimization problem. Further, most of the existing research in selective maintenance setting has considered the quality of imperfect maintenance actions as deterministic [40]. To the best of our knowledge, there has not been a study on selective maintenance of MSS composed of multi-state components considering stochastic imperfect maintenance. It is exactly the literature gap that we aim to address while formulating a selective maintenance model for MSS.

This chapter studies the selective maintenance optimization problem considering the S-dependency between multi-state components of a series system. Specifically, we generalize the approach presented in Dao and Zuo [40], which considers that the performance rate of each component only affects the performance rate and degradation rate of subsequent components in the series structure. They assumed that there exist only one-way interactions between directly connected components, i.e., a component is only affecting subsequent components, and a component can only be affected from its predecessors. However, there exist practical scenarios where the degradation state of a component may affect the degradation rate of its neighboring components, and may in turn be affected by the state of the other components, even when they are not in direct connection. For example, wear on a pulley may impact the rate of wear of a belt and vice versa. Moreover, there are many other factors such as the state of noncritical components and environmental/operational conditions that may affect the interaction as well. We, therefore, model the two-way interactions between multi-state components and capture the effects of unknown factors such as failure of non-critical components and environmental conditions on the degradation rate. This chapter also considers stochastic imperfect maintenance actions to capture uncertainty in the outcome of maintenance actions and its impact on MSS reliability in the next mission. The outcome of imperfect maintenance actions will be considered as a random variable with two possible states: success and not success. The uncertainty in the outcome of imperfect maintenance action is modeled as a function of the number of imperfect maintenance actions performed earlier, as well as the levels of maintenance actions carried out. In this way, we use the information of maintenance actions performed in the previous missions to make the best decision. Because of the uncertainty in the outcome of all imperfect maintenance actions, at the end of the maintenance break the system could be in a series of different possible states with certain respective probabilities. To further minimize the uncertainty in system reliability, we treat the variability in system reliability as a second objective in our selective maintenance optimization model. A nonlinear selective maintenance optimization model is, then, formulated to maximize the system reliability and minimize the variability (uncertainty) in system reliability in the next mission subject to maintenance time and cost constraints. The decision variables are the level of maintenance actions for selected components. The overall contributions in this chapter can be summarized in the following highlights:

- The proposed approach captures the S-dependence between components in a selective maintenance setting and models the two-way interactions as a function of the system performance rate and the number of influencing components. It also models the effects of unknown factors such as the state of non-critical components on the interaction by incorporating random effects.
- Along with the do-nothing and perfect maintenance actions, the proposed framework considers stochastic imperfect maintenance actions for MSS and models their probability of success as a function of the number of imperfect maintenance actions performed earlier.
- Finally, a selective maintenance optimization model is formulated considering a system reliability requirement and minimizing variability of system reliability as two objective functions to find robust solutions with higher reliability and reduced uncertainty.

3.2. Relevant Literature

The topic of selective maintenance for binary state system (BSS) considering a series-parallel system was first discussed by Rice et al. [52]. While considering two maintenance actions -- do-nothing or replace the failed component -- at each maintenance break, the authors developed a nonlinear selective maintenance optimization model to maximize system reliability in the next mission subject to a limited maintenance time. Cassady et al. [48] extended this work by relaxing the assumption of identical components and added maintenance cost as another constraint of the problem. The authors formulated three different selective maintenance models, namely, maximizing system reliability subject to both time and cost constraints, minimizing system repair costs subject to time and reliability requirement constraints, and minimizing total repair time subject to both cost and reliability requirement constraints. Cassady et al. [53] further studied the selective maintenance problem in a series-parallel system in which the lifetime of components follows a Weibull distribution. The authors considered different maintenance actions, such as replacement and minimal repair of failed components, and preventive replacement of surviving components. The optimization problem considered was like the one proposed by Rice et al. [30], where reliability maximization was considered as the objective function and maintenance time was treated as a constraint. Pandey et al.[54] also extended earlier work on selective maintenance of BSS by considering imperfect repairs as another available maintenance actions and showed that system reliability could be improved by performing imperfect maintenance actions.

Further, with the increasing complexity of many engineering systems, research interest has shifted from binary state to multi-state systems (MSS), which led to the proposal of several approaches for selective maintenance in MSS. Chen et al. [55] proposed the selective maintenance approach for a multi-state series-parallel system, where replacing the failed component with a new one was the only maintenance action. They developed an optimization model to minimize the total cost of maintenance activities subject to a minimum system reliability requirement. Liu and Huang [56] investigated MSS where the system consisting of binary state components functions at several output performance levels. They considered imperfect maintenance action as another maintenance action and used the age reduction coefficient approach to formulate the improvement after imperfect maintenance action. Recently, Khatab and Aghessaf [57] studied the selective maintenance problem for a series-parallel system composed of binary state components when the quality of the imperfect maintenance actions is stochastic. They presented a stochastic age reduction coefficient, which follows Beta distribution, to describe the maintenance improvement. Authors only considered binary state component assuming its age reduction coefficient is independent of the pervious imperfect maintenance actions performed on the component. However, in many real-world cases, the capability of improving the components will diminish with the number of imperfect maintenance actions performed in the previous missions. Moreover, system components continue to degrade with age and usage time and, hence, are expected to go through more than two states. Pandey et al. [58] extended selective maintenance to a MSS with multi-state components and proposed a generalized maintenance model considering deterministic imperfect maintenance actions along with the replacement and do-nothing actions. The authors formulated an optimization problem to achieve maximum system reliability during the next mission subject to limited maintenance resources. The components were assumed to be independent with constant transition rates between the component's states.

More recently, there have been studies on selective maintenance that are considering different forms of dependencies. For example, Dao et al. [59] proposed a selective maintenance model for a multi-state series-parallel system with identical and independent components with economic dependence. The authors adjusted repair cost and repair time for each of the components considering two types of economic dependency between multi-state components based on the shared -setting up and the advantage of repairing multiple identical components in each subsystem of the multi-state series-parallel system. Dao and Zuo [60] developed a selective maintenance optimization model for MSS, considering structural and economic dependencies with the goal of maximizing the system reliability. The authors calculated the total maintenance time and cost by adding up the maintenance time and cost of each component as well as the system disassembly time and cost, while subtracting the time and cost savings gained by simultaneously performing maintenance actions on multiple components. Considering S-dependence between system components when deciding on a selective maintenance policy has been attempted in recent years. More specifically, Maaroufi et al. [61] investigated a selective maintenance model for BSS considering that failure of a component leads to the immediate failure of other neighboring or functionally related components. In their work, the authors used a fault tree method to analyze the dependence between components and evaluate the overall system reliability. Dao and Zuo [40] studied selective maintenance for a MSS taking into account two types of S-dependencies: immediate failure dependence (Type 1) and gradual degradation dependence (Type 2). The authors developed a selective maintenance optimization model for MSS to maximize the total system profit, considering the required reliability and the maintenance time as constraints. However, in their work, the authors made some assumptions that limit the broader application of the methodology. These assumptions include the limitation that only direct connections between components lead to S-dependence, the existence of only one-way interactions between components, and that the state of each component only affects the state and degradation rate of subsequent components in the series structure.

In the proposed work, we extend the work of [40] by focusing on two-way interactions between multi-state components of the system even when they do not necessarily share a direct connection. Our work also models the effects of unknown factors on the interaction between components. In particular, this chapter presents a general selective maintenance model for MSS having S-dependent components and considers the influence of stochastic imperfect maintenance actions. These actions are not deterministic, but instead, have a probability of success which depends on previous maintenance actions performed on the components as well as the levels of maintenance actions carried out.

3.3. The System Model for S-Dependent Components

In this section, we describe the multi-state system model we use. More specifically, we discuss the degradation model and how its reliability is derived.

3.3.1. Degradation Model for MSS

In this work, we focus on a multi-state system that consists of N multi-state components in a series structure, which represents many real-life applications. Consider each component i $(i = 1,2,..,N)$ has $K + 1$ different states, each with different performance rate, g_k $(k = 0,1, \ldots., K)$. Using our notation, state K is the perfect functioning state, state 0 is the complete failure state, and all states k such that $0 < k < K$ are intermediate states for each component. Component degradation is assumed to follow a continuous time Markov process, with transition times between component states following an exponential distribution. It is also considered that system components degrade gradually, i.e., a component with current state k must visit state $k - 1$ before arriving at state $k - 2$. A state transition diagram of a multi-state component is shown in Figure 3.1. The parameter $\lambda_{i,k}$ is the intrinsic degradation or transition rate of component i from state k to its immediately lower state, $k - 1$: this parameter can be determined from historical data and expert elicitation.

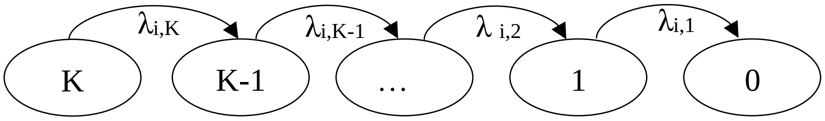


Figure 3.1: The state transition diagram of a multi-state component

In a series system, each component is critical for the system performance, and failure of a single component can lead to complete system failure. Therefore, the state of the system at time t, denoted by $S_s(t)$, is defined by the state of the most degraded component as:

$$S_s(t) = min\{s_1(t), s_2(t), \ldots., s_N(t)\} \qquad (3.1)$$

In our proposed model, we consider S-dependencies between components. Specifically, if a component degrades to a lower state, it may influence the degradation rates of other neighboring and/or functionally inter-related components. Initially, when all components are at the perfect functioning state and there is no interaction between components, all of them degrade at their intrinsic degradation rates. However, when a component degrades to a lower state, it may deteriorate the state of the system, which in turn can affect the degradation rates of the remaining components.

We define function $f(\cdot)$ to capture the S-dependence between all correlated components of the system. We denote the modified degradation or state transition rate of component i $(i = 1,2,..,N)$ from state k to $k-1$ at time t as $\lambda_{i,k}^{m}(t)$, and can be expressed as:

$$\lambda_{i,k}^{m}(t) = \lambda_{i,k} \cdot f(G_s(t), n_I(t), \delta), k = K, K-1, \ldots, 1 \tag{3.2}$$

The modified degradation rate $\lambda_{i,k}^{m}(t)$ consists of two elements: the intrinsic degradation rate $\lambda_{i,k}$, and the interaction effect on degradation rate $f(\cdot)$ caused by other degrading components. We note that $f(\cdot)$ is a function of the system performance rate $G_s(t)$ at time t ($G_s(t) = \min\{g_1(t), g_2(t), \ldots, g_N(t)\colon g_i(t) = performance\ rate\ of\ component\ i\}$), as well as the number of influencing components $n_I(t)$ transitioning to a lower state at time t. Further, the interaction effect is stochastic in nature, and is influenced by several other factors such as system environmental/operational conditions, including the state of non-critical components in the system. Hence, we include a parameter δ to capture the uncertainty caused by random variations in these factors. We assume that δ follows a normal distribution with a mean of zero and a standard deviation of σ. Since we assume that a random variation (uncertainty) caused by these factors lead to an increase in the degradation rates of the affected components, we only consider the absolute value of δ ($|\delta|$). Based on these considerations, the term $f(\cdot)$ in Equation (3.2) can be expressed as:

$$f(G_s(t), n_I(t), \delta) = \left(\frac{G_K}{G_s(t)} + |\delta|\right)^{n_I(t)/N}, G_s(t) \neq 0 \text{ and } f(\cdot) \geq 1 \tag{3.3}$$

When all components are in the perfect state, i.e., $n_I(t) = 0$ and $G_s(t) = G_K$, the value of $f(\cdot)$ is equal to 1, and $\lambda_{i,k}^m(t) = \lambda_{i,k}$. When the system degrades to a lower state at time t ($G_s(t) < G_K$) and the number of influencing components increases, the value of $f(\cdot)$ will be greater than 1 causing $\lambda_{i,k}^m(t) > \lambda_{i,k}$.

3.3.2. Reliability of MSS with S-Dependent Components

After modeling the degradation rates of multi-state components, this subsection discusses the reliability of the MSS in the next mission and how it can be evaluated when considering the degradation of its S-dependent components. We define the system reliability as the probability that the system successfully executes the next mission of duration τ. In other words, we associate the reliability with the probability that the system has not reached to a state with performance rate less than a specified demand level D. Therefore, the reliability of the MSS at time t is defined as:

$$R_s(t, D) = Pr\{G_s(t) \geq D\} \tag{3.4}$$

The system reliability can then be evaluated as the summation of probabilities of all acceptable states of the system:

$$R_s(t, D) = \sum_{k=0}^{K}[P_k(t) \cdot I(G_s(t) \geq D)], \tag{3.5}$$

where $P_k(t)$ is the probability that the MSS is in state k at time t and I is an indicator function with a value of 1, when the performance rate of the system is higher than the demand level ($G_s(t) \geq D$), and 0, in any other case. The states with a zero value at the indicator function can be regarded as system failure. The system state probability at time t is given as:

$$P_k(t) = Pr\{S_s(t) = k\} = \sum_{min\{s_i\}=k} P_{s_1, s_2, \ldots, s_N}(t) \tag{3.6}$$

where $P_{s_1,s_2,\ldots,s_N}(t)$ is the probability that the state of each component $i(= 1\ ,2, \ldots, N)$ at time t is s_i. The state of the system at time t is k, if and only if the state of at least one component is k and the state of the other components is k or higher than k.

The state probabilities of each component, $P_{s_1,s_2,\ldots,s_N}(t)$, can be computed by solving the Chapman–Kolmogorov system of differential equations, following the paradigm in [62]. Since the transition rates are not deterministic because of stochastic interaction effect between degrading components ($\delta \neq 0$), the method in [40] is not applicable to determine the state probabilities in Equation (3.6). Hence, to capture the stochastic interaction effects, we use two loop Monte Carlo simulation (MCS) approach to estimate the system reliability. For this purpose, a large sample of the system life history is generated, and the estimated system reliability at time t is expressed as the fraction of times the system is functional at the end of the mission. We carry out N_U iterations of MCS and at each time, we randomly select $\delta \sim N(0, \sigma^2)$ value to reflect the uncertainty associated with the S-dependence between components. For each generated δ, the system reliability is estimated using MCS with N_{MCS} runs, as discussed in the following paragraphs.

The system state at time t is shown as $S_s(t) = (s_1(t), s_2(t), \ldots, s_N(t))$. To compute the probability of the system transition from its current state to the next state, the mission time is divided into small time intervals Δt. This is a common issue with time interval selection, and we must choose Δt to be as small as possible so as to increase the accuracy of calculations. Then, at the end of each time interval $t + \Delta t$, the system can be in any one of the $N + 1$ states given the current state of the system $S_s(t) = (s_1, s_2, \ldots., s_N)$. Figure 3.2 illustrates these $N + 1$ different states that system can occupy at the end of each time interval.

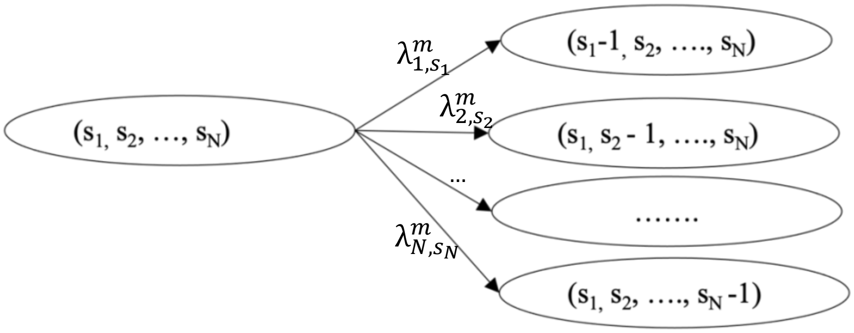


Figure 3.2: The state transition diagram of a multi-state component

In a continuous time Markov chain, the probability that the states of two components change simultaneously at the same time is zero [40]. Therefore, at each small time interval Δt, at most the state of one component will change, which can be illustrated as:

$$\begin{aligned} &Pr\{s_1(\mathrm{t}+\Delta\mathrm{t}) = s_1, s_2(\mathrm{t}+\Delta\mathrm{t}) = s_2, \dots, s_N(\mathrm{t}+\Delta\mathrm{t}) = s_N\ |S_s(t)\} \\ &\qquad + Pr\{s_1(\mathrm{t}+\Delta\mathrm{t}) = s_1 - 1, s_2(\mathrm{t}+\Delta\mathrm{t}) = s_2, \dots, s_N(\mathrm{t}+\Delta\mathrm{t}) = s_N\ |S_s(t)\} \\ &\qquad + Pr\{s_1(\mathrm{t}+\Delta\mathrm{t}) = s_1, s_2(\mathrm{t}+\Delta\mathrm{t}) = s_2 - 1, \dots, s_N(\mathrm{t}+\Delta\mathrm{t}) = s_N\ |S_s(t)\} + \cdots \\ &\qquad + Pr\{s_1(\mathrm{t}+\Delta\mathrm{t}) = s_1, s_2(\mathrm{t}+\Delta\mathrm{t}) = s_2, \dots, s_N(\mathrm{t}+\Delta\mathrm{t}) = s_N - 1\ |S_s(t)\} = 1 \end{aligned} \tag{3.7}$$

where $s_i(t)$ and $S_s(t)$ represent the state of component i and the system at time t, respectively.

We can also assume that using a small enough Δt results in the transition rates of all components being independent and constant during Δt. Therefore, the probability that the transition of component i $(i = 1,2,..,N)$ from its current state s_i to state $s_i - 1$ does not occur in the next time interval Δt can be approximated by:

$$p_i = Pr\{s_i(\mathrm{t}+\Delta\mathrm{t}) = s_i | s_i(\mathrm{t}) = s_i\} \approx e^{-\lambda^m_{i,s_i}(t) * \Delta\mathrm{t}} \tag{3.8}$$

We model the transition of the component i at each time interval generating a random number $U \sim Uniform\ (0,1)$. If U is greater than p_i, then a transition occurs during this small time interval and alters the state of component i from its current state, s_i, to state $s_i - 1$. When the state of each component changes, it may influence the degradation rates of the remaining components. To capture the interaction effects between components after each transition, we determine the state of the system and the number of influencing components to update the interaction effect $f(\cdot)$ and the transition rates of all components, as per Equation (3.2) and Equation (3.3). We then use these updated transition rates in Equation (3.8) to compute the transition probabilities of each component for the next time interval. The simulation stops when either time reaches the end of the mission, or the performance rate of the system falls below the demand level D. We run each simulation for N_{MCS} times and eventually calculate the reliability of the system given that the state of system at the beginning of the next mission is $S_s(0) = (s_1(0), s_2(0), \dots, s_N(0) = S$ as:

$$R_S^u|S = 1 - \frac{count}{N_{MCS}} \quad , \ \mathrm{u} = 1, 2, \dots, N_U \tag{3.9}$$

where count represents the number of times that system failed in N_{MCS} iterations of MCS. As noted, this MCS process is performed N_U times, and finally, the expected system reliability and its variability can be calculated as the following estimates:

$$E(R_S\dot{|}S) = \frac{\sum_{u=1}^{N_U} R_S^u|S}{N_U} \tag{3.10}$$

$$Var\ (R_S|S) = \frac{1}{N_U} \sum_{u=1}^{N_U} (R_S^u|S - E(R_S|S))^2 \tag{3.11}$$

3.4. Selective Maintenance Optimization Problem Considering Stochastic Imperfect Maintenance

As noted earlier, any given complex system is expected to work for consecutive missions of duration τ. This expectation essentially enforces carrying out all maintenance actions within the break between two consecutive missions to keep the system ready for next mission. However, maintenance resources (time, budget, and others) are limited and, hence, it is difficult to perform all desired maintenance actions on all the components with given resources. Therefore, it is important to determine the components to be maintained as well as the level of the maintenance actions (l_l) required for satisfying the mission requirements. In fact, in our work, the goal is to determine the maintenance action for each component in order to bring the system back to state $X = [x_1, x_2, .., x_N]$ at the beginning of the $m + 1^{th}$mission, given the state of the system at the end of m^{th} mission is known and denoted as $Y = [y_1, y_2, .., y_N]$. This is pictorially shown in Figure 3.3. We can now define the decision variables $a_i^{(l_i)}$ of the optimization problem as:

$$a_i^{(l_i)} = \begin{cases} 1, \text{if maintenance action with level } l_i \text{ performed on components } i \\ 0, \text{otherwise} \end{cases} \tag{3.12}$$

Since the outcome of an imperfect maintenance action is considered stochastic, the system at the end of the maintenance break can be in several different possible states with certain probabilities. This, in turn, influences the reliability of the system at the end of the next mission. Consequently, it is necessary to take into consideration the variance of system reliability as a second objective function, leading to simultaneously maximizing the expected system reliability while minimizing the variance of the system reliability. Detailed explanations on the expected value and the variance of the system reliability, and the optimization model considering stochastic imperfect maintenance are presented in the following subsections.

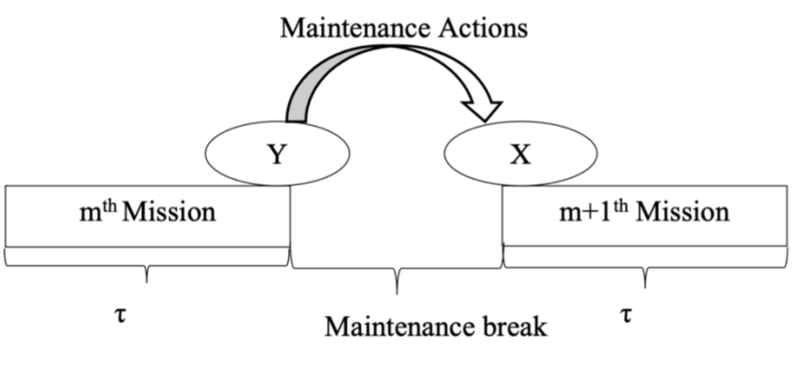


Figure 3.3: Two successive missions and the maintenance break between them.

3.4.1. Maintenance Actions and Resources

As soon as the system enters its maintenance break, multiple maintenance actions can be performed on each component, including do-nothing, replacement with new component as corrective or preventive maintenance action, and an imperfect maintenance action. The state x_i of the component i $(i = 1,2,..,N)$ after each maintenance action could be anywhere between its current state y_i and the perfect state K, which is represented as:

$$x_i = \begin{cases} y_i & \text{if } a_i^{(0)} = 1 \\ K & \text{if } a_i^{(K-y_i)} = 1 \\ \mathrm{y}_i \leq x_i < \mathrm{K} & \text{if } a_i^{(l_i)} = 1 \ , 0 < l_i < \ K - y_i \end{cases} \tag{3.13}$$

In Equation (3.13), $a_i^{(0)} = 1$ implies that component i is not selected for maintenance and, hence, the state of the component does not change. The perfect preventive maintenance action or replacement action is represented by $a_i^{(K-y_i)} = 1$ and restores the component health condition to its perfect state, rendering it as good as new. We consider the result of the perfect maintenance action is deterministic, i.e., the state of each component after the maintenance is as good as new with probability 1 ($P^{PM} = 1$). However, we note here that the cost of perfect maintenance actions is often high.

On the other hand, imperfect maintenance actions change the condition of components to a state that can be anywhere between the as good as new state and the as bad as current states. The cost of imperfect maintenance is typically lower compared to a perfect maintenance action. As the state of each component after imperfect maintenance depends on several factors such as the quality of the maintenance action, the current condition of the component, and the availability of maintenance resources; the outcome of imperfect maintenance action can be considered stochastic. Hence, after performing an imperfect maintenance action on component i with a specific maintenance level l_i ($a_i^{l_i} = 1$), the component can acquire a new state $x_i = y_i + l_i$ with probability P_i^{IM}, or no changes in the current state $x_i = y_i$ with probability $1 - P_i^{IM}$ if the imperfect action was not effective. For example, if the current state of a component is $y = 1$ and its perfect state is $K = 4$, then that component can be maintained to three possible states $x = 2, 3,$ or 4 with different maintenance costs. If we select to perform an imperfect maintenance action with level $l = 1$(or 2), then the new state would be $x = 2$ (or 3) with probability $P^{IM}(or\ \acute{P}^{IM})$, or it will remain at its current state $x = 1$ with probability $1 - P^{IM}(or\ 1 - \acute{P}^{IM})$.

Further, the probability of achieving the desired improved state is influenced by two main factors. First, it depends on the number of imperfect maintenance actions (z) performed on the component in the past, i.e., the probability of success for each imperfect maintenance level action decreases with an increase in the value of z. This essentially means that the capability of an imperfect maintenance action to improve upon the health state of a component to a desired level diminishes with each maintenance action performed on the component. Secondly, the effect of z on $P^{IM}(.)$ varies with the selected levels of the maintenance actions; in plain terms, the probability of success is lower if one expects to reach a higher level of improvement from an imperfect maintenance action. For example, the probability of achieving a 50% improvement in the health condition of a component from a maintenance action l_i is significantly higher than the probability of expecting a 90% improvement from a maintenance action $\acute{l_i}$ ($\acute{l}_i > l_i$). Therefore, to capture both the impact of the number of imperfect maintenance actions performed earlier (z_i) and the maintenance level (l_i), the probability of success for a maintenance action on component i ($i = 1,2,..,N$) , $P_i^{IM}(z_i, l_i)$, is updated as:

$$P_i^{IM}(z_i, l_i) = P_i^{IM} * \exp\left(-z_i * \frac{l_i}{K}\right) \tag{3.14}$$

where P_i^{IM} is the probability of successfully preforming the first imperfect maintenance action, which, as noted earlier, can be determined from a combination of historical data and expert knowledge.

Because of the uncertainty in the outcome of all imperfect maintenance actions, at the end of the maintenance break the system could be in different possible states with certain probabilities. For example, if imperfect maintenance actions are performed on M $(M \leq N)$ components of the system, the number of possible states the system can acquire is 2^M. Let us consider the following example of a series multi-state system consisting of three components. Each of the components has four states and the state of the system at the beginning of the maintenance break is $Y = [1, 0, 1]$. Further assume that the maintenance actions with level $l_1 = 1, l_2 = 3,$ and $l_3 = 1$ are selected for components 1, 2, and 3, respectively. As the actions performed on components 1 and 3 are imperfect maintenance actions, the state of components 1 and 3 at the end of maintenance break can be $x_1 = 1$ or 2 and $x_3 = 1$ or 2, respectively. However, the state of component 2 after performing perfect maintenance action will be 3 with probability 1 ($P_2^{PM} = 1$). Therefore, the number of possible system states is $2^2 = 4$ as: $S_1 =$ [1,3,1], $S_2 =$ [1,3,2], $S_3 =$ [2,3,1], or $S_4 =$ [2,3,2] with certain probabilities. The probability that system acquires any one of the above possible states can be derived by multiplying the probability of the state of its components. Since each component acquires state $x_i = y_i + l_i$ with probability $P_i^{IM}(.)$, or $x_i = y_i$ with probability 1-$P_i^{IM}(.)$, the probability that system will be in any one of the four states is given as the following expression:

$$p(X = S_j) = \begin{cases} \left(1 - P_1^{IM}(.)\right) * P_2^{PM} * \left(1 - P_3^{IM}(.)\right) & j = 1 \\ \left(1 - P_1^{IM}(.)\right) * P_2^{PM} * \left(P_3^{IM}(.)\right) & j = 2 \\ \left(P_1^{IM}(.)\right) * P_2^{PM} * \left(1 - P_3^{IM}(.)\right) & j = 3 \\ \left(P_1^{IM}(.)\right) * P_2^{PM} * \left(P_3^{IM}(.)\right) & j = 4 \end{cases} \quad \& \textstyle\sum_{j=1}^{4} p(X = S_j) = 1$$

Once the value of $p\big(X = S_j\big)$ is available, the estimates of expected system reliability and its variability in the next mission are calculated as:

$$E(R_S) = \sum_{j=1}^{2^M} E(R_S|S_j) * p(X = S_j) \tag{3.15}$$

$$Var\ (R_S) = \sum_{j=1}^{2^M} Var\ \big(R_S|S_j\big) * p\big(X = S_j\big) + \sum_{j=1}^{2^M} (E(R_S|S_j))^2 * p\big(X = S_j\big) - (\sum_{j=1}^{2^M} E(R_S|S_j) * p\big(X = S_j\big))^2 \tag{3.16}$$

where S_j is the j^{th} possible state of the system. Furthermore, associated with each maintenance action is a certain amount of cost and time. Maintenance cost $c_i(y_i, y_i + l_i)$ and time $t_i(y_i, y_i + l_i)$ represent resources required to perform the necessary maintenance action on component i, which, in turn, depend on its current state y_i and the chosen level of maintenance action l_i. If the selected action is do-nothing, which does not change the state of the component, both the associated cost and time will be zero. When replacing a degraded component by a new one, the state of the component changes to as good as new and the associated cost and time are represented as $c_i(y_i, K)$ and $t_i(y_i, K)$, respectively. Last, the cost and time required for imperfect maintenance actions are indicated as $0 < c_i(y_i, y_i + l_i) < c_i(y_i, K)$ and $0 < t_i(y_i, y_i + l_i) < t_i(y_i, K)$, respectively. The matrix forms of maintenance cost and time for component i $(i = 1,2,..,N)$ are given by:

$$T_i = \begin{bmatrix} 0 & t_i(0,1) & \cdots & t_i(0,K) \\ 0 & 0 & \cdots & t_i(1,K) \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & t_i(K-1,K) \\ 0 & 0 & \cdots & 0 \end{bmatrix} \tag{3.17}$$

$$C_i = \begin{bmatrix} 0 & c_i(0,1) & \cdots & c_i(0,K) \\ 0 & 0 & \cdots & c_i(1,K) \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & c_i(K-1,K) \\ 0 & 0 & \cdots & 0 \end{bmatrix} \tag{3.18}$$

The values of the matrix are positive for all $y_i + l_i > y_i$; otherwise, they are 0. Also we have that $t_i(y_i, y_i + l_i) < t_i(y_i, y_i + \acute{l}_\imath)$ and $c_i(y_i, y_i + l_i) < c_i(y_i + \acute{l}_\imath)$ if $l_i < \acute{l}_\imath$, as well as $t_i(y_i, y_i + l_i) > t_i(\acute{y}_\imath, \acute{y}_\imath + \acute{l}_\imath)$ and $c_i(y_i, y_i + l_i) > c_i(\acute{y}_\imath, \acute{y}_\imath + \acute{l}_\imath)$, if $y_i < \acute{y}_\imath$ and $y_i + l_i = \acute{y}_\imath + \acute{l}_\imath$.

Finally, the total cost incurred by performing maintenance on the selected components can be given as:

$$C_M = \sum_{i=1}^{N} c_i(y_i, y_i + l_i) \tag{3.19}$$

Similarly, the total maintenance time is the summation of all required time during the maintenance actions for each individual component, and is calculated as:

$$T_M = \sum_{i=1}^{N} t_i(y_i, y_i + l_i) \tag{3.20}$$

3.4.2. Selective Maintenance Optimization Model

We formulate the selective maintenance optimization model to find the level of maintenance actions required to maximize system reliability in the next mission with decreased uncertainty (variability). The decision variables are the level of maintenance actions required on individual components as defined in Equation (3.12) while the total maintenance time and cost are treated as constraints. The bi-objective optimization problem is formulated as follows:

$$Max\ E(R_S) = \sum_{S \in W} E(R_S|S) * p(X = S) \tag{3.21a}$$

$$Min\ Var\ (R_S) = \sum_{S \in W} Var\ (R_S|S) * p(X = S) + \sum_{S \in W} \big(E(R_S|S)\big)^2 * p(X = S) -$$

$$\left(\sum_{S \in W} E(R_S|S) * p(X = S)\right)^2 \tag{3.21b}$$

$$S.t.$$

$$\sum_{i=1}^{N} t_i(y_i, y_i + l_i) \le T_0 \tag{3.21c}$$

$$\sum_{i=1}^{N} c_i(y_i, y_i + l_i) \le C_0 \qquad (3.21d)$$

$$a_i^{(0)} + a_i^{(K-y_i)} + \sum_{l_i=1}^{K-y_i-1} a_i^{(l_i)} = 1 \qquad \forall\, i \in E \qquad (3.21e)$$

$$a_i^{(l_i)} \in \{0, 1\} \quad \forall\, i \in E \;\; ; \; l_i \in \{0, 1, .., K - y_i\}.$$

In the formulation, W is a set which reveals all different possible states of the system after performing each maintenance strategy and is formally defined as $W = \{X: x_i \in \{y_i\}, \forall\, i \in E_0; x_i \in \{K\}, \forall\, i \in E_1 \,;\, x_i \in \{y_i, y_i + l_i\}\,, \forall i \in E_2\}$. Note that $E = \{1,2, \dots, N\}$, $E_0 = \left\{i \in E: a_i^{(0)} = 1\right\}$, $E_1 = \left\{i \in E: a_i^{(K-y_i)} = 1\right\}$, and $E_2 = \left\{i \in E: \sum_{l_i=1}^{K-y_i-1} a_i^{(l_i)} = 1\right\}$ are three auxiliary sets representing the components selected for the three types of maintenance actions: do-nothing (E_0), replacement (E_1), and imperfect maintenance action (E_2), respectively. For the given example, we would have that $E_0 = \emptyset$, $E_1 = \{2\}$, and $E_2 = \{1,3\}$. In that same example, we have $W = \{(1,3,1), (1,3,2), (2,3,1), (2,3,2)\}$ and includes all four possible states ($2^M = 2^2 = 4$) of the system after the given maintenance strategy ($a_2^{(3)} = 1,\; a_1^{(1)} = 1$, and $a_3^{(1)} = 1$).

In the above optimization model, Equation (3.21a) and Equation (3.21b) are the two objective functions for system reliability and variability in system reliability, respectively. Equation (3.21c) and Equation (3.21d) represent the total time and budget constraints. Finally, Equation (3.21e) states that only one maintenance action with specified maintenance level l_i can be selected for each component. All decision variables are binary. As the optimization model is nonlinear and stochastic, we employ a genetic algorithm (GA) to find the best maintenance strategy. When the total number of the states of the system is small, we could use exact solution methods and enumerate all the states to find the optimal solution. Otherwise, in most real-life instances, we can use GA, as introduced by Holland [63]. GA is well suited to solve multi-objective optimization problems when the search space is large. The ability of GA to simultaneously search different regions of a solution space enables us with finding a diverse set of solutions for difficult problems. Moreover, most multi-objective genetic algorithms do not require the decision maker to prioritize, scale, or weigh objectives. We can use non-dominated sorting genetic algorithm II (NSGAII) to find the Pareto-optimal solutions for the bi-objective optimization problem. The concept of constrained-domination is considered in the modified-NSGAII to tackle constraints in the optimization problem [64][65]. Based on this concept, during various stages of the algorithm, the crowded comparison operator is used in the selection process. At each iteration, both objective function values (i.e., the mean and variance of system reliability) and the constraint violation (CV) of every individual of the population are considered in the selection process. The CV, which is equal to zero for feasible solutions, is calculated for every individual as follows:

$$CV = \frac{Max(C - C_0, 0)}{C_0} + \frac{Max\ (T - T_0\ , 0)}{T_0} \tag{3.22}$$

where C and T are the maintenance cost and time of each individual of the population, and C_0 and T_0 are the maintenance budget and time limit, respectively. As another solution approach, to efficiently solve the bi-objective optimization problem for large systems, we divide the solving approach into two phases like the work of Diallo et al. [66]. In the first phase, we find the system reliability for all feasible solutions ($CV = 0$). Then, we find the optimal strategy between the feasible solutions solving a bi-objective integer problem.

3.5. Numerical Studies and Results

3.5.1. Example 1

To demonstrate the applicability of the proposed model, we provide a numerical study of a two-component system ($i = 1{,}2$), where the components are stochastically dependent. In this example, we also assume that each component has four states $s_i = 0, 1, 2, 3$ with corresponding performance rates $g_i = 0, 1, 2, 3$. The transition rates of the components are provided in Table 3.2. The maintenance cost (in $1000) and time (in days) for each component are given as follows:

$$C_1 = \begin{bmatrix} 0 & 4.5 & 9 & 14.5 \\ 0 & 0 & 4 & 12 \\ 0 & 0 & 0 & 9.5 \\ 0 & 0 & 0 & 0 \end{bmatrix}, C_2 = \begin{bmatrix} 0 & 3 & 8.5 & 15 \\ 0 & 0 & 6 & 12 \\ 0 & 0 & 0 & 4.5 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$T_1 = \begin{bmatrix} 0 & 0.5 & 1 & 1.5 \\ 0 & 0 & 0.5 & 1 \\ 0 & 0 & 0 & 0.5 \\ 0 & 0 & 0 & 0 \end{bmatrix}, T_2 = \begin{bmatrix} 0 & 0.5 & 1 & 1.5 \\ 0 & 0 & 0.5 & 1 \\ 0 & 0 & 0 & 0.5 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Table 3.2: The transition rates of components

Component	Transition rate ($Year^{-1}$)		
	$\lambda_{i,3}$	$\lambda_{i,2}$	$\lambda_{i,1}$
1	0.12	0.18	0.150
2	0.09	0.15	0.05

The state of the system at the beginning of the maintenance break is $Y = [1, 1]$, and the number of imperfect maintenances performed on the components before the current maintenance break are $z_1 = 1$ and $z_2 = 1$. The initial probability of success for imperfect maintenance action on component 1 and component 2 are $P_1^{IM} = 0.9$ and $P_2^{IM} = 0.9$. Our goal is to find the best selective maintenance actions subject to a maintenance time limit of $T_0 = 2$ (in days) and a maintenance budget limit of $C_0 = 10$ (in $1000) to maximize the system reliability and minimize the variance of system reliability at the end of the next mission duration $\tau = 0.5$ years. In addition, we assume that $\delta \sim N(0, 0.1^2)$.

Using the above input data, the problem was coded and solved using Matlab R2016a. To show the impact of component dependency and the stochastic nature of the maintenance actions, we consider four different maintenance scenarios: (i) independent components and deterministic maintenance actions, (ii) independent components and stochastic imperfect maintenance actions, (iii) S-dependent components and deterministic actions, and (iv) S-dependent components and stochastic imperfect maintenance actions. The obtained results from these scenarios are compared based on the system reliability goal (mean and variance of reliability), maintenance cost, and time.

As noted earlier, a proper sample size needs to be found for the MCS. To find the best sample size for the $MCS,$ we first run the MCS considering the third scenario using different sample sizes. Our results indicate that the sample size of $N_{MCS} = 10000$ provides convergence of both mean and variance of system reliability for all nine possible states of the system $(X_j, for\ j = 1,2,\dots,9)$. The results for one such state, $X = [2,2]$, are shown in Figure 2.4a and Figure 2.4b. It is observed that there is no significant difference between the results obtained with $N_{MCS} = 10000$ and $N_{MCS} = 100000$. However, the simulation effort (number of iterations) required with $N_{MCS} = 100000$ is nearly 10 times more than with $N_{MCS} = 10000$. It clearly demonstrates that the sample size $N_{MCS} = 10000$ is sufficient to achieve same result, which essentially validates our decision to use $N_{MCS} = 10000$.

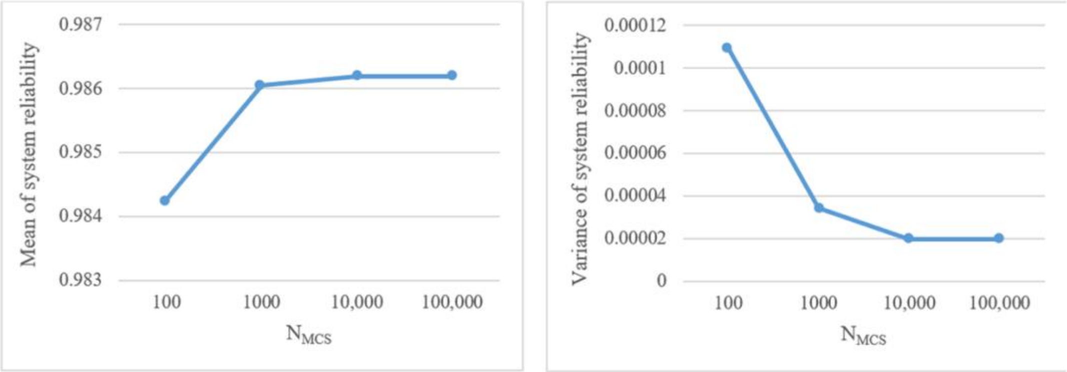


Figure 3.4: a) Mean of system reliability (left) b) variance of system reliability for different sample size of MCS (right)

In the first and second scenarios, the components are assumed to be independent of each other; however, in the second scenario the outcome of the maintenance action is treated as stochastic so as to capture the impact of uncertainty on the system reliability. When the imperfect maintenance actions are deterministic (scenario one), the maximum achievable system reliability is 0.9942 and is obtained by selecting the imperfect maintenance actions with level 1 for both components to achieve system state $X = [2, 2]$. In the second scenario, the maximum achievable system reliability is 0.9409 for selected maintenance level actions $L = [l_1, l_2] = [1, 1]$. This nearly 5% decrease in system reliability along with a certain amount of variability (0.00218) in reliability estimate are good indicators of the impact of the stochastic nature of imperfect maintenance outcomes. This seems to corroborate the assumption that deterministic considerations lead to unrealistic overestimations of system reliability. Table 3.3 presents all feasible maintenance strategies ($CV = 0$) of the second scenario. The maintenance strategy $L = [1, 1]$ results in the highest expected system reliability and the highest variance of system reliability between all feasible solutions. Despite the highest variance in system reliability, the maintenance strategy $L = [1, 1]$ still provides significantly higher reliability in comparison of other feasible maintenance strategies given in Table 3.3. It is, therefore, important that the decision makers also consider the variability in system reliability estimate and its impact on mission requirements before selecting the appropriate maintenance strategy to implement among all feasible solutions.

Table 3.3: Independent components & stochastic actions

l_1	l_2	Reliability	Variance of reliability	CV	CM	TM
0	0	0.8767	0	0	0	0
0	1	0.9152	0.00082	0	6	0.5
1	0	0.9044	0.00042	0	4	0.5
1	1	0.9409	0.00218	0	10	1

To understand the impact of S-dependence and the combined effect of S-dependence and the stochastic nature of imperfect maintenance on system reliability, we consider the last two scenarios. In the third scenario, the components are considered as S-dependent, and the outcome of imperfect maintenance actions is treated as deterministic. The results show that best maintenance strategy is $L = [1, 1]$ with system reliability 0.9826 and variance in system reliability as low as 0.00002. It is important to note that the variance estimate, though very small, is due to the uncertainty in S-dependence between components. A comparison to the first scenario also indicates a decrease in system reliability estimate due to the interaction or dependency between components, which essentially strengthens the importance of capturing and including the interactions among components of the system.

In the fourth scenario, we consider the S-dependence between components and the stochastic nature of imperfect maintenance to investigate a more realistic scenario, which captures all uncertainty introduced by several sources. As shown in Table 3.4, the optimization model provides the Pareto-optimal solutions including the maintenance strategy $L = [1, 1]$ with system reliability (0.8742) and variance (0.01110). Our comparative analysis with other scenarios clearly highlights the significant combined effect of both the S-dependence among components and the uncertainty in imperfect maintenance outcome on the obtained system reliability estimate. We notice a substantial drop in system reliability estimate accompanied by an increase in variability caused by the uncertainty stemming from these sources. This clearly supports our concern of studying a more realistic approach that captures interactions between system components and the uncertainty involved in the outcome of an imperfect maintenance action.

Table 3.4: S-dependent components & stochastic actions

l_1	l_2	Reliability	Variance of reliability	CV	CM	TM
0	0	07418	0.00023	0	0	0
0	1	0.7768	0.00085	0	6	0.5
1	0	0.8476	0.00631	0	4	0.5
1	1	0.8742	0.01110	0	10	1

To investigate the effect of other parameters on the system reliability estimate, we solve the optimization problem considering different values for these parameters. They include the random variation of the interaction effect caused by other factors, varying probabilities of success for imperfect maintenance actions, and the number of imperfect maintenance actions performed on each of the components in the past. There are several factors that do influence degradation behavior and system performance (e.g., non-critical components, or operating conditions) but are difficult to quantify. This is the reason why we first experiment with the effect of the random variation of the interaction effects on system reliability for the maintenance strategy $L = [1, 1]$ in scenario 3. To do that, we consider five different values for the standard deviation σ of the random parameter $\delta \sim N(0, \sigma^2)$ to document its impact on the obtained reliability estimates. The results of the analysis, as shown in Table 3.5, reveal the impact of random parameters on both system reliability and variance estimates. It is, therefore, important to capture uncertainty or randomness caused by these factors in system reliability estimates, whenever they appear to be present. Ignoring the uncertainty caused by these factors might lead to overestimating system reliability and risking failure of the mission.

Table 3.5: System reliability for $L = [1, 1]$ versus the random variation of S-dependence.

Random variation	$\delta = 0$	$\delta \sim N(0, 0.1^2)$	$\delta \sim N(0, 0.5^2)$	$\delta \sim N(0, 1^2)$	$\delta \sim N(0, 3^2)$
Mean of system reliability ($\bar{R}$)	0.9839	0.9826	0.9775	0.9674	0.9274
Variance of system reliability	-	0.00002	0.00004	0.00021	0.00296
Standard deviation of system reliability (σ_R)	-	0.00439	0.00639	0.014162	0.05445

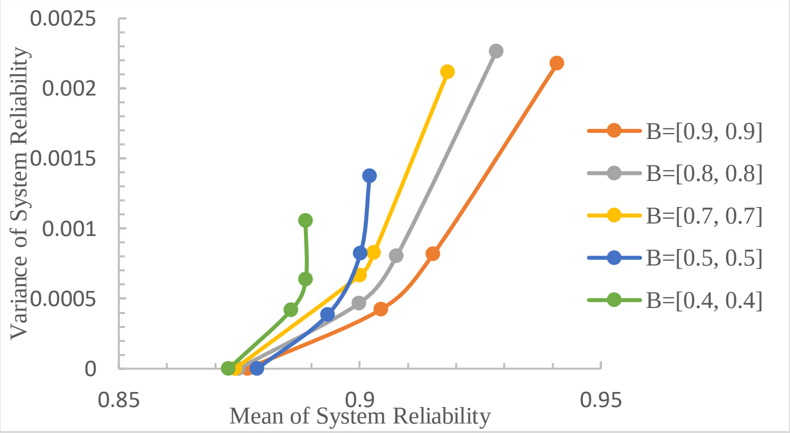


Figure 3.5: Pareto-optimal solutions for different probability of success of imperfect maintenance

As described, another important factor in our study is the probability of success of the first imperfect maintenance action carried out on a component. We now investigate the effect of this probability on the system reliability estimate and the maintenance strategy selection. This is done through solving the optimization problem in scenario 2 using different values for P_1^{IM} and P_2^{IM} denoted as $B = [P_1^{IM}, P_2^{IM}]$. Figure 3.5 represents the obtained Pareto-optimal solutions under varying probabilities of success. As can be seen in Figure 3.5, lower values of probability of success $B = [0.4, 0.4]$ not only result in the system reliability decreasing, but there is no significant difference between system reliability estimates of different optimal solutions. On the other hand, higher probabilities of success of imperfect actions $B = [0.9, 0.9]$ substantially improves system reliability in the next mission, while also providing a series of optimal solutions with significantly different reliability estimates. This understanding can help decision makers select an appropriate maintenance strategy from the available solutions. For example, if the probabilities of success of the first imperfect maintenance actions are considered to be $B = [0.9, 0.9]$, the optimal maintenance strategy $L = [1, 1]$, which consumes all available resources, has the highest reliability. Contrary to this, if the probabilities of success are in fact equal to $B = [0.4, 0.4]$, then the model provides multiple optimal solutions with reliability estimates that are not significantly different. Seeing as the resource requirements are different for these multiple optimal solutions, the decision makers could be better off selecting one which depletes fewer of the maintenance resources. Consider, as an example, two maintenance strategies $L_1 = [1, 1]$ and $L_2 = [0, 1]$, for which the system reliability estimates are not significantly different ($\bar{R}_1 = 0.8887$ and $\bar{R}_2 = 0.8888$). Since maintenance strategy L_1 requires two times more resources than strategy L_2, a decision maker would be better off selecting maintenance strategy L_2, as that would allow them to meet the reliability requirements in the next mission with fewer resources. The results imply that the probability of success of imperfect maintenance actions have a significant effect on system reliability and decision makers are encouraged to invest more resources to accurately determine these probabilities.

To show the effect of the number of imperfect maintenance actions performed in the past on each component, we consider three different values (0,1,2) for z_1 and z_2. The system reliability of maintenance strategy $L = [1,1]$ in scenario 4 is shown in Figure 3.6. As can be seen, this strategy is sensitive to the changes in z values. Even though the system reliability of the maintenance strategy L is higher than the reliability level 0.95 when $z_1 = z_2 = 0$, it starts decreasing for other values of z_1 and z_2. If we do not consider the past, then the system reliability is overestimated. Therefore, when there is not enough information available about the history of maintenance actions and when the unreliability cost is high, it is preferred to spend more maintenance resources and perform maintenance strategies that are less sensitive to the changes of z values such as $L = [2,0]$ and $L = [2,2]$.

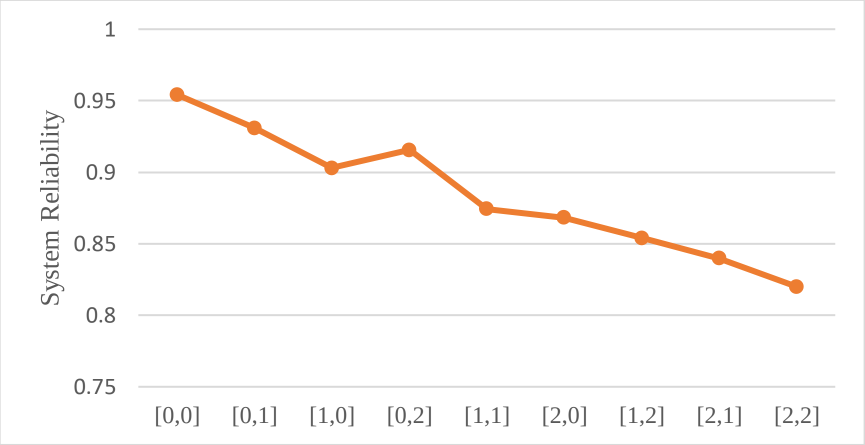


Figure 3.6: System reliability of $L = [1,1]$ with different values of z_1 and z_2

Our results of all four scenarios are presented in Table 3.6 and Figure 3.7. In this example, we get different estimations of the expected value and variance of the system reliability under maintenance strategy $L = [1,1]$ in each scenario. It is critical for some systems and shows the importance of considering the actual scenario to determine the optimal maintenance actions in real world problems.

Table 3.6: Selective maintenance results for different scenarios of example 1

	Maintenance Scenario	l_1	l_2	$\bar{R}$	σ_R^2	$\bar{R}+\sigma_R$	$\bar{R}-\sigma_R$
1	Independent components & deterministic actions	1	1	0.9942	–	0.9942	0.9942
2	Independent components & stochastic actions	1	1	0.9409	0.00218	0.9876	0.8942
3	S-dependent components & deterministic actions	1	1	0.9826	0.00002	0.9870	0.9783
4	S-dependent components & stochastic actions	1	1	0.8742	0.01110	0.9796	0.7688

In the fourth scenario the mean of system reliability is 0.8742 and its variance is higher compared to the other scenarios. If the desired level of the system reliability is 0.9, then maintenance strategy $L = [1, 1]$ would not be optimal. Therefore, the decision makers must increase their time and budget limitations to turn a more robust maintenance strategy with higher reliability, such as $L = [2,1]$ and $L = [1,2]$.

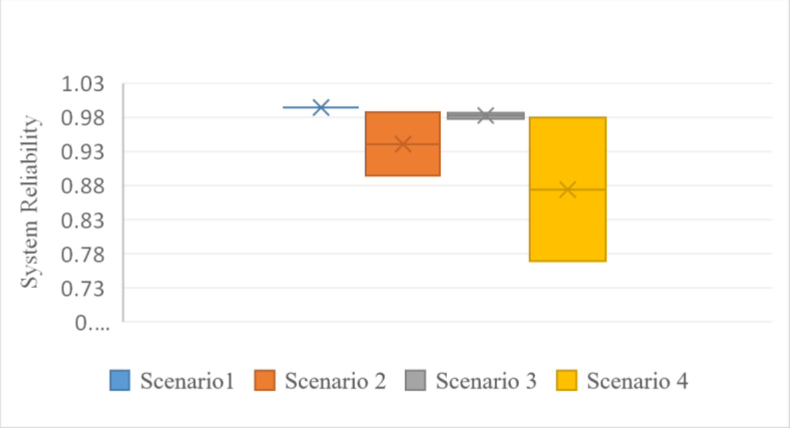


Figure 3.7: The mean system reliability and standard deviation for strategy $L = [1, 1]$ in four scenarios.

3.5.2. Example 2

In the second example, we consider an MSS consisting of five multi-state components in the series structure given in [40]. Each component can be in one of four possible states, $s_i =$

$0, 1, 2, 3$, with corresponding performance rates $g_i = 0, 1, 2, 3$. The intrinsic transition rates are given in Table 3.7. At the beginning of the maintenance break, the state of the system is $Y = [2,0,1,1,0]$. The number of imperfect maintenance actions performed on the components before the current maintenance break are $z_1 = z_2 = z_4 = z_5 = 0$, $z_3 = 1$, and the initial probability of success for imperfect maintenance actions are $P_1^{IM} = 0.9, P_2^{IM} = 0.99, P_3^{IM} = P_4^{IM} = 1$, and $P_5^{IM} = 0.9$. The maintenance cost (in \$1000) and time (in days) matrices for each component are given as follows:

$$C_1 = \begin{bmatrix} 0 & 4.5 & 9 & 14.5 \\ 0 & 0 & 4 & 12 \\ 0 & 0 & 0 & 9.5 \\ 0 & 0 & 0 & 0 \end{bmatrix}, C_2 = \begin{bmatrix} 0 & 3 & 8.5 & 16 \\ 0 & 0 & 6 & 12 \\ 0 & 0 & 0 & 4.5 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \ C_3 = \begin{bmatrix} 0 & 2.5 & 6 & 10 \\ 0 & 0 & 4 & 8 \\ 0 & 0 & 0 & 3.5 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$C_4 = \begin{bmatrix} 0 & 2.5 & 4.5 & 9 \\ 0 & 0 & 3 & 6.8 \\ 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \ C_5 = \begin{bmatrix} 0 & 3 & 6 & 10 \\ 0 & 0 & 3 & 5.5 \\ 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 \end{bmatrix}, T_1 = \begin{bmatrix} 0 & 1 & 2.5 & 3.5 \\ 0 & 0 & 1.5 & 2 \\ 0 & 0 & 0 & 1.5 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$T_2 = \begin{bmatrix} 0 & 1.5 & 2 & 5 \\ 0 & 0 & 1.5 & 3 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, T_3 = \begin{bmatrix} 0 & 1 & 2.5 & 5 \\ 0 & 0 & 2 & 3.5 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, T_4 = \begin{bmatrix} 0 & 1 & 3 & 4.5 \\ 0 & 0 & 1.5 & 3 \\ 0 & 0 & 0 & 1.5 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$T_5 = \begin{bmatrix} 0 & 1.5 & 2 & 3 \\ 0 & 0 & 1 & 2.5 \\ 0 & 0 & 0 & 1.5 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Table 3.7: The transition rates of components

Component	Transition rate (Year $^{-1}$)		
	$\lambda_{i,3}$	$\lambda_{i,2}$	$\lambda_{i,1}$
1	0.08	0.15	0.1
2	0.06	0.11	0.05
3	0.14	0.09	0.2
4	0.18	0.1	0.15
5	0.11	0.08	0.16

The goal is to find the optimal selective maintenance strategy, subject to the maintenance time limit of T_0=12 (in days) and the maintenance budget limit of C_0= 40 (in $1000). The goal is to maximize the system reliability and minimize the variance of system reliability at the end of the next mission of duration τ= 0.5 (in years). The additional system requirements are for a demand level $D=2$ and $\delta \sim N(0, 0.1^2)$.

Given that demand level D is equal to two, we consider the system states with performance rates higher than or equal to the demand level as acceptable states of the system at the beginning of the next mission. The total number of possible states that a system can acquire and are acceptable are 32 (each component can be in state 2 or 3). Out of these 32 possible states, only 20 states do not exceed the limit of maintenance time and budget constraints and, hence, compose our set of feasible solutions. First, we compute the system reliability for all feasible states using the same approach as discussed in the previous example. We solve the optimization model to find the maintenance actions providing maximum system reliability with minimum variance in system reliability estimate considering S-dependency between components and stochastic maintenance actions. We also solve the selective maintenance optimization problem assuming that the system components degrade independently. Table 3.8 provides a comparison of the results obtained from our proposed approach with other approaches.

Table 3.8: Selective maintenance results for example 2

Optimal maintenance strategy for the system with	System state	C_M	T_M	$\bar{R}$
Proposed approach (S-dependent components (two-way interaction) & stochastic actions)	$X_1 = [3, 2, 3, 2, 3]$	39	11.5	0.8696
Considering Independent components	$X_2 = [3, 2, 3, 3, 2]$	39	12	0.9069
Approach proposed in [40] (S-dependent components)	$X_3 = [3, 3, 2, 2, 2]$	38.5	12	0.8584

The proposed approach provides an optimal maintenance strategy with the highest system reliability (0.8696), which suggests replacing components 1, 3, and 5, and performing imperfect maintenance actions on components 2 and 4. The system reliability estimate provided by our proposed approach is higher than the one proposed in [40] but lower than the scenario where all components degrade independently, which, however, is not a realistic assumption. The maintenance resources used for all three system states are nearly the same. To further investigate the effectiveness and realism of the proposed approach, we estimated the reliability for system states $X_2 = [3, 2, 3, 3, 2]$ and $X_3 = [3, 3, 2, 2, 2]$ using the proposed approach (considering S-dependent components and stochastic actions). The new reliability estimates of these two systems states (X_2 and X_3) are 0.7834 and 0.5267, which are much lower than the reliability estimates of system state (X_1) provided by the proposed approach. This comparison and analysis clearly demonstrate that ignoring S-dependence between components and the effects of previous maintenance actions can provide an unrealistic estimate (overestimation) of system reliability that may lead to selecting a poor maintenance strategy.

We further investigate the capability of this model to deal with higher reliability requirements and resource constraints. Assuming that the required system reliability in the next mission must be greater than or equal to 0.9, then the current solution (see Table 3.7) obtained by our approach does not satisfy this requirement. The insight is that to meet a higher target of system reliability, one would need more maintenance resources. However, it is not always true that increasing maintenance resources will necessarily lead to higher reliability estimates. We investigate two different scenarios to understand how the proposed approach helps select a maintenance strategy within given resource constraints. For example, the reliability of system state $X = [3, 2, 3, 3, 3]$ is 0.9281 with total maintenance cost and time requirements as 42.8 and 13, respectively. On the other hand, to get the system in state $\acute{X} = [3, 3, 2, 3, 3]$, the maintenance strategy requires even more resources ($C_M = 46.3$ and $T_M = 14.5$) to implement as compared to the system state X. However, the system reliability for system state $\acute{X}$ is 0.6734, which is much lower than the reliability of system state X. This considerable difference in system state reliability estimates can be explained by the fact that in state $\acute{X}$, component 3 is selected for imperfect maintenance assigning lesser maintenance resource even though component 3 appears to be more critical. Its lower probability of success after scheduling an imperfect maintenance action leads to a lower system reliability estimate. It is important to note that for achieving a certain system reliability target for the next mission, both flexibility in increasing the maintenance resources and assigning them to critical components are very important. The proposed approach facilitates the selection of critical components for assigning more resources to achieve the given reliability target and provides more effective maintenance strategy.

3.6. Summary

In this chapter, we studied the selective maintenance problem for multi-component series systems with multi-state components. The system performs several successive missions separated by scheduled breaks during which maintenance of its components takes place. A component can receive different maintenance actions characterized by various levels and different probabilities of success. For the first time, we considered the history of maintenance actions performed in previous breaks and modeled their effects on the probability of success of imperfect maintenance actions. Because of stochastic imperfect maintenance action, the system can be in different states with certain probabilities which lead to uncertainty in the reliability of the system in the next mission. Moreover, the S-dependence between components of the system as well as the stochastic interaction effects were considered. We used MCS to compute the reliability of the system in the next mission considering the stochastic interaction between components. We also presented a selective maintenance optimization model to find the best maintenance strategy when maximizing the expected system reliability and minimizing the variance of system reliability in the next mission subject to time and cost constraints of the selected maintenance actions. We then demonstrated the importance of considering S-dependence between components and the stochastic imperfect maintenance actions solving the optimization problem for two different series systems. The examples and their results showed that stochastic imperfect maintenance actions can have significant effects on the obtained reliability estimates. Also, ignoring S-dependence can lead to overestimation of the system reliability. Overall, the outcome of this research provided a useful reference for selective maintenance optimization of multi-component series systems having an intractably large system state space.

4. PREDICTING REMAINING USEFUL LIFE OF A MULTI-COMPONENT SYSTEM BASED ON INSTANCE-BASED LEARNING[3]

Condition-based maintenance (CBM) is an effective maintenance strategy that increases the safety and reliability of engineering systems. Prognostics, which is the basis of CBM, deals with predicting the future behavior and remaining useful life (RUL) of systems. In many applications, the system is composed of multiple components that are stochastically dependent (S-dependent). Most of the earlier works that characterize RUL estimation of the multi-component systems are developed on the premise that components in a system are independent. The existing methods fail to perform when the components are interdependent. This chapter proposes a RUL prediction approach based on instance-based learning considering the S-dependency between multiple components of a given system. To show the effectiveness of the proposed approach, we used a simulated dataset. The superiority of the proposed method is demonstrated by comparing the obtained results with another popular approach and the state-of-the-art results on the same dataset [67].

4.1. Introduction

The prognostics and health management (PHM) discipline focuses mainly on predicting the remaining useful life (RUL) of a system based on condition monitoring data or degradation signals [12], [14], [68], [69]. RUL prediction is defined as predicting the duration from the current

[3] The present chapter is based on the following paper:
A. F. Shahraki, A. Roy, O. P. Yadav, and A. P. S. Rathore, “Predicting remaining useful life based on instance-based learning,” published in 2019 Annual Symposium on Reliability and Maintainability (RAMS)
Contribution of Ameneh Forouzandeh Shahraki: developing the mathematical models, analysis of the case study, discussion of the results, and drafting the paper. Contribution of Arighna Roy, Om Yadav, Ajay Rathore and: verification of the results and proofreading the draft paper.

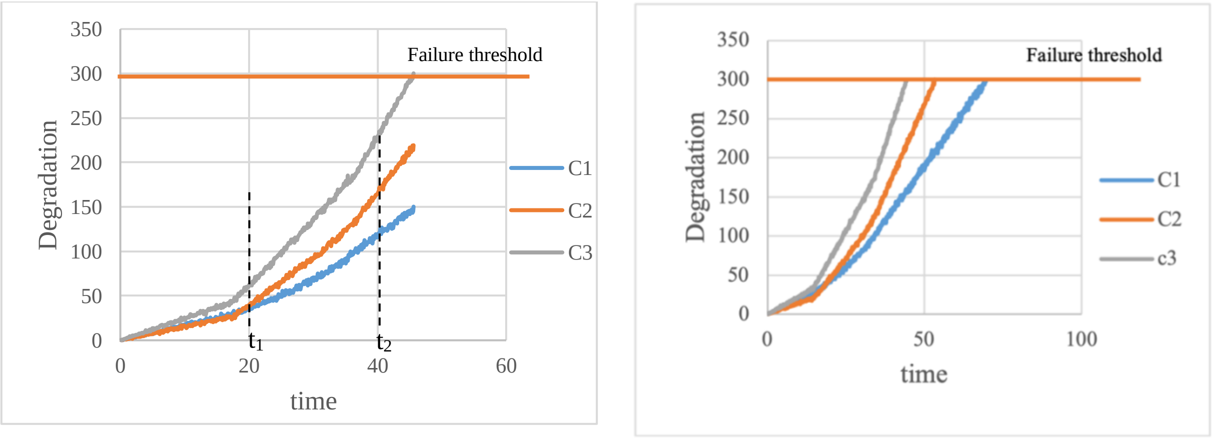
time to the end of the useful life of a system [12]. The reliable and accurate prediction of RUL helps formulate the best preventive maintenance strategy to increase system reliability and safety, and to avoid sudden system shutdowns [4], [70]. In general, RUL prediction can be implemented using model-based and/or data-driven approaches [12], [69]–[72]. The model-based approaches involve explicit mathematical functions, established empirical, or analytical models to capture the degradation and failure behavior of the system. On the other hand, data-driven approaches rely on historical data when it is impossible to apply the domain knowledge of system failure behavior, or the system complexity is very high. Data-driven approaches are relatively easy to be generalized and have recently become more attractive with advances in sensor technologies and data analyses methods.

Numerous data-driven approaches have been proposed to predict the RUL of a system [14]. Some approaches such as the general path models [73], [74] and stochastic process models [12] first monitor and predict the evolution of a degradation signal and then estimate the RUL as the time needed for the signal to reach the end of life criteria (failure threshold). These approaches require fixing and known failure thresholds. On the other hand, some approaches such as support vector machine (SVM) [75] and neural networks (NN) [76], [77], employ the time-to-failure data to learn a mapping between sensor signals and the corresponding RUL. These approaches are capable of obtaining promising results with abundant training data. However, degradation data are usually insufficient in real-world applications. The limitation serves as motivation for similarity-based/instance-based methods, which have been proven effective in predicting RULs with limited data [78], [79]. The similarity-based method addresses the prognostics problem by directly employing the historical time-to-failure data to estimate a test system's RUL. It is easier to implement with no necessity of having domain-specific knowledge to achieve satisfactory prognostic performance.

Existing literature on RUL prediction mostly has focused on the degradation process of a simple system, assuming a single degradation signal is able to fully characterize the degradation process of the system, and there is only a single failure mode. However, most of the engineering systems in real-world applications are complicated and composed of multiple components that lead to multiple failure modes. Prediction of the RUL of such complex systems is challenging, especially when the components of these systems are S-dependent, i.e., the failure or degradation of some components in the system could affect the failure or degradation of other components in the system [8], [10]. Although some works have focused on interactions caused by the failure of a component, the number of publications considering degradation dependency is still limited. Bian and Gebraeel [10] assumed that the degradation interactions occur at discrete states, and the degradation rate of a component increases when the degradation state of other interdependent components increases in amplitude or intensity. They extended their work the cases that the degradation interaction may occur in a continuous manner [9]. They have made some assumptions that limit their application in practice. For example, in [10], the authors assumed that when the degradation state of the influencing component increases in amplitude, the degradation rates of the affected components increase by a constant amount (δ). Although it helps to easily model the degradation process of each component, the assumption of constant δ is difficult to satisfy for many applications that the change in the degradation rate may depend on parameters such as the age or degradation states of the interdependent components. Moreover, developing a specific model to capture the degradation process of a multi-component system and the interaction between its components is not possible without having domain-specific knowledge. We propose a novel data-driven approach to directly estimate the RUL of a complex system using the similarity between the test and historical time-to-failure (or called here degradation-to-failure) data without relying on these assumptions. It is easier to implement with no necessity to involve domain-specific knowledge about the degradation trend of the historical data.

4.2. The Problem Description

We consider a system composed of N critical interdependent components, $\mathrm{C_i}\colon \mathrm{i} = 1,2,..,\mathrm{N}$. The system performance depends on the performance of its components that their health status gradually deteriorates over time. We assume that when component $\mathrm{C_i}$ transitions to a more severe degradation state, it causes the increase in the degradation rates of other interdependent components by an amount that depends on the degradation state of both components and is not necessarily constant and may change during the time. As an illustration, consider a system with three interdependent components, which the degradation state of each of them influences the degradation rates of the remaining ones. A possible path of system degradation is depicted in Figure 4.1 for a system with series structure (a) and parallel structure (b). As we see, there are different change points that the degradation rates of the components will change. For example, in Figure 4.1 (a) at time t_1 the degradation rates of C_3 and C_2 have changed because of the increase in the degradation level of C_1. The difficulty of the considered problem comes from the fact that the dependency between components leads to variable degradation rates of each component during its lifetime.



(a)

(b)

Figure 4.1: A sample path of the degradation process for a system with (a) series structure (b) parallel structure

4.3. The Proposed Methodology for RUL Prediction

We predict the RUL of a test system (operating system) based on RULs of other systems with similar degradation behaviors. At time $t = 0$, all components of the test system start from perfect condition with the initial degradation level $D_i(t = 0) = 0$ $(\forall i = 1,2, \ldots, N)$. The degradation of each component is accumulated during time and monitored until prediction time t. Due to the degradation interaction between the components of the complex system, we derive the RUL of components and the system by bringing the state of all components into consideration. The RUL of the operating system is determined using an approach based on K-Nearest-Neighbor (KNN) algorithm. We consider the historical degradation-to-failure data to predict the RUL of the test system as the weighted average of RUL of K most similar trajectories to the test trajectory.

The proposed methodology is composed of three steps to estimate the RUL of the system in operation. The first step is to evaluate the distance/similarity between the trajectories of the test system and the historical trajectories considering the pointwise difference between the sequences of observations of their components. In [80]–[82], only the last observations of the test trajectory are used to find the similarity level. Because of the importance of all previous observations, we consider all the observations from the beginning to prediction time t. The similarity measure between two different systems is calculated considering the similarities between their components. In the second step, we find the K most similar systems to the test system. Finally, the weighted average of the RUL of the K selected systems is computed as an estimation of the RUL of the test system. The weight of each system is related to its similarity to the test system. The algorithm of the prediction RUL of the test system is shown in Figure 4.2. Note that we need to determine the best value of parameter K in KNN to get the best possible prediction results.

```
Input: A set of historical trajectories ($\mathbb{T} = \{\boldsymbol{T}^r\}_{r=1}^{R}$) and the test trajectory until prediction time $t$ ($\mathbf{T}^{test}$)
Output: The predicted $RUL$ of test system at time $t$ ($\widehat{RUL_t}$)
for each $\boldsymbol{T}^r$ in $\mathbb{T}$ do
        separate the first $t$ time steps from the beginning of degradation process of all $N$ components
        for component $C_i$ of the $r^{th}$ system do
                compute the similarity between component $C_i$ of $r^{th}$ system and test system
        end for
        compute the similarity between $r^{th}$ system and test system ($\mathrm{s}_t^r$)
end for
find $K$ systems that have the lowest $\mathrm{s}_t^r$values
compute the $RUL$ of the test system $\widehat{RUL_t}$
```

Figure 1.2: The algorithm for predicting the RUL

The historical trajectories contain the monitored degradation-to-failure data of R systems. The components of each system may fail at different times that causes the time series of different components to have different lengths. For the series system, the failure of each component leads to the failure of the system as shown in Figure 4.1(a). Therefore, the time series of the failed component is complete, i.e. the degradation data until the failure time is available only for the failed component. $\mathbf{T}^{\mathrm{r}}_{0:\mathrm{FT_r}} = \left[\mathrm{T}^{\mathrm{r}}_{1,0:\mathrm{FT_r}}, \mathrm{T}^{\mathrm{r}}_{2,0:\mathrm{FT_r}}, \dots, \mathrm{T}^{\mathrm{r}}_{\mathrm{N},0:\mathrm{FT_r}}\right]$ ($\forall$ r $= 1,2,\dots,$R) represents N time series data of r^{th} system from the beginning until the failure time ($\mathrm{FT_r}$). $\mathbf{T}^{\mathrm{test}}_{0:\mathrm{t}} =$

$[\mathrm{T}^{\mathrm{test}}_{1,0:\mathrm{t}}, \mathrm{T}^{\mathrm{test}}_{2,0:\mathrm{t}}, \dots, \mathrm{T}^{\mathrm{test}}_{\mathrm{N},0:\mathrm{t}}]$ is time series data of N test components from the beginning until time t. The pointwise difference between $\mathbf{T}^{\mathrm{test}}_{0:\mathrm{t}}$ and $\mathbf{T}^{\mathrm{r}}_{0:\mathrm{FT}_\mathrm{r}}$ is calculated as follows:

$$\mathrm{s}^r_t = \sum_{i=1}^{N} \sum_{x=0}^{\min\,(lenght(T_i^{test}),t)} (T^{test}_{ix} - T^r_{ix})^2 \qquad \forall \mathrm{r} = 1, .., \mathrm{R} \tag{4.1}$$

The RULs of the K most similar system to test system, i.e., the systems that have the lowest s^r_t values, are used to get their weighted average as an estimation of RUL ($\widehat{RUL_t}$). The weight of r^{th} system (w^r) and $\widehat{RUL_t}$ are calculated as:

$$w^r = \exp\,(-\mathrm{s}^r_t) \tag{4.2}$$

$$\widehat{RUL_t} = \frac{\sum_{j=1}^{K}(w^j * RUL^j_t)}{\sum_{j=1}^{K} w^j} \tag{4.3}$$

To determine the RUL of each system (RUL^j_t), we need to consider its structure. For a series system, failure of any of the components leads to system failure. Therefore, the life of the system depends on the first failed component. On the other hand, in a parallel structure, the life of the system depends on the last failed component. Note that for a parallel system that we have complete data for all components, we can estimate RUL of each component and plan their maintenance separately.

4.4. Numerical Study and Results

Different simulated data sets are used to verify the performance of our proposed approach. We consider a system consisted of three components that are degrading over time. We use the same baseline parameter values as the work of Bian and Gebraeel [10]. The generated dataset is composed of multiple multivariate time series signals. Each time series represents a degradation path of a component of the system. The components are interdependent, and their degradation processes are simulated using the Equation (4.4). The baseline parameters values [10] are given in Table 4.1.

Table 4.1: Baseline parameter values [10]

Component Index	Component C_1	Component C_2	Component C_3
d_i	300	300	300
M_i	3	3	3
μ_i	1.8	1.4	2.3
τ_i	0.2	0.3	0.1
ξ_i	92	92	92
θ_i	9.1	9.1	9.1
$\delta_{1,j}$	0	0.12	0.23
$\delta_{2,j}$	0.04	0	0.02
$\delta_{3,j}$	0.05	0.07	0

$$D_i(t) = D_i(0) + \int_0^t [k_i + \sum_{j\neq i} \delta_{j,i} h_j(D_j(v))]dv + \varepsilon_i\,(t) \tag{4.4}$$

- $D_i(t)$ and $D_i(0)$ are the degradation signals of the component C_i at time t and zero, respectively.
- $k_i + \sum_{j\neq i} \delta_{j,i} h_j(D_j(v))$: degradation rate of component C_i, is a linear function of its inherent degradation rate (k_i) and the degradation states of other influencing components $h(D(t))$.
- h$(D(t))$ is a function of the degradation state of all components.
- $\delta_{j,i}$ represents the incremental change in the degradation rate of component C_j when component C_i transitions to a more severe state.
- $\varepsilon_i(t)$ is used to model the noise level of the degradation signal.
- $\varepsilon_i(t) = B_i(t)$, $B_i(t)$ follows a stationary Brownian motion process with diffusion parameter σ_i^2, i.e.,$B_i(t){\sim}N(0, \sigma_i^2$t).
- k_i, σ_i^2 are stochastic and change from one system to another: $\sigma_i^2{\sim}\Gamma^{-1}(\xi_i, \theta_i)$, $k_i|\sigma_i^2{\sim}N(\mu_i, \tau_i\sigma_i^2)$
- $(\theta_i, \delta_{i,j}) \in \{(m_1 \times \theta_i, m_2 \times \delta_{i,j})): m_1, m_2 = 1,2, \ldots ,20\}$.

- M_i shows the number of states for each component.

We also investigate the impact of two critical factors on the performance of RUL prediction approach. These two factors, the level of the degradation signal noise and the value of the degradation interaction, are changed by changing the values of m_1 and m_2. We simulate degradation paths of three components for different systems considering different parameter settings.

The simulated degradation data set for three components of different systems is divided into two groups. One group is the historical degradation dataset and another one is the test degradation dataset for different systems. The first group will be used to compute the RUL of the test systems. First, we investigate the impact of the different levels of K and training size on the RUL prediction of a system. Table 4.2 shows the MSE computed at three life percentiles (50%, 70%, and 90%) for different value of K. For example, the 50th life percentile implies that 50% of the system's lifetime was attained at the time the prediction was evaluated. As we see, the $K = 2$ is the best value for KNN algorithm that leads to minimum MSE values.

Table 4.2: Prediction error for different K values in a parallel system at three life percentiles

	MSE		
K	50%	70%	90%
1	29.5532	25.627	21.0386
2	16.0416	15.7266	14.2064
3	18.6172	18.594	17.0316
4	19.6778	17.4674	16.2812
5	24.3358	20.9458	17.5464
6	26.5228	22.1388	17.7522

Moreover, to verify the capability of the proposed approach to provide accurate RUL predictions, we compute the MSE for 50 test trajectories considering different number of training trajectories. Although increasing the training size leads to smaller MSE, the value of MSE in case of limited number of training trajectories is still good enough (Figure 4.3). To determine the optimal training size, we must consider both the cost of collecting more training data and the value of the getting more accurate RUL prediction. As we see, for training size bigger than 40, the change in MSE value is not significant. Therefore, the decision makers must choose the optimal training size based on the degree of importance of the system RUL prediction and the available budget.

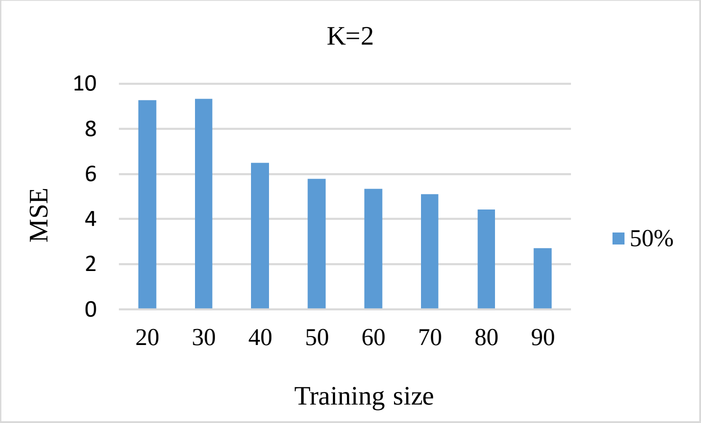


Figure 4.3: Comparison of the MSE value for different training size when $K = 2$

After determining the best value of K, we predict the RUL for different test systems. As shown in Figure 4.4 for one series system at different life percentiles $(10\%, 20\%, 30\%, \ldots, 90\%)$, the estimated RUL is reasonably close to the true RUL. Moreover, for 50 test systems, their predicted RUL and true RUL at 50% life percentile are shown in Figure 4.5.

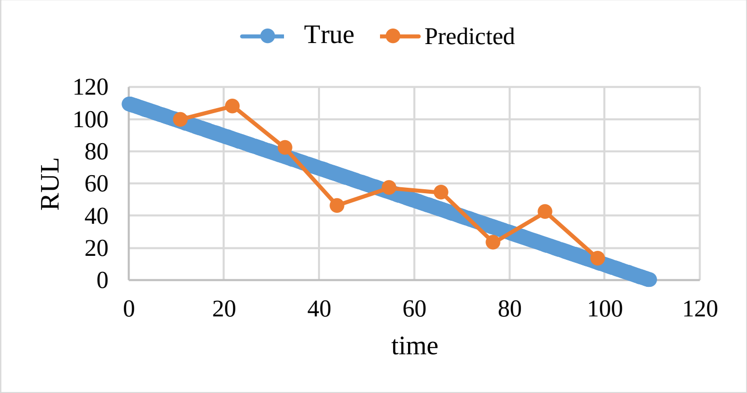


Figure 4.4: True and predicted RUL for a series system

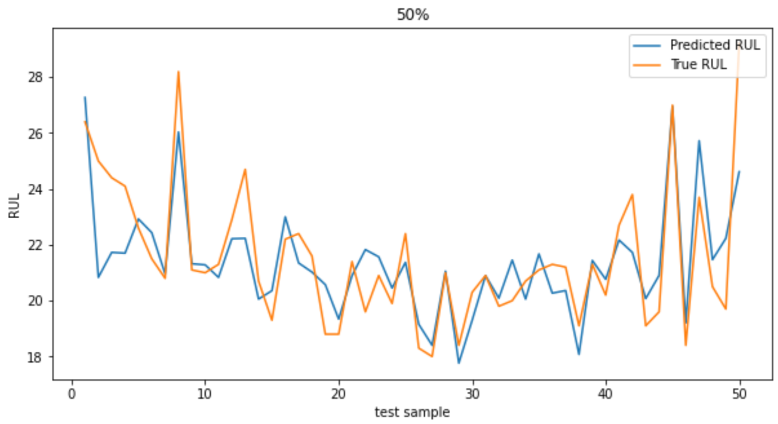


Figure 4.5: True and predicted RUL for 50 test systems at 50th life percentile

As mentioned before, the level of the degradation signal noise and the value of the degradation interaction are two main model parameters. Therefore, we evaluate the prediction errors for different levels of these two parameters as shown in Figure 4.6. The prediction errors are computed for 50 test systems for 20 different levels of parameters. As we see, the MSE values increase as the value of signal noise increases (m_1). However, a different trend of the change in the MSE values can be seen when the value of interaction (m_2) increases. Interestingly, our approach performs better when the effects of degradation interactions become more significant. We believe that one of the primary reasons for this improvement is that one component becomes critical component, i.e., most of the time this component (for example, C_3) fails first and leads to failure of the series system.

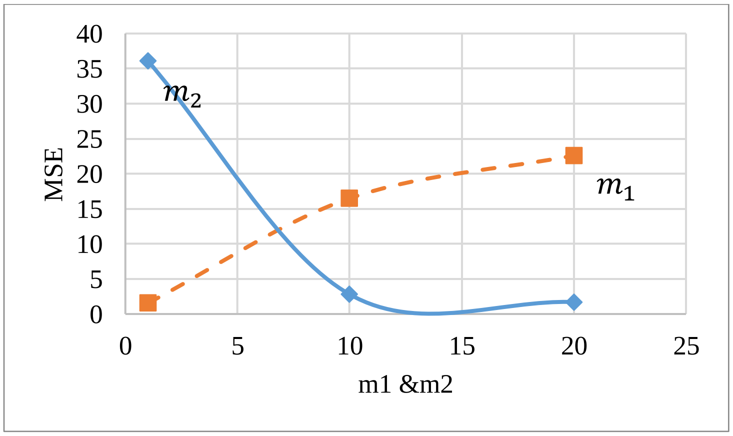


Figure 4.6: MSE values for different values of m_1 and m_2

To demonstrate the effectiveness of our approach, we calculated the mean parentage error (MPE) using the method in [10] and compared the obtained results with their results as shown in Table 4.3. The MPE for different life percentiles using our approach is close or smaller than their MPE. We can conclude that our approach outperforms their approach even. Another advantageous of our approach is that we did not make any assumption about the interaction between components. Therefore, it can be used when the degradation interaction is not necessarily constant and may change during the time. We solved the problem for a series system when the interaction between components is a function of age of the system. The mean prediction error for 50th, 70th, and 90th of life of the system are less than 1%. It shows that our approach is capable of successfully predicting the RUL of system with different types of interaction between its components using historical degradation-to-failure.

Table 4.3: Comparison results for three life percentiles when $(m_1, m_2) = (20,20)$

Life percentile	50%	70%	90%
MPE of our approach	2.6%	2.29%	2.01%
MPE of approach in [10]	25%	12%	7%

4.5. Summary

In this chapter, we presented a data-driven approach for RUL estimation of a complex system composed of S-dependent components using a similarity-based matching method. The proposed approach was successfully applied to predict the RUL of simulated systems. The results showed that the approach can be applied for RUL prediction of highly complex systems without prior expertise on the system behavior and the failure threshold. One limitation of our approach is the need to have big training trajectories, which can be a problem in some systems with limited history.

5. SELECTIVE MAINTENANCE OPTIMIZATION FOR MULTI-STATE SYSTEMS OPERATING IN DYNAMIC ENVIRONMENTS [4]

This chapter deals with the selective maintenance of a multistate series system working under time-varying environmental/operating conditions. The operating conditions are evolving dynamically during the mission and influence the degradation rate of each component and the whole system [83].

5.1. Introduction

Many complex systems consisted of multiple components such as wind turbines and power generation systems have to perform multiple consecutive missions [48]. As the components of the system deteriorate with usage, their reliability decreases. During the interval between any successive missions, the maintenance actions can be performed to keep the system in normal state and increase the probability of successfully executing the next mission. Since there are limited resources for the maintenance activities during the maintenance break, it is important to decide on the best maintenance strategy (e.g., the subset of components to maintain, the level of maintenance actions, etc.) considering the system's requirements. Research on selective maintenance has been focused on developing optimization frameworks and statistical models to intelligently distribute maintenance resources across all components of a system [40], [48], [54], [56], [60].

[4] The present chapter is based on the following paper:
A. F. Shahraki and O. P. Yadav, “Selective Maintenance Optimization for Multi-State Systems Operating in Dynamic Environments,” published in 2018 Annual Symposium on Reliability and Maintainability (RAMS).
Contribution of Ameneh Forouzandeh Shahraki: developing the mathematical models, analysis of the case study, discussion of the results, and drafting the paper. Contribution of Om Yadav verification of the results and proofreading the draft paper.

Most of the studies on selective maintenance have assumed that the operating conditions remain constant during the entire mission time or have no effect on the degradation and failure processes of the system. In practice, the operating conditions vary stochastically during the mission time and the system may be exposed to different operating conditions [84]. Under such circumstances, the degradation and failure process of the system can greatly decelerate or accelerate. For example, temperature changes directly affect the characteristics of electromagnetic relays and increase their degradation rate. For another example, an air conditioning system used in a warehouse operates at different power levels, depending on the external environment, to keep the warehouse temperature at a certain level. Generally, if a system works at a higher load, it will have a higher degradation rate [85]. Therefore, it is important to model the evolution of operating condition and capture its influence on the degradation process of the system.

Observing that the operating condition often varies alternatively and cyclically, some researchers introduced a regime switching model to depict systems operating under time-varying operating condition, where the evolution process of the operating condition is modeled by a Markov process. In [86] a Markov regime switching model was used to describe two alternative environments. Later, this model was extended by considering a system that operates cyclically in multiple environments [81],[83],[84]. Generally, the effect of the operating condition is considered by modeling that one or more degradation parameters will vary with operating condition. Cekyay and ¨Ozekici [87] considered a multi-state system, where the operating condition would affect the transition rate matrix of system states. Zhu et al. [89] adopted a Gamma process to model the degradation process and assumed that the operating conditions would affect the shape parameter of the Gamma process. Bian et al. [88] introduced the future operating profile into the drift coefficient in the Wiener process. In these studies, the consequence of operating condition on the degradation process is independent of the current system degradation state. However, in many situations, the consequence of operating condition could vary at different health states due to varying strength of the system. Taking the fatigue crack growth for example, the consequence of loading condition is lower in the initial period of fatigue crack compared to next periods.

Recently, a load and state dependent degradation rate model was proposed in [90] that captures the effects of the load applied on a component and its degradation state on the degradation rate of the component. One drawback of the defined model is that a degraded component compared to a new one has a lower degradation rate. However, as the component's degradation level increases, the degradation rate of the component might increase, and the component resistance to failure might be reduced. Moreover, it was assumed that the load applied on the component is comprised of two parts: a known base-load profile that represents the load trend in the next mission and a random variation that captures the uncertainty associated with the future loading conditions. For many operating conditions like usage profiles, it is difficult to clearly determine the base function in the next mission.

In this chapter, we extend the previous models by capturing the effects of age and degradation state (cumulative damage) of each component, as well as the effects of dynamic operating conditions on the degradation rate. A convenient method to integrate these effects in the degradation modeling is applying the proportional hazards (PH) model. PH model has been widely applied in the area of reliability analysis for various critical engineering systems [14][91]. The PH model uses internal and/or external covariates to compute the hazard rate function or instantaneous risk of failure over time [92]. We treat the degradation state as the internal covariate in the PH model. The degradation process of each component is modeled as a discrete degradation process by a multi-state deteriorating Markov process. When a component degrades gradually from one state to another one, its cumulative damage starts increasing, which influences the degradation rate of the component. On the other hand, we consider the operating condition of the system as the external covariate in PH model. The dynamic operating condition in the next operating mission is modeled as a homogeneous Continuous-Time Markov Chain (CTMC) whose state dwell times are random. Since the degraded components compared to new ones are more vulnerable to harsh operating conditions, we consider that the effect of operating condition on the degradation rate of each component depends on its degradation state. Our model overcomes the identified limitations of previously studied models. We use the proposed model to determine the best maintenance strategy of a multi-state series system working under dynamic operating condition.

5.2. Proposed Approach

We consider a complex system that consists of N multi-state components connected in series structure that reflects many real-world applications. Each component i $(i = 1,2,.., N)$ has $K_i + 1$ different states $s_i(= 0,1, \dots., K_i)$ with different performance rates, g_{i,s_i}. The state K_i is the perfect functioning state, the state 0 is the failure state, and the states between 0 and K_i are the intermediate states of the i^{th}component. It is considered that transitions are only left-to-right, and a component goes from state k to its next lower state $k - 1$ before moving to state $k - 2$. In a series system, failure of one component leads to the system failure. The state of the system at each time is defined as the state of its most degraded component.

The system operates under dynamic operating conditions. The operating conditions vary stochastically according to CTMC with M states. At each time, the system' operating condition occupies one of the M states. We assume that the total number of operating states is $M = 3$. The state transition diagram of a multi-state component working in multi-state operating condition is shown in Figure 5.1. $\mu_{m,n}$ is the transition rate of the operation condition from state m to state n

$(m \neq n)$ and $\mu_m = \sum_{m \neq n} \mu_{m,n}$ is the total rate of leaving state m $(m = 0,1,2)$. The transition rates of the operating condition can be obtained from historical data of previous missions. $\lambda_{i,k,z}$ is the transition rate or failure rate of i^{th} component from state k to lower state $k-1$ in condition state $z(\text{z} = 0,1,\ldots,\text{M})$. The transition rate of each component depends on two states involved in the transition, the age of the component, and the state of the operating condition that their effects are captured using PH model.

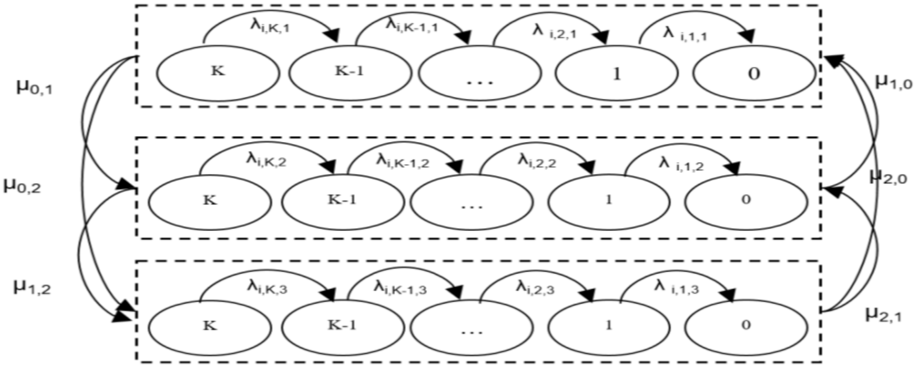


Figure 5.1: The state transition diagram of a multi-state component in CTMC operating conditions with 3 states

The PH model is a convenient and widely used method to model hazard depending on covariates and recently has received attention from researchers in the reliability field. The basic PHM is given as:

$$\lambda(t; \boldsymbol{z}(t)) = h_0(t) \exp\left(\boldsymbol{\gamma z}(t)\right) \tag{5.1}$$

where:

- $\lambda(t; \boldsymbol{z}(t))$ is the hazard rate at time t for a component with covariates $\boldsymbol{z}(t)$.
- $\boldsymbol{z}(t)$ is a vector of time dependent covariates such as temperature, humidity, and pressure
- exp $(\boldsymbol{\gamma z}(t))$ is an adjusting functional term that considers the key covariates and their weights.

- $\boldsymbol{\gamma}$ is the vector of regression coefficients indicating the degree of influence of each covariate on the hazard function.
- $h_0(t)$ represents the baseline hazard rate function.

For many electrical components such as intermediate relays, the failure rate at time t depends on the degradation state (internal covariate) of the relay at time t and the operating conditions experienced by them such as temperature (external covariate). Moreover, a degraded relay compared to a new one is more vulnerable to a higher temperature, i.e., the coefficient of external covariate depends on the degradation state. Therefore, we integrate the effects of both internal and external covariates at time t by modifying the Equation (5.1) as follows:

$$\lambda_i^r\left(t, s_i(t), z(t)\right) = \lambda_{0,i}(\mathrm{t})\exp\left(\theta_i * (K_i - s_i(t)\right)\exp\left(f\left(K_i - s_i(t)\right) * z(t)\right) \tag{5.2}$$

where $\exp\left(\theta_i * (K_i - s_i(t)\right)$ captures the effects of cumulative damage, which is the difference between the perfect state K_i and the current state $s_i(t)$, on the failure rate, and θ_i is the regression coefficient indicating the degree of influence of the cumulative damage of component i on its failure rate. $\exp\left(f\left(K_i - s_i(t)\right) * z(t)\right)$ captures the effect of operating condition ($z(t)$) on the failure rate at time t, and $f(K_i - s_i(t))$ is non-decreasing function of cumulative damage that shows the influence of operating condition on the failure rate at time t. Here, we assume the baseline hazard function of component i follows an exponential distribution with constant rate $\lambda_{0,i}$. The exponential terms in Equation (5.2) can be replaced by any function depending on the physics of failure.

After modeling the transition rates of multi-state components considering internal and external covariates, we estimate the reliability of MSS at the end of mission. The system reliability can be defined as the probability that it successfully completes the next mission with duration τ. In the series structure, the system fails when any of its components fails. As the components are independent, the reliability of each component ($R_i(t)$) and the reliability of the system ($R_s(t)$) at time t are shown as:

$$R_i(t) = \Pr(s_i(t) \geq 1) = \sum_{k=1}^{K_i} \Pr\,(s_i(t) = k) \tag{5.3}$$

$$R_s(t) = \prod_{i=1}^{n} R_i(t) = \prod_{i=1}^{n} \Pr(s_i(t) \geq 1) = \prod_{i=1}^{n} \sum_{k=1}^{K_i} \Pr\,(s_i(t) = k) \tag{5.4}$$

Since the operating condition is not stable during the mission, it is not possible to find the state probabilities by developing the system of differential equations as in [62]. The state of each component varies stochastically during the mission. Thus, a Monte-Carlo simulation (MCS) algorithm is proposed to estimate the reliability of system working in dynamic operating condition. The MCS method is based on the generation of random numbers repeated many times, and the occurrence number of a specific condition of interest is counted. Given the state of operating condition at the beginning of the mission, we simulate for N_{MCS}^1 times the operating conditions that the system may encounter during the mission. For each simulated condition, the sequence of operating states in the next mission is known, and only the dwell time for each state (time spent in each state) is a random variable, which is exponentially distributed by the mean of $1/\mu_m$. Then, we compute the reliability of system for each deterministic operating condition by dividing the mission time into smaller time intervals Δt. The state of the system and its components at the beginning of the mission are known $S_s(0) = X = [x_1, x_2, \dots, x_N]$. At each time interval, component i stays in its current state s_i with probability $p_i(\Delta t, s_i)$, and goes to lower sate by the probability $1 - p_i(\Delta t, s_i)$. This probability for component i ($i = 1,2, \dots, N$) can be approximated as follows:

$$p_i(\Delta t, s_i) = Pr\{s_i(\mathrm{t} + \Delta\mathrm{t}) = s_i | s_i(\mathrm{t}) = s_i, z(t) = m\,\} \approx e^{-\lambda_i^r(t, s_i, m) * \Delta\mathrm{t}} \tag{5.5}$$

It is assumed that the state of the operating condition does not change in a very small interval Δt. We model the transition of component i at each time interval by generating a random number $U \sim\ Uniform\ (0{,}1)$. If U is greater than p_i, the transition occurs in the small time interval and changes the state of the component i from state s_i to state $s_i - 1$. When the state of each component and/or the state of operating condition changes, it will influence the transition rate of the component. Therefore, we update the transition rates using Equation (5.2) and use them in Equation (5.5) to compute the transition probabilities of each component for the next time interval. The process stops when the time reaches the end of mission or any of the components fail. We run the simulation for N_{MCS}^2 times and calculate the reliability of the system for each operating condition given that the state of the system at the beginning of the next mission is $S_s(0) = X$ as:

$$R_S^u|X = 1 - \frac{number\ of\ times\ system\ failed}{N_{MCS}^2},\ \mathrm{u} = 1, \dots, \mathrm{N}_{MCS}^1 \tag{5.6}$$

Then, the expected system reliability in the next mission is calculated as:

$$\bar{R}_s|X = \frac{\sum_{u=1}^{\mathrm{N}_{MCS}^1}(R_S^u|X)}{\mathrm{N}_{MCS}^1} \tag{5.7}$$

In selective maintenance problems, maintenance actions can be performed to make the system ready for the next mission. During the maintenance break, one of these maintenance actions can be performed on each component: do-nothing ($x_i = y_i$), replacement with new component as corrective or preventive maintenance actions ($x_i = K_i$), and imperfect maintenance actions ($y_i < x_i < K_i$). The cost $c_i(y_i, x_i)$ and time $t_i(y_i, x_i)$ of each maintenance action performed on component i depend on the current state y_i and state after maintenance x_i. Given the state of the system at the beginning of the maintenance break ($\boldsymbol{Y} = (y_1, y_2, \dots, y_N)$), the goal is to find the states of the system at the beginning of the next mission ($\boldsymbol{X} = (x_1, x_2, \dots, x_N)$) in order to maximize the expected system reliability $\overline{(R_S})$ in the next mission considering the maintenance time and budget limitations (T_0, C_0). The integer nonlinear optimization problem to find the best maintenance strategy is formulated as:

$$\begin{cases} Maximize\ \bar{R}_S \\ Subject\ to: \\ T_M(\boldsymbol{X}) = \sum_{i=1}^{N} t_i(y_i, x_i) \leq T_0 \\ C_M(\boldsymbol{X}) = \sum_{i=1}^{N} c_i(y_i, x_i) \leq C_0 \\ y_i \leq x_i \leq K_i \qquad \forall\ i = 1,2, \dots N \\ x_i\ is\ integer \end{cases} \tag{5.8}$$

where T_M(X) and C_M(X) are the total maintenance time and cost, respectively. Solving the above model is time-consuming, especially for large systems, due to the difficulty of estimating the reliability of the system. Therefore, we use a two-phase solution approach. First, we generate all feasible patterns of the components, feasible combination of components, and related maintenance actions to be performed. Then, the evolutionary algorithm is used to find the best maintenance strategy that maximizes the system reliability.

5.3. Numerical Example and Results

To illustrate the practical value of the proposed approach, we apply the present model to find the optimal selective maintenance strategy for a series system composed of two components ($i = 1,2$). Each component has four states ($s_i = 0,1,2,3$) with corresponding performance rates ($g_i = 0,1,2,3$). The basic transition rate $\lambda_{0,i}$ and the value of θ_i for each component are provided in Table 5.1. The state of the components of the system at the beginning of the maintenance break is $Y = [1,1]$. The demand level for this system is $D = 1$.

In the next mission, the operating condition evolves dynamically according to CTMC by the transition rates given in matrix form as Equation (5.9). The state of environmental condition at the beginning of the operational mission is $z(t) = 0$.

$$Q = \begin{bmatrix} -0.4 & 0.2 & 0.2 \\ 0.2 & -0.3 & 0.1 \\ 0.1 & 0.3 & -0.4 \end{bmatrix} \tag{5.9}$$

In our study, the function f in the Equation (5.2) is defined as Equation (5.10) and the value of β_i for each component is given in Table 5.1.

$$f\big(K_i - s_i(t)\big) = \frac{\beta_i K_i}{s_i(t)} \tag{5.10}$$

Table 5.1: The transition rates of components & PH model parameters

Component	$i = 1$	$i = 2$
Transition rate $\lambda_{0,i}$ (/hours)	0.01	0.02
θ_i	0.05	0.04
β_i	0.5	0.4

The maintenance cost and time for each component are given as follows:

$$C_1 = \begin{bmatrix} 0 & 4.5 & 9 & 14.5 \\ 0 & 0 & 4 & 12 \\ 0 & 0 & 0 & 9.5 \\ 0 & 0 & 0 & 0 \end{bmatrix}, C_2 = \begin{bmatrix} 0 & 3 & 8.5 & 15 \\ 0 & 0 & 6 & 12 \\ 0 & 0 & 0 & 4.5 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$T_1 = \begin{bmatrix} 0 & 0.5 & 1 & 1.5 \\ 0 & 0 & 0.5 & 1 \\ 0 & 0 & 0 & 0.5 \\ 0 & 0 & 0 & 0 \end{bmatrix}, T_2 = \begin{bmatrix} 0 & 0.5 & 1 & 1.5 \\ 0 & 0 & 0.5 & 1 \\ 0 & 0 & 0 & 0.5 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

The goal is to find the best selective maintenance actions considering the maintenance time limit $T_0 = 1$ (in hours) and maintenance budget limit $C_0 = 12$ (in \$1000) to get the maximum expected system reliability in the next mission with duration $\tau = 15$ (hours).

To show the impact of variable operating conditions on the failure rate and the dependence between operating conditions and the degradation state of the components, we consider three different scenarios. In the first scenario, the operating condition is stable in the next mission ($z = 0$). In the second and third scenarios, the operating conditions are variable and the sequence of the operating condition states in the next mission is known as $0 \rightarrow 1 \rightarrow 2 \rightarrow 0$. In the second scenario, the effect of operating conditions on the failure rate of each component is independent of its degradation state and the function f is defined as $f(.) = \beta_i$. However, in the third scenario, the dependence between operating conditions and the degradation state of the components is captured using Equation (5.10).

Using the given input data, the problem is solved using Matlab R2016a. To compute the reliability of the system, we use MCS with sample size $N_{MCS}^{1} = 100$ and $N_{MCS}^{2} = 1000$. The results of three scenarios are shown in Table 5.2 for all nine different possible states of the system at the beginning of the next mission. As can be observed, the estimated system reliability in the first scenario is higher than the second and third ones. It shows that assuming the operating condition is stable during the next mission when it is varying stochastically, leads to overestimation of the system reliability. In addition, comparing the expected system reliability in the scenario 2 with scenario 3 shows the importance of considering the dependence between operating condition and the degradation state of the components.

Table 5.2: The expected system reliability for three scenarios

s_1	s_2	Scenario1	Scenario 2	Scenario3	C_M	T_M
1	1	0.7151	0.6318	0.3219	0	0
1	2	0.8283	0.7693	0.4342	6	0.5
1	3	0.8499	0.8	0.5214	12	1
2	1	0.842	0.7823	0.6846	4	0.5
2	2	0.9773	0.9583	0.8933	10	1
2	3	0.988	0.9758	0.9238	16	1.5
3	1	0.8605	0.8105	0.7945	12	1
3	2	0.9906	0.9797	0.9737	18	1.5
3	3	0.9992	0.998	0.9934	24	2

Based on the maintenance time and budget constraints, only some of the nine possible states are feasible solutions. The system state $\boldsymbol{X} = [2,2]$, performing imperfect maintenance actions on both components, is the optimal solution with reliability of 0.8933. If the decision makers are looking for a solution with higher system reliability, they must increase the budget and/or time limitation during the maintenance break. For example, if the desired system reliability is 0.9, they must increase the maintenance budget limit to 16 unit to get the state $X = [2,3]$ with reliability 0.923.

We assumed that the sequence of operating states in the next mission is known. In some cases, we just know the number of times that the system might encounter different states of operating condition and their order is unknown. For this case, we use our approach considering all the possible sequences to compute the average of system reliability. We consider that the number of visits to states 0, 1, and 2 are two, one, and one, respectively. Therefore, there are four different sequences given the beginning state is $z = 0$ as:

- Sequence 1: $0 \rightarrow 1 \rightarrow 2 \rightarrow 0$,
- Sequence 2: $0 \rightarrow 1 \rightarrow 0 \rightarrow 2$,
- Sequence 3: $0 \rightarrow 2 \rightarrow 0 \rightarrow 1$,
- Sequence 4: $0 \rightarrow 2 \rightarrow 1 \rightarrow 0$.

Table 5.3: The expected system reliability for four sequences

s_1	s_2	Sequences				$\bar{R}_S$
		1	2	3	4	
1	1	0.3219	0.5182	0.5999	0.6289	0.5172
1	2	0.4342	0.6929	0.7432	0.7729	0.6608
1	3	0.5214	0.7241	0.7619	0.7985	0.7014
2	1	0.6846	0.7118	0.7572	0.7754	0.7322
2	2	0.8933	0.9276	0.9418	0.958	0.9301
2	3	0.9238	0.9564	0.9672	0.9754	0.9557
3	1	0.7945	0.7527	0.7829	0.8063	0.7841
3	2	0.9737	0.964	0.9712	0.9766	0.9713
3	3	0.9934	0.9941	0.9961	0.9968	0.9951

The system reliability for each sequence and their average are given in Table 5.3. The best maintenance strategy between all feasible solutions is performing imperfect maintenance actions on both components to get $\boldsymbol{X} = [2, 2]$ with reliability 0.9301. The system reliability has different values for each sequence. If the operating conditions are controllable, we can find the best sequence to increase the system reliability. In this case, the order of operating conditions could be another decision variable.

5.4. Summary

This chapter considered the selective maintenance problem for multi-component series systems with multi-state components functioning in multi-state operating conditions. The effects of operating condition and cumulative damage on the transition or failure rates were discussed. We presented a selective maintenance optimization model to capture the mean of system reliability as an objective function subjected to time and cost of maintenance actions. MCS was used to compute the expected reliability of the system in the next operating mission.

The numerical study showed that the proposed approach can effectively capture the effects of dynamic operating conditions to get a reasonable estimation of system reliability and find the best maintenance strategy. Indeed, the results showed that neglecting the effect of dynamic operating conditions leads to overestimating the system reliability. Note that the problem discussed in this chapter can be considered a special case of a phased mission system (PMS) with two random variables, the sequence of the phases and the duration of phases.

6. USING LSTM NEURAL NETWORK TO PREDICT REMAINING USEFUL LIFE OF ELECTROLYTIC CAPACITORS IN DYNAMIC OPERATING CONDITIONS [5]

A critical aspect of prognostics and health management is predicting the remaining useful life (RUL). The existing RUL prediction techniques for aluminum electrolytic capacitors mostly assume the operating conditions remain constant for the entire prediction timeline. In practice, the electrolytic capacitors experience large variations in operating conditions during their lifetime that influence their degradation process and RUL. This chapter proposes a method based on deep learning for RUL prediction. The proposed framework uses the original condition monitoring and operating condition data without the necessity of assuming any particular type of degradation process and, therefore, avoiding the requirement of establishing a link between model parameters and operating conditions. The proposed framework first identifies the degrading point and then develops the Long Short-Term Memory (LSTM) model to predict the RUL of capacitors. The LSTM-based method can reduce computational time and complexity while ensuring high prediction performance. Its effectiveness is demonstrated by utilizing the simulated degradation process and temperature condition time series of aluminum electrolytic capacitors used in electric vehicle powertrains [93].

[5] The present chapter is based on the following paper:
A. F. Shahraki, S. Al-Dahidi, A. R. Taleqani, and O. P. Yadav, "Using LSTM neural network to predict remaining useful life of electrolytic capacitors in dynamic operating conditions,"published in Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability, 2022.
Contribution of Ameneh Forouzandeh Shahraki: developing the mathematical models, analysis of the case study, discussion of the results, and drafting the paper. Contribution of Sameer Al-Dahidi, Ali RahimTaleqani, and Om Yadav: analysis and verification of the results and proofreading the draft paper.

6.1. Introduction

The prognostics and health management (PHM) discipline focuses mainly on predicting the remaining useful life (RUL) of a system based on condition monitoring data or degradation signals [12][69]. The reliable and accurate prediction of RUL helps formulate the best preventive maintenance strategy to increase system reliability and safety, and to avoid sudden system shutdowns [70][4]. In general, RUL prediction can be implemented using model-based and/or data-driven approaches [14][71][72]. The model-based approaches involve explicit mathematical functions, established empirical, or analytical models to capture the degradation and failure behavior of the system. On the other hand, data-driven approaches rely on historical data when it is impossible to apply the domain knowledge of system failure behavior, or the system complexity is very high Data-driven approaches are relatively easy to be generalized and have recently become more attractive with advances in sensor technologies and data analyses methods.

More recently, the use of power electronic modules in various critical engineering systems has grown significantly [94][95]. This has resulted in increased focus on the design and development of more reliable power electronic devices. At the same time, research on reliability analysis of power electronic systems and RUL prediction of their components has gained momentum to support these efforts and ensure safe and reliable applications. Among the critical components of power electronic system, the aluminum electrolytic capacitor is responsible for almost 30% of the total failure incidents [94][96]. It is, therefore, vital to develop an effective approach for predicting RUL of capacitors to prevent any catastrophic failures of critical engineering systems.

The existing literature on RUL prediction of electrolytic capacitors often relies on building degradation models using observed signals and subsequently estimating RUL given a predefined failure threshold. For example, Celaya et al. [97] developed a RUL prediction approach of electrolytic capacitors using Kalman filter. The authors applied a nonlinear least-squares regression algorithm to estimate the parameters of an exponential degradation model [98]. Later, Qin et al. [99] proposed an adaptive and robust prediction method to estimate the health state of electrolytic capacitors and predict RUL using a combination of Verhulst and exponential models. The unscented Kalman filter was applied to generate proposal distribution of the particle filter to track degradation path. In general, these studies have three critical limitations. First, most have considered a simple exponential model to capture the degradation process limiting their use in modeling the degradation phenomena of complex systems [99]. Another cause of concern is the use of experimental data from controlled laboratory conditions to estimate the model parameters. These data may differ significantly from data of actual field operating conditions because of ignoring different sources of uncertainty. Finally, the RUL prediction performance is impacted by assuming constant operating conditions during the lifetime of the electrolytic capacitors and/or difficulties in developing specific links between the degradation model parameters and dynamic operating conditions [84], [88], [100]–[105]. In reality, the operating conditions such as ambient temperature, humidity, or operating profiles vary significantly during the usage time and therefore, affect the life of the system. For example, the degradation rate of electrolytic capacitors is greater at higher temperatures leading to shorter lifespans compared to electrolytic capacitors at lower temperatures [96][106]. On the other hand, the equivalent series resistance (ESR) of the capacitor increases during the aging process resulting in rising core temperatures and causing further acceleration of the aging process. These interactions among the capacitor degradation process, core temperature, and the operating conditions cannot be easily captured through mathematical modeling.

This chapter proposes a framework to predict the RUL of the aluminum electrolytic capacitors considering dynamic operating conditions and degradation signals. The proposed RUL prediction model adopts methods that can capture the complex relationships between inputs and outputs, as well as the sequential dependencies in time series data. To process the sequential data, recurrent neural networks (RNN) have been used recently in RUL prediction approaches [107], [108]. RNNs preserve, learn, and record historical information in sequence data through the periodic connection of the hidden layer nodes. However, one major drawback of standard RNNs is that error gradients vanish as they back propagate through multiple time steps making the learning of long-term relationships nearly impossible. The LSTM network, as a variant of RNN, is designed to handle these long-term dependencies more efficiently than standard RNNs [109]. The gating mechanism in LSTM regulates the flow of information through time and enables the network learning to keep, remove, and update information. LSTM cell ensures the cell states keep both long- and short-term memories and will not be directly renewed as RNN does.

There have been some efforts in the past using LSTM for RUL [110] [111]. For example, Wu et al. [111] used the LSTM network to predict RUL of four types of problems of aircraft turbofan engines. They found that the LSTM network outperformed other benchmark models such as multilayer perceptron, standard RNN, and gated recurrent unit (GRU). Ma et al. [112] presented a grid LSTM model to study the long-term and short-term fuel cell aging experiments. Liu et al. [113] used the LSTM model for predicting the RUL of a PEMFC system. Huang et al. [114] proposed a bidirectional LSTM based framework for RUL prediction of engines under multiple operating conditions. Despite the high accuracy of LSTM network for RUL prediction, existing prediction models mostly assume that the system degradation starts from the moment operations are initiated, and therefore, RUL decreases linearly with time from the beginning of system's life.

However, in practice, there are several systems including an electrolytic capacitor that have degradation free phase. Most of these systems start degrading after a certain usage time, which is called a degrading point (DP). Thus, it is important to detect degrading points and capture the degradation phenomenon while training the model to ensure it reflects the true behavior of the failure process.

To address the identified concerns in the current *RUL* prediction approaches for electrolytic capacitors and overcome the limitation in existing LSTM models in the application of *RUL* prediction, we propose a general LSTM-based framework. Since a particular model for the degradation process and any domain-specific prior knowledge are not necessary, the proposed framework provides a general approach that can be easily adopted for complex systems working in dynamic operating conditions. This framework first develops a method to detect the degrading points of capacitors and then uses the degradation data beyond the identified degrading points to establish the *RUL* prediction model. The incorporation of the degradation free phase improves the *RUL* prediction performance and accelerates the prediction speed by ignoring irrelevant data. The proposed method then uses the original degradation and operating condition data to directly predict the *RUL* of the capacitor at each time step. Hence, it removes the need of a hand-crafted feature extraction procedure, which can be an inexact and time-consuming exercise for complex systems. Thereafter, a weighted *RUL* prediction is calculated to get stable and realistic results. The weighted *RUL* prediction considers information closest to the prediction time as well as information from earlier times to reduce the excessive influence of the dynamic operating conditions. To the best of our knowledge, it is the first LSTM-based framework to predict *RUL* of the aluminum electrolytic capacitors considering dynamic operating conditions. The performance of the proposed approach is examined with respect to evaluation metrics from the literature and computational efforts. The obtained results show the superiority of the proposed framework.

6.2. Aluminum Electrolytic Capacitor Degradation Process

The primary degradation process of aluminum electrolytic capacitors is the vaporization of the electrolyte caused by the change in the core temperature of the capacitor. This causes regions of the capacitor plates to dry out decreasing the effective contact surface area between electrode and electrolyte resulting in an increase of the *ESR* of the capacitor. The continued increase in the *ESR* value indicates the capacitor degradation state. For many applications, capacitor failure is considered to occur when the *ESR* increases by 200% of its initial value [115] [116].

In Zhou et al.[117] and Abdennadher et al.[118], the exponential model is used to capture the aging of capacitor at a constant temperature *T* as:

$$ESR_t(T) = ESR_0(T).\, e^{C(T).t} \tag{6.1}$$

where $ESR_0(T)$ is the initial *ESR* value of a capacitor at T, t is the age of the capacitor, and $C(T)$ describes a temperature-dependent coefficient (degradation rate) of the capacitor. $C(T)$ is expressed as:

$$C(T) = \frac{C_0.e^{-E_{a_2}}}{k.T} \tag{6.2}$$

where C_0 is base degradation rate, E_{a_2} is the activation energy, and k is the Boltzmann constant. $ESR_0(T)$ is related to the capacitor’s geometry and basic material properties and expressed as:

$$ESR_0(T) = \frac{d}{A_{cap}.\sigma} \tag{6.3}$$

where A_{cap} is the surface area of the electrolytic capacitor, d is the average distance, and σ is the conductivity which is a function of temperature.

Interestingly, the initial increase in *ESR* will cause further increase in the core temperature of the capacitor due to more dissipation power loss. This additional increase in core temperature further accelerates the degradation process, i.e., the capacitors with higher core temperatures will have higher degradation rate. Hence, it is important to consider both the increase in *ESR* and the core temperature to evaluate the performance of a capacitor. In Sun et al. [106], the core temperature of the electrolytic capacitor is considered to continuously rise as a function of time during the aging process. To get the accumulated *ESR* with consideration of the time-variable temperature, the Equation (6.1) is modified as $ESR_t(T(t)) = ESR_0(T). e^{\int_0^t C(T(t)).t}$. The operating temperature is assumed to be constant, and *ESR* is measured at constant temperature ignoring the effect of the measurement temperature. However, besides the degradation phenomena of the electrolytic capacitor, the measured *ESR* value can also change with the operating temperature even at its perfect state. This change can be explained by the increase of the electrolyte conductivity, which leads to the reduction in the *ESR* at higher temperature [119]. Therefore, the measured *ESR* value is not a suitable degradation indicator for a capacitor working in dynamic temperature conditions, especially at high temperatures. In Rigamonti et al.[96], the normalized *ESR* value (ESR_t^{norm}), which is independent of the measurement temperature, is used as a degradation indicator. This indicator shows the relative variation of the measured *ESR* at time t ($ESR_t(T)$) with respect to $ESR_0(T)$ at the same temperature *T* and is defined as:

$$ESR_t^{norm} = \frac{ESR_t(T)}{ESR_0(T)} \tag{6.4}$$

It is assumed that for a new capacitor, the relationship between the measured *ESR* and measurement temperature is known or can be easily obtained by performing a series of controlled laboratory tests. However, this relationship is either unknown, or finding a model that can capture the real relationship is not possible. Moreover, the core temperature of the electrolytic capacitors is also influenced by operating temperature in addition to the power dissipation and thermal resistance of the capacitors [96]. The interactions between the capacitor degradation process, changes in the core temperature, and the operating temperature make *RUL* prediction of the capacitors a challenging issue that we address in the proposed framework.

6.3. Proposed LSTM-Based Framework

Given the original condition monitored sensor data from *N* electrolytic capacitors, the goal is to develop a *RUL* prediction model. Suppose $\boldsymbol{S} = [S_1, S_2, \dots, S_N]^T$represents the multivariate time series data of *N* capacitors. $S_i = [\boldsymbol{s}_{i,1}, \boldsymbol{s}_{i,2}, \dots, \boldsymbol{s}_{i,l(S_i)}]^T$contains the degradation data of capacitor i and the operating conditions experienced by this capacitor during its lifetime, $l(S_i)$ time steps. For a test capacitor, $S_{test} = [\boldsymbol{s}_{test,1}, \boldsymbol{s}_{test,2}, \dots, \boldsymbol{s}_{test,t_p}]^T$ are recorded from the beginning of its life until prediction time t_p. In the *RUL* prediction problem, more information can be obtained from the temporal sequence data compared with the data sampled at a single time step [120]. Therefore, we adopt a time window of size L to use multivariate temporal sequence data and predict the *RUL* at t_p as:

$$\widehat{RUL}_{t_p} = f(\boldsymbol{s}_{test,t_p-L+1}, \boldsymbol{s}_{test,t_p-L+2}, \dots, \boldsymbol{s}_{test,t_p-1}, \boldsymbol{s}_{test,t_p}) \tag{6.5}$$

To solve the described problem and find $f(.)$, we establish a direct relationship between the original condition monitored sensor data and corresponding *RUL* in the proposed LSTM-based framework. As shown in Figure 6.1, the training stage of the framework has two main steps. Since the sensor data before the degrading point indicate no degradation and hence provide little information about the capacitor degradation process, we first identify the degrading points. Given that the capacitors work in dynamic operating conditions, one can expect large sensor data variation making it difficult to detect and establish the degrading points. Therefore, we first set a degradation level (*L_d*) for sensor data. We, then, apply the “m/M” rule [121], if *m* out of *M* consecutive points exceeds L_d, the degrading point is detected. The goal of setting this rule is to overcome the high variability of sensor data. The optimal values for L_d, m, and M are found by a grid search technique to obtain the best prediction performance on the training and validation sets.

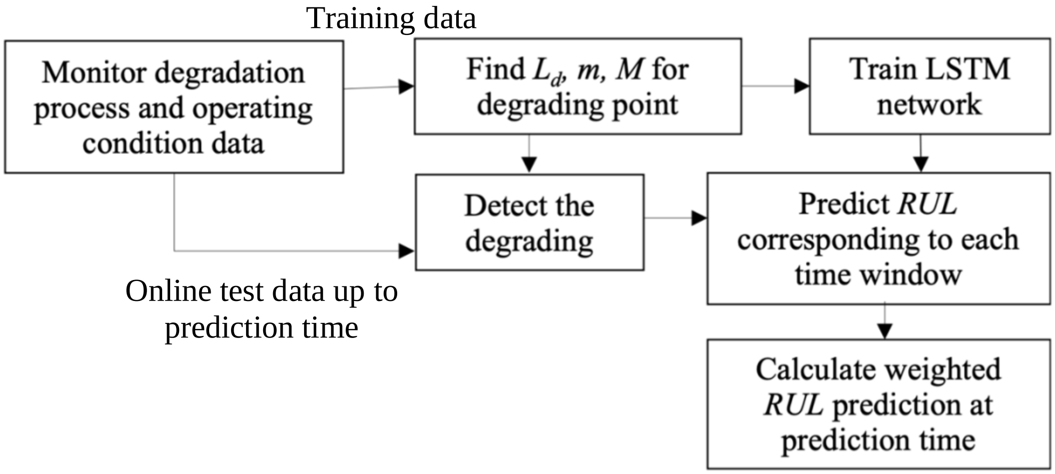

Training data
Monitor degradation process and operating condition data
Find L_d, m, M for degrading point
Train LSTM network
Detect the degrading
Predict RUL corresponding to each time window
Online test data up to prediction time
Calculate weighted RUL prediction at prediction time

Figure 6.1: Proposed framework for RUL prediction of electrolytic capacitors.

Next, the training sensor data beyond the degrading points are fed into the LSTM input layer via a data preparation process, where the sensor data is first normalized using Min-Max scaling such that each data point is within the range of [0,1]. Thereafter, the feature vectors for the training capacitors are generated using the time window of size L, while a label vector is generated for the corresponding RUL of the capacitor. The k^{th} window ($k = 1,2,..,n_i$) of the multivariate time series S_i is $TW_{ik} = [\boldsymbol{s}_{i,k}, \dots, \boldsymbol{s}_{i,(L+(k-1))}]$. The time windows overlap because each window is obtained by sliding the previous window by one time step. We assume that there is no change in RUL of capacitors before the degrading point (RUL_{early}) but after the degrading point, the RUL decreases linearly until it reaches to zero. Therefore, the training samples of the i^{th} capacitor will be $\{(TW_{i1}, RUL_{i1}), (TW_{i2}, RUL_{i2}), \dots, (TW_{ik}, RUL_{ik}), \dots, (TW_{in_i}, RUL_{in_i})\}$, where time window TW_{ik} contains all sensor data within the k^{th} window of the i^{th} capacitor and RUL_{ik} represents the RUL from the last moment of the associated window. The total number of windows (n_i) depends on $l(S_i)$ and L. The proper value of L depends on the characteristics of the degrading capacitor e.g., the average lifetime or data sampling frequency. We investigate historical degradation data to find the appropriate value for L in Section 6.4.2.

After preparing the training data set, we model the relationship between the sensor data of each window and its label using LSTM. The LSTM cell is composed of a cell state, a hidden state, an input gate, a forget gate, and an output gate. The gating mechanism in LSTM regulates the flow of information through time and enables the network learning to keep, remove, and update the information. For more details about the LSTM cell structure, hyper parameters, and model parameters of the network, readers can refer to literature that leveraged LSTM network for RUL prediction[122][111][113][123]. The number of LSTM hidden layers and the number of neurons in each of these layers are two main hyper parameters of the network to control its architecture. We find the best network architecture by tuning its hyper parameters considering the specific properties of the problem and defined evaluation metrics. The optimal hyper parameters will be used to determine the network parameters to minimize the desired loss function. The mean squared error (*MSE*) function is adopted as the network loss function to calculate the average error for N_T training samples as: $MSE = \frac{1}{N_T}\sum_{i=1}^{N_T} {\delta_i}^2$, where $\delta_i = \widehat{RUL}_i - RUL_i$ represents the difference between predicted $\widehat{RUL}_i$ and the true RUL_i. Resilient mean square backpropagation (RMSprop) is used to adaptively minimize the loss function of LSTM. The learning rate of RMSprop optimizer is another hyper parameter of the LSTM. Moreover, dropout technique and early stopping mechanism are used to reduce the risk of network overfitting problem. Dropout ignores randomly selected neurons during training, and hence reduces the sensitivity to the specific weights of individual neurons. The dropout rate (p) is a hyper parameter related to dropout technique, which is defined as the probability of retaining each hidden unit in the hidden layer. In addition, an early stopping mechanism monitors the performance improvement on a validation set. It stops the training process when the validation error begins to rise with some patience (i.e., after some epochs). The details of tuning the hyper parameters of the LSTM model are discussed in Section 6.4.2.

Finally, the proposed framework is used for detecting the degrading point and predicting the RUL of a new test capacitor in real-time. Since RUL prediction results are used for making logistical decisions such as formulating the optimal maintenance strategy or designing a spare parts inventory plan, it is important to have reasonable predictions with minimum variability. Therefore, to reduce the influence of sudden changes in operating conditions, we calculate the weighted RUL prediction considering the RUL of all previous time windows before the prediction time. For this purpose, we estimate the failure time corresponding to each time window as: $\hat{f} = \widehat{RUL} + last\ time\ instant\ of\ the\ window$. The difference between the weighted estimation of failure times and prediction time is considered as the weighted RUL prediction of the test capacitor.

6.4. The Case Study

In this section, the effectiveness of the proposed method is demonstrated by considering a case study properly designed to mimic the degradation behavior of aluminum electrolytic capacitors used in electric vehicle powertrain [96][104] whose operation is characterized by continuously varying temperatures [124]. The performance of the proposed method for providing more realistic RUL predictions is compared with some existing benchmark methods.

6.4.1. Dataset Description

As stated previously, the ESR^{norm} is considered as a degradation indicator of the electrolytic capacitors [96]. The degradation process of 100 electrolytic capacitors is simulated by resorting to a physics-based model of the electrolyte vaporization represented by a first-order Markov process as [96][104][124]:

$$ESR_t^{norm} = ESR_{t-1}^{norm}.e^{C(T_{t-1})} + p_{t-1} \tag{6.6}$$

where ESR_t^{norm} and ESR_{t-1}^{norm} are the normalized ESR value at time t and $t-1$, respectively. $C(T_{t-1})$ is the degradation rate as a function of temperature condition at time $t-1$, and $p_{t-1} \sim Normal(0, 0.02)$ is the corresponding process noise.

During usage time, two types of sensor data, the degradation signal ($ESR^{measured}$) and the temperature conditions (T) experienced by the capacitor, are gathered at a regular time interval. Following the guidelines reported in Al-Dahidi et al. [125], the simulation of ESR^{norm}, $ESR^{measured}$, and T were carried out. ESR^{norm} was simulated by iteratively applying Equation (6.6) (with a time step of 1 hour), assuming an initial value equal to 100 ohms. The capacitor failure time was defined as the time when the ESR^{norm} signal value reaches to a predetermined failure threshold of 200 ohms. $ESR^{measured}$ was simulated by iteratively using Equation (6.7) linking $ESR^{measured}$ to ESR^{norm} as:

$$ESR_t^{measured} = ESR_t^{norm}.(a + b.e^{-\frac{(T_t^{ESR}-273.15)}{c}}) + o_t \tag{6.7}$$

where a, b, and c are the measurement parameters, T_t^{ESR} is the temperature experienced by the capacitor measured at time t, and $o_t \sim Normal(0, 0.002)$ is the measurement noise at time t. The temperature conditions experienced by the capacitor were simulated to show the seasonality pattern that arises in the operating temperature and the effects of aging on the core temperature [125].

Figure 6.2 shows the simulated $ESR^{measured}$ and T of three different capacitors from the normal state to failure. As shown in Figure 6.2, the projected run-to-failure data of these samples are different ranging from 10000 to 35000 hrs. This large variation in failure times is mainly caused by variation in temperature conditions. It can be easily recognized that capacitor 3 (blue color) has shorter life than the other two capacitors since it experiences higher temperature conditions during its usage time causing early and faster degradation. It is also clear that during the initial phase of life, the measured ESR values are small and remain nearly unchanged until a certain usage time. We believe this is the degradation free phase or possibly the degradation initiation phase. Once the degradation process starts, it propagates at a relatively faster rate, as indicated by increasing values of $ESR^{measured}$.

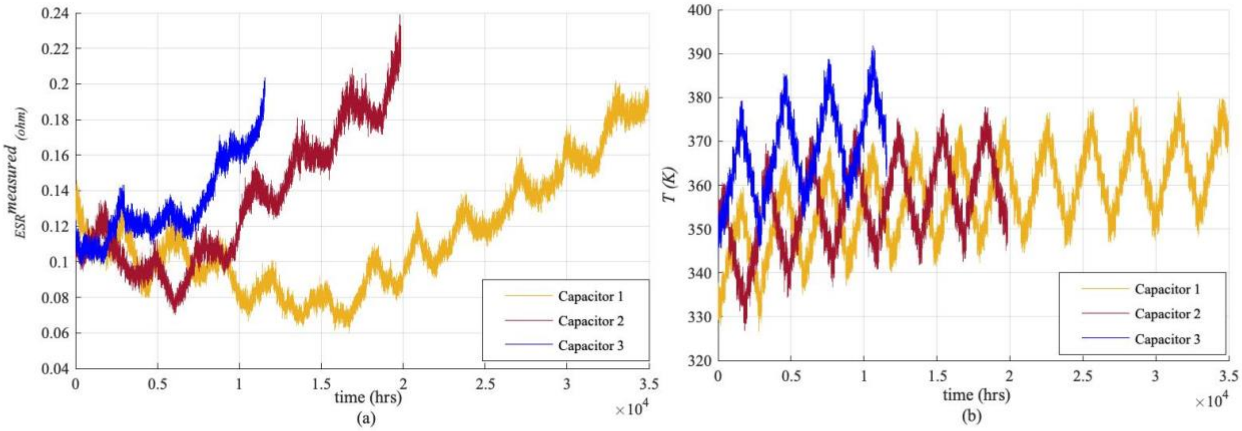


Figure 6.2: (a) *ESR* measurements ($\text{ESR}^{\text{measured}}$), and (b) temperature conditions (T)

6.4.2. Parameter Determination and Results Analysis

To structure the prediction model, we divided the simulated run-to-failure data of 100 electrolytic capacitors into two random parts: the data of 75 capacitors as training set and the data of 25 capacitors as test set. Further, 20% samples of the training set were randomly designated as cross validation set. The training and validation sets were used to build and improve the performance of the *RUL* prediction model while the trained model was used to predict *RUL* of test capacitors. In these sets, the time series data of some capacitors is composed of a large number of samples, for example 100000 samples. Generating deep learning models such as LSTM by direct use of these data as inputs incur too much computational burden. Therefore, the standard down-sampling process is performed to reduce the data samples by representing a number of consecutive discrete time steps by only one longer time step [126]. For this case, the time series of hourly data was sampled down to time series consisting of 50-h time steps. Comparing the training time and prediction results of original and down-sampled data, we concluded that the computational time efforts were significantly reduced without seriously affecting the precision of the results. Note that we use the down-sampled time series data as the input data to all methods in this chapter.

In order to separate out degradation data from initial non-degrading data, we determined the parameters of the method to detect the degrading points. The best combination of parameters was found by grid search on $L_d = (0.1, 0.11, 0.12, 0.13), M = (10,15,20,25), m = M - 3, M - 4, M - 5$. The best parameter combination is $L_d = 0.13, m = 12,$ and $M = 15$. Considering the degrading points, we divided the gathered degradation data into two different groups: data before and after the degrading points (see Figure 6.3 a-c). The data after the degrading points (see Figure 6.3 (b)), which reflects the degradation behavior of the capacitors, is used to model the relation between input sensor data and RUL. Using the approach proposed in Heimes[127], we set the initial RUL before the degrading points as $RUL_{early} = 210$ considering the average length of the training time-series.

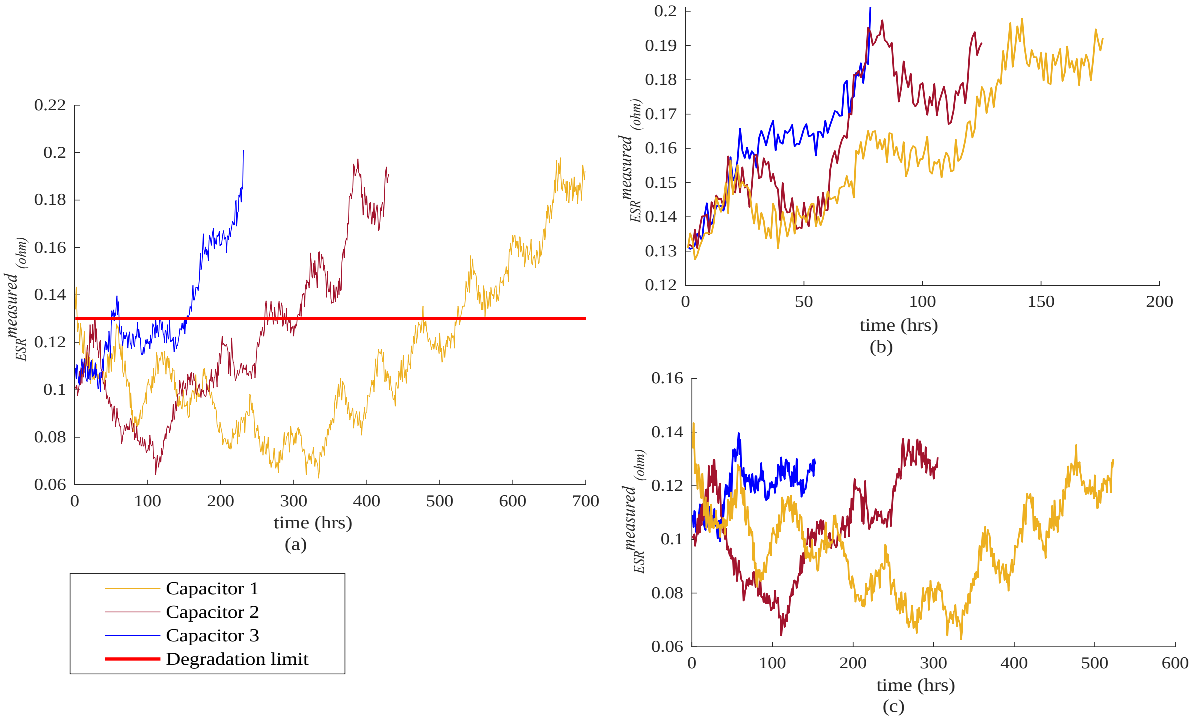


Figure 6.3: (a) ESR$^{\text{measured}}$ for three sample capacitors, (b) ESR$^{\text{measured}}$ after degrading points, (c) ESR$^{\text{measured}}$ before degrading points.

In the second step, we first normalized the condition monitored sensor data using Min-Max scaling as: $\acute{s}_{ijl} = \frac{s_{ijl}-\min(s_l)}{\max(s_l)-\min(s_l)}$, where s_{ijl} and $\acute{s}_{ijl}$ represent for l^{th} sensor ($ESR^{measured}$ or T) the j^{th} data point of the i^{th} training capacitor before and after normalization, respectively. The scaling parameters, $\min(s_l)$ and $\max(s_l)$, represent the minimum and maximum values of the l^{th} sensor across all time points of all training capacitors, respectively. Consequently, the range of data for both $ESR^{measured}$ and T is between 0 and 1 as shown in Figure 6.4 for one sample training capacitor. It should be noted that the obtained scaling parameters are used later to scale the sensor data of test capacitors. Thereafter, the input to the LSTM network was generated using the sliding time window procedure with size $L = 10$. The effects of time window size to prediction performance and model training time will be discussed later in this section.

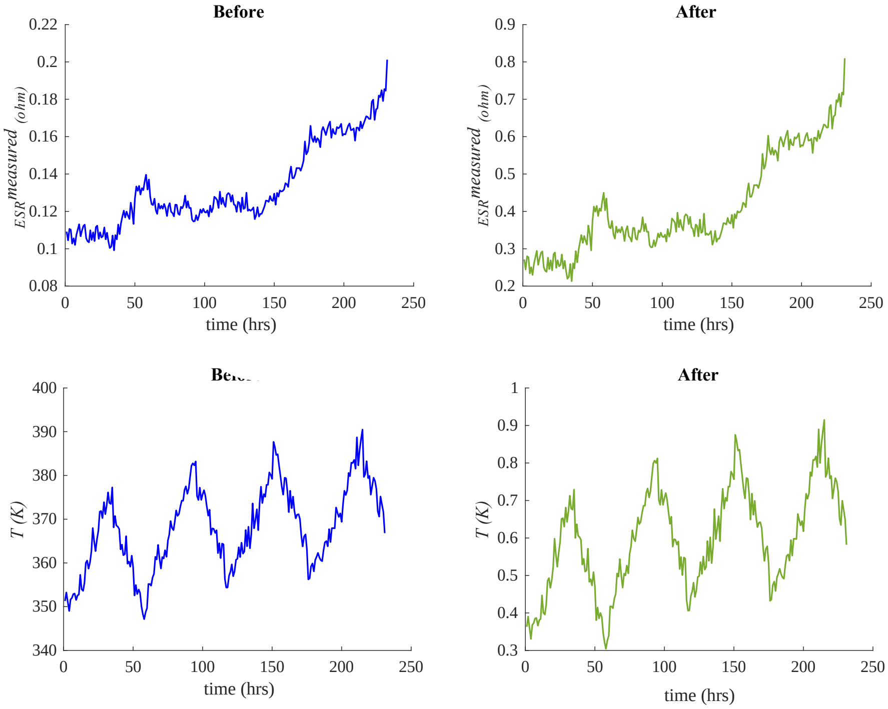


Figure 6.4: $ESR^{measured}$ and T before and after normalization for training capacitor 1

To assess the performance of proposed approach for RUL prediction, the commonly applied evaluation metrics [128][111][123], root mean square error ($RMSE$) and scoring function (SF), were used. Smaller values of these evaluation metrics correspond to smaller errors between true and predicted values of the $RULs$, hence better prediction performance. The metrics are given below:

$$RMSE = \sqrt{\frac{\sum_{k=1}^{M_T} \delta_i^2}{M_T}} \tag{6.8}$$

$$SF = \sum_{i=1}^{M_T} SF_i \, , SF_i = \begin{cases} e^{\frac{-\delta_i}{13}} - 1, if \ \delta_i = \widehat{RUL}_i - RUL_i < 0 \\ e^{\frac{\delta_i}{10}} - 1, if \ \delta_i = \widehat{RUL}_i - RUL_i \geq 0 \end{cases} \tag{6.9}$$

where SF is the total score, SF_i is the score of ith test capacitor, and M_T is the total number of the test capacitors. $RMSE$ equally penalizes underestimates ($\delta_i < 0$) and overestimates ($\delta_i > 0$) of the same magnitude. For electrolytic capacitors, however, $\delta_i < 0$ is preferred over $\delta_i > 0$ to avoid their failures. Therefore, the scoring function was defined to emphasize more on prediction errors when the predicted RUL is larger than the true RUL. The scoring function and the value of its parameters can be modified based on the system requirements. If we consider the prediction results for more than one time step of each test capacitor, we first use Equation (6.8) and Equation (6.9) to calculate the metrics for each test capacitor, and then calculate their average as the final metric for all test capacitors.

We constructed the LSTM network using the python deep learning library, Keras. The model training was executed with one Intel(R) Corei7 CPU 2.6 GHz processor and 16 GB RAM. As the hyper parameter interactions in LSTM are quite small [111], we tuned them independently. In the proposed framework, the effects of three main hyper parameters were investigated: the number of hidden layers of the LSTM network, the number of hidden neurons per hidden layers, and the network optimizer. It should be noted that we used different values of the other hyper parameters such as learning rate and dropout probability. However, it was found that small changes in these parameters did not have a significant effect on the prediction performance. To illustrate the effects of the main hyper parameters, the value of one parameter was varied while others were kept fixed. We first trained the model using RMSprop optimizer with learning rate of 0.001, dropout probability 0.2, batch size 100, and epoch 200 with different number of hidden layers. The number of hidden neurons per hidden layer was optimized via a 10-fold cross validation process. Similarly, the number of layers was set to two when different optimizers were tried. We ran the model for each set of hyper parameters 10 times and selected the best combination of hyper parameters considering the evaluation metrics and training time.

Table 6.1 shows the effect of number of hidden layers on the model training time and evaluation metrics for the validation set. We observed that it takes a longer time to train the model with more hidden layers. Considering all three indicators, LSTM with two hidden layers provides the optimal performance, the lowest *SF* and *RMSE*. Although increasing the network depth can capture more complex patterns, it leads to a larger number of parameters and is more likely to overfit. Figure 6.5 shows the training loss curve with various optimizers: Adagrad, AdaDelta, Adam, and RMSprop. We noticed that AdaDelta and AdaGrad optimizers do not converge during the training process causing their poor performance. On the other hand, the RMSprop and Adam converged quickly before the 25th and 75th epoch, respectively. The prediction performance of RMSprop and Adam optimizers was nearly the same and therefore, we selected the RMSprop as it converged faster, and the network reached steady state in less time. To further reduce the training time and prevent overfitting, we used the early stopping technique by setting patience as 10 epochs.

Table 6.1: Performance with different number of hidden layers

Hidden layer numbers	1	2	3	4	5	6
RMSE	17.43	16.28	17.05	17.72	18.34	18.57
SF	1893	1283	2085	1772	3721	2854
Training time (second)	127.34	175.32	223.88	275.74	329.84	385.31

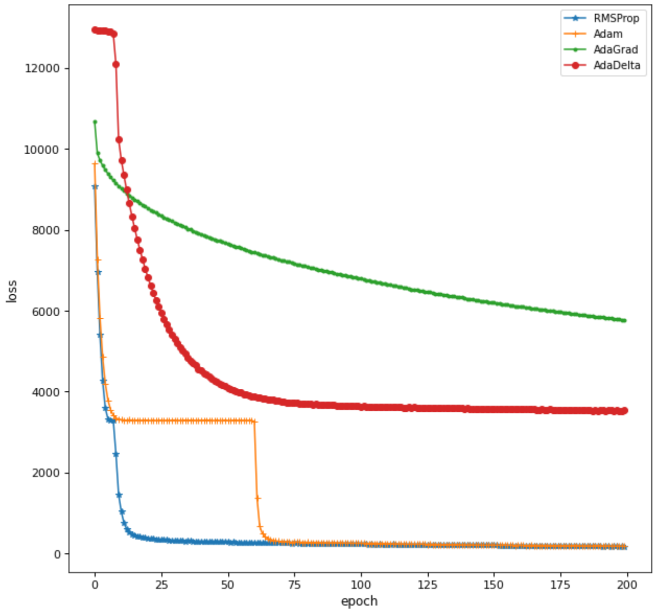


Figure 6.5: Comparison of the training loss evolution with various optimizers

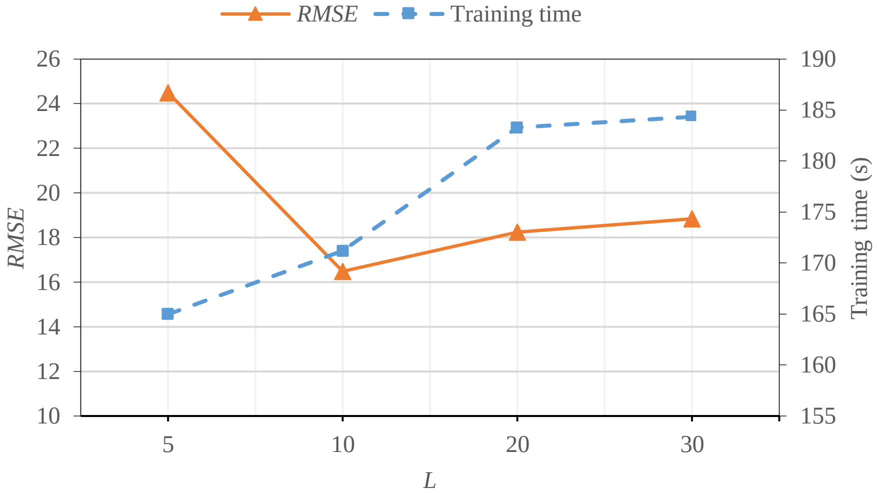


Figure 6.6: Performance with different time window size (L)

Moreover, we investigated how the time window size (L) affects the prediction performance and training time when the number of hidden layers was set to two. As seen in Figure 6.6, the model training time increases as more information is included in one training sample, i.e., a larger time window. It can be observed that there is significant reduction in *RUL* estimation error when L increases from 5 to 10. However, no further improvement in the prediction performance is achieved for $L > 10$. The results of the *SF* were not shown since they presented very similar trends to the results of the *RMSE*. Finally, the optimized LSTM network consisted of two hidden layers each with 50 and 25 neurons was used to build the prediction model. The trained model was then used for the *RUL* prediction of test capacitors. *RUL* prediction results of four sample test capacitors are illustrated in Figure 6.7. The predicted *RUL* in the early period is close to $RUL_{early} = 210$, and afterward, the *RUL* prediction results follow almost a linear trend until the end of the life. Although, there is some variation between the predicted and true *RUL*, we get high-precision *RUL* prediction results.

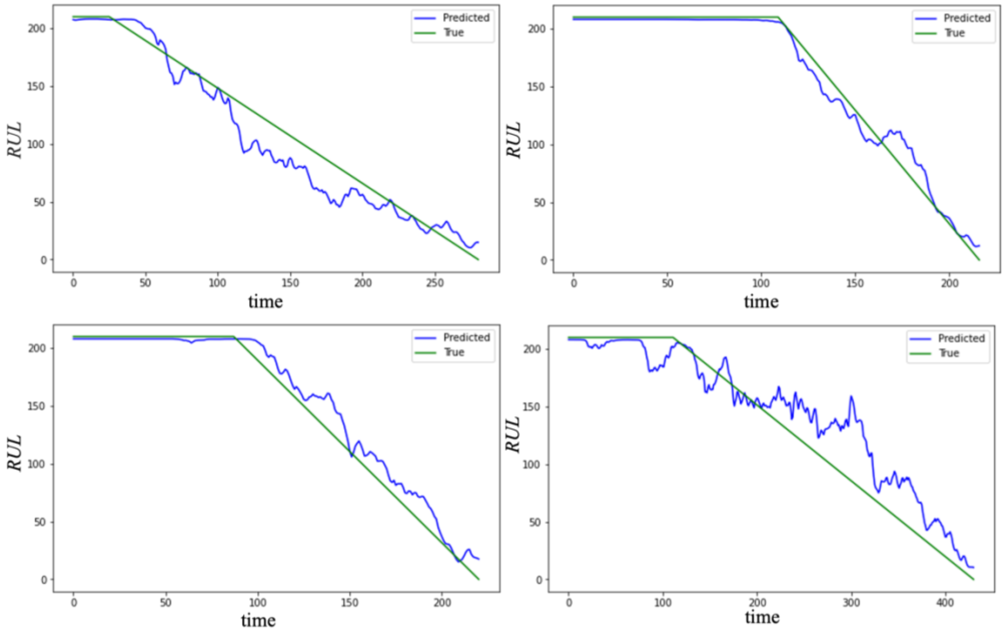


Figure 6.7: Four test capacitors RUL prediction results

To further demonstrate the effectiveness of the proposed framework, we compared the performance of our proposed framework with some existing benchmark methods using the same training and test sets. The first method is the LSTM network that assumes there is no degradation free phase. We considered the same hyper parameters used in our framework. The second method is the Fuzzy Similarity-Based (FSB) approach. In the FSB approach, the similarity between the features of the test capacitor and all capacitors in the training set is evaluated to find the *RUL* of the test capacitor. Since the FSB approach does not require modeling of the degradation process and operating conditions, it is an appropriate candidate for comparison. The details on the FSB approach can be found in Maio and Zio [129]. The third method is a simple artificial neural network (ANN) with one layer consisting of 32 hidden neurons [130]. We also considered other structures of recurrent neural networks such as RNN [127] and GRU [131]. Table 6.2 shows the results of comparison of proposed framework with all other benchmark methods. The comparison was carried out based on *SF* and *RMSE* at different monitoring times with different remaining time steps to failure. All the results were averaged across 10 runs to reduce the effect of randomness.

For the proposed framework, the boxplot of *RMSE* on different monitoring times shows small variations in the *RMSE* between the 10 runs (Figure 6.8).

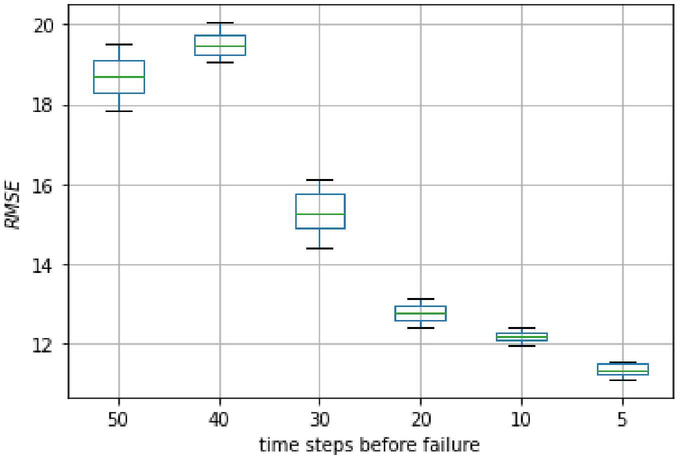


Figure 6.8: Boxplot of RMSE on different monitoring times

Table 6.1: Evaluation metrics of all methods on the test set

Metric	Method	Time steps before failure					
		50	40	30	20	10	5
RMSE	LSTM without DP	21.14	23.82	21.05	16.33	14.23	13.30
	FSB	20.40	18.95	17.19	13.94	13.03	12.15
	ANN	25.36	25.01	19.79	15.01	14.61	11.82
	RNN	18.04	18.80	18.74	17.55	17.37	18.24
	GRU	18.57	20.41	18.06	15.94	14.50	14.92
	Proposed framework	18.69	19.57	15.28	12.78	12.19	11.34
SF	LSTM without DP	468.17	377.90	261.26	116.85	107.97	79.85
	FSB	209.43	180.76	131.66	74.11	77.06	60.93
	ANN	527.38	751.11	199.04	99.75	100.53	57.93
	RNN	332.08	210.27	180.82	132.97	133.37	147.63
	GRU	366.21	303.03	159.18	117.79	91.56	92.80
	Proposed framework	342.09	269.72	105.59	87.56	63.96	55.39

It can be observed that the proposed framework provides better prediction performance during most of the monitoring times. Since most of the methods show a downward trend for the evaluation metrics, we can expect more accurate *RUL* predictions as the capacitors approach the end of life. We believe, one possible reason is that more data is collected and used in the analysis for each capacitor as they approach the failure time. Comparing the performance of the proposed framework with the first method (LSTM without DP) shows the advantage of detecting the degrading points. The consideration of degrading point in proposed framework helps obtain prediction results with better performance in shorter time period. ANN shows higher *RMSE* and *SF* values than the proposed framework and other methods during most of the monitoring times. It indicates that adding depth to the network and considering the dependency between sequence data in the proposed framework are crucial to achieve satisfactory prognostic performance in complex systems. Further, comparing the performance of proposed framework with RNN and GRU, we conclude that LSTM can handle the short- and long-term dependencies within complex structure of this study better than RNN and GRU.

Based on the comparative study, the FSB approach seems to be the second-best method after the proposed framework. We, therefore, further compared the performance of the proposed framework with FSB over the entire lifetime of the test sample data. For this comparative study, we considered additional metrics such as Mean Absolute Error (*MAE*), Weighted Mean Absolute Error (*WMAE*), and Accuracy Index (*AI*) as defined in Al-Dahidi et al.[104]. Smaller values of these metrics indicate better prediction performance. The following equations are used to calculate these metrics:

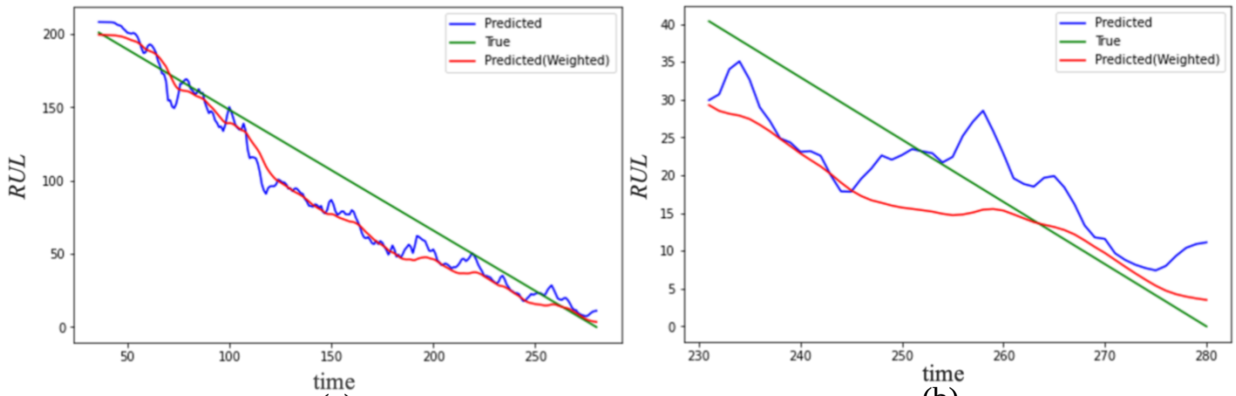
$$\begin{cases} MAE = \frac{\sum_{i=1}^{M_T} |\delta_i|}{M_T} \\ WMAE = \frac{\sum_{i=1}^{M_T} |\delta_i|}{\sum_{i=1}^{M_T} RUL_i} \\ AI = \sum_{i=1}^{M_T - 1} \frac{|\delta_i|}{RUL_i} \end{cases} \tag{6.10}$$

The results of comparative analysis are presented in Table 6.3. Comparing the prediction performance based on these metrics, we concluded that the proposed framework improves the prediction performance and takes significantly less computational time for model training and RUL prediction.

Table 6.3: Evaluation metrics and time comparison on test set

	RMSE	SF	MAE	WMAE	AI	Time (second)
Proposed framework	14.52	884.69	12.01	0.12	0.32	186.05
FSB	29.76	53768.07	23.87	0.25	0.44	4820886.645

Finally, we calculated the weighted RUL predictions considering the results of all time windows before the prediction time. The exponential weighted moving average ($EWMA$) [132] of estimated failure times at time t was computed recursively as $EWMA_t = \alpha \times f_t + (1 - \alpha) \times EWMA_{t-1}$, where f_t is the estimated failure time at time *t* and α is the smoothing factor (a positive number less than 1). To get an appropriate value of smoothing factor α, we used the equation $\alpha = 2/(span + 1)$ and considered $span\ = 20$. It is important to note that we selected $EWMA$ and set the value of *span* for this study based on the prediction performance on the validation set. Based on $EWMA_t$, the weighted RUL prediction was calculated as the difference between the EWMA_t and prediction time *t.* Figure 6.9 shows the weighted *RUL* predictions and the predictions obtained using the last time window for one sample test capacitor. It can be seen that using the information of all time windows up to prediction time leads to more stable and smoother predictions. As shown in Figure 6.10, the weighted results provide slight improvement in the *RMSE* of most test capacitors.



(a) (b)

Figure 6.9: RUL prediction results of one test capacitor from (a) degrading point until failure, (b) last 50 time steps until failure

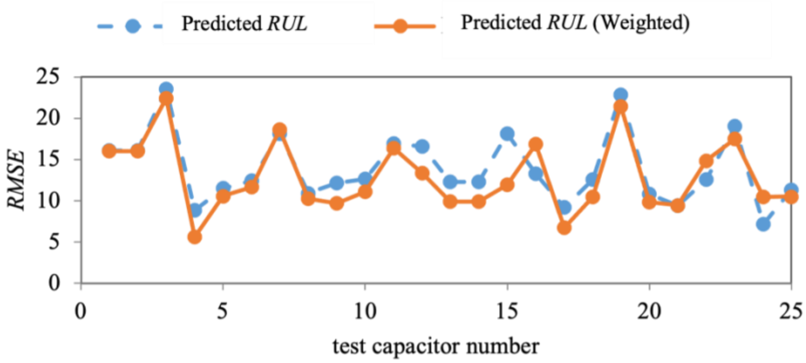


Figure 6.10: $RMSE$ for all test capacitors

6.5. Summary

A novel LSTM-based framework combining degrading point detection and RUL prediction model was proposed to predict the RUL of electrolytic capacitors working in dynamic operating conditions. While the approach was tested on the electrolytic capacitors, it is general enough and can be easily adapted to a variety of complex systems. The proposed framework uses original degradation and operating condition data beyond the degrading points. Therefore, it retains most of the information and does not rely on any specific model for the degradation process and the availability of domain-specific prior knowledge. We also calculated the weighted RUL prediction that considers the information closest to the prediction time and information of earlier times. This helped reduce the excessive influence of the dynamic operating conditions to improve the accuracy of the prediction results for important decisions such as maintenance and spare part inventory planning.

The case study on the degradation of electrolytic capacitors showed that the proposed framework outperforms the existing benchmark methods by achieving prediction results with higher accuracy in a reasonable computational time.

7. EFFICIENT ALGORITHM FOR RELIABILITY EVALUATION OF K-OUT-OF-N PHASED-MISSION SYSTEMS CONSIDERING THE IMPERFECT FAULT COVERAGE

In this chapter, an efficient algorithm is proposed for reliability analysis of the phased-mission systems (PMS) with imperfect fault coverage. The system is composed of several statically independent and non-identical *k*-out-of-*n* subsystems, where each subsystem contains multiple potentially non-identical components. The PMS behavior changes at different phases during the mission as it can have different configurations and success criteria, and experience various operational and environmental conditions in each phase. The proposed method considers both the dynamic of the system in each phase and the statistical dependence of components states across the phases. It also accounts for imperfect fault coverage for the components to get accurate reliability analysis.

7.1. Introduction

A PMS is often required to accomplish multiple non-overlapping phases or tasks of the operation in sequence to accomplish a mission task [133]. Examples of PMS in real-world applications include wireless sensor networks [134], the Mars orbiter mission system [135], body sensor networks [136], the space tracking, telemetry and command system [137], distributed computing system [138], modular multiprocessor system in a space station [139], and aircraft fleets [140]. For instance, an aircraft system needs to undergo various phases such as taxi, take-off, ascent, level flight, descent, and landing phases to successfully accomplish its mission. During each phase, a PMS may be subject to different operational stresses, environmental conditions, and reliability requirements. Thus, the component behavior, the reliability requirement, and the success criteria may vary from phase to phase, leading to complex and dynamic PMS [133]. In addition, there exist statistical dependencies across different phases for each component. Specifically, the state of a component at the end of a phase should be identical to the state at the beginning of the next phase in non-repairable PMS [133], [138], [141]. Because of dependencies across the phases and the system dynamics, the reliability analysis of PMS is a challenging and complex problem.

In the past, considerable efforts have been made to deal with the reliability modeling and assessment of the PMS. Generally, these studies can be divided into two main groups: the simulation and analytical methods [142], [143]. Simulation methods such as Petri-net based methods and Monte Carlo simulation-based methods can deal with systems having complex structures. However, they can be computationally inefficient and only offer approximate reliability results, which are difficult to use for further analysis [133], [137]. In contrast, analytical methods such as combinatorial methods, state-space model, and modular method can often provide accurate results with lower computational cost [144]. Despite considerable research efforts have been dedicated for developing analytical methods [135], [138], [145]–[147], until recently they are mostly limited to small-scale PMS models and are not appropriate for large-scale PMS with complex structure [148]. Recently, Amari et al. [148], [149] proposed an efficient recursive method for exact reliability evaluation of large-scale PMS. Although this method provides a fast and accurate analysis of system reliability, it is only applicable for a system with identical components. Moreover, it was assumed that the failures of any redundant component can be perfectly detected and recovered.

In majority of PMS applications such as flight control and space mission, fault tolerance has been an essential architectural attribute to achieve high reliability [148]. Fault tolerance is generally achieved by using redundancy concepts. For a fault-tolerant system, the mechanism of fault detection, location, isolation, recovery, and reconfiguration play a critical role as a not-covered fault may lead to a system failure despite existence of adequate redundancy. The *k*-out-of-*n:G* structure is the common form of redundancy in which a system consisting of *n* components functions if at least *k* components function properly. For example, an aircraft flight with two engines, in which a minimum of one engine must be functioning in the taxi phase, forms a *1*-out-of-*2* system in this phase. Both series and parallel systems are special cases of *k*-out-of-*n* systems. The *k*-out-of-*n* system redundancy has been extensively studied in the literature. However, most of them only considered single-phase mission. In many practical situations, the system operates in different phases during a mission and the number of required working components may change for each phase. For aircraft flight example, the system has *2*-out-of-*2* redundancy structure in the take-off phase despite having *1*-out-of-*2* redundancy structure in the taxi phase. Moreover, the engines experience different stress levels during take-off phase compared to taxi phase. Therefore, the aircraft flight can be seen as an example of *k*-out-of-*n* PMS.

For fault-tolerant PMS with *k*-out-of-*n* redundancy structure, the efficiency of automatic recovery and reconfiguration mechanisms influences the system reliability [150], [151]. Two-engine aircraft requires at least one engine during the taxi phase. If the first engine fails, the automatic recovery and reconfiguration mechanisms isolate the failed engine and switch in the second one so that the aircraft can operate correctly in this phase. However, if the recovery mechanism could not successfully manage the occurred faults, the aircraft system may fail despite having a spare engine. Thus, to get an accurate reliability analysis of the *k*-out-of-*n* PMS, it is important to consider the probability of successfully recovering from a fault given that the fault has occurred, which is known as the fault coverage factor. For the imperfect fault coverage (IFC) systems that the recovery mechanism cannot successfully manage all faults, the fault coverage factor becomes less than unity [152].

Some researchers studied the reliability of PMS subject to IFC [135], [139], [153]–[155]. Xing [135] and Xing and Dugan [153] proposed a binary decision diagram based algorithm for incorporating IFC into the analysis of PMS. Later Xing et al. [154] improved the computational time and memory requirements of previous works by presenting an efficient recursive formula to compute the overall mission reliability. Although the proposed method has no limitation on the type of failure distributions for the system components, it is only applicable for a system with identical components. Recently, Wang et al. [155] proposed two recursive algorithms based on record values that have less computational complexity compared to Xing et al. [154]. However, these studies are limited to systems composed of identical components and are not computationally efficient for large scale systems.

This chapter proposes an exact and efficient method for the reliability analysis of PMS with several statically independent and non-identical *k*-out-of-*n* subsystems consisting of multiple potentially non-identical components. Since there is no possibility to do any manual interventions during the mission in some critical-mission systems such as space systems, we consider the components of PMS are non-repairable, i.e., the components cannot recover from a failure or any unusable conditions during the mission. However, we consider automatic recovery and reconfiguration mechanisms, which can be subjected to IFC, i.e., all the component failures cannot be perfectly detected and recovered [156]. The efficiency of the proposed method is demonstrated through the analysis of different systems with various scales. Our work makes the following new contributions:

(i) the PMS consists of multiple non-identical subsystems and each subsystem has *k*-out-of-*n* structure

(ii) the components of each subsystem are not necessarily identical and may have different characteristics, particularly, different types of time-to-failure distribution or parameter values, aging rates, and performances.

(iii) the recovery and reconfiguration mechanisms are subjected to IFC

7.2. Description of PMS Model with Imperfect Fault Coverage

A PMS performs a mission consisting of *M* non-overlapping phases that must be accomplished in sequence. The considered system consists of several statistically independent fault-tolerant *k*-out-of-*n* subsystems. Each subsystem has multiple components, which are statistically independent but not necessarily identical. The minimum number of required components for each subsystem may vary from phase to phase. The components have phase-dependent and time-varying failure rates and are not repairable during the mission. If a component enters the failure state, it remains in this state for the rest of the mission time. Although no manual repairs are allowed, the model allows incorporation of automatic recovery and reconfiguration mechanisms. Since the automatic recovery and reconfiguration mechanisms are subject to IFC, the failures are covered with probability *c* (fault coverage factor). Any not-covered failure will cause the mission failure even when the remaining redundancy is still adequate. When a subsystem is not required for a specific phase, it is kept idle in this phase and its components are in a warm standby mode. The failure of idle subsystem components can cause the subsystem and the overall mission failure if imperfect fault coverage happens in this phase. The failure of any required subsystem because of not having required working components or the occurrence of any not-covered failure causes the mission failure.

To analyze the reliability of fault-tolerant PMS, it is important to know its behavior in response to a fault. This behavior is analyzed through a fault handling model, which is also called coverage model [135], [150], [153], as shown in Figure 7.1. The occurrence of any fault can have three possible outcomes: transient restoration (R), permanent coverage (C), and single-point failure (S).

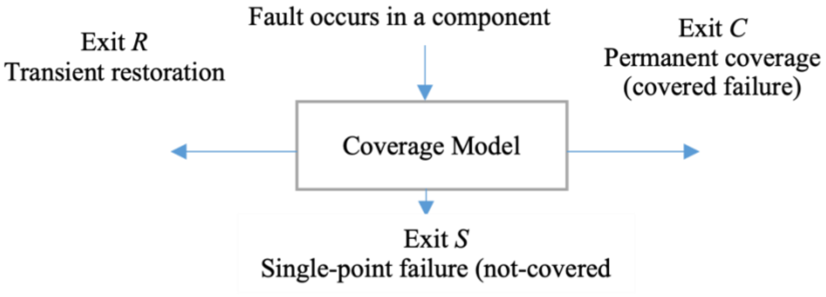


Figure 7.1: The general structure of coverage model

When the fault is transient and can be handled without discarding the component, exit R is taken. If the fault causes system failure, the exit S is taken. On the other hand, exit C is taken if the fault is determined to be permanent and successfully isolated and recovered. The three exits are mutually exclusive and complete. Therefore, the sum of their conditional probabilities given the fault has occurred is $P_R + P_S + P_C = 1$. These probabilities can be determined by solving an appropriate coverage model [157]. As the exit R and C are considered successful actions of fault handling mechanism, the fault coverage factor is: $P_R + P_C$. Since the state of a component is not changed with the events of fault activations that lead to R exit, the model is modified by ignoring R outcome. In this modified model, the failure rate of the component is multiplied by $1/(1 - P_R)$ to get the effective failure rate. Then, the probabilities of two outcomes C and S are modified as: $\acute{P}_C = P_C/(1 - P_R)$ and $\acute{P}_S = P_S/(1 - P_R)$. In this case, any failed component will be detected, isolated, and recovered with the fault coverage factor $c = \acute{P}_C$. Otherwise, it causes the overall system failure. In many cases, the probability of exit R is zero that leads the same results for modified and unmodified models. In the next section, we propose a method to evaluate the reliability of a PMS subject to the IFC that consists of multiple subsystems where each subsystem uses a *k*-out-of-*n* structure.

7.3. Mission Reliability Evaluation

The considered PMS fails if at least one of its required subsystems fails in a phase. The failure of the system in any one of the mission phases, leads to the overall mission failure. Because of the dynamic system structure, the statistical dependency across the phases for a given component, and the existence of not-covered failures, reliability analysis of the considered PMS is a complex and challenging problem. Therefore, we first use the modularization method to reduce the computational complexity [148], [158]. This method is not limited to any specific type of redundancy structure, and different subsystems can use different redundancy structures, which makes it applicable for many problems. By employing the modularization method, the overall mission reliability of the system is calculated as the product of mission reliabilities of individual subsystems as:

$$R_{PMS} = \prod_{s=1}^{N} R_s \tag{7.1}$$

The reliability analysis of a *k*-out-of-*n* subsystem subject to IFC is a complex problem as discussed in Amari et al. [150]. The analysis becomes even more complicated when the subsystem needs to operate in consecutive mission phases with different operating conditions and requirements. To reduce complexity, we consider two mutually exclusive failure modes: covered failure and not-covered failure modes. If at least one component fails in the not-covered state, it causes the failure of the subsystem despite the presence of sufficient redundancies. Alternatively, the subsystem fails in the covered state if the number of working components is less than the minimum required components in any phases, given no component fails in the non-covered state. We use the simple and efficient algorithm (SEA) proposed in Amari et al. [150] for incorporating the IFC. SEA reduces the problem complexity and produces simple closed-form solutions [150], [153]. By considering two mutually exclusive failure modes, the reliability of subsystem s ($\forall \mathrm{s} = 1,2,\dots,\mathrm{N}$) is calculated according to the total probability theorem as:

$$\mathrm{R_s} = \mathrm{Pr}\ (\mathrm{A})\ \times \mathrm{Pr}\ (\mathrm{E_1}) + \mathrm{Pr}\ (\mathrm{D}) \times \mathrm{Pr}\ (\mathrm{E_2}) = 0 \times [1 - \mathrm{P_{sc}}] + \mathrm{R_{sc}} \times \mathrm{P_{sc}} = \mathrm{R_{sc}} \times \mathrm{P_{sc}} \quad (7.2)$$

$\mathrm{A} =$ (susbsysem s functions | at least one *not-covered* failure)

$\mathrm{E_1} =$ (at least one *not-covered* failure)

$\mathrm{D} = (\text{susbsysem s functions}|\text{no } not - covered \text{ failure})$

$\mathrm{E_2} =$ (no *not-covered* failure)

For a k-out-of-n subsystem s, we only need to calculate the conditional reliability of subsystem over the entire mission when there is no not-covered failure (R_{sc}), and the probability that no component of the subsystem *s* experiences not-covered failure (P_{sc}). A recursive algorithm is proposed to calculate R_{sc} in next section. In the following, the way of calculating P_{sc} is explained in detail.

Each component of the subsystem can be in one of $M + 2$ states: covered failure in phase j ($j = 1,2,\dots,M$), not-covered failure in any mission phase, or no failure during the mission (or failure in phase $M + 1$). The probability of each state depends on the condition of each mission phase. To get the probability of each state, we use the cumulative exposure model (CEM) to account for the effects of phase-dependent stress on the failure properties of components [159]. Using the accelerated failure time model (AFTM) to model the life-stress relationship, the stress-dependent failure distribution of component *i* in phase j, denoted by F_{ij}, can be represented as:

$$F_{ij} = F_i(\alpha_{ij}t) \quad \forall i = 1,2,\dots,n\ \ , j = 1,2,\dots,M \quad (7.3)$$

where $F_i(.)$ is the baseline failure distribution function of component *i* and α_{ij} is an acceleration factor of component *i* during phase *j*. According to CEM, the cumulative failure probability of component *i* at the end of phase *j* can be calculated as:

$$Q_{ij} = F_i(\alpha_{i1}\tau_1 + \cdots + \alpha_{ij}\tau_j) \tag{7.4}$$

where τ_j is the duration of phase *j*. As the component *i* is working at the beginning of the mission, thus $Q_{i0} = 0$. In addition, the component *i* either fails during the mission or after the mission (i.e., in phase $M + 1$), so $Q_{i,M+1} = 1$. The probability that component *i* first fails in phase *j* is:

$$f_{ij} = Q_{ij} - Q_{i,j-1} \tag{7.5}$$

If component *i* fails in phase *j*, it will be covered by the probability c_j (fault coverage rate). The computation of c_j depends on the fault handling mechanism [156]. Thus, the probability that failure of component *i* occurs in phase *j* and it is covered, f_{ijc}, can be calculated by Equation (7.6). On the other hand, the probability that failure of component *i* occurs in phase *j* and it is not-covered, f_{iju}, can be calculated by Equation (7.7).

$$f_{ijc} = f_{ij} * c_j \tag{7.6}$$

$$f_{iju} = f_{ij} * (1 - c_j) \tag{7.7}$$

These failures are mutually exclusive. Hence, the probability that component *i* fails in not-covered mode during the mission is:

$$S_{iu} = \sum_{j=1}^{M} f_{iju} \tag{7.8}$$

The obtained probabilities for $M + 2$ states of component i are summarized in Table 7.1. Since the failures of a *k*-out-of-*n* subsystem are independent of each other, the probability that no component experiences a not-covered failure during the mission can be calculated as:

$$P_{sc} = \prod_{i=1}^{n}(1 - S_{iu}) \tag{7.9}$$

Table 7.1: The states and state probabilities of components i

State	1	…	j	….	M	$M+1$	$M+2$
State probability	f_{i1c}	…	f_{ijc}	….	f_{iMc}	$1-(S_{iu}+\sum_{j=1}^{M} f_{ijc})$	S_{iu}

To calculate the R_{sc} in the next section, we need to obtain the conditional failure probabilities of components in each phase. Let g_{ij} be the conditional probability that failure of component *i* occurs in phase *j* given no not-covered failure happens during the mission. The matrix of g_{ij} values, denoted as $\boldsymbol{G}$**,** with the size $n \times (M+1)$ is calculated as:

$$\boldsymbol{G} = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1(M+1)} \\ g_{21} & g_{22} & \cdots & g_{2(M+1)} \\ \vdots & \vdots & \cdots & \vdots \\ g_{n1} & g_{n2} & \cdots & g_{n(M+1)} \end{bmatrix} \begin{cases} g_{ij} = \frac{f_{ijc}}{1-S_{iu}} & \forall\, i = 1,2,\ldots,n, j = 1,2,\ldots,M \\ g_{i(M+1)} = 1 - \frac{\sum_{j=1}^{M} f_{ijc}}{1-S_{iu}} & \forall\, i = 1,2,\ldots,n \\ \sum_{j=1}^{M+1} g_{ij} = 1 \end{cases} \qquad (7.10)$$

7.4. Subsystem Conditional Reliability Analysis

Any required *k*-out-of-*n* subsystem, e.g., subsystem *s*, fails in the covered failure mode if the number of working components is less than the minimum number of required components. Let x_j $(j = 1,2,\ldots,M)$ be the number of components of *k*-out-of-*n* subsystem *s*, that have failed before the completion of phase *j*. If $x_j < m_j = n - k_j + 1$ for all value of *j*, then the subsystem mission is successful. Hence, the subsystem conditional reliability can be evaluated as the sum of the probabilities of all combinations of x_j values as:

$$R_{sc} = Pr\{x_1 < m_1; \ldots; x_j < m_j; \ldots; x_M < m_M\} \qquad (7.11)$$

One approach to evaluate R_{sc} is to get the summation of probabilities of all combinations of x_j values, where $x_j < m_j\ \forall\, j = 1,2,\ldots,M$. Since the number of combinations increases exponentially, this approach is computationally inefficient [148]. Another approach is using the relationship between PMS and generalized multi-state *k*-out-of-*n* system (GMSknS) to get a more efficient solution. In fact, GMSknS can be considered as dual for the PMS model. We propose a recursive method closely related to the method proposed for the reliability evaluation of GMSknS with independent components in Tian et al. [160].

For the k-out-of-n subsystem s, the minimum number of required components for each phase may be different. For the cases that $k_j \leq k_{\acute{j}} \; \forall \, \acute{j} > j$, satisfying the requirements of phase $\acute{j}$ guarantees satisfying the requirements of phase j. Therefore, we can merge these phases to have strictly decreasing k values (or strictly increasing m values). The phase merging helps to improve the computational efficiency of our proposed approach. As an example, consider a subsystem composed of 5 non-identical components that must function in a mission with 7 phases with vector of k values as: $\boldsymbol{k} = [k_1, k_2, k_3, k_4, k_5, k_6, k_7] =$ [1,5,2,2,4,3,1]. The vector of k values can be simplified as: $[\{1{,}5\}, \{2{,}2{,}4\}, 3{,}1] = [5{,}4{,}3{,}1]$ and the number of phases is reduced to 4. The corresponding conditional failure probabilities of merged phases are added together and $\boldsymbol{G}$ will be updated to a new matrix with size 5×5. As another example, the vector of k values $\boldsymbol{k} = [1{,}5{,}2{,}4{,}3{,}4{,}1]$ is simplified as $[\{1{,}5\}, \{2{,}4{,}3{,}4\}, 1] = [5{,}4{,}1]$ after phase merging. The detailed algorithm to merge the phases is presented in Figure 7.2.

After phase merging, we calculate the subsystem conditional reliability $R_{sc}(n, \boldsymbol{m}, \boldsymbol{G})$, using the recursive function as follows:

$$R_{sc}(n, \boldsymbol{m}, \boldsymbol{G}) = \sum_{j=1}^{M+1} g_{nj} \times R_{sc}(n-1, \boldsymbol{m^j}, \boldsymbol{G^n}) \qquad (7.12)$$

$R_{sc}(n, \boldsymbol{m}, \boldsymbol{G})$ is a function of n, $\boldsymbol{m}$, and $\boldsymbol{G}$. n is the number of components of the subsystem s, $\boldsymbol{m} = [m_1, \dots, m_j, \dots, m_M]$ is the vector of $m_j = n - k_j + 1$ values, and $\boldsymbol{G}$ is the matrix of failure probabilities of n components during the mission phases and phase $(M+1)$, which is calculated using the Equation (7.10). Note that the values of $n, \boldsymbol{m}$, and $\boldsymbol{G}$ are different for each subsystem.

```
Inputs: $M, \boldsymbol{k} = [k_1, k_2, \dots, k_M]$, $\boldsymbol{G}$
Outputs: Updated values of $M, \boldsymbol{k}, \boldsymbol{G}$ after merging phases

if $M = 1$ , stop
for $j = M - 1$ down to 1
        if $\boldsymbol{k}(j+1) = 0 \; or \; \boldsymbol{k}(j) <= \boldsymbol{k}(j+1)$
                $\boldsymbol{G}(:, j+1) = \boldsymbol{G}(:, j) + \boldsymbol{G}(:, j+1)$;
                $\boldsymbol{k}(j+1) = \max(\boldsymbol{k}(j), \boldsymbol{k}(j+1))$;
                $\boldsymbol{k}(j) = []$; $\boldsymbol{G}(:, j) = []$
        end if
end for
```

Figure 7.2: Algorithm 1 for merging phases to get strictly decreasing *k* values

First, we consider all states of the nth component: (1) the cases where n^{th} component has failed in any of the mission phases, or (2) stayed in phase $(M+1)$ – survived until the end of the mission. If we know this information about this component, we can express the reliability of a subsystem with n components via evaluating the reliability of $(M+1)$ different simpler subsystems where each subsystem has only $(n-1)$ components. Hence, we need to update $\boldsymbol{m}$ and $\boldsymbol{G}$ to get $\boldsymbol{m^j}$ and $\boldsymbol{G^n}$ for each state of the component *n*. When component-*n* fails in phase j $(j = 1,2,\dots,M)$, it means that it has worked in all phases before phase *j* but failed to support the subsystem in all subsequent phases starting phase *j*. Alternatively, the failure of component *n* in phase $(M+1)$ conceptually means that it has satisfied the mission requirement (failed after completing its mission). Thus, the vector $\boldsymbol{m^j}$ is the same as the vector $\boldsymbol{m}$**,** except that the values of its elements are decreased by one for $j \leq$ indexes $\leq$ M as shown in Equation (7.13). In addition, $\boldsymbol{G^n}$ is obtained by deleting the *nth* row of the matrix $\boldsymbol{G}$ as shown in Equation (7.14).

$$\boldsymbol{m^j} = \left[m_1^j, \dots, m_{(j-1)}^j, m_j^j, \dots, m_M^j\right] = [m_1, \dots, m_{(j-1)}, m_j - 1, \dots, m_M - 1] \qquad (7.13)$$

$$\boldsymbol{G} = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1(M+1)} \\ g_{21} & g_{22} & \cdots & g_{2(M+1)} \\ \vdots & \vdots & \dots & \vdots \\ g_{(n-1)1} & g_{(n-2)2} & \cdots & g_{(n-1)(M+1)} \end{bmatrix} \qquad (7.14)$$

To improve the computational efficiency, we perform phase merging during the recursive method to get strictly decreasing k values (increasing m values). We need to check two special cases when updating $\boldsymbol{m^j}$ and $\boldsymbol{G^n}$. For the first case, if $m_M^j > n$, i.e., the *k* value for the phase *M* is zero, this phase will be absorbed by the adjacent upper phase. Thus, m_M^j is deleted from $\boldsymbol{m^j}$. Also, the values of last two columns of $\boldsymbol{G^n}$ are added together as new value for the last column and the column *M* will be deleted. The next special case happens when any two phases have the same *k* values (or *m* values). We merge these phases to get strictly decreasing *k* values.

To show the tasks of updating the vector of m values and phase merging during the recursive method, we consider a sample subsystem composed of 5 non-identical components with *k* values as $\boldsymbol{k} = [5,4,3,1]$, $\boldsymbol{m} = [1,2,3,5]$. The results for 5th and 4th components are discussed and showed in Table 7.2. We first consider all states of 5th component. This component either has failed in covered mode in phase 2, 3, or 4, or survived until the end of the mission. As shown in Table 7.2, the values of vector $\boldsymbol{m}$ are updated based on the component state and merged to get increasing *m* values. Similarly, all states of 4th component given the state of 5th component are shown in Table 7.2. As we see for all states, it is needed to perform at the maximum only one phase merging.

Beside the special cases for phase merging, the recursive method has two boundary conditions. For the first boundary condition (BC_1), $R_{sc}(n, \boldsymbol{m}, \boldsymbol{G}) = 0$ when $m_M = 0$. It means that the minimum number of working components required in phase M is bigger than the number of components. For the second boundary condition (BC_2), the PMS is reduced to a single-phase system when $M = 1$.

Table 7.2: States of components and *m* values before and after phase merging

States of 5th component: $\boldsymbol{m} = [1,2,3,5]$, $M = 4$, $n = 5$					
j	Phase 1	Phase 2	Phase 3	Phase 4	Phase 5
$\boldsymbol{m}$	NA	$\boldsymbol{m}^{\boldsymbol{2}} = [1,1,2,4]$	$\boldsymbol{m}^{\boldsymbol{3}} = [1,2,2,4]$	$\boldsymbol{m}^{\boldsymbol{4}} = [1,2,3,4]$	$\boldsymbol{m}^{\boldsymbol{5}} = [1,2,3,5]$
merged $\boldsymbol{m}$	NA	$\boldsymbol{m}^{\boldsymbol{2}} = [1,2,4]$	$\boldsymbol{m}^{\boldsymbol{3}} = [1,2,4]$	$\boldsymbol{m}^{\boldsymbol{4}} = [1,2,3,4]$	$\boldsymbol{m}^{\boldsymbol{5}} = [1,2,3]$
States of 4th component given the 5th component has failed in phase 2: $\boldsymbol{m} = [1,2,4]$, $M = 3, n = 4$					
j	Phase 1	Phase 2	Phase 3	Phase 4	
$\boldsymbol{m}$	NA	$\boldsymbol{m}^{\boldsymbol{2}} = [1,1,3]$	$\boldsymbol{m}^{\boldsymbol{3}} = [1,2,3]$	$\boldsymbol{m}^{\boldsymbol{4}} = [1,2,4]$	
merged $\boldsymbol{m}$	NA	$\boldsymbol{m}^{\boldsymbol{2}} = [1,3]$	$\boldsymbol{m}^{\boldsymbol{3}} = [1,2,3]$	$\boldsymbol{m}^{\boldsymbol{4}} = [1,2]$	
States of 4th component given the 5th component has failed in phase 3: $\boldsymbol{m} = [1,2,4]$, $M = 3, n = 4$					
j	Phase 1	Phase 2	Phase 3	Phase 4	
$\boldsymbol{m}$	NA	$\boldsymbol{m}^{\boldsymbol{2}} = [1,1,3]$	$\boldsymbol{m}^{\boldsymbol{3}} = [1,2,3]$	$\boldsymbol{m}^{\boldsymbol{4}} = [1,2,4]$	
merged $\boldsymbol{m}$	NA	$\boldsymbol{m}^{\boldsymbol{2}} = [1,3]$	$\boldsymbol{m}^{\boldsymbol{3}} = [1,2,3]$	$\boldsymbol{m}^{\boldsymbol{4}} = [1,2]$	
States of 4th component given the 5th component has failed in phase 4: $\boldsymbol{m} = [1,2,3,4]$, $M = 4, n = 4$					
j	Phase 1	Phase 2	Phase 3	Phase 4	Phase 5
$\boldsymbol{m}$	NA	$\boldsymbol{m}^{\boldsymbol{2}} = [1,1,2,3]$	$\boldsymbol{m}^{\boldsymbol{3}} = [1,2,2,3]$	$\boldsymbol{m}^{\boldsymbol{4}} = [1,2,3,3]$	$\boldsymbol{m}^{\boldsymbol{5}} = [1,2,3,4]$
merged $\boldsymbol{m}$	NA	$\boldsymbol{m}^{\boldsymbol{2}} = [1,2,3]$	$\boldsymbol{m}^{\boldsymbol{3}} = [1,2,3]$	$\boldsymbol{m}^{\boldsymbol{4}} = [1,2,3]$	$\boldsymbol{m}^{\boldsymbol{5}} = [1,2,3]$
States of 4th component given the 5th component has survived in the mission: $\boldsymbol{m} = [1,2,3]$, $M = 3, n = 4$					
j	Phase 1	Phase 2	Phase 3	Phase 4	
$\boldsymbol{m}$	NA	$\boldsymbol{m}^{\boldsymbol{2}} = [1,1,2]$	$\boldsymbol{m}^{\boldsymbol{3}} = [1,2,2]$	$\boldsymbol{m}^{\boldsymbol{4}} = [1,2,3]$	
merged $\boldsymbol{m}$	NA	$\boldsymbol{m}^{\boldsymbol{2}} = [1,2]$	$\boldsymbol{m}^{\boldsymbol{3}} = [1,2]$	$\boldsymbol{m}^{\boldsymbol{4}} = [1,2,3]$	

There are several algorithms to compute the reliability of single-phase *k*-out-of–*n* system with non-identical components [161]. In this work, we consider two algorithms proposed in [162], [163] and compare their performance. The details of the algorithms are shown in Figure 7.3. While algorithm 2-A is a recursive algorithm, the algorithm 2-B is an iterative algorithm that only saves $k+1$ values of R in one dimensional array P.

Algorithm 2-A

```
Inputs: $n, m, \boldsymbol{G}$
Output: $R = R_{sc}(n, m, \boldsymbol{G})$
    $k = n + 1 - m$
    if $k > n$
            $R = 0$
    else if $k = 0$
            $R = 1$
    else
            $R = \boldsymbol{G}(n,1) * R_{sc}(n-1, m, \boldsymbol{G}) + \boldsymbol{G}(n,2) * R_{sc}(n-1, m-1, \boldsymbol{G})$
    end if
```

Algorithm 2-B

```
Inputs: : $n, m, \boldsymbol{G}$
Output: $R = R_{sc}(n, m, \boldsymbol{G})$
    $k = n + 1 - m$
    P= []
    $P(1) = 1$
    for $j = 2\ to\ k + 1$
            $P(j) = 0$
    end for
    for $i = 1\ to\ n$
            for $j = k + 1\ downto\ 2$
                    $P(j) = \boldsymbol{G}(n,2) * P(j-1) + \boldsymbol{G}(n,1) * P(j)$
            end for
    end for
    $R = P(k+1)$
```

Figure 7.3: Algorithm 2-A and algorithm 2-B for subsystem conditional reliability evaluation when $M = 1$

The detailed description of the recursive algorithm to compute the conditional reliability of subsystem s (R_{sc}) is presented in Figure 7.4.

```
Inputs: $n, \boldsymbol{m}, \boldsymbol{G}$
Output: $R = R_{sc}(n, \boldsymbol{m}, \boldsymbol{G})$
  $M = lenght(\ \boldsymbol{m})$
  if $BC_1$ is satisfied ($m_M = 0$)
        $R = 0$
  else if $BC_2$ is satisfied ($M = 1$)
        Compute $R$ using the algorithm 2A or 2B
  else
        $R = 0$
.       for $j = \ 1\ to\ M + 1$
              # Update the matrix $\boldsymbol{G}$
              $\boldsymbol{G^n} \ = \boldsymbol{G}(1\!:\!n-1,:)$
              # Update the vector $\boldsymbol{m_s}$
              $\boldsymbol{m^j} = \boldsymbol{m}$
              if $j < (M + 1)$
                    for $i = j\ to\ M$
                          $m_{\boldsymbol{i}}^{\boldsymbol{j}} = \boldsymbol{m_i^j} - 1$
                    end for
              end if
              # Merge the phases to get strictly decreasing $k > 0$ values
              if $\ m_M^j > n$
                    Merge phase $M$ and $M + 1$ (algorithm 1)
              else
                    for $\ h = 1\ to\ M - 1$
                          if $m_h^j = m_{h+1}^j$
                                Merge the phases with the same $m$
                                values (algorithm 1)
                          end if
                    end for
              end if
        $R = R + \boldsymbol{G}(n, j) * R_{sc}(n - 1, \boldsymbol{m^j}, \boldsymbol{G^n})$
        end for
  end if
```

Figure 7.4: Algorithm 3 for subsystem conditional reliability evaluation

7.5. Numerical Examples and Results

In this section, two examples are provided to illustrate the application and efficiency of the proposed method. In the examples, the reliability of the different PMS is calculated for system with different numbers of subsystems and components that need to work in a mission with different phases.

7.5.1. Example 1

Consider a PMS with 4 subsystems that must accomplish a mission with 4 phases. The number of components of each subsystem (n), the baseline failure distribution of each subsystem (F), the parameter values of baseline failure distributions are shown in Table 7.3. The baseline failure distribution of the components in each subsystem are different. For subsystem 1, the initial ages of components are $[85.4, 137.3, 186.9, 236.8]$. Except components of subsystem 1, the initial ages of other components are zero. The cumulative distribution function for the Weibull, Gamma, Lognormal, and Exponential are shown in the following:

- $Weibull(\eta, \beta)$: $F(t, \eta, \beta)) = 1 - exp(-(t/\eta)^\wedge\beta)$
- $Gamma(\eta, \beta)$: $F(t, \eta, \beta)) = \text{gammainc}\left(\frac{t}{\eta}, \beta\right)$, gammainc is incomplete gamma function
- $Lognormal(\mu, \sigma)$: $F(t, \mu, \sigma) = ½ + 1/2\text{erf}\ \left(\left(lnt - \mu\right)/\sqrt{2\sigma^2}\right)$, erf (•) is error function
- $Exponential(\eta)$: $F(t, \eta) = 1 - exp(-t/\eta)$

Table 7.3: Parameters of four subsystems

	n	F	Distribution parameters for each component	
			Scale η or μ	Shape β or σ
Subsystem 1	4	$Weibull(\eta, \beta)$	[1100,1200,1300,1400]	1.8
Subsystem 2	5	$Gamma(\eta, \beta)$	[2200,2400,2600,2800,3000]	1.5
Subsystem 3	3	$Lognormal(\mu, \sigma)$	[8.5953, 8.6823, 8.7624]	2.5
Subsystem 4	5	$Exponential(\eta)$	[5500,6000,6500,7000,7500]	NA

The duration (τ), fault coverage factor (c), and the phase-dependent parameters (k and α values) of all subsystems for each phase are given in Table 7.4. Subsystem 2 and subsystem 3 are idle during phases 2 and 3, respectively. As shown in Table 7.4, the k values for these subsystems in phase 2 and phase 3 are zero. The components in subsystem 2 can still fail in phase 2, even though the subsystem is kept idle during this phase because $\alpha = 0.4$.

Table 7.4: Phase-dependent requirements and parameters

		Phase 1	Phase 2	Phase 3	Phase 4
τ		20	50	75	40
c		0.99	0.98	0.97	0.99
phase-dependent parameters					
Subsystem 1	k	3	3	2	1
	α	1	1.2	2	1.5
Subsystem 2	k	3	0 (Idle)	4	2
	α	2	0.4	4	3
Subsystem 3	k	1	2	0 (Idle)	2
	α	1	2	0	0.25
Subsystem 4	k	3	4	3	2
	α	1	4	3	2

Based on the given parameters, we first calculated the conditional failure probabilities of components during the four phases as shown in Table 7.5. Then, for each subsystem, P_{sc} and R_{sc} were calculated using Equation (7.9) and the recursive algorithm 3, respectively. According to the Equation (7.2), the reliability of subsystem s (R_S) is the product of P_{sc} and R_{sc}. The mission reliability of the system was calculated as the product of the reliabilities of individual subsystems as in Equation (7.1), $R_{PMS} = 0.9323$. For comparison, we also calculated the reliability of the subsystems and system considering the perfect coverage, $R_{PMS} = 0.9585$. As shown in Table 7.6, if we consider perfect fault coverage while in fact there is IFC, we get an overestimated values for subsystems and system reliability.

The CPU time for solving this problem using MATLAB 2020 on a personal computer is 0.0079 seconds using algorithm 2-A and 0.0481 seconds using algorithm 2-B. Therefore, we choose algorithm 2-A to compute the reliability when $M = 1$.

Table 7.5: Conditional component failure probabilities at each phase

Subsystem	Component	Phase 1	Phase 2	Phase 3	Phase 4
Subsystem 1	C_1	0.0046	0.0178	0.0665	0.0344
	C_2	0.0055	0.0197	0.0663	0.0328
	C_3	0.0061	0.0206	0.0651	0.0312
	C_4	0.0063	0.0210	0.0635	0.0298
Subsystem 2	C_1	0.0018	0.0015	0.0407	0.0220
	C_2	0.0016	0.0013	0.0360	0.0196
	C_3	0.0014	0.0012	0.0321	0.0176
	C_4	0.0013	0.0010	0.0289	0.0160
	C_5	0.0011	0.0009	0.0262	0.0145
Subsystem 3	C_1	0.0124	0.0503	0	0.0041
	C_2	0.0114	0.0473	0	0.0039
	C_3	0.0104	0.0445	0	0.0037
Subsystem 4	C_1	0.0036	0.0349	0.0374	0.0132
	C_2	0.0033	0.0321	0.0345	0.0122
	C_3	0.0030	0.0297	0.0320	0.0113
	C_4	0.0028	0.0276	0.0298	0.0106
	C_5	0.0026	0.0258	0.0279	0.0099

Table 7.6: Subsystems and system reliability

	Imperfect fault coverage			Perfect fault coverage
	P_{sc}	R_{sc}	R_S	R_S
Subsystem 1	0.9889	0.9940	0.9829	0.9937
Subsystem 2	0.9884	0.9939	0.9823	0.9877
Subsystem 3	0.9966	0.9887	0.9854	0.9884
Subsystem 4	0.9913	0.9886	0.9799	0.9881
	R_{PMS}			R_{PMS}
System	0.9323			0.9585

7.5.2. Example 2

In this example, we consider a larger scale PMS consisted of large number of subsystems that needs to operate in a mission with many phases. Large scale PMS can exist in many applications such as computer networks, computer clusters, and cloud computing systems. For example, the PMS can correspond to a large computer cluster with large number of connected computers that work together to accomplish many sequential tasks with different requirements and operating condition. The parameters of simulated PMS are defined in a way to make it possible for verification and future research comparisons.

The considered PMS has 8 subsystems ($s = 1\ to\ 8$). The *k*-out-of-*n* subsystem *s* has $n_s = 4 + s$ components. The system must accomplish a mission with $M = 10$ phases. The phase-dependent requirements and parameters are given in Table 7.7. We use modulo operator to define the duration of phase j ($j = 1,2,\ldots,10$) as $\tau_j = (1 + mod(j,5)) * 20$. Hence, the total duration of the mission is $T = 600$ time units. The fault coverage factors during phase j are $c_j = 1 - 0.02 \times mod(j,10)$. Using the ceiling function, the k and α values for subsystem *s* during phase *j* are defined as: $k_j = n_s – ceil(0.6 * n_s * j/M) + 1$ and $\alpha_j = 2 * (ceil(0.6 * n_s * 1/M)/ceil(0.6 * n_s * j/M) * (\tau_1/\tau_j)$. As an example, for subsystem 7 $\boldsymbol{k} = [11, 10, 10, 9, 8, 8, 7, 6, 6, 5]$ and $\boldsymbol{\alpha} = [2.0, 0.67, 0.5, 0.27, 1.0, 0.5, 0.27, 0.17, 0.13, 0.57]$.

Table 7.7: Phase-dependent requirements and parameters

	Phase j	
τ	$(1 + mod(j,5)) * 20$	
c	$1 - 0.02 \times mod(j,10)$	
phase-dependent parameters of subsystem *s*		
Subsystem *s*	k	$n_s – ceil(0.6 * n_s * j/M) + 1$
	α	$2 * (ceil(0.6 * n_s * 1/M) * \tau_1/ceil(0.6 * n_s * j/M) * \tau_j)$

For simplicity, we consider that the components of each subsystem can be categorized into two different groups. The parameters of subsystem s are given in Table 7.8. The number of components in group-1 of the subsystem s is: $n_{s,1} = ceil(n_s/3)$ and the remaining components are in group-2. For example, there are 11 components in subsystem 7. Hence, there are $ceil(11/3) = 4$ components in group-1 and $11 - 4 = 7$ components in group-2. The baseline failure time distribution for all components is $Weibull(\eta, \beta)$, where the distribution parameters for components belonging to group-1 of subsystem s are: $\eta_{s,1} = 5 * n_s * T$ and $\beta_{s,1} = 1 + 0.1 * mod(n_s, 10)$. Weibull parameters for group-2 are the same as group-1, except the scale parameter is $\eta_{s,2} = 1.2 * \eta_{s,1}$. For example, in case of subsystem 7, we have: $\beta_{7,1} = \beta_{7,2} = 1.1$, $\eta_{7,1} = 33{,}000$, and $\eta_{7,2} = 1.2 * 33{,}000$. All components are brand new, i.e., their initial age is zero. Table 7.9 shows the reliability of 8 subsystems and system for imperfect and perfect fault coverage cases. The CPU time for solving this problem is 46.58 seconds for imperfect fault coverage case.

Table 7.8: Parameters of subsystem s

	n	F	Distribution parameters for each component	
			Scale η_s	Shape β_s
Group-1	$n_{s,1} = ceil(n_s/3)$	Weibull	$\eta_{s,1} = 5 * n_s * T$	$\beta_{s,1} = 1 + 0.1 * mod(n_s, 10)$
Group-2	$n_{s,2} = n_s - n_{s,1}$	Weibull	$\eta_{s,2} = 1.2 * \eta_{s,1}$	$\beta_{s,2} = \beta_{s,1}$

Table 7.9: Reliability of subsystems and system

	Imperfect fault coverage			Perfect fault coverage
	P_{sc}	R_{sc}	R_S	R_S
Subsystem1	0.9980	0.9917	0.9897	0.9914
Subsystem 2	0.9990	0.9975	0.9965	0.9974
Subsystem 3	0.9994	0.9986	0.9980	0.9985
Subsystem 4	0.9997	0.9992	0.9989	0.9992
Subsystem 5	0.9999	0.9999	0.9998	0.9999
Subsystem 6	0.9941	0.9765	0.9707	0.9760
Subsystem 7	0.9964	0.9873	0.9838	0.9871
Subsystem 8	0.9980	0.9933	0.9914	0.9932
	R_{PMS}			R_{PMS}
System	0.9306			0.9438

7.6. Summary

In this chapter, we presented a recursive method for the reliability analysis of PMS subject to IFC behavior. The proposed method is computationally efficient and applicable for subsystems with different components with any failure distribution. As demonstrated through numerical examples, the proposed mission reliability evaluation algorithm can be used to compute the reliability of large-scale PMS with any structure in reasonable CPU time. The proposed method facilitates determining the optimal number of components for each subsystem to get the best configuration of the system with maximum reliability.

8. CONCLUSION AND FUTURE WORK

This dissertation has presented new predictive analytics methodologies for effective prognostics and health management, and reliability assessment of complex systems with interdependent components that are operating in dynamic operating conditions. The developed frameworks and methods led to improved maintenance planning. The main research results and new contributions of this dissertation are summarized as follows.

New reliability models were proposed, and the optimal maintenance strategy was determined to fulfill the system requirement within the available maintenance resources. In chapter 3, we captured the S-dependency between components of a complex system in a selective maintenance setting and modeled the interactions between components as a function of the system performance rate and the number of influencing components. We also captured the effects of unknown factors, such as the state of non-critical components, on the interaction. The results for two different series systems showed that stochastic imperfect maintenance actions could significantly affect the reliability of system. Also, ignoring S-dependence leads to overestimating the system reliability and improper maintenance actions. Our research contribution provided a useful reference for selective maintenance optimization of multi-component series systems.

In chapter 4, we adopted the instance-based method to directly predict the RUL using the historical degradation process of similar systems. The proposed approach was successfully applied to predict the RUL of simulated systems. The results showed that the approach can be applied for RUL prediction of highly complex systems where it is hard or even impossible to capture true interactions between interdependent components through specific models.

Moreover, we considered three different cases for predictive analytics of a system functioning under dynamic operating conditions. In chapter 5, we modeled the dynamic operating condition as a homogeneous CTMC, and the degradation rate as a function of the age and degradation state of the system as well as the dynamic operating conditions experienced by the system. We accounted for the reality that the influence of the operation condition on the degradation process depends on the current degradation state of the system. Because of the uncertainty of future operating conditions, the MCS algorithm was used to estimate the system reliability. The numerical study showed that the proposed approach can effectively capture the effects of dynamic operating conditions to get a reasonable estimation of system reliability. Moreover, we presented a selective maintenance optimization model to find the optimal maintenance strategy considering the mean of system reliability as the objective function and time and cost of maintenance actions as two constraints of the optimization model.

In chapter 6, to predict the RUL of the complex systems working in dynamic operating conditions, we proposed a novel LSTM-based framework that does not need any specific model for the degradation process, the operating condition, and the effects of the operating conditions on the degradation process. The proposed method used the original degradation and operating condition data to directly predict the RUL of the system in two steps: finding the degrading point and using LSTM to predict the RUL. While the effectiveness of the proposed framework was showed using a simulated data set of electrolytic capacitors, this general framework can be easily adopted to a variety of complex systems.

In chapter 7, we considered PMS that its behavior changes at different phases during the mission as it can have different configurations and success criteria, and experience various operational and environmental conditions in each phase. We proposed an exact and efficient method for the reliability analysis of a non-repairable PMS with several statically independent and different k-out-of-n subsystems consisting of multiple non-identical components. We also considered the IFC, i.e., the component failures cannot be perfectly detected and covered. We demonstrated the effectiveness of this method by computing the reliability of two different systems. The proposed recursive method for the reliability analysis is computationally efficient and applicable for subsystems with different components with any failure distribution.

In the future, one important and interesting research opportunity is to extend the current predictive modeling framework to other practical scenarios. In the following, several future scopes have been discussed briefly.

- Our work in Chapter 3 provides two future directions. (1) To extend the proposed approach to find the optimal maintenance strategy considering all operational missions during the lifetime of the system instead of one mission. (2) To find an efficient method to solve the selective maintenance problem for large serial k-out-of-n systems.
- One extension of the work in Chapter 4 is to investigate the effect of variability of the other parameters of the model, such as the inherent degradation rate of different components, on the performance of the RUL prediction results.
- In chapter 6, LSTM has been considered for the RUL prediction of complex systems. More sophisticated approaches, such as graph neural networks that can capture the spatio-temporal dependency between condition-based sensor data, can be investigated to get more accurate prediction of system RUL. Moreover, the proposed framework provides only point estimation of RUL. However, accurate interval estimations of the RUL are crucial to understand the stochastic nature of degradation processes and perform reliable risk analysis and maintenance decision-making.

- In Chapter 7, we considered PMSs with the following failure dependencies: (1) phase-dependent failure characteristics, (2) dependencies of component states across the phases, and (3) imperfect fault coverage that causes system level failures. However, in some systems, other types of failure dependencies can exist. For example, a failure of a component will result in a higher load on each of the surviving components, thereby inducing a higher failure rate for them. Therefore, another future work is to compute the reliability of PMS considering the dependency between load-sharing components.

REFERENCES

[1] E. Zio, “Prognostics and health management of industrial equipment,” in *Diagnostics and Prognostics of Engineering Systems: Methods and Techniques*, 2012.

[2] “Lac-Mégantic runaway train and derailment investigation summary-Railway Investigation Report-Trnasportation Saftey Board of Canada. Avaliable: http://www.tsb.gc.ca/eng/rapports-reports/rail/2013/r13d0054/r13d0054-r-es.html,” 2014.

[3] “2021 Texas power crisis, Avaliable: https://en.wikipedia.org/wiki/2021_Texas_power_crisis.” .

[4] A. K. S. Jardine, D. Lin, and D. Banjevic, “A review on machinery diagnostics and prognostics implementing condition-based maintenance,” *Mechanical Systems and Signal Processing*. 2006

[5] Z. W. Birnbaum, J. D. Esary, and S. C. Saunders, “Multi-Component Systems and Structures and Their Reliability,” *Technometrics*, 1961.

[6] A. Heng, S. Zhang, A. C. C. Tan, and J. Mathew, “Rotating machinery prognostics: State of the art, challenges and opportunities,” *Mechanical Systems and Signal Processing*. 2009.

[7] R. P. Nicolai and R. Dekker, “Optimal Maintenance of Multi-component Systems: A Review,” in *Springer Series in Reliability Engineering*, 2008.

[8] M. C. A. Olde Keizer, S. D. P. Flapper, and R. H. Teunter, “Condition-based maintenance policies for systems with multiple dependent components: A review,” *European Journal of Operational Research*. 2017.

[9] L. Bian and N. Gebraeel, “Stochastic framework for partially degradation systems with continuous component degradation-rate-interactions,” *Nav. Res. Logist.*, 2014.

[10] L. Bian and N. Gebraeel, “Stochastic modeling and real-time prognostics for multi-component systems with degradation rate interactions,” *IIE Trans. (Institute Ind. Eng.*, 2014.

[11] J. H. Cha and J. Mi, “Study of a Stochastic Failure Model in a Random Environment,” *J. Appl. Probab.*, 2007.

[12] X. S. Si, W. Wang, C. H. Hu, and D. H. Zhou, “Remaining useful life estimation - A review on the statistical data driven approaches,” *European Journal of Operational Research*. 2011.

[13] Z. Zhang, X. Si, C. Hu, and X. Kong, “Degradation modeling-based remaining useful life estimation: A review on approaches for systems with heterogeneity,” *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*. 2015.

[14] A. F. Shahraki, O. P. Yadav, and H. Liao, “A review on degradation modelling and its engineering applications,” *International Journal of Performability Engineering*. 2017.

[15] M. A. Freitas, M. L. G. De Toledo, E. A. Colosimo, and M. C. Pires, “Using degradation data to assess reliability: A case study on train wheel degradation,” *Qual. Reliab. Eng. Int.*, 2009.

[16] H. F. Yu, “Designing an accelerated degradation experiment with a reciprocal Weibull degradation rate,” *J. Stat. Plan. Inference*, 2006.

[17] S. J. Bae and P. H. Kvam, “A nonlinear random-coefficients model for degradation testing,” *Technometrics*, 2004.

[18] S. J. Bae, W. Kuo, and P. H. Kvam, “Degradation models and implied lifetime distributions,” *Reliab. Eng. Syst. Saf.*, 2007.

[19] G. A. Whitmore and F. Schenkelberg, “Modelling Accelerated Degradation Data Using Wiener Diffusion with a Time Scale Transformation,” *Lifetime Data Anal.*, 1997.

[20] A. Elwany and N. Gebraeel, “Real-time estimation of mean remaining life using sensor-based degradation models,” *J. Manuf. Sci. Eng. Trans. ASME*, 2009.

[21] X. Wang, N. Balakrishnan, and B. Guo, “Residual life estimation based on a generalized Wiener degradation process,” *Reliab. Eng. Syst. Saf.*, 2014.

[22] M. Abdel-Hameed, “A Gamma Wear Process,” *IEEE Trans. Reliab.*, 1975.

[23] J. M. van Noortwijk, “A survey of the application of gamma processes in maintenance,” *Reliab. Eng. Syst. Saf.*, 2009.

[24] X. Wang and D. Xu, “An inverse gaussian process model for degradation data,” *Technometrics*, 2010.

[25] H. Qin, S. Zhang, and W. Zhou, “Inverse Gaussian process-based corrosion growth modeling and its application in the reliability analysis for energy pipelines,” *Front. Struct. Civ. Eng.*, 2013.

[26] Z. S. Ye, N. Chen, and Y. Shen, “A new class of Wiener process models for degradation analysis,” *Reliab. Eng. Syst. Saf.*, 2015.

[27] R. Moghaddass and M. J. Zuo, “A parameter estimation method for a condition-monitored device under multi-state deterioration,” *Reliab. Eng. Syst. Saf.*, 2012.

[28] R. Moghaddass, M. J. Zuo, and X. Zhao, “Modeling multi-state equipment degradation with non-homogeneous continuous-time hidden semi-markov process,” in *Diagnostics and Prognostics of Engineering Systems: Methods and Techniques*, 2012.

[29] J. P. Kharoufeh, C. J. Solo, and M. Y. Ulukus, “Semi-Markov models for degradation-based reliability,” *IIE Trans. (Institute Ind. Eng.)*, 2010.

[30] N. Z. Gebraeel and M. A. Lawley, “A neural network degradation model for computing and updating residual life distributions,” *IEEE Trans. Autom. Sci. Eng.*, 2008.

[31] A. Malhi, R. Yan, and R. X. Gao, “Prognosis of defect propagation based on recurrent neural networks,” *IEEE Trans. Instrum. Meas.*, 2011.

[32] O. Fink, E. Zio, and U. Weidmann, “Predicting component reliability and level of degradation with complex-valued neural networks,” *Reliab. Eng. Syst. Saf.*, 2014.

[33] M. S. Kan, A. C. C. Tan, and J. Mathew, “A review on prognostic techniques for non-stationary and non-linear rotating systems,” *Mechanical Systems and Signal Processing*. 2015.

[34] E. Zio and F. Di Maio, “A data-driven fuzzy approach for predicting the remaining useful life in dynamic failure scenarios of a nuclear system,” *Reliab. Eng. Syst. Saf.*, 2010.

[35] Z. Fagang, C. Jin, G. Lei, and L. Xinglin, “Neuro-fuzzy based condition prediction of bearing health,” *JVC/Journal Vib. Control*, 2009.

[36] V. Crk, “Reliability assessment from degradation data,” *Proc. Annu. Reliab. Maintainab. Symp.*, 2000.

[37] P. Wang and D. W. Coit, “Reliability prediction based on degradation modeling for systems with multiple degradation measures,” 2004.

[38] Z. Pan and N. Balakrishnan, “Reliability modeling of degradation of products with multiple performance characteristics based on gamma processes,” *Reliab. Eng. Syst. Saf.*, 2011.

[39] S. Eryilmaz, “Modeling dependence between two multi-state components via copulas,” *IEEE Trans. Reliab.*, 2014.

[40] C. D. Dao and M. J. Zuo, “Selective Maintenance for Multistate Series Systems with S-Dependent Components,” *IEEE Trans. Reliab.*, 2016.

[41] A. F. Shahraki, O. P. Yadav, and C. Vogiatzis, “Selective maintenance optimization for multi-state systems considering stochastically dependent components and stochastic imperfect maintenance actions,” *Reliab. Eng. Syst. Saf.*, 2020.

[42] H. Wang, “A survey of maintenance policies of deteriorating systems,” *Eur. J. Oper. Res.*, 2002.

[43] J. Barata, C. G. Soares, M. Marseguerra, and E. Zio, “Simulation modelling of repairable multi-component deteriorating systems for ‘on condition’ maintenance optimisation,” *Reliab. Eng. Syst. Saf.*, 2002.

[44] D. I. Cho and M. Parlar, “A survey of maintenance models for multi-unit systems,” *European Journal of Operational Research*. 1991.

[45] H. P. Hong, W. Zhou, S. Zhang, and W. Ye, “Optimal condition-based maintenance decisions for systems with dependent stochastic degradation of components,” *Reliab. Eng. Syst. Saf.*, 2014.

[46] N. Rasmekomen and A. K. Parlikad, “Condition-based maintenance of multi-component systems with degradation state-rate interactions,” *Reliab. Eng. Syst. Saf.*, vol. 148, pp. 1–10, 2016.

[47] H. Li, E. Deloux, and L. Dieulle, “A condition-based maintenance policy for multi-component systems with Lévy copulas dependence,” *Reliab. Eng. Syst. Saf.*, 2016.

[48] C. R. Cassady, E. A. Pohl, and W. P. Murdock, “Selective maintenance modeling for industrial systems,” *J. Qual. Maint. Eng.*, 2001.

[49] L. M. Maillart, C. R. Cassady, C. Rainwater, and K. Schneider, “Selective maintenance decision-making over extended planning horizons,” *IEEE Trans. Reliab.*, 2009.

[50] Q. Feng, L. Jiang, and D. W. Coit, “Reliability analysis and condition-based maintenance of systems with dependent degrading components based on thermodynamic physics-of-failure,” *Int. J. Adv. Manuf. Technol.*, 2016.

[51] X. Zhuang and O. P. Yadav, “A new reliability assessment model for power electronic modules,” 2016.

[52] W. F. Rice, C. . Cassady, and J. . Nachlas, “Optimal maintenance plans under limited maintenance time,” *Proc. seventh Ind. Eng. Res. Conf.*, pp. 1–3, 1998.

[53] C. Richard Cassady, W. Paul Murdock, and E. A. Pohl, “Selective maintenance for support equipment involving multiple maintenance actions,” *Eur. J. Oper. Res.*, 2001.

[54] M. Pandey, M. J. Zuo, R. Moghaddass, and M. K. Tiwari, “Selective maintenance for binary systems under imperfect repair,” *Reliab. Eng. Syst. Saf.*, 2013.

[55] C. Chen, M. Q. H. Meng, and M. J. Zuo, “Selective maintenance optimization for multi-state systems,” 1999.

[56] Y. Liu and H. Z. Huang, “Optimal selective maintenance strategy for multi-state systems under imperfect maintenance,” *IEEE Trans. Reliab.*, 2010.

[57] A. Khatab and E. H. Aghezzaf, “Selective maintenance optimization when quality of imperfect maintenance actions are stochastic,” *Reliab. Eng. Syst. Saf.*, 2016.

[58] M. Pandey, M. J. Zuo, and R. Moghaddass, “Selective maintenance modeling for a multistate system with multistate components under imperfect maintenance,” *IIE Trans*, 2013.

[59] C. D. Dao, M. J. Zuo, and M. Pandey, "Selective maintenance for multi-state series-parallel systems under economic dependence," *Reliab. Eng. Syst. Saf.*, 2014.

[60] C. D. Dao and M. J. Zuo, "Selective maintenance of multi-state systems with structural dependence," *Reliab. Eng. Syst. Saf.*, 2017.

[61] G. Maaroufi, A. Chelbi, and N. Rezg, "A selective maintenance policy for multi-component systems with stochastic and economic dependence," *9th Int. Conf. Model. Optim. Simul.*, 2012.

[62] G. L. Anatoly Lisnianski, A. Lisnianski, and G. Levitin, "Multi-state system reliability: assessment, optimization and applications," *Ser. Qual. Reliab. Eng. Stat.*, 2003.

[63] J. H. Holland, *Adaptation in Natural and Artificial Systems*. 2019.

[64] S. Favuzza, M. G. Ippolito, and E. R. Sanseverino, "Crowded comparison operators for constraints handling in NSGA-II for optimal design of the compensation system in electrical distribution networks," *Adv. Eng. Informatics*, 2006.

[65] A. Forouzandeh Shahraki and R. Noorossana, "Reliability-based robust design optimization: A general methodology using genetic algorithm," *Comput. Ind. Eng.*, 2014.

[66] C. Diallo, U. Venkatadri, A. Khatab, and Z. Liu, "Optimal selective maintenance decisions for large serial k-out-of-n: G systems under imperfect maintenance," *Reliab. Eng. Syst. Saf.*, 2018.

[67] A. F. Shahraki, A. Roy, O. P. Yadav, and A. P. S. Rathore, "Predicting remaining useful life based on instance-based learning," 2019.

[68] G. Vachtsevanos, F. Lewis, M. Roemer, A. Hess, and B. Wu, *Intelligent Fault Diagnosis and Prognosis for Engineering Systems*. 2007.

[69] B. Rezaeianjouybari and Y. Shang, “Deep learning for prognostics and health management: State of the art, challenges, and opportunities,” *Meas. J. Int. Meas. Confed.*, 2020.

[70] Y. Lei, N. Li, L. Guo, N. Li, T. Yan, and J. Lin, “Machinery health prognostics: A systematic review from data acquisition to RUL prediction,” *Mechanical Systems and Signal Processing*. 2018.

[71] D. An, N. H. Kim, and J. H. Choi, “Practical options for selecting data-driven or physics-based prognostics algorithms with reviews,” *Reliability Engineering and System Safety*. 2015.

[72] J. Lee, F. Wu, W. Zhao, M. Ghaffari, L. Liao, and D. Siegel, “Prognostics and health management design for rotary machinery systems - Reviews, methodology and applications,” *Mech. Syst. Signal Process.*, 2014.

[73] C. J. Lu and W. O. Meeker, “Using degradation measures to estimate a time-to-failure distribution,” *Technometrics*, 1993.

[74] N. Gebraeel, “Sensory-updated residual life distributions for components with exponential degradation patterns,” *IEEE Trans. Autom. Sci. Eng.*, 2006.

[75] R. Khelif, B. Chebel-Morello, S. Malinowski, E. Laajili, F. Fnaiech, and N. Zerhouni, “Direct Remaining Useful Life Estimation Based on Support Vector Regression,” *IEEE Trans. Ind. Electron.*, 2017.

[76] J. Zhu, N. Chen, and W. Peng, “Estimation of Bearing Remaining Useful Life Based on Multiscale Convolutional Neural Network,” *IEEE Trans. Ind. Electron.*, 2019.

[77] S. Zheng, K. Ristovski, A. Farahat, and C. Gupta, “Long Short-Term Memory Network for Remaining Useful Life estimation,” 2017.

[78] T. Wang, “Trajectory Similarity Based Prediction for Remaining Useful Life Estimation,” *PhD thesis*, 2011.

[79] A. Wan *et al.*, “Prognostics of gas turbine: A condition-based maintenance approach based on multi-environmental time similarity,” *Mech. Syst. Signal Process.*, 2018.

[80] E. Zio, F. Di Maio, and M. Stasi, “A data-driven approach for predicting failure scenarios in nuclear systems,” *Ann. Nucl. Energy*, 2010.

[81] P. Baraldi, F. Di Maio, S. Al-Dahidi, E. Zio, and F. Mangili, “Prediction of industrial equipment Remaining Useful Life by fuzzy similarity and belief function theory,” *Expert Syst. Appl.*, 2017.

[82] L. L. Li, D. J. Ma, and Z. G. Li, “Residual Useful Life Estimation by a Data-Driven Similarity-Based Approach,” *Qual. Reliab. Eng. Int.*, 2017.

[83] A. F. Shahraki and O. P. Yadav, “Selective Maintenance Optimization for Multi-State Systems Operating in Dynamic Environments,” 2018.

[84] L. Bian and N. Gebraeel, “Stochastic methodology for prognostics under continuously varying environmental profiles,” *Stat. Anal. Data Min.*, 2013.

[85] J. Shen and L. Cui, “Reliability performance for dynamic multi-state repairable systems with K regimes,” *IISE Trans.*, 2017.

[86] A. G. Hawkes, L. Cui, and Z. Zheng, “Modeling the evolution of system reliability performance under alternative environments,” *IIE Trans. (Institute Ind. Eng.*, 2011.

[87] B. Çekyay and S. Özekici, “Optimal maintenance of systems with Markovian mission and deterioration,” *Eur. J. Oper. Res.*, 2012.

[88] L. Bian, N. Gebraeel, and J. P. Kharoufeh, “Degradation modeling for real-time estimation of residual lifetimes in dynamic environments,” *IIE Trans.* 2015.

[89] W. Zhu, M. Fouladirad, and C. Bérenguer, “Condition-based maintenance policies for a combined wear and shock deterioration model with covariates,” *Comput. Ind. Eng.*, 2015.

[90] C. D. Dao and M. J. Zuo, “Optimal selective maintenance for multi-state systems in variable loading conditions,” *Reliab. Eng. Syst. Saf.*, 2017.

[91] N. Gorjian, L. Ma, M. Mittinty, P. Yarlagadda, and Y. Sun, “A review on reliability models with covariates,” 2009.

[92] D. W. Wightman and A. Bendell, “The practical application of proportional hazards modelling,” *Reliab. Eng.*, 1986.

[93] A. F. Shahraki, S. Al-Dahidi, A. R. Taleqani, and O. P. Yadav, “Using LSTM neural network to predict remaining useful life of electrolytic capacitors in dynamic operating conditions,” *Proc. Inst. Mech. Eng. Part O J. Risk Reliab.*, 2022.

[94] H. Wang, M. Liserre, and F. Blaabjerg, “Toward reliable power electronics: Challenges, design tools, and opportunities,” *IEEE Ind. Electron. Mag.*, 2013.

[95] J. Falck, C. Felgemacher, A. Rojko, M. Liserre, and P. Zacharias, “Reliability of Power Electronic Systems: An Industry Perspective,” *IEEE Ind. Electron. Mag.*, 2018.

[96] M. Rigamonti, P. Baraldi, E. Zio, D. Astigarraga, and A. Galarza, “Particle Filter-Based Prognostics for an Electrolytic Capacitor Working in Variable Operating Conditions,” *IEEE Trans. Power Electron.*, 2016.

[97] J. R. Celaya, C. S. Kulkarni, G. Biswas, and K. Goebel, “Towards a model-based prognostics methodology for electrolytic capacitors: A case study based on electrical overstress accelerated aging,” *Int. J. Progn. Heal. Manag.*, 2012.

[98] C. S. Kulkarni, G. Biswas, J. R. Celaya, and K. Goebel, “Physics based degradation models for electrolytic capacitor prognostics under thermal overstress conditions,” *Int. J. Progn. Heal. Manag.*, 2013.

[99] Q. Qin, S. Zhao, S. Chen, D. Huang, and J. Liang, “Adaptive and robust prediction for the remaining useful life of electrolytic capacitors,” *Microelectron. Reliab.*, 2018.

[100] J. Kharoufeh, “Explicit results for wear processes in a Markovian environment,” *Oper. Res. Lett.*, vol. 31, no. 3, pp. 237–244, 2003.

[101] X. S. Si, C. H. Hu, X. Kong, and D. H. Zhou, “A residual storage life prediction approach for systems with operation state switches,” *IEEE Trans. Ind. Electron.*, 2014.

[102] J. A. Flory, J. P. Kharoufeh, and N. Z. Gebraeel, “A switching diffusion model for lifetime estimation in randomly varying environments,” *IIE Trans. (Institute Ind. Eng.*, 2014.

[103] T. Liu, Q. Sun, J. Feng, Z. Pan, and Q. Huangpeng, “Residual life estimation under time-varying conditions based on a Wiener process,” *J. Stat. Comput. Simul.*, 2017.

[104] S. Al-Dahidi, F. Di Maio, P. Baraldi, and E. Zio, “Remaining useful life estimation in heterogeneous fleets working under variable operating conditions,” *Reliab. Eng. Syst. Saf.*, 2016.

[105] T. Tao, E. Zio, and W. Zhao, “A novel support vector regression method for online reliability prediction under multi-state varying operating conditions,” *Reliab. Eng. Syst. Saf.*, 2018.

[106] B. Sun, X. Fan, C. Qian, and G. Zhang, “PoF-Simulation-Assisted Reliability Prediction for Electrolytic Capacitor in LED Drivers,” *IEEE Trans. Ind. Electron.*, 2016.

[107] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[108] L. Guo, N. Li, F. Jia, Y. Lei, and J. Lin, “A recurrent neural network based health indicator for remaining useful life prediction of bearings,” *Neurocomputing*, vol. 240, pp. 98–109, 2017.

[109] M. Yuan, Y. Wu, and L. Lin, “Fault diagnosis and remaining useful life estimation of aero engine using LSTM neural network,” *IEEE/CSAA International Conference on Aircraft Utility Systems*, 2016.

[110] Q. Wu, K. Ding, and B. Huang, “Approach for fault prognosis using recurrent neural network,” *J. Intell. Manuf.*, 2020.

[111] Y. Wu, M. Yuan, S. Dong, L. Lin, and Y. Liu, “Remaining useful life estimation of engineered systems using vanilla LSTM neural networks,” *Neurocomputing*, 2018.

[112] R. Ma, T. Yang, E. Breaz, Z. Li, P. Briois, and F. Gao, “Data-driven proton exchange membrane fuel cell degradation predication through deep learning method,” *Appl. Energy*, vol. 231, pp. 102–115, 2018.

[113] J. Liu, Q. Li, W. Chen, Y. Yan, Y. Qiu, and T. Cao, “Remaining useful life prediction of PEMFC based on long short-term memory recurrent neural networks,” *Int. J. Hydrogen Energy*, 2019.

[114] C. G. Huang, H. Z. Huang, and Y. F. Li, “A Bidirectional LSTM Prognostics Method Under Multiple Operational Conditions,” *IEEE Trans. Ind. Electron.*, 2019.

[115] P. Venet, H. Darnand, and G. Grellet, “Detection of faults of filter capacitors in a converter. Application to predictive maintenance,” in *INTELEC,(Proceedings)*, 1993, pp. 229–234.

[116] D. A. Hewitt, J. E. Green, J. N. Davidson, M. P. Foster, and D. A. Stone, “Observation of electrolytic capacitor ageing behaviour for the purpose of prognostics,” in *IECON* 2016.

[117] Y. Zhou, X. Ye, and G. Zhai, “Degradation model and maintenance strategy of the electrolytic capacitors for electronics applications,” in *2011 Prognostics and System Health Management Conference, PHM-Shenzhen 2011*, 2011.

[118] K. Abdennadher, P. Venet, G. Rojat, J. M. Rétif, and C. Rosset, “A real-time predictive-maintenance system of aluminum electrolytic capacitors used in uninterrupted power supplies,” *IEEE Trans. Ind. Appl.*, vol. 46, no. 4, pp. 1644–1652, 2010.

[119] K. W. Lee, M. Kim, J. Yoon, S. Bin Lee, and J. Y. Yoo, “Condition monitoring of DC-link electrolytic capacitors in adjustable-speed drives,” *IEEE Trans. Ind. Appl.*, vol. 44, no. 5, pp. 1606–1613, 2008.

[120] D. Laredo, Z. Chen, O. Schütze, and J. Q. Sun, “A neural network-evolutionary computational framework for remaining useful life estimation of mechanical systems,” *Neural Networks*, 2019.

[121] Z. Hameed, Y. S. Hong, Y. M. Cho, S. H. Ahn, and C. K. Song, “Condition monitoring and fault detection of wind turbines and related algorithms: A review,” *Renewable and Sustainable Energy Reviews*. 2009.

[122] M. Yuan, Y. Wu, and L. Lin, “Fault diagnosis and remaining useful life estimation of aero engine using LSTM neural network,” 2016.

[123] J. Li, X. Li, and D. He, “A Directed Acyclic Graph Network Combined With CNN and LSTM for Remaining Useful Life Prediction,” *IEEE Access*, 2019.

[124] C. Kulkarni, G. Biswas, X. Koutsoukos, J. Celaya, and K. Goebel, “Integrated diagnostic/prognostic experimental setup for capacitor degradation and health monitoring,” in *AUTOTESTCON (Proceedings)*, 2010.

[125] S. Al-Dahidi, F. Di Maio, P. Baraldi, and E. Zio, “A locally adaptive ensemble approach for data-driven prognostics of heterogeneous fleets,” *Proc. Inst. Mech. Eng. Part O J. Risk Reliab.*, vol. 231, no. 4, pp. 350–363, 2017.

[126] M. Hoffmann, L. Kotzur, D. Stolten, and M. Robinius, “A review on time series aggregation methods for energy system models,” *Energies*. 2020.

[127] F. O. Heimes, “Recurrent neural networks for remaining useful life estimation,” in *2008 International Conference on Prognostics and Health Management, PHM* 2008, pp. 1–6.

[128] L. Guo, N. Li, F. Jia, Y. Lei, and J. Lin, “A recurrent neural network based health indicator for remaining useful life prediction of bearings,” *Neurocomputing*, 2017.

[129] Di Maio F and Z. Enrico, “Failure prognostics by a data-driven similarity-based approach,” *Int. J. Reliab. Qual. Saf. Eng.*, vol. 20, no. 01, p. 1, 2013.

[130] A. K. Mahamad, S. Saon, and T. Hiyama, “Predicting remaining useful life of rotating machinery based artificial neural network,” in *Computers and Mathematics with Applications*, 2010, pp. 1078–1087.

[131] J. Chen, H. Jing, Y. Chang, and Q. Liu, “Gated recurrent unit based recurrent neural network for remaining useful life prediction of nonlinear deterioration process,” *Reliab. Eng. Syst. Saf.*, vol. 185, pp. 372–382, 2019.

[132] Pandas, “Pandas Documantation , the pandas development team, 2020, https://pandas.pydata.org/pandasdocs/stable/reference/api/pandas.DataFrame.ewm.html?highlight=ewm#pandas-dataframe-ewm.

[133] A. S. Xing, Liudong, “Reliability of Phased-mission Systems,” in *Handbook of Performability Engineering*, pp. 349–368, 2008.

[134] C. Wang, L. Xing, A. E. Zonouz, V. M. Vokkarane, and Y. L. Sun, "Communication Reliability Analysis of Wireless Sensor Networks Using Phased-Mission Model," *Qual. Reliab. Eng. Int.*, vol. 33, no. 4, 2017.

[135] L. Xing, "Reliability evaluation of phased-mission systems with imperfect fault coverage and common-cause failures," *IEEE Trans. Reliab.*, vol. 56, no. 1, 2007.

[136] Y. Wang, L. Xing, G. Levitin, and N. Huang, "Probabilistic competing failure analysis in phased-mission systems," *Reliab. Eng. Syst. Saf.*, vol. 176, 2018.

[137] X. Yang and X. Wu, "Mission reliability assessment of space TT&C system by discrete event system simulation," *Qual. Reliab. Eng. Int.*, vol. 30, no. 8, 2014.

[138] Y. Dai, G. Levitin, and L. Xing, "Structure Optimization of Nonrepairable Phased Mission Systems," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 44, no. 1, 2013.

[139] J. L. Bricker, "A Unified Method for Analyzing Mission Reliability for Fault Tolerant Computer Systems," *IEEE Trans. Reliab.*, 1973.

[140] D. Yang, H. Wang, Q. Feng, Y. Ren, B. Sun, and Z. Wang, "Fleet-level selective maintenance problem under a phased mission scheme with short breaks: A heuristic sequential game approach," *Comput. Ind. Eng.*, vol. 119, 2018.

[141] R. Peng, Q. Zhai, L. Xing, and J. Yang, "Reliability analysis and optimal structure of series-parallel phased-mission systems subject to fault-level coverage," *IIE Trans. (Institute Ind. Eng.*, vol. 48, no. 8, 2016.

[142] D. Wu, R. Peng, and L. Xing, "Recent advances on reliability of phased mission systems," in *Communications in Computer and Information Science*, vol. 1102, 2019.

[143] L. Xing and S. V. Amari, *Binary Decision Diagrams and Extensions for System Reliability Analysis*. 2015.

[144] D. Wu, R. Peng, and L. Xing, “Recent advances on reliability of phased mission systems,” in *Communications in Computer and Information Science*, 2019.

[145] C. Wang, L. Xing, R. Peng, and Z. Pan, “Competing failure analysis in phased-mission systems with multiple functional dependence groups,” *Reliab. Eng. Syst. Saf.*, vol. 164, 2017.

[146] C. Wang, L. Xing, S. V. Amari, and B. Tang, “Efficient reliability analysis of dynamic k-out-of-n heterogeneous phased-mission systems,” *Reliab. Eng. Syst. Saf.*, vol. 193, 2020.

[147] Y. Mo, L. Xing, and S. V. Amari, “A multiple-valued decision diagram based method for efficient reliability analysis of non-repairable phased-mission systems,” *IEEE Trans. Reliab.*, vol. 63, no. 1, 2014.

[148] S. V. Amari, “A practical method for reliability analysis of phased-mission systems,” 2011.

[149] S. V. Amari, C. Wang, L. Xing, and R. Mohammad, “An efficient phased-mission reliability model considering dynamic k-out-of-n subsystem redundancy,” *IISE Trans.*, vol. 50, no. 10, 2018.

[150] S. V. Amari, J. B. Dugan, and R. B. Misra, “A Separable Method for Incorporating Imperfect Fault-Coverage into Combinatorial Models,” *IEEE Trans. Reliab.*, vol. 48, no. 3, 1999.

[151] S. V. Amari, A. F. Myers, A. Rauzy, and K. S. Trivedi, “Imperfect Coverage Models: Status and Trends,” in *Handbook of Performability Engineering*, 2008.

[152] T. F. Arnold, “The Concept of Coverage and Its Effect on the Reliability Model of a Repairable System,” *IEEE Trans. Comput.*, vol. C–22, no. 3, 1973.

[153] L. Xing and J. B. Dugan, “Analysis of generalized phased-mission system reliability, performance, and sensitivity,” *IEEE Trans. Reliab.*, vol. 51, no. 2, 2002.

[154] L. Xing, S. V. Amari, and C. Wang, “Reliability of k-out-of-n systems with phased-mission requirements and imperfect fault coverage,” *Reliab. Eng. Syst. Saf.*, vol. 103, 2012.

[155] G. Wang, R. Peng, and L. Xing, “Reliability evaluation of unrepairable k-out-of-n: G systems with phased-mission requirements based on record values,” *Reliab. Eng. Syst. Saf.*, vol. 178, 2018.

[156] S. V. Amari, “Optimal design configurations of fault-tolerant systems,” 2013.

[157] J. B. Dugan and S. A. Doyle, “Incorporating imperfect coverage into a BDD solution of a combinatorial model,” *J. Eur. des Syst. Autom.*, vol. 30, no. 8, 1996.

[158] R. Mohammad, A. Kalam, and S. V. Amari, “Reliability evaluation of phased-mission systems with load-sharing components,” 2012.

[159] S. V. Amari and G. Dill, “A new method for reliability analysis of standby systems,” 2009.

[160] Z. Tian, M. Zuo, and R. Yam, “Multi-state k-out-of-n system and their performance evaluation,” *IIE Trans. (Institute Ind. Eng.*, vol. 41, no. 1, 2009.

[161] S. V. Amari, M. J. Zuo, and G. Dill, “O(kn) Algorithms for Analyzing Repairable and Non-repairable k-out-of-n:G Systems,” in *Handbook of Performability Engineering*, 2008.

[162] Y. Dutuit and A. Rauzy, “New insights into the assessment of k-out-of-n and related systems,” *Reliab. Eng. Syst. Saf.*, vol. 72, no. 3, 2001.

[163] A. M. Rushdi, “Utilization of symmetric switching functions in the computation of k-out-of-n system reliability,” *Microelectron. Reliab.*, vol. 26, no. 5, 1986.