

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/50253040>

System-level health assessment of complex engineered processes

Article

Source: OAI

CITATIONS

6

READS

171

1 author:



[Manzar Abbas](#)

Georgia Institute of Technology

8 PUBLICATIONS 140 CITATIONS

SEE PROFILE

SYSTEM LEVEL HEALTH ASSESSMENT OF COMPLEX ENGINEERED PROCESSES

A Thesis
Presented to
The Academic Faculty

by

Manzar Abbas

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
December 2010

SYSTEM LEVEL HEALTH ASSESSMENT OF COMPLEX ENGINEERED PROCESSES

Approved by:

Dr. George J. Vachtsevanos, Advisor
School of Electrical and Computer
Engineering
Georgia Institute of Technology

Dr. Jennifer Michaels
School of Electrical and Computer
Engineering
Georgia Institute of Technology

Dr. Aldo A. Ferri
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Gregory Durgin
School of Electrical and Computer
Engineering
Georgia Institute of Technology

Dr. Thomas Habetler
School of Electrical and Computer
Engineering
Georgia Institute of Technology

Date Approved: 21st October 2010

Humbly dedicated to the heir of Hussain, the 12th Imam, Al-Mahdi

ACKNOWLEDGEMENTS

First and foremost, i would like to acknowledge my Creator who blessed me whatever I have. Then, my thanks go to my beloved Prophet Muhammad and his pure Progeny for being a source to acknowledge the realities of existence, both seen and unseen.

I am thankful to my advisor D. George Vachtsevanos who provided me continuous support and advice. I want to thank the memebers of my PhD committee especially Dr. Jennifer Michaels for taking out time to thoroughly read my dissertation and providing worthy guidelines for improving my work. I also want to acknowledge the support of my colleagues in the Intelligent Control Systems Lab (ICSL). My thanks and acknowledgement goes to the community of Pakistani friends inside and outside of Georgia Tech.

Lastly, I want to thank my family. My father passed away in Pakistan while during the course of my PhD. I want to thank my mother and brothers for continuously providing me with emotional support throughout these years and my wife and daughter for sharing the burdens of my studies.

Contents

DEDICATION	iii
ACKNOWLEDGEMENTS	iv
LIST OF TABLES	ix
LIST OF FIGURES	x
SUMMARY	xii
I INTRODUCTION	1
1.1 Motivation	1
1.2 Problem Statement and Significance	1
1.3 Background of the Problem	4
1.3.1 Prognosis Modeling	4
1.4 Types of Metamodeling Methods	8
1.4.1 Response Surface Models (RSM)	9
1.4.2 Kriging	11
1.4.3 Multivariate Adaptive Regression Splines (MARS)	12
1.4.4 Artificial Neural Network (ANN)	12
1.5 Metamodeling Applications in the System-level Prognostics and Health Management (PHM) Domain	13
1.5.1 Model Approximation	13
1.5.2 Problem Formulation	13
1.5.3 Design-space Exploration	14
1.6 Types of Metamodeling Strategies	17
1.6.1 Comparison between RSM and ANN	17
1.7 Organization of the Thesis	20
II HEALTH ASSESSMENT METHODOLOGY FOR A GENERAL CASE: COMPONENT- LEVEL AND SUBSYSTEM-LEVEL MODELING	21
2.1 Architecture of the Methodology	24

2.1.1	System Hierarchy	24
2.1.2	Block Diagram of the Methodology	24
2.2	Types of Variables in the Architecture	25
2.2.1	Stress Factors	25
2.2.2	Load Variables	27
2.2.3	Subsystem-level Variables	27
2.2.4	System-level Variables	28
2.3	Examples	28
2.3.1	Electromechanical System	28
2.3.2	Vehicle Electrical System	29
2.3.3	Turbomachinery	30
2.4	Types of Models in the Methodology	31
2.4.1	Failure Mechanism Model ($h_1(.)$)	31
2.4.2	Load-stress Model ($h_2(.)$)	38
2.4.3	Subsystem-level Model ($g(.)$)	41
2.4.4	System-level Metamodel ($f(.)$)	42
2.5	Assumptions	43
2.5.1	System-related Assumptions	43
2.5.2	Failure Mechanism Modeling	44
2.5.3	Load-stress Model	44
III	SYSTEM-LEVEL METAMODELING FOR A GENERAL CASE . . .	45
3.1	Defining the input space	46
3.1.1	Parameterization of the system-level variables and subsystems' health parameters	47
3.1.2	Defining the range of parameters	48
3.2	Screening	49
3.3	Strategies for Designing Experiments	51
3.3.1	Classical Experimental Design	53
3.4	Polynomial Regression Modeling Using Method of Least Squares . .	59

3.4.1	Estimation of the parameters in regression models using least squares	59
3.4.2	Orthogonality in Response Surface Models	61
IV	CASE STUDY: COMPONENT-LEVEL AND SUBSYSTEM-LEVEL MODELING	63
4.1	Hierarchical Architecture of a Gas Turbine Engine	63
4.2	Failures and Degradation in Gas Turbine Engines	65
4.3	Failure Mechanism Modeling	68
4.3.1	Damage (creep) Model	69
4.3.2	Damage Accumulation Rule	70
4.3.3	Creep Failure Criteria	71
4.4	Load-stress Model	72
4.5	Subsystem-level Model	73
V	CASE STUDY: SYSTEM-LEVEL METAMODELING OF A GAS TURBINE ENGINE	75
5.1	Simulation Platform	77
5.2	Defining the input space	79
5.2.1	Parametrization of Mission Profiles	80
5.2.2	Defining the Range of Mission Profile Parameters	82
5.2.3	Parametrization of Subsystem Health	83
5.2.4	Defining the Range of Subsystem Health Parameters	85
5.3	Design of Experiments (DOE) for Screening	86
5.3.1	Experimental Design for Mission Profile Parameters Screening	87
5.3.2	Experimental Design for Subsystems' Health Parameters Screening	87
5.4	Building Models for Screening	89
5.4.1	Effects Estimation of the Mission Profile Parameters	89
5.4.2	Effects Estimation of Subsystem Health Parameters	91
5.5	Parameter Screening	92

5.6	Modeling	96
5.6.1	System-level Metamodel for Takeoff Phase	97
5.6.2	System-level Metamodel for Climb Phase	97
5.6.3	System-level Metamodel for Cruise Phase	101
5.6.4	The Overall System-level Metamodel	103
5.7	Model Verification and Performance Comparison	106
5.7.1	Artificial Neural Network (ANN) Models	108
5.8	Summary of Results	109
5.8.1	Data Requirement vs. Number of Variables	109
5.8.2	Model Building Procedure and Model Accuracy: RSM vs. ANN	112
5.8.3	Model Representation	112
VI	CONCLUSIONS AND SUGGESTED FUTURE WORK	113
Appendix A	SCREENING STAGE: DESIGN MATRICES AND RESPONSE VALUES	116
Appendix B	CLIMB PHASE METAMODELING: DESIGN MATRIX AND RESPONSE VALUES	120
Appendix C	CLIMB PHASE METAMODELING: COEFFICIENT VALUES OF ALL TERMS	123
Appendix D	CRUISE PHASE METAMODELING: DESIGN MATRIX AND RESPONSE VALUES	124
VITA	125
REFERENCES	126

List of Tables

1.1	Summary of comparison between RSM and ANN.	19
2.1	Variables in the proposed architecture: examples in different types of systems.	32
3.1	An example of resolution III design.	55
3.2	Data for linear regression.	60
5.1	Ranges of the mission profile parameters.	84
5.2	Subsystems' health parameters and their definitions.	85
5.3	Range of subsystems' health parameters.	86
5.4	Design matrix for screening takeoff parameters and the observed values.	88
5.5	Design matrix for screening subsystems' health parameters during each phase.	90
5.6	Coefficient values (coded) of mission profile parameters obtained from screening experiments.	90
5.7	Coefficient values (coded) of subsystems' health parameters obtained from screening experiments.	91
5.8	Summary of significant parameters.	96

List of Figures

1.1	Examples of metamodeling methods.	9
1.2	Architecture of ISO-13374 standard for condition monitoring and diagnostics. [1]	16
1.3	Organization of the thesis.	20
2.1	Organization of Chapter 2.	22
2.2	An example of the system hierarchy in a complex system.	25
2.3	Hierarchical arrangement of parameters used in the architecture. . .	26
2.4	Decreasing strength due to stress factors.	27
3.1	Block diagram of building a response surface metamodel	46
3.2	An example of time-series data parametrization.	48
3.3	Screening based on linear RSE: (top) monotonic process (bottom) non-monotonic process.	50
3.4	Types of central composite designs (CCD).	58
4.1	The modules implemented in Chapter 4 are shown inside the circle. .	64
4.2	Hierarchical architecture of a gas turbine engine.	65
4.3	Notional creep behavior.	66
5.1	Block diagram of development procedure of system-level metamodel using subsystem level model.	76
5.2	Block diagram of the system-level response surface metamodeling procedure.	78
5.3	Subsystem-level construction of a turbofan engine.	79
5.4	Phases of a typical flight cycle of a commercial aircraft.	81
5.5	Damage accumulation during various phases of a nominal flight cycle. .	82
5.6	Mission profile parameters used to characterize mission of a commercial aircraft.	83
5.7	A central composite inscribed (CCI) design for 3 control factors. . .	88
5.8	Flowchart of the screening process.	93
5.9	Bar plot of estimates of mission profile parameters.	94

5.10	Bar plot of estimates of subsystems's health parameters.	95
5.11	Histogram of estimates of RSM model for climb phase	99
5.12	Coefficients of pruned system-level RSM model for climb phase. . . .	100
5.13	Comparison of prediction accuracy of full model and pruned model: climb phase.	101
5.14	Coefficients of pruned system-level RSM model for cruise phase. . . .	102
5.15	Comparison of prediction accuracy of full model and pruned model: cruise phase.	103
5.16	System level metamodel of the gas turbine engine represented as poly- nomial.	104
5.17	Graphical representation of the system-level metamodel.	105
5.18	System-level metamodel applied to estimate the damage accumulated during a flight cycle.	107
5.19	Performance of the system-level RSM model for the test cases. . . .	107
5.20	Testing error vs. number of neurons in the hidden layer: Climb phase ANN model.	110
5.21	Testing error vs. number of neurons in the hidden layer: Cruise phase ANN model.	111
5.22	Accuracy comparison of RSM model vs. ANN model: Test cases . .	111

SUMMARY

Condition-Based Maintenance (CBM) and Prognostics and Health Management (PHM) technologies aim at improving the availability, reliability, maintainability, and safety of systems through the development of fault diagnostic and failure prognostic algorithms. In complex engineering systems, such as aircraft, power plants, etc., the prognostic activities have been limited to the component-level, primarily due to the complexity of large-scale engineering systems. However, the output of these prognostic algorithms can be practically useful for the system managers, operators, or maintenance personnel, only if it helps them in making decisions, which are based on system-level parameters. Therefore, there is an emerging need to build health assessment methodologies at the system-level.

Fault diagnosis/prognosis research at the system-level has been approached via data-driven and model-based methods. Data-driven methods do not provide any insight into the effects of system parameters on fault growth, thus offer little assistance in planning future operations or optimizing maintenance schedules. Model-based methods have been focused primarily on component-level prognostic models. Some researchers have borrowed concepts from the artificial intelligence domain and developed Model-Based Reasoning (MBR) methods but these are qualitative approaches and not useful for most of the practical application domains.

This thesis presents a methodology that is based on a hierarchical architecture consisting of three layers; system-level, subsystem-level, and component-level. Based on this hierarchy, four types of variables are defined, i.e., system-level variables, subsystem-level variables, load variables, and stress factors. These variables are interconnected with each other through four types of models, i.e., failure mechanism

model, load-stress model, subsystem-level model, and system-level metamodel. At the lowest level (component-level), damage accumulation models that do not require antecedent diagnostic activities, are used to estimate the damage in critical components. At the highest level of the hierarchy, response surface metamodeling methods are used to build a system-level health assessment model. These methods originated from the statistical theory of design of experiments (DOE), and focus on planning experiments to reduce the number of total observations by locating data points in “meaningful” regions. Therefore, a large dimensional input space can be modeled using a small number of data points. The methodology is then tested, using a turbofan engine simulator developed by NASA. The Response Surface Model (RSM) results are compared with those of Artificial Neural Networks (ANNs), which are another type of metamodeling methods. For a given amount of data, RSM results are found to be almost as accurate as the ANN results. In contrast to ANN, which is a black box type of model, RSM is represented as a polynomial. The model representation in a polynomial form is particularly useful in this application since the coefficient values represent the relative effect of input variables on the system-level health, thereby, facilitating the decision making process at the system-level.

A brief overview of the main contributions of this thesis is given below.

- Development of a hierarchical framework for using component-level prognostic models in assessing system-level health.
- Development of a methodology for capturing and representing the coupling between subsystems’ degradation.
- Development of a methodology for modeling the effects of system’s usage pattern on system-level health.
- Application of these methodologies to a complex system for demonstration purposes.

Chapter I

INTRODUCTION

1.1 Motivation

Recent advances in the prognostics and health management (PHM) technologies, have largely focused on component-level modeling. These models are useful for simple systems, in which the number of variables that interact with the model parameters are small. In the case of complex engineering systems, such as aircrafts, power plants, and the like, the constituent components are small elements in a much larger structure. Over the years, well-established procedures have been developed to build prognostic models of critical components in such systems. However, the output of these prognostic algorithms can be practically useful for the system managers, operators, or maintenance personnel only if it helps them in making decisions, which are based on system-level parameters. The prognostic algorithms, on the other hand, deal with low-level variables. Thus, a gap exists between the component-level prognostic modules and system-level applications. To make use of these prognostic results and build a system-level health assessment methodology that can assist the decision makers, this gap needs to be filled.

1.2 Problem Statement and Significance

A complex system can be considered to be a three-layered architecture, consisting of system, subsystems, and components. The component-level layer is the lowest layer in this hierarchy. It is true that most of the faults are initiated at this level, and failure prognostic technologies have focused on component-level faults. This thesis presents a procedure to develop a system-level model that is built using the component-level

prognostic information. First, the basic requirements that should be met by this model, are mentioned. It is, then followed by the issues that need to be addressed, and their proposed solution.

Complex engineering systems such as aircraft, power plant, etc. consist of several subsystems, which are interconnected with each other through the closed loop control system. A performance degradation in any of these subsystems, causes an overall reduction in the system's performance. To meet the performance criteria, the control system shifts the operating setpoints at the subsystem-level. This shift at subsystem-level translates into a change in the component-level parameters and subsequently causes different level of stresses in the components. In other words, a degradation in any of the subsystems might affect the life consumption rates of components in any of the subsystems, due to their interactions through the closed loop control system. A system-level health assessment methodology, therefore, should include subsystem-level diagnostic information in the model formulation.

The second requirement related to the modeling of the system-level health is discussed now. One of the objectives of developing this methodology is to model the relation between the system-level information and the component-level fault growth. The system-level information (e.g. usage pattern), however, is generally recorded and represented as time-histories. To utilize it in our modeling scheme, the system-level time-series data should be reduced into a set of parameters that can represent this information.

The third requirement concerns the representation of the model itself. The motivation for building this model is that it should assist the system managers, operators, and maintenance-related people in making appropriate decisions. The output of the model should be represented in such a way that all these categories of decision makers can readily benefit from the model representation.

In regard to the first two requirements, the fundamental issue is the size of the

problem, in terms of number of variables involved in the model. A complex engineering system such as an aircraft, power plant, and the like, consist of several subsystems, each of which consists of many components. These components are under the influence of several variables. In short, large number of variables are active in a complex system, at all levels. The variables themselves and their interactions render the number of constituent elements in the model extremely large. The problem is further complicated by the fact that the system's usage pattern is also included as input to the model. It means that the dimensionality of input space is prohibitively large. This issue has two practical ramifications. Firstly, the amount of data required to build a model depends on the dimensionality of the input space. Secondly, visualization of the modeling results keeps on getting obscure as the number of dimensions increase. In this dissertation, response surface methods (RSM) are proposed as a solution to both of these issues, i.e., amount of data and visualization of large-dimensional input space.

RSM are one of the metamodeling strategies that have been used in systems engineering, for quite some time. They are shown to be robust to noise (experimental variability) and have been successfully used in the fault-diagnostics problems [2]. For our work, two important characteristics of these methods are that the amount of data required to build these models is much smaller than the other modeling methods, and that these models are represented as polynomials. The model representation in a polynomial form is particularly attractive in our application as it reveals not only the relative significance of input variables but also interactions among them. Therefore, the representation of this information as a polynomial model can act as a helping aid in decision making at the system-level by the system managers, operators and maintainers.

1.3 Background of the Problem

In this dissertation, a methodology to assess system-level health is developed. At its foundation, the methodology is built on component-level prognostic models. Subsequently, these models are used in the system-level metamodeling framework. Hence, the methodology is confluence of two areas of study; prognostic modeling and metamodeling for engineering design. Extensive literature exists in both of these fields. In this section, background that is relevant to our problem, is discussed for each of these topics.

1.3.1 Prognosis Modeling

Foretelling the future has been attracting the interest of researchers and practitioners in a wide area of disciplines from finance to weather forecasting, since decades. In maintenance engineering disciplines, failure prognostic is defined as an ability to predict the future condition of a component and/or system of components [3]. In the condition-based maintenance (CBM) and prognostics and health management (PHM) technologies, it is considered to be the Achilles' heel [4].

Failure prognostics has been approached via a variety of techniques ranging from Bayesian estimation and other probabilistic/statistical methods to artificial intelligence tools. In general, prognosis approaches can be categorized broadly into model-based and/or data-driven methods.

1.3.1.1 Model-based prognosis

Model-based prognosis schemes include those that employ a model of the process being predicted. These schemes attempt to incorporate physical understanding of the system into the estimation of remaining useful life (RUL). Model-based approaches differ from data-driven approaches in that they can make remaining useful life (RUL) estimates in the absence of any recorded data. Component/system modeling is usually accomplished either at the micro level or the macro level.

At the micro level (also called the material level), physical models are embodied by a series of dynamic equations that define relationships, at a given time or load cycle, between the damage of a component, and environmental/operational conditions under which the component is being operated. For example, the Paris crack growth model relates the crack growth rate in materials to stress intensity factor [5], and Yu and Harris’s fatigue life model for ball bearings [6] relates the fatigue life of a bearing to the induced stress. Since measurements of critical damage properties (such as stress or strain of a mechanical component) are rarely available, sensed system parameters have to be used to infer the stress/strain values. Another strategy that is typically used to build physics-based models of a component is finite-element analysis (FEA), in which the element is fully defined in its geometry and/or properties, and the problem is solved numerically. While the micro level modeling approach is useful for offline tasks, it is not feasible to be used in the online health assessment of the component. Another drawback of this approach is that it does not provide any understanding of the effects of system-level operating conditions on fault evolution.

In many cases involving complex systems, deriving models based on all the physical processes involved is difficult. Macro level models are based on relatively simplified mathematical relationships among system input variables, system state variables, and system output variables [7]. The trade-off is between increased coverage of degradation modes and reduced accuracy of a particular degradation mode. For prognosis applications, macro level models have been used in Bayesian estimation techniques, autoregressive moving average (ARMA) techniques, and Kalman/particle filtering methods. Bayesian estimation techniques combine information from fault growth models and online sensor data to predict the posterior probability density function of fault growth. The Bayesian learning framework offers the advantage of explicitly incorporating and propagating uncertainty in the prediction models [8]. The core idea is to construct a probability density function (PDF) of the state based on all the

available information. For a linear system with Gaussian noise, the method reduces to a Kalman filter. For nonlinear systems or non-Gaussian noise, no general analytical (closed-form) solution exists for the state-space PDF, and an extended Kalman filter (EKF) is the most popular solution to recursive nonlinear state estimation problem [9]. In this case, the desired PDF is approximated by the Gaussian form, which may have significant deviation from the true distribution, causing the filter to diverge. In contrast, in the particle filtering (PF) approach [10], the PDF is approximated by a set of particles (points) representing sampled values from the unknown state space and a set of associated weights denoting discrete probability masses. The particles are generated and recursively updated based on a probabilistic model as well as measurements. In other words, PF is a technique for implementing a recursive Bayesian filter using Monte Carlo (MC) simulations and is also known as the sequential MC (SMC) method. Recently, a particle filtering-based approach has been used for analyzing crack growth length in a planetary gear plate of a helicopter [11] and battery health monitoring in a spacecraft [12].

The Bayesian learning scheme has been reported to provide a robust framework for long-term prognosis. Though appropriate for online tasks, this scheme fails to take into consideration faults in other subsystems. For example, a planetary gear plate is a subsystem in the helicopter transmission [11]. Similarly, in [12], a battery is a subsystem in a large system (spacecraft). RUL estimate provided by [11] and [12] may be accurate if all the other subsystems are performing according to their design specifications. In field conditions, however, the other subsystems also undergo health degradation. Any system level prognostic methodology must take into consideration the effects of health degradation of other subsystems as well.

1.3.1.2 Data-driven prognosis

As the name implies, data-driven techniques utilize the monitored operational data related to system health. In many instances, fault data in terms of the time plots of various signals are available. In such cases, data-driven prediction techniques such as artificial neural networks (ANN), fuzzy-logic systems, and other computational intelligence methods have been used. ANNs are well suited for practical problems, for which it is easier to record data than to gather knowledge governing the underlying system being studied. A dynamic version of ANN, dynamic or recurrent wavelet neural networks, has been used in the prognosis of bearing crack size and industrial chiller [13]. The recurrent neural network has also been reported to be effective in predicting RUL in the 2008 data challenge problem [14], in which run-to-failure data of an unspecified engineered system were provided, and the RULs of a set of test units were to be estimated. The algorithm used was based on the recurrent neural network architecture, and no attempt was made to analyze the data to understand or to extract the underlying features. Another data-driven approach uses Kalman filter ensembles of neural network models to determine RUL [15]. The method involved constructing multilayer perceptrons and radial basis function networks for regression. A linear Kalman filter was then used to filter and mix the ANN models as an ensemble.

Recently, a similarity based approach for estimating the system remaining useful life (RUL) has been proposed [16]. This approach was used to tackle the data challenge problem defined by the 2008 PHM data challenge competition. Results showed that the similarity-based approach can be effective in performing RUL estimation. In this approach, first the features are combined to produce a single health indicator (HI). Then, HIs calculated from each cycle of the training unit form a one-dimensional time series, which is used to build a model depicting the pattern of performance degradation from normal to failure. In this way, a library of models is stored for multitudes of test units. The HI of the test unit is then calculated, and a distance

measure (e.g., Euclidian) between the test unit and the training units is defined. Once a distance measure has been defined, one RUL is estimated against each model in the library. The final RUL of the test unit is estimated through the weighted sum of the obtained RULs by assigning weights based on the distance between the test unit and each model in the library.

Data-driven algorithms can continue to learn as they operate, ideally making their assessments more reliable with each fault detection attempt. This approach and other data-driven methods avoid the need to understand the underlying physical mechanisms that describe the behavior of a system. However, data-driven methods fail to incorporate certain pieces of information, which are often available. For example, results of system FMECA studies contain information about severity of certain fault modes. This information, if ignored, can be relatively insignificant as far as accuracy in RUL prediction is concerned but is extremely vital from the perspective of system level health assessment. Another drawback of data-driven methods is that they do not provide physical insight into the effects of system parameters on fault growth. It is obvious that such methods will be of little assistance to system managers and operators in planning future operations.

In our methodology, a prognostic model of any type can be used at the component-level. Obviously, the more accurate the model, the better will be the performance of the overall algorithm. The only requirement is that the inputs of this component-level model are compatible with the higher-level parameters in the system's hierarchy.

1.4 Types of Metamodeling Methods

A model is an abstraction of a real-world phenomenon. The size of a complex system, however, prohibits the development of comprehensive system models. To save time and money spent on expensive physical experiments or computer simulations, many methods or heuristics have been proposed by researchers to facilitate the modeling

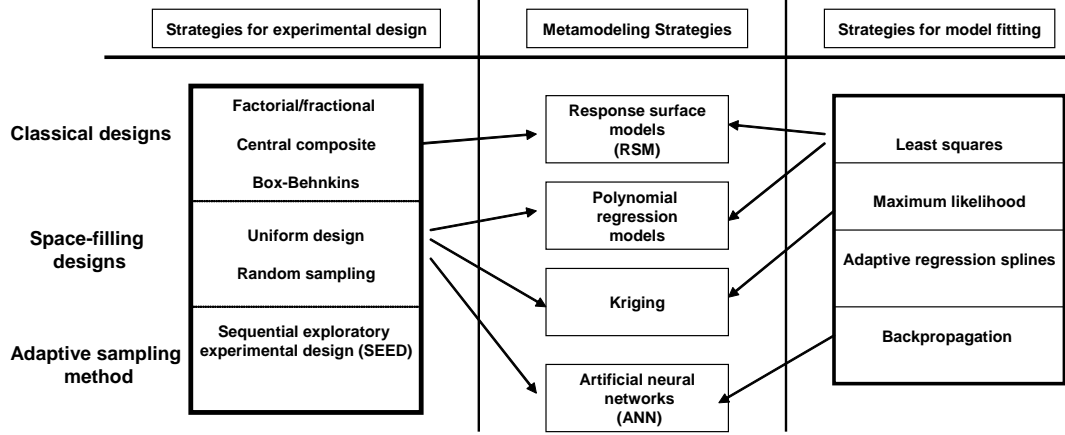


Figure 1.1: Examples of metamodeling methods.

processes. Metamodels or “surrogate” models are such types of models that are another level of abstraction and highlight the properties of the model itself. By using metamodels, engineers agree not to focus on the details of fluctuations of the responses, but try to grasp an approximated response-changing tendency with efficient and effective abstractions. The aim of these methods is to reduce the number of total observations or locate data points in “meaningful” regions. Metamodeling has been increasingly used in system design in recent years as a result of the never-ending need to move the abstraction level of design ever higher [17].

Response surface modeling (RSM), artificial neural networks (ANNs), and Kriging are well-known metamodeling strategies. The metamodeling procedure, in general, can be divided into two parts; designing experiments to generate data; and fitting the model on the generated data. Each of these methods has its own strategies for experimental design and model fitting. Figure 1.1 highlights a few of the well-known metamodeling methods alongwith their corresponding experimental designs and model fitting strategies.

1.4.1 Response Surface Models (RSM)

Response surface methodology (RSM) has been defined in several ways. In [18], RSM is defined as “the body of techniques by which one experimentally seeks an optimum

set of system conditions”. In [19], it is stated that “response surface methodology comprises a group of statistical techniques for empirical model building and model exploitation. By careful design and analysis of experiments, it seeks to relate a response, or output variable to the levels of a number of predictors, or input variables, that affect it.” In [20], RSM is defined as “a collection of tools in design or data analysis that enhance the exploration of a region of design variables in one or more responses.” Like the other metamodeling methods, RSM consists of two components, i.e., experimental design and model fitting. For model fitting, least squares methods are used to create polynomial regression (PR) models, while the experimental design is based on classical sampling methods.

Classical sampling methods originated from the statistical theory of design of experiments (DOE), and focus on planning experiments so that the “appropriate” data that can be analyzed by statistical methods is collected, resulting in valid and objective conclusions. While applying these methods, the variables/factors involved in the designs are categorized as response factors, control factors, nuisance factors, and noise factors.

Response factors are the variables that provide useful information about the process under study. When the experiments are designed to develop prognostic models, the fault dimension or the features related to the fault dimension are included as the response variables.

The factors that may influence the performance of a process or system, can be classified as either control factors or nuisance factors. The control factors are those factors that the experimenter may wish to vary in the experiment. Nuisance factors, on the other hand, may have large effects that must be accounted for, yet the experimenter may not be interested in them in the context of the present experiment. While developing component-level metamodels, the objective is to estimate the relation between the loads on the damage acceleration in a component. Hence, in this

case, the control factors are the load variables that the experimenter is interested in exploring their effects on the damage. Some of the load variables might have an effect on the damage acceleration but might not be measurable or of no practical value. Such variables will be nuisance factors in experimental design. In such a case, experiments are designed in such a way that the effects of the nuisance variables are minimized in the final model.

When a factor is present that varies naturally and uncontrollably in the process, it is called noise factor. It can come from various sources, like the measurement tools or the process itself.

1.4.2 Kriging

Kriging is named after D. G. Krige, a South African mining engineer who, in the 1950's, developed empirical methods for determining true ore grade distributions from distributions based on sampled ore grades [21]. These metamodels are extremely flexible due to the wide range of correlation functions that can be chosen for building the metamodel. Furthermore, depending on the choice of the correlation function, the metamodel can either “honor the data,” providing an exact interpolation of the data, or “smooth the data,” providing an inexact interpolation [22].

Kriging postulates a combination of a polynomial model and departures of the form:

$$y(x) = f(x) + Z(x), \tag{1.4.1}$$

where $y(x)$ is the unknown function of interest, $f(x)$ is a known polynomial function of x , and $Z(x)$ is the realization of a stochastic process with zero mean, variance σ^2 , and nonzero covariance. The $f(x)$ term in 1.4.1 is similar to the polynomial model in a response surface, providing a “global” model of the design space. In many cases $f(x)$ is simply taken to be a constant term [23]. While $f(x)$ “globally” approximates

the design space, $Z(x)$ creates “localized” deviations [17].

In general the Kriging models are more accurate for nonlinear problems but difficult to obtain and use because a global optimization process is applied to identify the maximum likelihood estimators. Kriging is also flexible in either interpolating the sample points or filtering noisy data. On the contrary, a polynomial model is easy to construct, clear on parameter sensitivity, and cheap to work with but is less accurate than the Kriging model [24].

1.4.3 Multivariate Adaptive Regression Splines (MARS)

It is known that pre-specified parametric models are limited in flexibility and accuracy since accurate estimates are usually only possible when the true function is close to the pre-specified parametric one. Thus, when the form of the underlying true function is unknown, statisticians prefer methods like MARS that can adaptively create a statistical model. MARS is essentially a linear model with a forward and backward stepwise algorithm to select the terms to include in the model. The piecewise-linear MARS approximation is a linear combination of linear basis functions that are truncated at knots. The knots determine where the approximation bends to model curvature, and one of the objectives of the forward stepwise algorithm is to select appropriate knots. After a reasonable piecewise-linear MARS approximation has been constructed, there is an option to smooth the approximation to achieve first derivative (or higher) continuity. MARS is both flexible and straightforward to implement with the computational effort primarily dependent on the number of basis functions added to the model. This approach has been successfully used in modeling the objective function in large-scale dynamic programming problems [25].

1.4.4 Artificial Neural Network (ANN)

ANN models have been very popular for modeling a variety of physical relationships. Mathematically, an ANN model is a nonlinear statistical model, and a nonlinear

method, e.g., backpropagation, is used to estimate the parameters (weights) of the model. There are two main issues in building a network:

1. Specifying the architecture for the network.
2. Training the network to perform well with reference to a training set.

To a statistician, these issues are equivalent to the following steps, respectively [25]:

1. Specifying a regression model.
2. Estimating the parameters of the model given a set of data.

If the architecture is made large enough, a neural network can be a nearly universal approximator [26].

1.5 Metamodeling Applications in the System-level Prognostics and Health Management (PHM) Domain

1.5.1 Model Approximation

The goal of model approximation is to achieve a global metamodel as accurate as possible to the actual process, at a reasonable cost. Metamodeling for model approximation is generally performed as a two step process, namely data generation by performing experiments and fitting the model on the generated data.

1.5.2 Problem Formulation

In a complex system, a large number of variables are involved at various levels. In most of the cases, it is not practical to include all the variables in the final structure of the problem. Therefore, a sequential approach is used to keep track of these variables at various levels. While a sequential approach helps in managing engineering problems for complex systems, engineers need a systematic way to reduce the number of variables as the design/analysis advances further. Some of the metamodeling

methodologies offers screening strategies, which are based on systematic procedures to reduce the number of variable that are significant in the subsequent modeling stages.

It is worth-mentioning here that all types of metamodeling methodologies can not be used in such types of applications. ANN, for example, is a black-box type of metamodeling strategy. To quantitatively determine the relative importance of the model inputs using ANN models, one needs to perform further analysis. Such type of metamodeling strategies are therefore not vary attractive for formulating problems for complex systems.

1.5.3 Design-space Exploration

The relationship between the factors and the response is usually embedded in complex equations or models. Engineers, by experience, often only have a vague idea about such relationships. A common method an engineer uses to understand a design problem is through sensitivity analysis and “what-if” questions. Sensitivity analysis, however, is based on a fixed condition with the variation of one variable. An engineer still cannot have an idea of the interaction between the variables. Some types of metamodeling approaches can assist the engineer to gain an insight into the problem, through two channels. The first is through the metamodel itself. Given a metamodel, one can analyze the properties of the metamodel to gain a better understanding of the problem. A good example is for the polynomial metamodel, in which the interaction effects are represented as separate terms. The magnitude of the coefficients in these terms indicates the significance of the interaction between the corresponding variables.

The second way of enhancing the understanding is through visualization. Visualization of multi-dimensional data alone has been an interesting topic, and many methods have been developed over the years [27]. To bring this issue in the context of prognostics and health management (PHM) framework of a complex system,

Figure 1.2 shows an the open systems architecture for condition-based maintenance (OSA-CBM) [1]. In the hierarchy suggested in this architecture, the high-level layer is the advisory generation (AG) layer, which is responsible for taking actions, based on the inputs provided by the prognostic assessment (PA) layer.

Considering the types of actions that can be taken by the AG layer, there can be three categories:

- Maintenance actions: These are the actions that are to be taken by the maintenance personnel who are concerned with such type of actions as the scheduling the next maintenance action, e.g., scheduling the non-destructive inspection (NDI) of turbine blade .
- Actions by the system-operators: Operators are responsible for such types of action that can enhance the safety of the system, or increase the cost-effectiveness of the operations, during the real-time functioning of the system, e.g. altering the aircraft power settings, changing the altitude .
- Management actions: These types of actions are taken while the operations are still being planned, i.e., the system is offline. For example, different aircrafts in a fleet can be scheduled for different types of missions depending upon their subsystem's health state.

Obviously, each type of action requires that the output of the PA layer be visualized in such a way that the relevant group of people can benefit from the prognostic assessment results. These results are based on the output generated by the health assessment (HA) layer. The methodology for system-level metamodeling developed in this dissertation, comes into play at the HA-level. It becomes, therefore, critical that the information generated by the health-assessment layer be represented in such a way that can be utilized at the PA-level, and subsequently at the AG -level, by each group of people. This scenario restricts the types of metamodeling strategies that

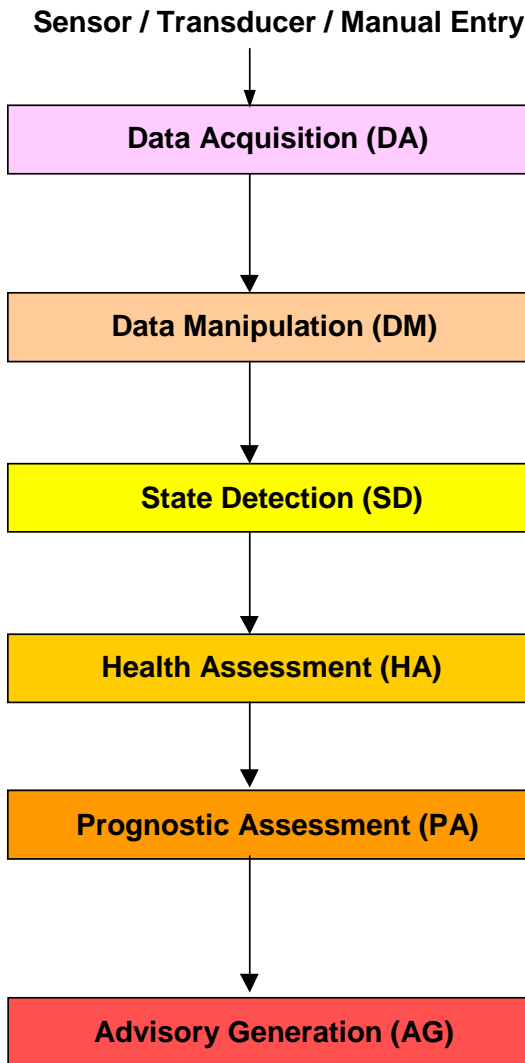


Figure 1.2: Architecture of ISO-13374 standard for condition monitoring and diagnostics. [1]

can be used in system-level health assessment task. This subject will be brought into discussion again in the next section, in which the two popular metamodeling strategies, i.e. RSM and ANN, are compared for their advantages and disadvantages.

1.6 Types of Metamodeling Strategies

1.6.1 Comparison between RSM and ANN

Metamodeling approaches, like other modeling methods, can be compared using several criteria. Response surface methodology (RSM) and artificial neural network (ANN) methods are two well-known approaches for constructing metamodels [28], and are compared from the following aspects.

1.6.1.1 Handling nonlinearities in the process

ANN is more suitable for those problems that are characterised by high nonlinearities. In [29], the authors claim that an ANN model can capture any degree of non-linearity that exists between the response and input process parameters while exhibiting good generalization capability. RSM, on the other hand, are mostly used for linear, or at the most, quadratic problems [30].

1.6.1.2 Amount of data requirement

For any modeling method, the required amount of data, to a large extent, is dependent on the nature of the problem (e.g. dimensionality, nonlinearity, required accuracy, and the like). RSM is mostly used for linear and quadratic models. For a linear model, 2^k (where k = number of factors/dimensions) samples obtained at specified locations in the model space are sufficient, while $2^k + 2k + 1$ samples are used to build a quadratic model. For the same number of input factors, the required number of data points are much larger. However, these points can be used from random locations in the model space. This fact is further elaborated in the next point.

1.6.1.3 Suitability for archived data

The model fitting component of RSM is based on the method of least squares (regression analysis), while the data generation component of RSM is based on classical DOE methods, which require pre-planned experiments at specific system settings. The ANN, on the other hand, is not restricted in this sense. In other words, once the metamodels are to be fitted using the archived data, RSM cannot be used in its entirety, i.e. only the regression part can be used in this case.

1.6.1.4 Expert knowledge required in building the model

ANN modeling is a type of data-driven methods that avoid the need to understand the underlying physics of the system. RSM offers a large flexibility in this aspect, and this is due to the fact that RSM is built in many stages. In the initial stage, no expert knowledge about the system is expected. As the experimentation proceeds further, the results from the previous stage provides knowledge, which can be incorporated to refine the model. If the knowledge about the system is already available before any experimentation is initiated, it can help in skipping some of the tests. In other words, ANN is purely a data-driven method, in which knowledge about the system is neither required not useful in the modeling process. RSM, in this respect, is a flexible approach. An attractive feature of this approach is that the expert knowledge is not required beforehand. However, it facilitates the modeling process, if available.

1.6.1.5 Enhancing the understanding of the process

RSM are represented as polynomial models, in which the coefficients values reveal useful information about the underlying process. This information includes the relative significance of the inputs on the outputs, the significance of interactions between the inputs, and the extent of nonlinearity present in the process. ANNs, on the other hand, are “black-box” models [31], which do not reveal much knowledge about the underlying problem. Of course, sensitivity analysis can be performed using an ANN

Table 1.1: Summary of comparison between RSM and ANN.

	RSM	ANN
Nonlinearity handling ability	low	high
Amount of data required	low	high
Suitability for archived data	no	yes
Expert knowledge required to build the model	no	no
Expert knowledge facilitates the modeling process	yes	no
Enhance the understanding of the underlying process	yes	no

model but that requires another level of experimentation.

Table 1.1 summarizes the points discussed in the above paragraphs. Both approaches have their pros and cons. There are some situations, however, where any one of the advantages of an approach dominates all its disadvantages. For example, in the system-level health assessment methodology developed in this work, the focus is on building a model that can assist the decision makers. Despite its drawbacks, RSM is well-suited for this purpose because of the following two advantages. Firstly, the amount of data required to build these models is much smaller than that for ANN. Secondly, the model representation in a polynomial form reveals not only the relative significance of input variables but also interactions among them. Therefore, the representation of this information as a polynomial model can act as a helping aid in decision making at the system-level by the system managers, operators and maintainers.

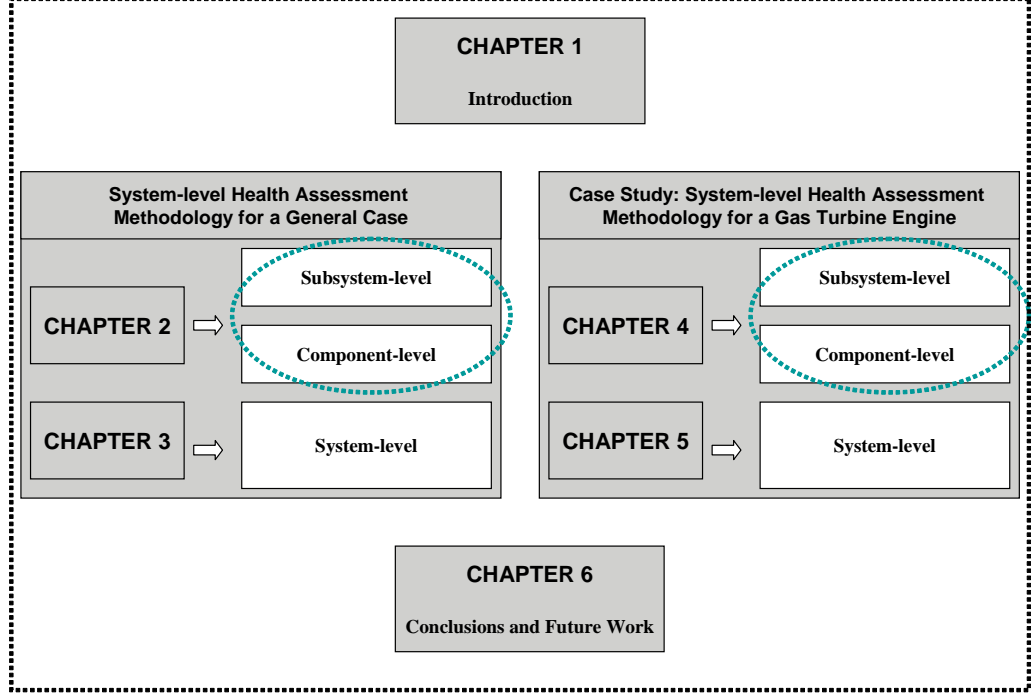


Figure 1.3: Organization of the thesis.

1.7 Organization of the Thesis

The outline of the thesis is given in Figure 1.3. The procedure to develop the meta-model for a general application is explained Chapter 2 and Chapter 3. The methodology is then tested on a gas turbine engine of turbo fan type, using an engine simulator developed by NASA, and the results are discussed in Chapter 4 and Chapter 5.

Chapter II

HEALTH ASSESSMENT METHODOLOGY FOR A GENERAL CASE: COMPONENT- LEVEL AND SUBSYSTEM-LEVEL MODELING

This chapter contains details of the development procedure of the system-level health assessment methodology, for a general case. As shown in Figure 2.1, the chapter is organized as follows.

- Architecture of the methodology: The present work focuses on a complex system. The methodology developed in this work arranges such a system into a hierarchy consisting of three layers: system-level, subsystem-level, and component-level. Based on this hierarchy, four types of variables are defined at various levels, i.e, system-level variables, subsystem-level variables, load variables, and stress factors. These variables are interconnected with each other through four types of models, i.e, failure mechanism model, load-stress model, subsystem-level model, and system-level metamodel.
- Types of variables in the methodology: Section 2.2 discusses the types of variables used in this methodology.
- Examples of these variables in different types of systems: In section 2.3, three examples of different types of systems (electromechanical, vehicle electrical system, and turbomachinery) are used to explain the arrangement of these variables.
- Types of models and their development procedure: The following four types of models are present in this architecture.

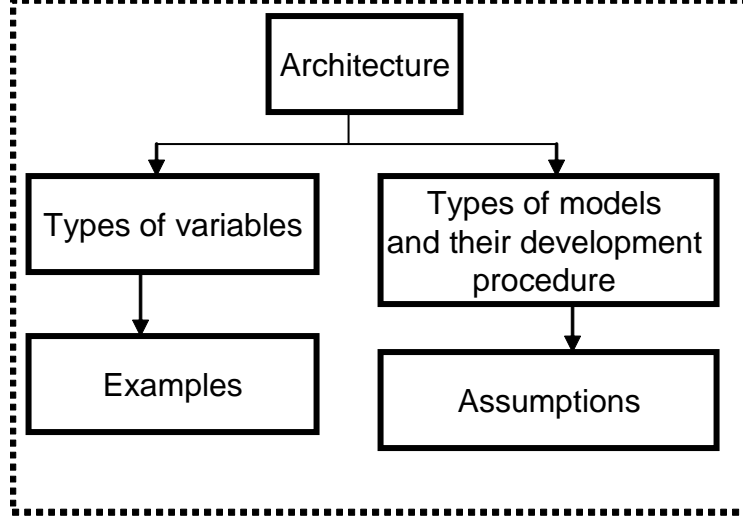


Figure 2.1: Organization of Chapter 2.

1. Failure mechanism model.
2. Load-stress model.
3. Subsystem-level model.
4. System-level model.

Each of these models are explained in section 2.4. A brief introduction to these models is presented here.

At the lowest level (component-level) failure mechanism models are used to estimate the damage under the effect of various stress factors active in the component. A large body of literature exists on failure mechanism models of various types. In this thesis, damage accumulation models that do not require prior diagnostic activities have been used, although the methodology can be adapted for systems in which diagnostic updates are available. Damage accumulation models have also been termed as life consumption monitoring models [32]. Such models are particularly useful during the design phase of the prognostic system when the field data are not available. These models have three components:

- (a) Damage model
- (b) Damage accumulation rule
- (c) Acceptable limits of damage accumulation

Each of these components are explained in section 2.4.1

The next higher-level model is the load-stress model, which translates the load variables into stress factors. These models can be classified into two broad categories: analytical and numerical. Numerical types of models, e.g. finite element (FE), although usually more accurate than the analytical ones, are quite complex to understand and represent. Hence, these results are not appropriate to be used in applications like ours where simplicity in representing the inputs outputs relation is a key requirement. In system design, FEA results are simplified using “surrogate models” or “metamodels”. Obviously, the simplification is achieved at the cost of accuracy.

Load-stress models and failure mechanism models are component-level models. Since a large number of components are present in a system, it is not feasible to take measurements at the component-level. Subsystem-level is the lowest-level at which parameters can be monitored online in a feasible manner. These parameters are translated into component-level degradation by a subsystem-level model that is built on top of the component-level models (load-stress model and failure mechanism model).

A system-level metamodel aims at representing the effect of system-level variables on the degradation in the critical components of the system. These variables, however, are not the only source of variation in the subsystem-level variables. Another such source is the health of the constituent subsystems. As any of these subsystems undergo deterioration in their health, the overall system’s performance might be affected. To compensate for the loss in performance,

the closed loop control system readjusts the setpoint, and causes a change in the subsystem-level variables. Since the relation between the subsystem-level variables and the component-level degradation is already known through the subsystem-level model, the relation between the system-level variables and the component-level degradation is obtained.

- Assumptions: In developing this system-level health assessment methodology, a series of assumptions is made. Some of these assumptions are related to the availability of system-related information. The rest are related with the development of models used at various levels. All of these assumptions are discussed in the section 2.5.

2.1 Architecture of the Methodology

2.1.1 System Hierarchy

A system is defined as a group of interrelated, interacting or interdependent constituents (components) forming a complex whole [33]. Many engineering systems are comprised of hundreds or thousands of components. Intermediate groupings, or various levels of subsystems, are necessary to describe or depict these systems manageably. Such an engineering system that requires one or more levels of definition intermediate to system and component is characterized as a complex system in this dissertation. Thus, a complex system is a system composed of a number of subsystems, each of which is embodied by a particular set of components, or sub-subsystems. In Figure 2.2, an aircraft engine is presented as an example of a complex system, which comprises of subsystems (e.g., LPC, HPT). Further down the hierarchy, the subsystems are composed of components (e.g., HPT blades).

2.1.2 Block Diagram of the Methodology

From a PHM perspective, a fault can be stated as a reduction in performance and failure means an inability to perform the intended function [34]. At the lowest level

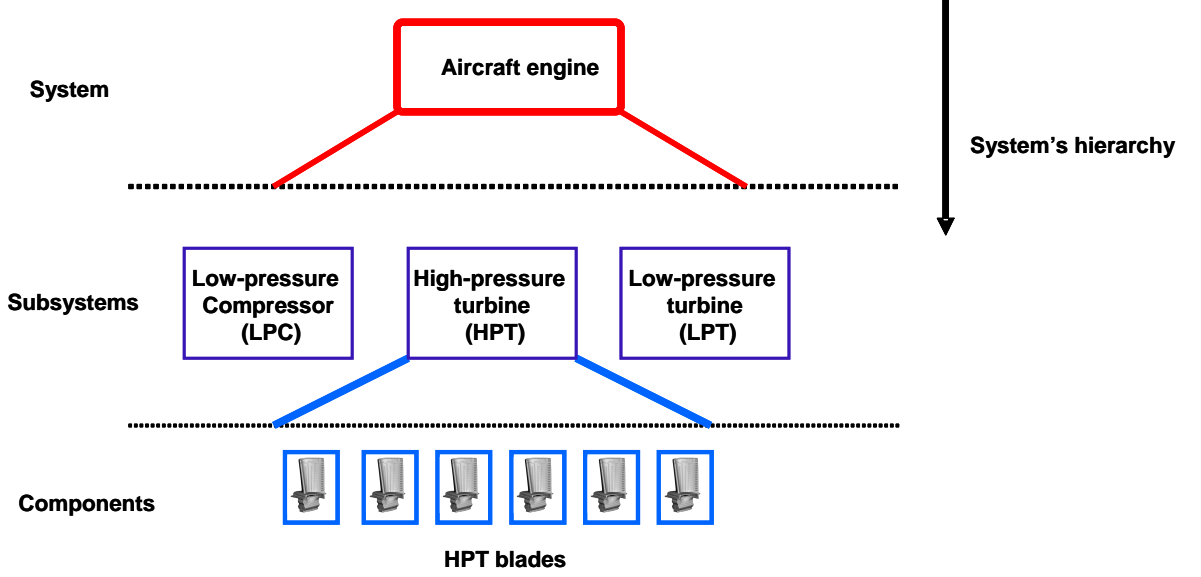


Figure 2.2: An example of the system hierarchy in a complex system.

of the hierarchy, a complex system is comprised of components. In most cases, faults originate at the component-level. A component-level fault might appear at subsystem-level, and subsequently at system-level. While component-level fault at the lowest level and system-level failure at the highest level, there are four types of parameters in between, i.e., stress factors (s), load variables (W), subsystem-level variables (V), and system-level variables (U). Another type of variable that is necessary to be considered is the subsystems' health parameters (X), which are used to account for any possible degradation in the health of constituent subsystems. The hierarchical arrangement of these parameters is shown in Figure 2.3 and explained in the following section.

2.2 *Types of Variables in the Architecture*

2.2.1 Stress Factors

2.2.1.1 *Why things fail?*

A component fails when the applied load exceeds its strength [35]. As time passes, the product can become weaker. Failure mechanisms are the physical processes that cause deterioration in the product. These physical processes are activated at microscopic

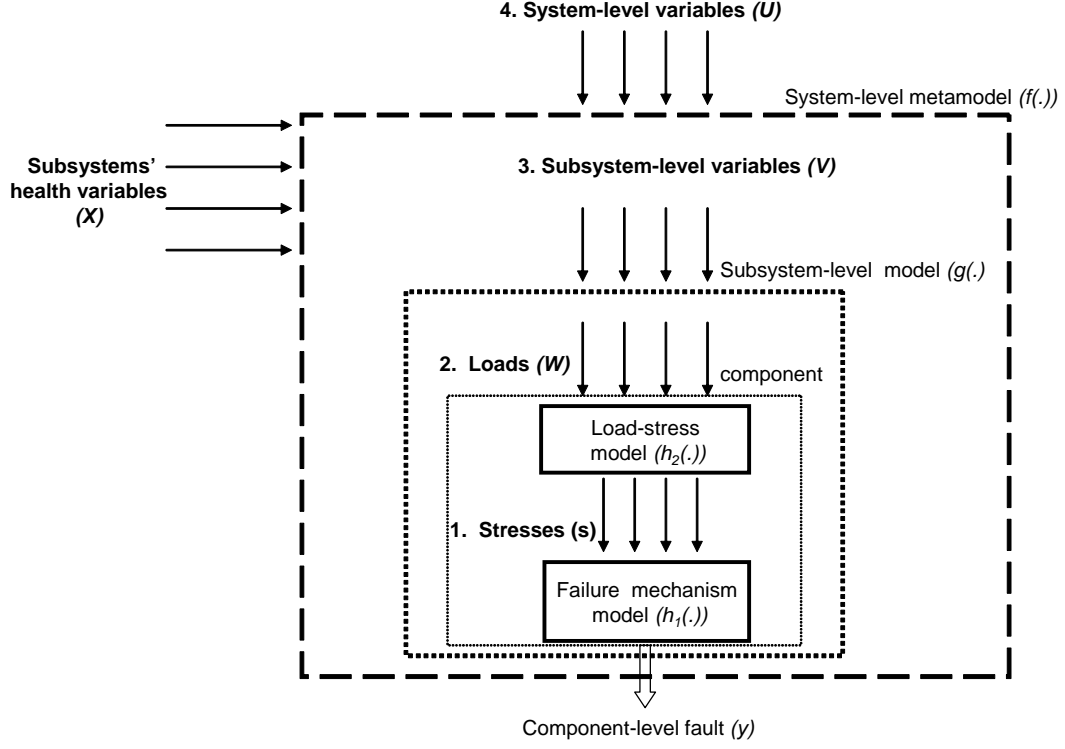


Figure 2.3: Hierarchical arrangement of parameters used in the architecture.

scale and cause the strength curve moves to the left, meaning that more products will fail (Figure 2.4). The parameters that directly drive or accelerate these failure mechanisms are termed stress factors in this work. A few example of stress factors are the following [35]:

- thermal (temperature/rate of change of temperature)
- electrical
- vibrational

For a particular product, many other product-unique stresses or parameters may be defined as appropriate. These may include humidity, PH of a solution in a product, and the like.

It may be noted here that in the discipline of solid mechanics, the term "stress" is categorically used for the parameters that determine the intensity of the internal

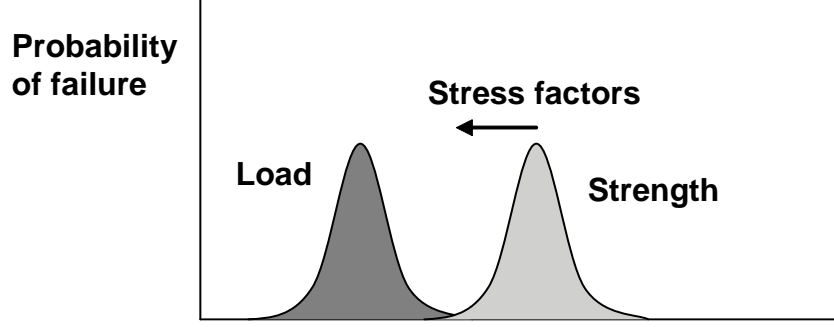


Figure 2.4: Decreasing strength due to stress factors.

forces acting between particles of a deformable body [36], usually represented by the symbol σ . In this dissertation, however, the term used is stress factor to distinguish it from the term stress. Here, the term stress factor is broadly used for the parameters that accelerate the failure mechanism at material-level, and is denoted by the symbol s .

2.2.2 Load Variables

The terms stress and load are used for different types of parameters in different fields. In this dissertation, load variables are defined here as the physically measurable parameters that can generate stresses at the component-level. Stress factors, in general, are not directly measurable using sensors. For example, temperature of a metal (load variable) is measurable using a sensor. The resulting thermal stress is an inferred quantity. The same temperature in different metal pieces will generate different levels of stress variable. In this thesis, load variables are represented by the symbol W .

At the next higher level, subsystem-level parameters have been categorized as subsystem-level operating conditions.

2.2.3 Subsystem-level Variables

The load variables on the components are generated by subsystem-level parameters, which have been termed as subsystem-level variables, and are denoted by the symbol

\mathbf{V} in this work. These are the lowest-level variables that can be measured directly using sensors. This is because it is not generally feasible to install sensors at the component-level, owing to their large number.

2.2.4 System-level Variables

At the highest level of the hierarchy, the layer of parameters consists of system-level variables (\mathbf{U}). These include the parameters through which the operators interact with the system (e.g., aircraft speed), and the parameters that specify the state of the system (e.g., aircraft altitude).

As shown in Figure 2.3, these parameters are linked with each other through four models, which are explained in Section 2.4.

2.3 Examples

In this section, three examples from different types of systems are used to demonstrate the hierarchy in the complex systems, as well as, variables in such a hierarchy.

2.3.1 Electromechanical System

- **System:** In this example, the system is a variable speed electric motor drive.
- **Subsystems:** This system consists of three subsystems; an electric machine, power converter (rectifier, inverter, etc.), and controller.
- **Components:** Each of these subsystem consists of several components. Stator windings and bearings are examples of components in an electric machine (subsystem). Similarly, insulated gate bipolar transistor (IGBTs) is an example of a component in a power converter (subsystem).

This system can fail in several ways, e.g., IGBT latchup fault, insulation failure in stator winding, and the like.

Winding insulation is one of the weakest components (both electrically and mechanically) in any electrical machine [37]. Focusing only on an insulation failure, following is a list of variables in hierarchically ascending order.

- Stress factors: There are many different stress factors that can affect the rate of degradation in stator and rotor winding. Broadly speaking there are thermal, electrical, ambient, and mechanical stress factors [38], the more important being the thermal and electrical.

Corresponding to each of these stress factors, there may be one or more load variable that generate these stresses.

- Load variables: In machine windings, one of the primary source of thermal stress is I^2R loss, which is driven by the current in the windings. Similarly, the switching frequency of the PWM (pulse width modulation) pulses increases electrical stress in the winding [39]. Hence, these parameters, i.e., current and switching frequency are the load variables for thermal and electrical stress factors, respectively.
- Subsystem-level variables: Current drawn by the machine causes a variation in the I^2R losses in the machine, and hence the thermal stress factor will be changed. Similarly, speed of the motor will affect the electrical stress factor on the machine by changing the switching frequency. Current drawn by the machine and its speed are subsystem-level variables.
- System-level variables: Examples of system-level variables are the torque supplied by the drive, speed of the drive, and the like.

2.3.2 Vehicle Electrical System

- System: In this example, the system is a vehicle electrical system.

- Subsystems: In general, this system consists of the following subsystems: a) battery, b) generator, and c) electrical loads.
- Components: Each of these subsystems has many components. Generator, for example, consists of rotor, stator, rectifier, and regulator. In the case of a battery, cell is the basic unit at its component-level.

This system can fail in several ways. In the case of batteries, grid corrosion is a common failure mode [40]. It results in an increased internal resistance of the battery [41].

- Stress factors: The three main stress factors that drive grid corrosion are the electrical stress, chemical stress, and thermal stress. The sources of these stress factors are load variables.
- Load variables: The electrical stress factor is generated by the cell voltage, which is the load variable for this type of stress factor. Similarly, increased acid concentration (load variable) increases the corrosion rate by increasing the chemical stress factor. Thermal stress factor is generated by the internal temperature of the cell (load variable).
- Subsystem-level variables: Under-the-hood temperature and current drawn from/to the battery are examples of subsystem-level variables that control the cell temperature and cell voltage (load variables).
- System-level variables: Ambient temperature, vehicle speed, system electrical load are examples of the system-level variables.

2.3.3 Turbomachinery

- System: In this example, the system is a turbofan engine of an aircraft.

- Subsystem: It consists of the following subsystems: a) fan, b) low-pressure compressor (LPC), c) high-pressure compressor (HPC), d) combustor, e) high-pressure turbine (HPT), and f) low-pressure turbine (LPT).
- Components: Each of these subsystems consists of several components, e.g, blades in HPT.

This system can fail in several ways. Creep rupture in the turbine blades is the most common failure mode of a turbofan engine.

- Stress factors: Creep in HPT blades is driven by two types of stress factors, i.e, Von Mises stresses and the blade temperature.
- Load variables: Blade temperature and its rotational speed are examples of load variables that might generate these stress factors.
- Subsystem-level variables: Turbine-entry-temperature (TET) and turbine rotational speed are examples of system-level variables that can affect the blade temperature and blade rotational speed.
- System-level variables: Ambient temperature, aircraft speed, altitude are examples of the system-level variables.

Table 2.1 summarizes some of the variables and their hierarchical arrangement, discussed in the above examples.

2.4 Types of Models in the Methodology

2.4.1 Failure Mechanism Model ($h_1(.)$)

Before discussing the topic of failure mechanism modeling, well-known failure mechanisms are briefly discussed as follows.

Table 2.1: Variables in the proposed architecture: examples in different types of systems.

	System		
	Electromechanical	Vehicle electrical system	Turbomachinery
Failure mechanism	insulation failure in windings	grid corrosion in battery cells	creep rupture in turbine blades
Stress factor	electrical	thermal	centrifugal
Load variable	PWM switching frequency	cell temperature	blade rotational speed
Subsystem-level variable	motor rotational speed	under-the-hood temperature	turbine core speed
System-level variable	drive speed	ambient temperature	aircraft speed

2.4.1.1 Failure Mechanisms

Failure mechanisms are the physical processes by which damage is caused to the components ultimately leading to failure [42]. Failure mechanisms can be broadly grouped into overstress mechanisms and wearout mechanisms. Overstress mechanisms are activated when the stresses acting on the component increase beyond a certain threshold, causing the component failure. In mechanical systems, examples of overstress failure mechanisms include brittle fracture, yield, buckling . In electrical system, an example of overstress failure mechanism is electric overstress (EOS). In this thesis, however, the focus is on the wearout failure mechanisms, which cause gradual or "graceful" degradation in components. These mechanisms are driven by stresses acting on the components. Examples of wearout failure mechanisms include wear, fatigue, creep, and corrosion.

Wear is the erosion of material when two surfaces in contact experience relative sliding motion under the action of a contact force. Wear rate is usually a material property and directly related to the hardness of the material. Surface treatment to increase hardness can therefore increase wear resistance.

Fatigue is a leading cause of wearout failures in engineering hardware [43]. When cyclic stresses are applied to a material, failure of the material occurs at stresses much below the ultimate tensile strength of the material, due to accumulation of damage. The fatigue process consists of the initiation and subsequent propagation of a crack.

At elevated temperatures, most materials can fail at a level of mechanical stress which is much lower than its ultimate strength measured at ambient temperature due to a phenomenon known as creep. It is modeled as time-dependent deformation and thereby is mathematically distinct from elastic and plastic deformation [44]. Elastic and plastic deformations are mathematically modeled as instantaneous deformations occurring in response to applied stresses. In reality, all deformations are time dependent, but the characteristic times for elastic and plastic deformations are orders of magnitude smaller than those for creep.

Corrosion is the process of chemical or electrochemical degradation of materials. Common forms of corrosion are uniform, galvanic and pitting corrosion [42]. Uniform corrosion is a chemical, or electrochemical, reaction occurring at the metal-electrolyte interface uniformly all over the surface. Galvanic corrosion occurs when two or more different metals are in contact. Each metal has a unique electrochemical potential. Hence, when two metals are in contact, the metal with the higher electrochemical potential becomes the cathode (where a reduction reaction occurs) and the other metal becomes the anode (where an oxidation reaction, or corrosion, occurs). Thus a galvanic cell forms. Pitting corrosion occurs at localized areas causing the formation of pits. The corrosion conditions produced inside the pit accelerate the corrosion process.

In general, failure mechanisms can be broadly categorized as time-based and cycle-based. The classification based on this criterion can be useful in building the failure mechanism model.

Time-based failure mechanisms Time-based failure mechanisms are those , whose damage is a function of time-duration of the applied stresses. This type of failure-mechanisms is characterized by the diffusional movement of materials at the atomic or

sub-atomic scale, and the effect of stress factor is expressed as time to failure. In mechanical systems, creep and corrosion are examples of time-based failure mechanisms. In electronics, electromigration is an example of such type of failure mechanism.

Cycle-based failure mechanisms Cycle-based failure mechanisms, on the other hand, does not depend on the duration of time during which the stresses act on the material. Rather, the damage caused is a function of number of cycles, however, short or long. Fatigue is a well-known example of cycle-based failure mechanism.

In different engineering systems, several types of models have been developed that predict the effect of stress factors on the resulting damage evolution. Predicting damage evolution is, in general, a very demanding problem. [3] identifies this problem as a “grand challenge” requiring a multidisciplinary approach, including engineering mechanics, reliability engineering, electrical engineering, computer science, information science, material science, statistics, and mathematics.

2.4.1.2 Classification of failure mechanism models

Failure mechanism models are used to estimate the damage caused to the component and subsequently its remaining useful life (RUL). These models can be classified into two broad categories.

Some models provide the accumulated damage/RUL estimates based on a priori knowledge of life expectancy and expected usage patterns. This prediction approach does not involve damage monitoring or diagnostic information. Such models are called as damage accumulation models.

Another category of models is based on the diagnostic information, and thus requires system status monitoring. [45] refers to these two kinds of performing prognostics as vertical and horizontal, respectively.

This thesis focuses on damage accumulation models that do not require prior diagnostic activities, i.e., using the vertical approach, although the approach can be

valid for systems in which diagnostic updates are available. These models have also been termed as life consumption monitoring models [32]. Such models are particularly useful during the design phase of the prognostic system when the field data are not available. These models have three components.

1. Damage model
2. Damage accumulation rule
3. Acceptable limits of damage accumulation

2.4.1.3 Damage model

Damage models (also known as stress models or named by the respective failure mechanisms) involve the description of the damage caused to the component through mathematical models of the physical laws governing its behavior. A disagreement on the definition of damage is, however, found in the literature. Some authors have defined damage as the extent of a part or product's degradation or deviation from its normal operating state [32]. On the other hand, some authors have insisted on using damage models in those cases where a degradation does not show itself in the component's performance [42]. In our view, the definition of damage to be used depends on the type of failure mechanism. In some cases, the effect of the failure mechanism is not obvious. For example, creep in a turbofan engine does not cause a significant degradation in the turbine's performance prior to failure. On the other hand, grid corrosion in a battery starts showing itself well before the failure.

An example of a damage model is the effect of temperature and Von Mises stresses (σ) on the rate of creep damage ($\dot{\epsilon}$) given as:

$$\epsilon_{creep} = \int_0^t \beta \sigma^2, \quad (2.4.1)$$

where ϵ_{creep} is creep strain, β is a temperature-dependent constant, σ are the Von Mises stresses, and t is time over which the loading conditions continued. This model translates the stress factors (Von Mises stresses and temperature) into component degradation.

2.4.1.4 Damage accumulation rules

The model discussed above determines the damage caused at a constant level of stress factors. There are only a few practical situations where they actually remain constant. Many models have been proposed for adding damage increments to predict failure under multi-loading conditions. These models can be classified as linear or nonlinear [32].

Due to its simplicity, Miner's cumulative linear damage rule [46], given in Equation 2.4.2, is commonly used to predict fatigue life damage under multiple loadings/environments. For time-based failure mechanisms such as creep [47], Miner's rule is expressed as:

$$CDI = \sum_{i=1}^m \frac{t_i}{L_i}, \quad (2.4.2)$$

where CDI is the cumulative damage index, t_i is the time interval at i_{th} level of stress, and L_i is the creep life at i_{th} level.

Similarly, for cycle-based failure mechanisms,

$$CDI = \sum_{i=1}^m \frac{n_i}{N_i},$$

n_i is actual number of applied cycles for the i_{th} load step, and N_i is the number of cycles to failure for the i_{th} load step. CDI ranges from 0 to 1.0, with 0 being the undamaged state and 1.0 being the fully damaged state. Failure is typically defined when the CDI exceeds a critical value of 0.7 [48].

Miner's cumulative linear damage rule is widely used in the disciplines of electronics reliability engineering and engineering mechanics. A few examples are; solder

joint reliability [49], predicting fatigue life of electronic components under thermal and vibration loading ([50] and [48]), and sequence effect of high-low vs low-high stresses on the creep fatigue life of Sn-Ag-Cu (SAC) solder [51].

However, the usefulness of Miner’s rule due to its linear nature has been called into question for the following reasons [52]:

1. The order of loading does not come into play in Miner’s Rule. The effect of sequence can be important. For example, [51] showed that for creep tests a high-low load sequence was more detrimental than a low-high load sequence. The sequence effect has been shown extensively for metals in general [53].
2. It does not capture the influence of stress levels on the damage accumulation rate. Damage is assumed to accumulate at the same rate at a given stress level without regard to past history, although experimental results have shown that damage can accumulate in a nonlinear manner [32]. Many researchers simply lower the CDI, however this is somewhat arbitrary and inconsistent for different loadings.

As a result, many nonlinear damage theories have been proposed to account for the nonlinearity in damage accumulation ([54], [52]) and to account for sequence effects and other abnormalities. However, such approaches do not always result in greater accuracy across a broad spectrum of environments and loadings, and they require extensive material testing and a great number of experiments. Such extensive material testing and data gathering is often impractical in most of the cases.

In general, Miner’s rule is recommended for its simplicity, versatility, and reasonable accuracy [55].

2.4.1.5 Acceptable limits of damage accumulation

The choice of the acceptable limit of damage depends on a variety of factors, such as the user’s application and the safety level associated with it, the type of failure the user

is interested in, and the user's definition of failure. For example, if the application involves human participation (such as aircraft or spacecraft) or may compromise the safety of personnel (such as machinery in a factory), a lower limit of damage accumulation may be chosen, but if the application is known to be fairly reliable (such as systems with multiple redundancy), a higher limit of damage accumulation may be selected [32].

The failure mechanism models enable us to model the effects of stresses on deterioration caused in the materials and help us understand the underlying physics of failure. The stresses, however, are generated by the loads being applied on the components. In this thesis, the relation between the load variables and the stress factors is expressed by load-stress model.

2.4.2 Load-stress Model ($h_2(.)$)

Each stress being generated in the component, is the result of some type of force acting on the component. For example, centrifugal stress in a rotating turbine is generated by the rotational force. The rotational speed is, thus, a load variable in this case. In this thesis, such a force that generates stresses in the components is termed as "load variable". Translation of the load variables into stress factors is achieved by certain models, which are termed as load-stress model in this thesis. The relation between load variables and stress factors, however, not only depend on the loads being applied on the component but also on its material properties and geometry. Here, it is pertinent to introduce the difference between the terms "uncertainty" and "variability".

2.4.2.1 Variability and uncertainty

Variability represents diversity or heterogeneity in a well characterized population. In other words, variability arises due to differences in the value of a quantity among different members of a population. Fundamentally a property of nature, variability

is usually not reducible through further measurement or study [56]. For example, different people have different body weights, no matter how carefully we measure them.

Uncertainty represents partial ignorance or lack of perfect information about poorly-characterized phenomena or models. Generally used by risk analysts, uncertainty is sometimes reducible through further measurement or study. For example, even though a risk assessor may not know the body weights of every person now living in a city, he or she can certainly take more samples to gain additional (but still imperfect) information about the distribution.

Stresses generated in a component might depend on applied loading conditions, its material properties and geometry. In a given component, its material properties and geometry do not vary. Hence, they can give rise to variability in the load-stress model but uncertainty in the model is unaffected by the material properties and geometry and depends only on the loading conditions. From a system-level perspective, interest lies in only those variables that vary due to changes in system-level parameters. Therefore, in a load -stress model, input parameters are limited to loading conditions only.

2.4.2.2 Types of load-stress models

These models can be classified into two broad categories: analytical and numerical. Analytical models have been used to model thermal stresses in electronic packages with orthotropic properties, interfacial compliance, and thermal loading as model inputs [57]. These models, though simple to use, are generally less accurate than the numerical models. An example of a tool that is oftenly used to develop numerical models is finite-element analysis (FEA). In this analysis, the structure is modeled as group of tiny elements meshed together. The inputs to this model are the geometry

definition of the structure, its material properties, and the boundary conditions applied on the structure, while the output is the resulting stress. In the context of our work, the applied boundary conditions are the load variables and the output is the stress factors that accelerate the failure mechanism. It means that the two models, i.e., the failure mechanism model $h_1(.)$ and the load-stress model $h_2(.)$ can be combined to determine the effect of load variables W on the component damage y , while stress factors s are the intermediate variable.

In other words, the failure mechanism model is given as:

$$y = h_1(s), \quad (2.4.3)$$

and the load-stress model as:

$$s = h_2(W). \quad (2.4.4)$$

Both of these models given in Equations 2.4.3 and 2.4.4 are component-level models, which can be combined as:

$$y = h_1(h_2(W)) = h(W). \quad (2.4.5)$$

Having said this, there are a few issues involved with this approach. In most cases, there are many variables that affect the component's boundary conditions and the relation between the applied boundary conditions and the resulting stresses is not simple. The complex geometrical configuration of the component means that even constant boundary conditions will result in a complex distribution of stresses. In short, the results produced by the finite-element models, though accurate, are quite complex to understand and represent. Hence, these results are not appropriate to be used in most of the applications, like ours, where simplicity in representing the inputs-outputs relation is a key requirement. In system design, FEA results are simplified

using “surrogate models” or “metamodels”. Obviously, the simplification is achieved at the cost of accuracy.

Load-stress models and failure mechanism models are component-level models. Due to the large number of components in a system, it is not feasible to take measurements at the component-level. Subsystem-level is the lowest-level at which parameters are monitored. These parameters are translated into component-level degradation by a subsystem-level model that is built on top of the component-level models (load-stress and failure mechanism).

2.4.3 Subsystem-level Model ($g(\cdot)$)

The load variables in the load-stress model are component-level parameters. As we move up in the system’s hierarchy, they originate from certain subsystem-level variables. By modeling the relation between these load variables and their sources at subsystem-level, a subsystem-level model is built. This model transforms the subsystem-level parameters into component-level degradation by using the results of the load-stress model and the failure mechanism model. Mathmatically, if we can map the subsystem-level variables (V) onto load variables (W), i.e., $V \rightarrow W$, then using Equation 2.4.5, a subsystem-level model can be constructed as,

$$y = g(V). \quad (2.4.6)$$

A subsystem-level model relates the effects of subsystem-level variables to the component’s degradation. These subsystem-level variables, however, are being generated by system-level parameters. Generally, any interaction with the system by the operators and managers takes place through these parameters. A system-level meta-model relates these parameters to subsystem-level variables, and thus, component degradation.

2.4.4 System-level Metamodel ($f(\cdot)$)

A system-level metamodel aims at representing the effect of system-level variables on the degradation in the critical components of the system. These variables, however, are not the only source of variation in the subsystem-level variables V . Another such source is the health of the constituent subsystems. As any of these subsystems undergo deterioration in their health, the overall system's performance might be affected. To compensate for the loss in performance, the closed loop control system readjusts the setpoint, and causes a variation in the subsystem-level variables. This implies that there are two sources that can cause variation in the subsystem level parameters, namely system-level variables and subsystems' health parameters. In this dissertation, the system-level variables have been denoted by U , and the subsystems' health parameters by X . The system-level metamodel aims at representing the relation between these vectors (U and X) and the component-level degradation y as:

$$y = f(U, X) \tag{2.4.7}$$

Since the component-level degradation is assumed to be the limiting factor of the system-level health, this expression is an implicit representation of the relation between the system-level variables, subsystems' health and the system-level health. In this dissertation, it is assumed that degradation of a single type is the life-limiting factor in the system. In those cases where failure mode effects and criticality analysis (FMECA) reveals otherwise, i.e, there are multiple dominant failure modes in the system, the component-level degradation models like the one in Equation 2.4.5, should be developed for each component and later combined together to build a single system-level metamodel.

A system-level metamodel can be built using several approaches. In this dissertation, response surface methodology (RSM) is used for this purpose. RSM is a type

of metamodeling strategies, and it has been used in engineering design of complex systems for many years. It is shown to be robust to noise (experimental variability) and has also been used in the fault-diagnostics problems [2]. Two attractive features of this method, related to the system-level health assesment metamodeling of complex systems, is its small amount of data requirement and their representation in polynomial form. The model representation in a polynomial form can act as a helping aid in decision making at the system-level since it reveals not only the relative significance of input variables but also interactions among them. The system-level response surface metamodeling procedure is discussed in Chapter 3.

2.5 *Assumptions*

2.5.1 System-related Assumptions

1. A complex system can fail in a multiple ways. It is being assumed in this work that the components that are most likely to fail are already known through FMECA study or expert knowledge. FMECA studies identify potential failure of a component/subsystem, determine the effects of this failure, and identify actions that can eliminate or reduce the likelihood of potential failures to occur [58]. For example, in the case of a gas turbine engine, it is known that most likely the HPT blades will fail before any other component of the engine.
2. It is also assumed that the failure mechanisms of each of these components are known a priori. This information can be obtained through the failure modes, mechanisms, and effects analysis (FMMEA) [59] of the system. The purpose of FMMEA is to identify potential failure mechanisms for all potential failures modes, and to prioritize failure mechanisms.
3. It is assumed that the health of the critical components is a representative of the health of the overall system. Referring again to the example of the gas turbine engine, it is assumed that the remaining useful life (RUL) of the HPT

blades is representative of the RUL of the engine as a whole.

2.5.2 Failure Mechanism Modeling

1. The first assumption in failure mechanism modeling is related to the interaction between the failure mechanisms. In a testing environment, it is relatively easy to estimate the effects of a failure mechanism separate from the other mechanisms. In real-life, however, multiple failure mechanisms are simultaneously playing their roles and are interacting with each other. Sometimes it is hard to even mark a clear boundary between their domains. For example, in turbomachinery, creep and low-cycle fatigue (LCF) are the dominant failure mechanisms. Creep is activated at high temperatures while LCF occurs as a result of thermal gradients. The two processes occur simultaneously when temperatures are high and non-constant. In this work, it is assumed that the failure mechanism model can capture the interaction effects.
2. The second assumption is related to the non-constant nature of applied stresses. These models are usually constant-stress models while in the actual operations, the stresses are non-constant. Hence, damage rules, which can account for non-constant stresses using constant-stress models, are needed. It is assumed that the damage rules can account for the random variations in the stress values.

2.5.3 Load-stress Model

Finite-element analysis (FEA) is used to model the relation between the load variables and the stress factors. For a given loading, the stresses in the component vary over a wide range. The most likely failure location is the one having the highest level of stress on it. As the loading changes, the location might change as well. However, in our methodology, it is assumed that the location remains the same irrespective of the changes in the load variables, i.e., there is a single “hotspot” in the component.

Chapter III

SYSTEM-LEVEL METAMODELING FOR A GENERAL CASE

As discussed in Chapter 1, “metamodels” or “surrogate models” are used in those cases where simplicity in representing the input/output relation is more important than the accuracy of the model. The inputs in our case are the system-level variables U and the subsystem health parameters X , while the model output is the component degradation y . Since a large number of inputs are generally present, a screening procedure is performed before building the model to ensure that only “significant” variables are retained in the final model.

The metamodel building procedure, in general, is depicted as a block diagram, as shown in Figure 3.1. The procedure consists of three main stages.

1. Identifying the input space: In this step, first the input factors to be studied are selected, and then the respective range of interest is identified. In our application, these factors are the system-level variables (U) and subsystems’ health variables (X).
2. Screening procedure: During screening, first sample points are chosen in the input space. These samples are chosen based on statistical theory of design of experiments (DOE). Then, physical experiments or simulations are performed at each of the sample points, and data are generated. The data are used to estimate the relative effects of the input factors, which are screened by comparing to a threshold value.
3. Modeling procedure: After the insignificant factors have been screened out, the

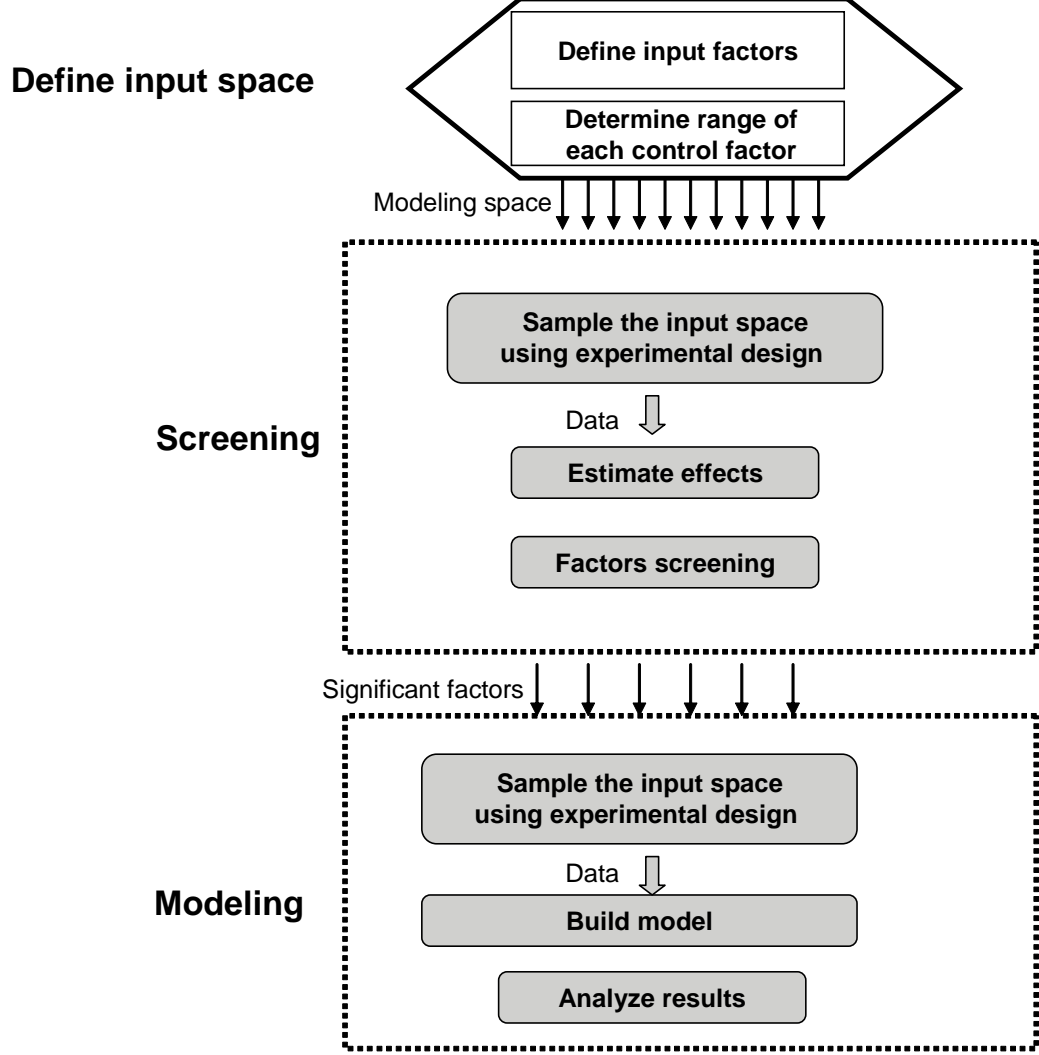


Figure 3.1: Block diagram of building a response surface metamodel

remaining variable make a new input space. DOE methods are then used to sample the space, and data are generated. A metamodel is built using the data, and the results of the metamodel are analyzed to verify that its performance meets the required criteria.

3.1 *Defining the input space*

The input space for the system-level metamodel is constituted by the two types of variables, i.e. system-level variables (U) and subsystems' health parameters (X). For each of these types, the input space is defined in two steps, as follows.

3.1.1 Parameterization of the system-level variables and subsystems' health parameters

The system-level metamodel aims at explaining the effect of the system-level variables (U) and the subsystems health parameters (X) on the degradation in the critical components. The information about these parameters, however, is recorded as time-histories, and thus should be reduced into a set of parameters that can characterize this information.

The simplification of time-series data into a set of parameters offers many advantages, a few of which are the following.

- Reduction in storage space and reduction in the calculation time: Data reduction is useful in life consumption monitoring to reduce data storage space and to reduce the time for damage calculations [60].
- A better understanding of the data: The time-historical representation, in general, is not helpful in understanding the data. The parametrization of the data can help in enhancing its understanding.
- A response surface metamodel (RSM) can only be developed if the information about the variables is represented appropriately. In other words, the data simplification is a requirement in building an RSM.

Many data simplification/reduction methods have been proposed by several authors. Example of such methods are the ordered overall range (OOR) [32] and rainflow-counting [61].

In this work, the the system-level variables have been parametrized while keeping in view the following:

1. The parametrization should help the managers in planning future operations.
2. It should help in determining the effects of operator actions on the system's health.

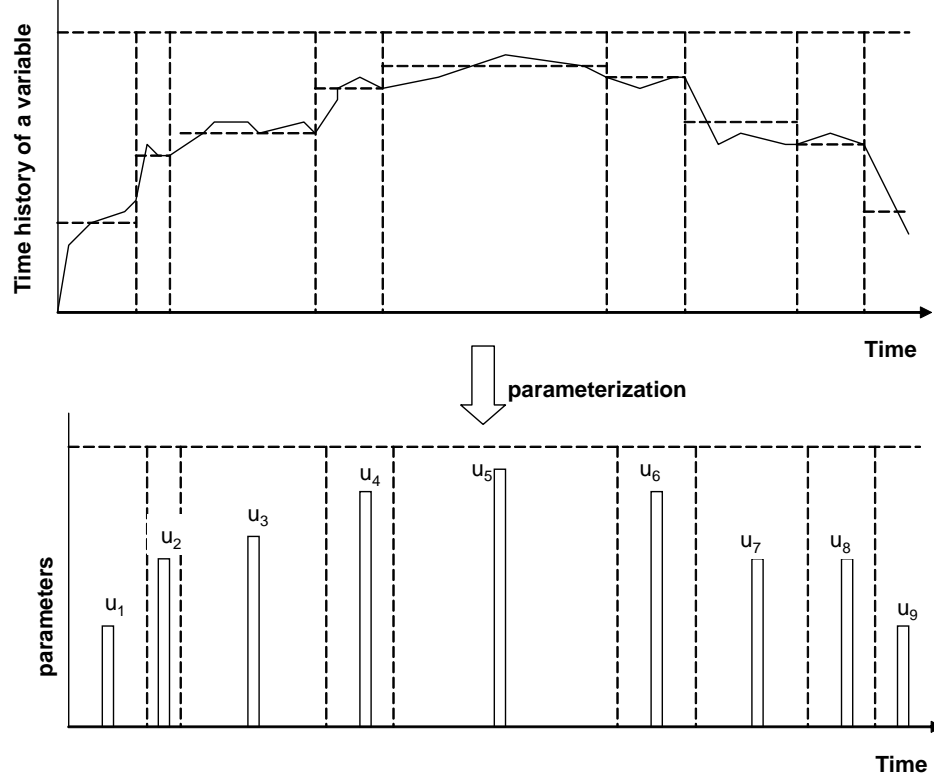


Figure 3.2: An example of time-series data parametrization.

To meet these criteria, our proposed parameterization technique breaks the time-histories into specialized segments. Then, each segment is represented by an average value. For example, in the case of an aircraft, a system-level variable (e.g., aircraft altitude, speed) is recorded as time-series data (Figure 3.2), which is arbitrarily broken down into segments, each of which is represented by a parameter.

This parameterization technique is based on an “averaging” method, and can be used for those failure mechanisms which are not sensitive to transients in the stress factors, e.g. creep, corrosion.

3.1.2 Defining the range of parameters

Once the time-histories of the system-level variables (U) and the subsystems’ health variables (X) have been transformed into a representative set of parameters, the ranges over which the model is to be built are determined based on the system expert

knowledge.

Once the input space has been defined, the next step is carrying out the screening experiment to reduce the set of factors to those that are most significant with respect to the damage in the component.

3.2 Screening

The number of experiments required to build a second or higher-order response surface equation (RSE) increases rapidly with the number of variables. When the number of factors is large or when experimentation is expensive, screening experiments are used to reduce the set of factors to those that are most influential on the response(s) being investigated. Usually, just a few variables contribute most of the variability in a response. Therefore, a screening test can be used to identify the most important independent variables. In a typical screening test, a low-level DOE is used to construct a low-order RSE with only the main effects considered. Based on this RSE, the sensitivity and relative importance of the response with respect to the independent variables are calculated. Next the variables are ranked in accordance with their importance to identify the major contributors. Then, according to desired fidelity, the major contributors can be identified.

The above procedure for a screening test is based on a linear RSE, and if the true function between response and a variable while all other variables are fixed is non-monotonic, a wrong decision to eliminate that variable may be made. A 1-dimensional case is shown in Figure 3.3. Here, a 2-level DOE is used to obtain a linear RSE, assuming that it approximates the true relationship between the control variable and the response variable. Based on this approximation, the control variable is eliminated from the subsequent modeling stages. As shown in Figure 3.3, there could have been two possibilities. If the actual underlying process is monotonic (top), the decision of eliminating the variable is a valid one. On the other hand, in the case of (bottom),

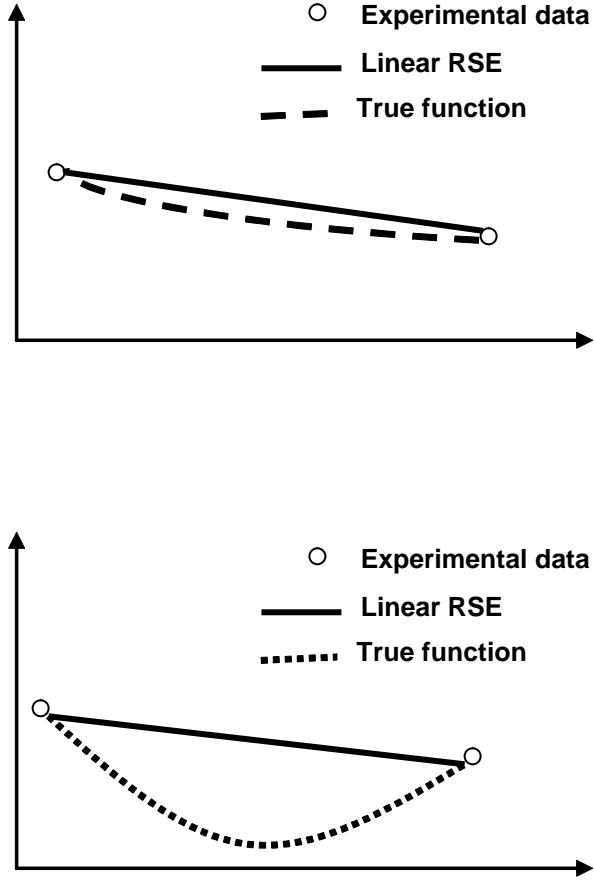


Figure 3.3: Screening based on linear RSE: (top) monotonic process (bottom) non-monotonic process.

the actual process is non-monotonic, The linear RSE is still flat which may lead to a decision that this variable is not important. But the response varies significantly with the variable, and hence its elimination based on the linear screening test is erroneous. This problem can be avoided by building higher-order models but this dramatically increases the number of experiments. For example, for $k = 4$ (the number of control variables), a first-order model requires $2^4 = 16$ experiments, while a second-order model requires $3^4 = 81$ experiments.

An efficient approach is to first go through the screening procedure using a first-order RSE. If the relative importance analysis shows the variable is a major contributor, then no further action is needed in the screening test and the variable is included in the subsequent modeling procedure. If the result from a screening test shows that a

potential variable is not important, 2 or 3 values for the variable are randomly picked, while fixing all other variables at arbitrary values. Then check if the response varies significantly with the variable. If the answer is yes, then it is an important variable and should not be eliminated.

A similar but more systematic approach is to use central composite designs (CCD) [62], which involves the use of a two-level factorial or fraction-factorial design combined with axial or star points that allow estimation of curvature. The well-known types of CCDs are discussed in Section 3.3.1.3.

The screening procedure is just like the metamodeling procedure in the sense that it consists of two steps: designing experiments and fitting the model. The difference is that objectives of designing experiments are different in the two cases. In the case of screening, the focus is on reducing the number of experiments. This, of course, is achieved at the cost of accuracy. In the case of metamodeling, accuracy of the model becomes more important.

In the methodology developed in this thesis, response surface methods (RSM) are used as the metamodeling method. These methods use classical experimental designs as a strategy for designing experiments and the method of least squares as the model fitting method. The following section briefly introduces a few well-known strategies for designing experiments while classical experimental designs are discussed in detail. The model fitting method, that is, the method of least squares, is discussed in the subsequent section.

3.3 Strategies for Designing Experiments

In traditional engineering experiments, the effects of varying a single parameter are observed with the other parameters held constant. This strategy can be termed as “one-factor-at-a-time” approach. After all the tests are performed, a series of graphs are constructed showing how the response variable is affected by varying each

factor with all the other factors held constant. The major disadvantage of one-factor-at-a-time strategy is that it fails to consider any possible interactions between the factors. An interaction is the failure of one of the factors to produce the same effects on the response at different levels of another factor. Interactions between factors are particularly common, and if they occur, the one-factor-at-a-time strategy will usually produce poor results. Another strategy of experimentation is the “best-guess” approach, which is frequently used in practice by engineers and scientists. Based on long experience with the system, some of the factors are ignored as they are known to have a relatively insignificant effect on the process being studied. Then, an arbitrary combination of the significant factors is selected, and the test is performed. Based on the outcome of the current test, the levels of one or two factors are changed while holding the other factors at the same levels as used previously. This approach can be continued almost indefinitely in the same way. This approach might work reasonably well if the experimenters have a great deal of technical and theoretical knowledge of the process as well as considerable practical experience. However, this approach is greatly dependent on expert knowledge and does not provide a systematic procedure for designing experiments.

To overcome these drawbacks, a large number of systematic approaches for experimental design have been proposed in the fields of engineering design and quality management. These approaches can be classified into the following categories.

Space filling designs In the case of deterministic computer experiments, many researchers advocate the use of ‘space-filling’ designs. The popular types of space-filling designs include uniform design and random sampling design. Uniform design methodology borrows concepts from the factorial design to choose a set of points uniformly distributed on the experimental domain [63]. The Monte Carlo simulation (MCS), which is a random sampling method, is still popular in industry, regardless

of its inefficiency [24]. It is probably because the adequate and yet efficient sample size at the outset of the experimentation is unknown for any black-box function.

Sequential and adaptive approaches Mainly due to lack of a priori knowledge of the “appropriate” sampling size, sequential and adaptive sampling has gained popularity in recent years [24]. In these methods, information from previous data points and metamodels is used as a guide in identifying new data points [17]. While classical experiments are also designed in a sequential manner, information from previous data points and metamodels is not used as a guide in identifying new data points. In [17], sequential exploratory experiment design (SEED) method is proposed to sequentially generate new sample points. Simulated annealing [64] and Bayesian methods [65] have been used to generate optimal sampling points in a sequential order.

3.3.1 Classical Experimental Design

Choice of an experimental design for metamodeling procedure depends on several factors including the number of control variables, types of variables, the degree of nonlinearity present in the underlying process, and accuracy required in the metamodel. In general, the metamodeling is an iterative process. First, an experimental design is selected, data are collected, a model is built, and results are analyzed. Then, the design is augmented based on the results and the desired performance. This procedure is explained in Chapter 5, in which a case study is used to verify the methodology.

Factorial or fractional-factorial, central composite design (CCD), and Box-Behnken design (BBD) are the commonly used classical experimental designs.

3.3.1.1 Fractional/factorial Design

Many experiments involve the study of the effects of two or more control factors on the response variable. In general, factorial designs are most efficient for this

type of experiment and make efficient use of the experimental resources. This is an experimental strategy in which the factors are varied together, instead of one at a time. The factorial experimental design concept enables the experimenter to investigate not only the individual effects of each factor (called main effects) but also interactions among the factors. The number of runs required in the experiment depends on the number of factors and the number of levels for each factor. Generally, if there are k factors, each at two levels, the two-level factorial design will require 2^k runs. For example, a 10-factor experiment with all factors at two levels will require 1024 runs. Since there are only two levels, only linear effects can be estimated. Theoretically, there can be any number of levels in a factorial design; however, in practice, it is limited to two levels. This is because the number of experiments required in the case of a high-level design makes it infeasible.

The two-level design is mostly used in the initial stage of the metamodeling procedure. At this stage, the primary focus is on reducing the number of control factors that are significant for the next stage of experiments. This stage is called screening, during which the assumption of linearity is often reasonable. As the number of factors in 2^k factorial design increases, the number of runs rapidly outgrows the resources of most experimenters. However, most of these runs are used to estimate only high-order interactions. For example, in a 6 factor two-level design which needs $2^6 = 64$ runs, only $\frac{6!}{5!1!} = 6$ of the 63 degrees of freedom correspond to main-effects, $\frac{6!}{4!2!} = 15$ degrees of freedom correspond to two-factor interactions, while the remaining 42 degrees of freedom are associated with three-factor or higher interactions. In most of the cases, it is reasonable to assume that certain high-order interactions are negligible. In that case, information on main effects and low order interactions may be obtained by running only a fraction of full factorial experiments. These are called fractional-factorial designs and are mostly used in screening experiments when large number of factors are involved.

Table 3.1: An example of resolution III design.

Design: 2^{**}(11-7), Resolution III											
Run	A	B	C	D	E	F	G	H	I	J	K
1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	-1	1	-1	-1	-1	-1	1	1
3	1	1	-1	1	-1	-1	-1	1	-1	1	-1
4	1	1	-1	-1	-1	1	1	-1	1	1	-1
5	1	-1	1	1	-1	-1	1	-1	-1	-1	1
6	1	-1	1	-1	-1	1	-1	1	1	-1	1
7	1	-1	-1	1	1	1	-1	-1	1	-1	-1
8	1	-1	-1	-1	1	-1	1	1	-1	-1	-1
9	-1	1	1	1	-1	1	-1	-1	-1	-1	-1
10	-1	1	1	-1	-1	-1	1	1	1	-1	-1
11	-1	1	-1	1	1	-1	1	-1	1	-1	1
12	-1	1	-1	-1	1	1	-1	1	-1	-1	1
13	-1	-1	1	1	1	-1	-1	1	1	1	-1
14	-1	-1	1	-1	1	1	1	-1	-1	1	-1
15	-1	-1	-1	1	-1	1	1	1	-1	1	1
16	-1	-1	-1	-1	-1	-1	-1	-1	1	1	1

3.3.1.2 Resolution in fractional-factorial designs

A technical description of how fractional factorial designs are constructed is beyond the scope of this introduction. Detailed accounts of how to design $2^{(k-p)}$ experiments can be found, for example, in [19, 66]. In general, they successively "use" the highest-order interactions to generate new factors. For example, consider the following design that includes 11 factors but requires only 16 runs (observations). The following discussion aims at introducing the concepts of confounding and resolution, which are important from the perspective of choosing the experimental design

The design in Table 3.1 is described as a $2_{III}^{(11-7)}$ design of resolution *III* (three). This means that overall $k = 11$ factors (the first number in parentheses) are studied; however, $p = 7$ of those factors (the second number in parentheses) are generated from the interactions of a full $2^{(11-7)} = 2^4$ fractional-factorial design. As a result, the design does not give full resolution; that is, there are certain interaction effects that

are confounded with (identical to) other effects. In general, a design of resolution R is the one in which no l -way interactions are confounded with any other interaction of order less than $R - l$. In the current example, resolution (R) is equal to 3. Here, no $l = 1$ level interactions (i.e., main effects) are confounded with any other interaction of order less than $R - l = 3 - 1 = 2$. Thus, no main effects in this design are confounded with any other interactions of order less than 2. This means that some of the main effects interactions are confounded with two-way interactions; and consequently, all higher-order interactions are equally confounded. If 64 runs are included to generate a $2^{(11-5)} = 2^6$ design, the resultant resolution would have been $R = IV$ (four). Thus, no $l=1$ -way interaction (main effect) is confounded with any other interaction of order less than $R - l = 4 - 1 = 3$. In this design, main effects are not confounded with two-way interactions, but confounded with three-way and higher-order interactions. What about the two-way interactions? No $l=2$ -way interaction is confounded with any other interaction of order less than $R-l = 4-2 = 2$. Thus, the two-way interactions in that design are confounded with each other.

3.3.1.3 Central Composite Design (CCD)

The practical deployment of a central composite design (CCD) often arises through sequential experimentation, i.e., a two level full-factorial or fractional-factorial design has been used to fit a first-order model, this model exhibits a lack of fit, and the axial runs are then added to allow the quadratic terms to be incorporated into the model. The CCD is a very efficient design for fitting the second-order model. Generally, the CCD consists of a 2^k factorial or fractional-factorial design with n_F runs, $2k$ axial or star runs, and n_C center runs. Figure 3.4 shows three well-known types of CCD with $k = 2$ factors, $n_F = 4$ (factorial runs), $2k = 4$ axial runs, and 1 center run. Blue squares represent the tests run in a two-level full factorial design, whereas red stars show the axial runs and green circle shows the center run added as part of CCD.

Therefore, using CCD approach allowed us to get a quadratic model from a linear model just by adding 5 runs.

Central composite circumscribed (CCC) designs are the original form of the central composite design. The star points are at distance α from the center based on the properties desired for the design and the number of factors in the design. The star points establish new extremes for the low and high settings for all factors. These designs have circular, spherical, or hyperspherical symmetry depending upon the number of factors involved, and require 5 levels for each factor. Augmenting an existing factorial or fractional-factorial design with star points, can produce this design.

In central composite face-centered (CCF) design, the star points are at the center of each face of the factorial space, so $\alpha = \pm 1$. This variety requires 3 levels of each factor. Augmenting an existing factorial or fractional-factorial design with appropriate star points can also produce this design.

For those situations in which the limits specified for factor settings are truly limits, the central composite inscribed (CCI) design uses the factor settings as the star points and creates a factorial or fractional-factorial design within those limits (in other words, a CCI design is a scaled down CCC design with each factor level of the CCC design divided by alpha to generate the CCI design). This design also requires 5 levels of each factor.

3.3.1.4 Box-Behnken Designs (BBD)

The Box-Behnken design is an independent quadratic design in that it does not contain an embedded factorial or fractional factorial design. In this design, the sample points are at the midpoints of edges of the process space and at the center. These designs are rotatable (or near rotatable) and require 3 levels of each factor. The geometry of this design suggests a sphere within the process space such that the surface of the sphere protrudes through each face with the surface of the sphere tangential

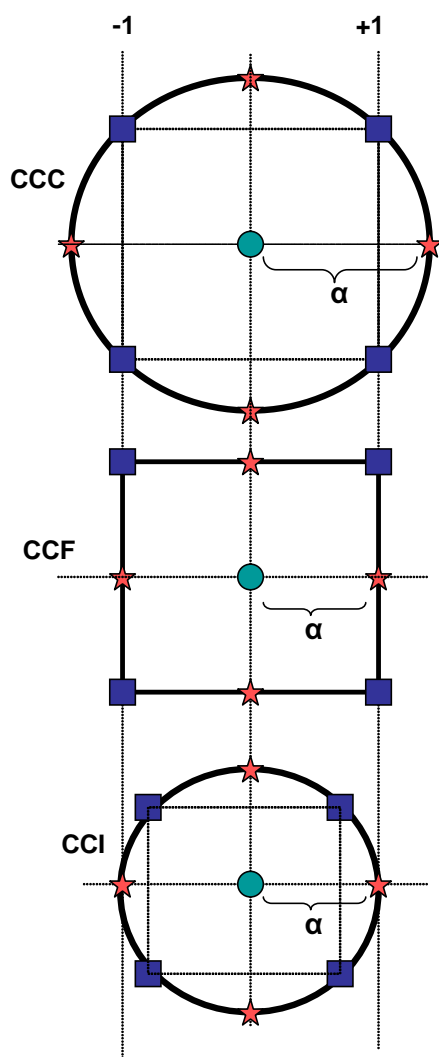


Figure 3.4: Types of central composite designs (CCD).

to the midpoint of each edge of the space. In general, for 3 factors or less, the Box-Behnken design offers some advantage in requiring a fewer number of runs. For 4 or more factors, this advantage disappears.

3.4 Polynomial Regression Modeling Using Method of Least Squares

Polynomial models estimate and predict the shape of response values over a range of input parameter values. Polynomial models are a great tool for determining which input factors drive responses and in what direction. These are the most common models used for analysis of designed experiments. A quadratic (second-order) polynomial model for two control factors has the form of the equation below.

$$y = b_0 + b_1x_1 + b_2x_2 + b_{12}x_{12} + b_{11}x_1^2 + b_{22}x_2^2 \quad (3.4.1)$$

In this equation, b_0 is called the intercept, b_1 and b_2 are called the main effects, b_{12} is called the interaction effect, and b_{11} and b_{22} are called the quadratic effects.

In the DOE terminology, effects mean that how changing the settings of a factor changes the response. The first-order effect of a single factor is called a main effect, while the second-order effect is called a quadratic effect. Interaction effects occur when the effect of one factor on the response depends on the level of another factor (second-order interaction) or factors (higher-order interactions).

Coefficients of these effects in the polynomial model are determined using the least square (LS) estimation by finding numerical values for the parameters that minimize the sum of the squared deviations between the observed responses and the functional portion of the model.

3.4.1 Estimation of the parameters in regression models using least squares

The method of least squares is typically used to estimate the regression coefficients in a regression model given as the following:

Table 3.2: Data for linear regression.

y	x₁	x₂	x_k
y ₁	x ₁₁	x ₁₂	x _{1k}
y ₂	x ₂₁	x ₂₂	x _{2k}
.
.
y _n	x _{n1}	x _{n2}	x _{nk}

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \epsilon \quad (3.4.2)$$

This model is called a multiple linear regression model with k variables. Suppose that $n > k$ observations on the response variable are available, say y_1, y_2, \dots, y_n . Along with each observed response y_i , there is an observation on each control factor, and let x_{ij} denotes the i th observation or level of variable x_j . Table 3.2 shows the data layout.

Assuming that the error term ϵ in the model has $E(\epsilon) = 0$ and $Var(\epsilon) = \sigma^2$ and that the $\{\epsilon_i\}$ are uncorrelated random variables, the model equation may be written in terms of observations in Table 3.2 as,

$$y = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_k x_{ik} + \epsilon_i$$

that is,

$$y = \beta_0 + \sum_{j=1}^k \beta_j x_{ij} + \epsilon_i, \quad i = 1, 2, \dots, n \quad (3.4.3)$$

The method of least squares chooses the coefficients in the above equation so that the sum of squares of the errors, ϵ_i , are minimized. The least square function is,

$$L = \sum_{i=1}^n \epsilon_i^2 = \sum_{i=1}^n (y_i - \beta_0 - \sum_{j=1}^k \beta_j x_{ij})^2 \quad (3.4.4)$$

The function L is to be minimized with respect to $\beta_0, \beta_1, \dots, \beta_k$. The least square estimators, say b_0, b_1, \dots, b_k are found to be,

$$\mathbf{b} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \quad (3.4.5)$$

$$\text{where } \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1k} \\ 1 & x_{21} & x_{22} & \dots & x_{2k} \\ \vdots & \vdots & \vdots & & \vdots \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{nk} \end{bmatrix}, \mathbf{b} = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_k \end{bmatrix}, \text{ and } \boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

In matrix notation, the fitted regression model is

$$\hat{\mathbf{y}} = \mathbf{X}\mathbf{b} \quad (3.4.6)$$

In scalar notation, the fitted model is

$$\hat{y}_i = b_0 + \sum_{j=1}^k b_j x_{ij}, \quad i = 1, 2, \dots, n \quad (3.4.7)$$

3.4.2 Orthogonality in Response Surface Models

Like other metamodeling methods, RSM also consists of two components, i.e. experimental design and model fitting. The experimental design component of RSM is based on classical DOE methods, while the model fitting is based on least square

methods. This combination of classical DOE and least square methods helps in minimizing the variances of the regression coefficients. A key concept that makes RSM superior to the ordinary least square regression is that of “orthogonality”.

A first-order orthogonal design is the one for which $\mathbf{X}'\mathbf{X}$ is a diagonal matrix. This definition implies that the columns of \mathbf{X} are mutually orthogonal. In other words, if we write

$$\mathbf{X} = [\mathbf{1}, \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k] \quad (3.4.8)$$

where \mathbf{x}_j is the j th column of \mathbf{X} , then a first-order orthogonal design is such that $\mathbf{x}_i'\mathbf{x}_j \neq 0$ for all $i = j$, and $\mathbf{1}\mathbf{x}_j = 0$ for $j = 1, 2, \dots, k$. If two columns are orthogonal, the levels of the two corresponding variables are linearly independent. The implication is that the roles of the two variables are being assessed independent of each other. This underscores the virtues of orthogonality [66]. It means that RSM is not only helpful in analyzing the effects of input variables on the response variables but it also helps in collecting the right data that minimizes the variance of regression coefficients. This characteristics distinguishes the RSM from other regression methods.

Chapter IV

CASE STUDY: COMPONENT-LEVEL AND SUBSYSTEM-LEVEL MODELING

In Chapter 2, the system-level health assessment methodology was developed for a general case. In this chapter and the next chapter, a gas turbine engine of turbofan type is used as a case study. Figure 4.1 shows a block diagram of this methodology. The component-level and subsystem-level modules (shown in red circle) are implemented in this chapter. These modules are subsequently used by the system-level metamodeling procedure, and will be discussed in the next chapter.

4.1 Hierarchical Architecture of a Gas Turbine Engine

The main components of a gas turbine engine are compressor, combustion chamber and turbine, connected together. To obtain a thermal efficiency, a high-pressure ratio is essential. One way of achieving a high-pressure ratio is to divide the compressor into two or more sections [67]. When the compressors are mechanically independent, each will require its own turbine. The low-pressure compressor is driven by the low-pressure turbine and the high-pressure compressor by the high-pressure turbine. This configuration is usually referred to as twin-spool engine. In the case of a turbofan engine, fan is also considered to be part of the engine. Therefore at the subsystem-level, the engine is constructed of the following:

1. Fan
2. Low-pressure compressor (LPC)
3. High-pressure compressor (HPC)

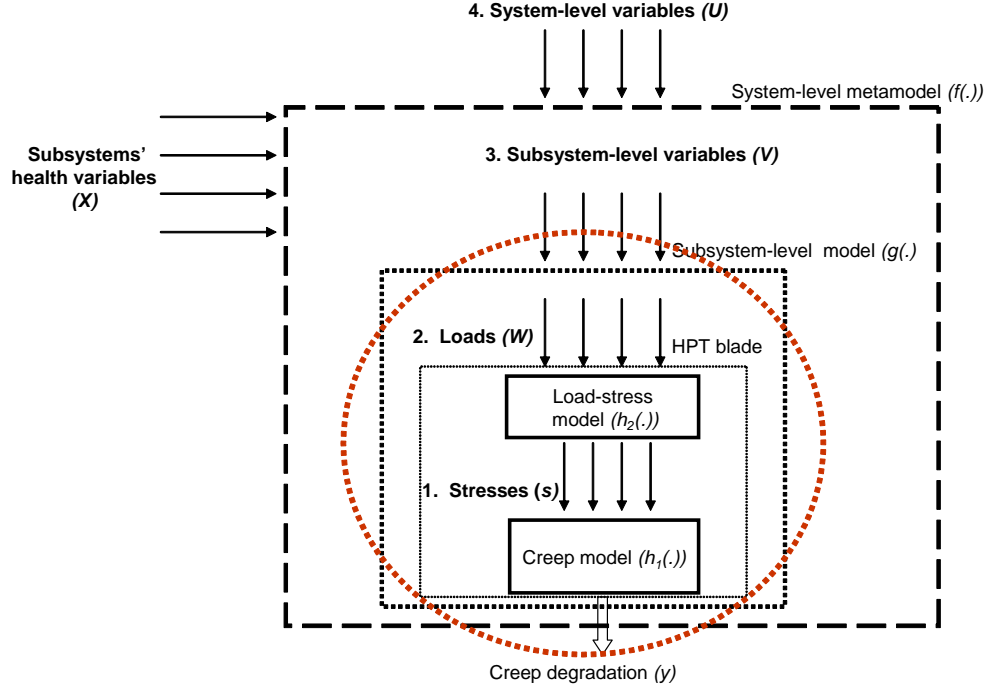


Figure 4.1: The modules implemented in Chapter 4 are shown inside the circle.

4. High-pressure turbine (HPT)

5. Low-pressure turbine (LPT)

Each of these subsystems is further composed of several components. It means that the architecture of a gas turbine engine can be viewed as consisting of three layers; system-level, subsystem-level, and component-level (Figure 4.2).

System-level layer is the topmost layer of the architecture. It is followed by subsystem-level layer and component-level layer, respectively. The system-level layer consists of engine itself, the subsystem-level layer consists of five subsystems (fan, LPC, HPC, HPT, and LPT), and the component-level layer consists of components in each of these subsystems. It is obvious that as we move down in the hierarchy, the number of constituent elements increase. For example, the system-level layer has only one element (engine), the subsystem-level layer has five elements, and the component-level layer consists of hundreds of elements.

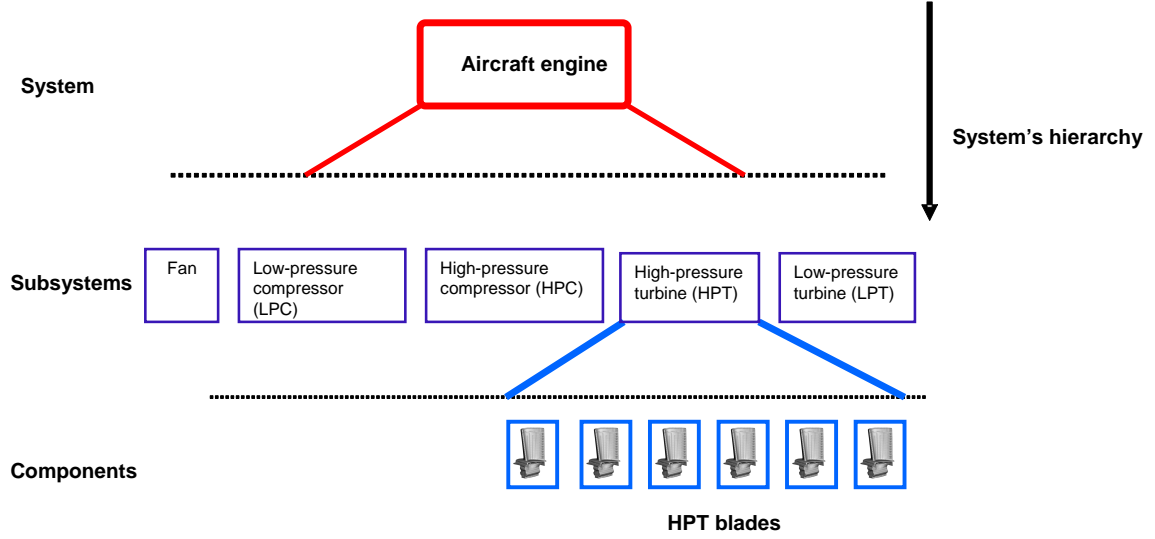


Figure 4.2: Hierarchical architecture of a gas turbine engine.

4.2 Failures and Degradation in Gas Turbine Engines

In a physical system, most of the faults that are manifested as system-level failures, are initiated at the component-level, and a gas turbine engine is no exception. In the engine, there are a large number of components, each of which can have multiple failure modes. Furthermore, each failure mode is a product of many failure mechanisms that are simultaneously active. In short, there can be a large number of failure scenarios in the engine. Ideally, a system-level health assessment methodology should take all these possibilities into consideration. However, in most of the practical cases, it is not possible to cover all of these cases.

Blades in the high-pressure turbine (HPT) are exposed to an extremely harsh environment, and are one of the most critical component in the system. In this thesis, it is assumed that the remaining life of HPT blades is a representative of the remaining life of the entire engine (for a general case, these assumptions are discussed in Section 2.5).

At this point, it is important to mention the term “degradation”, which has been used in this dissertation to explain those processes that cause a reduction in the

engine's performance but are not critical to cause failure. The most important mechanisms that cause degradation in engines are compressor fouling, turbine erosion, and corrosion [68]. In this work, it is assumed that, though there can be many degradation mechanisms going on in the engine, they are not directly causing the engine failure. In other words, these degradation mechanisms affect the life consumption rate of the HPT blades but are not a direct cause of the engine failure.

Generally, the following failure mechanisms are active in a turbine blade [69]:

1. Creep: As shown in Figure 4.3, at a constant stress, the strain initially increases swiftly with time (primary or transient deformation), then increases more slowly in the secondary region at a steady rate (creep rate). Finally a fast increase in strain leads to failure in the tertiary region. In this work, it is assumed that the largest portion of the blade's life is consumed in the secondary region at a constant creep rate.

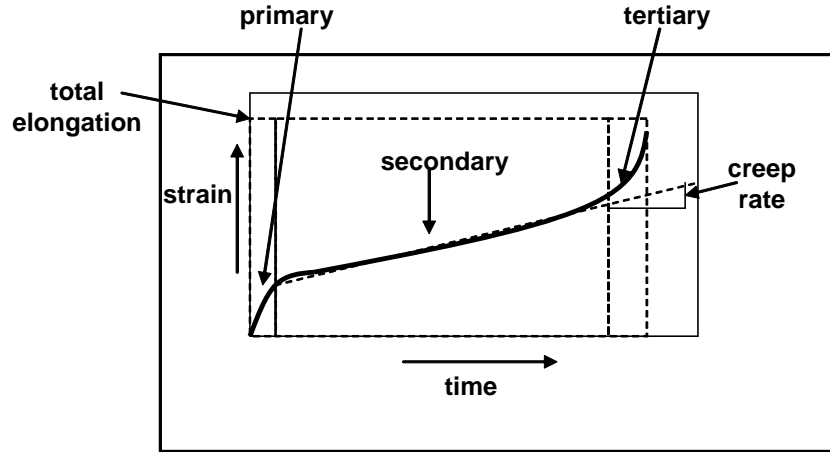


Figure 4.3: Notional creep behavior.

1. Low-cycle fatigue: This is caused by the imposition of varying loads (and hence stresses) upon a component of an aero-engine. If these are high enough, they will lead to failure of the component, even though the maximum stress is lower than the static strength of the material [70].

2. Oxidation : This is the reaction between the blade materials and the oxidants present in the hot gases from the combustion process. This effect varies greatly depending on the material composition and operating temperature [71].
3. Hot corrosion : Surface reactions with salts deposited in the vapor phase, gradually erode away at metallic or coated surfaces [72]. It is one of the most severe environments faced by man made materials, which over time leads to degradation of the aerodynamic performance of the blades.

With the use of suitable coatings, oxidation and hot corrosion can be suitably controlled [69]. However, the problems with creep and LCF persist, especially when the operating temperature approaches the metal melting point and the blade rotation causes considerable stresses.

In the present work, a commercial aircraft engine is being used as an application example. In the case of a commercial aircraft engine, creep is considered to have a dominant role. Therefore, it is assumed that creep is the only failure mechanism that is active. The effects of the other failure mechanisms are not included in the present work.

In the above discussion, many assumptions are stated, and are summarized as follows:

System-related assumptions

1. The gas turbine engine can fail in a multiple ways. It is being assumed in this work that the HPT blades will fail before than any other component of the engine. Hence, the remaining life of HPT blades is a representative of engine's remaining life.
2. It is also assumed that, although there can be many degradation mechanisms active in the engine, but they are not directly causing the engine failure. In

other words, these degradation mechanisms affect the life consumption rate of the HPT blades but are not a direct cause of the engine failure.

Failure mechanism-related assumptions

1. It is assumed that creep is the only failure mechanism that is responsible for the blade failure.
2. It is also assumed that the largest portion of the blade's life is consumed in the secondary region at a constant creep rate.
3. It is assumed that the order of loading does not have an effect on the creep rate.

4.3 Failure Mechanism Modeling

As discussed in the previous section, the HPT blades are assumed to be the life-limiting component of the engine, and creep is assumed to be the only failure mechanism that is active. In Chapter 2, damage accumulation models were classified as one of types of failure mechanism models. This type of model provides the accumulated damage/RUL estimates based on a priori knowledge of life expectancy and of expected usage patterns. This prediction approach does not involve damage monitoring or diagnosis information. Damage accumulation models are particularly useful during the design phase of the prognostic system, when the field data are not available. These models have three components.

1. Damage model
2. Damage accumulation rule
3. Acceptable limits of damage accumulation

4.3.1 Damage (creep) Model

Creep is a time dependent, thermally assisted deformation of components under load (stress) [73]. As a consequence of such deformation, unacceptable dimensional changes and distortions as well as final rupture of the components can occur. Depending on the component, the final failure may be limited either by deformation or by fracture.

Creep life is highly sensitive to Von Mises stresses and the blade temperature [47]. According to the terminology used in this dissertation, Von Mises stresses and temperature are the stress factors (s). It is worth mentioning here that in many other disciplines, e.g. solid mechanics, the term "stress factor" is used categorically for the parameters that measure the intensity of the internal forces acting between particles of a deformable body [36]. For example [69], the following three types of stresses are combined together as Von Mises stresses:

1. gas bending stresses (due to the bending moments in the blade due to the aerodynamic loads),
2. centrifugal stresses (due to solid objects rotating at high speeds),
3. thermal stresses (due to the difference in temperature across the metal).

In our thesis, the term "stress factor" is used in general, for any force that drives the failure mechanism. As a rule of thumb, for every decrease in material temperature of 10°-15° C, the available creep life is doubled [74]. For example, in the case of GTD111, which is an advanced super alloy widely used as a turbine material, at 150Mpa stress, decreasing the material temperature from 900°C to 850°C causes the creep life to increase from 10,000 hours to 100,000 hours. In this case, with 50°C decreasing in temperature, creep life increases to 10 times of the original one.

Creep properties are generally determined by means of a test in which a constant uniaxial load or stress is applied and the resulting strain is recorded as a function of

time. Figure 4.3 showed three stages of creep deformation under a constant loading. Creep deformation increases progressively in the primary stage, then stabilizes at a near constant creep damage accumulation rate in the secondary stage, and increases rapidly in tertiary stage till the creep rupture occurs. In this work, it is assumed that the largest portion of the blade's life is consumed in the secondary region at a constant creep rate (ϵ_{creep}).

[47] uses a simple creep model to estimate creep remaining life assessment,

$$\epsilon_{creep} = \int_0^t \beta \sigma^2, \quad (4.3.1)$$

where ϵ_{creep} is creep strain, β is a temperature-dependent constant, σ are the Von Mises stresses, and t is time over which the loading conditions continued. This model translates the stress factors (Von Mises stresses and temperature) into component degradation.

4.3.2 Damage Accumulation Rule

The model in Equation 4.3.1 calculates creep at constant levels of stress factors. In the actual application, however, their levels vary. The simplest, and the most common way to account for a variable loading, is by using a life-fraction model, also known as Robinson's rule, accumulation of creep rule [75], and Palmgren-Miner rule [76]. This rule is also particularly convenient, since it allows taking into account, degradation from the combined effects of creep and fatigue [77]. The life-fraction rule for variable loading assumes that at any time interval, accumulated damage is independent of the damage at other time intervals. Mathematically, it can be expressed as:

$$\sum_{i=1}^m \frac{t_i}{L_i} = D, \quad (4.3.2)$$

t_i = time interval at constant loading for i^{th} case ,

L_i = Creep life at constant loading,

D = material dependent constant ([78],[77]) (usually taken to be 1, as in this case),
 m = number of intervals at constant loading.

4.3.3 Creep Failure Criteria

In remaining creep life assessments, one of the following failure criteria may be used [47]:

1. Creep elongation: Typical numbers of creep strain as this failure criteria are from 0.2% to 1%. It is a conservative failure criterion, and usually there is still significant creep life remaining in a component when these criteria have been reached. ([79]).
2. Creep rupture strain: This criterion will yield much longer remaining life compare to using creep elongation [47]. Usually, the majority of a component's creep life is consumed while reach this failure criteria. However, most creep analysis Software such as ANSYS can not model creep till this criterion is reached. Thus, this criterion should not be used unless certain fidelity correction factor can be addressed.
3. Crack growth criteria. Once a crack initiated, the break of the component will not be far away.

The failure mechanism model used in this work is taken from [47], in which creep elongation is used as the creep failure criterion. It is assumed that the entire blade life is consumed when the creep strain reaches 0.5%.

Both stress factors in the creep model (Von Mises stresses and blade temperature) are result of some type of forces acting on the component. These forces are termed as load-variables in this dissertation and are translated into stress factors by using load-stress model.

4.4 Load-stress Model

Several tools can be used to model the effects of load variables on the stress factors. An example of such a tool is finite-element analysis (FEA). In this analysis, the structure is modeled as group of tiny elements meshed together. The inputs to this model are the geometry definition of the structure, its material properties, and the boundary conditions applied on the structure. According to the terminology adopted in this dissertation, these inputs are load variables (W). The output of FEA can be the resulting stress or creep strain if the creep model is included in FEA. It means that the two models, i.e., the failure mechanism model and the load-stress model (FEA) can be combined to determine the effect of the load variables on the creep strain, while the stress factors as the intermediate variable.

In [47], eight variables are identified as the potential load variables, and response surface methodology/design of experiments (RSM/DOE) is used to approximate the relationship between these load variables and creep life using creep model. These load variables are the following:

1. temperature of hot gas flow
2. temperature of cooling flow
3. blade rotational speed
4. diameter of cooling hole
5. Young's modulus
6. thermal conductivity
7. thermal expansion at 0 degree to grain axis
8. specific heat

In [47], a series of experiments, based on finite-element methods, is performed to develop a quadratic RSM of the form:

$$R = b_0 + \sum_{i=1}^k b_i x_i + \sum_{i=1}^k \sum_{j=i+1}^{k-1} b_{ij} x_i x_j + \sum_{i=1}^k b_{i,i} x_i^2,$$

where R is the response (creep life) and x_i 's are the load variables.

The response is defined as \log (creep life at a constant loading) for better fitness to the complex, high order true function as follows:

$$\log(R) = 5.3211 - 1.6939(w_1) - 1.3007(w_3). \quad (4.4.1)$$

In this equation, w_1 is the temperature of hot gas and w_3 is the blade rotational speed. The model in Equation 4.4.1 has been represented in the architecture shown in Figure 4.1 as $h_2(\cdot)$.

The inputs used in the load-stress model are component-level parameters. Any variation in these parameters occurs due to varying subsystem-level variables.

4.5 *Subsystem-level Model*

Subsystem-level model relates the subsystem-level variables (V) into the creep life, using the load variables (W). From Equation 4.4.1, component-level variables that determine the creep life of the HPT blade are hot gas temperature w_1 and blade rotational speed w_3 . At the subsystem-level, the source of hot gas temperature is the turbine-entry-temperature (TET), and the source of blade rotational speed is the HPT core speed, denoted by v_1 and v_2 respectively. Since the blade rotates at the same speed as the turbine, therefore $v_2 = w_3$. The hot gas temperature of the blade, depends not only on TET, but also on many other factors. However, it is assumed in this work that the blade temperature is uniformly distributed, and has the same value as that of TET, i.e., $v_1 = v_2$. With this assumption, the component-level model in Equation 4.4.1, can be translated into subsystem-level model as:

$$\log(R) = 5.3211 - 1.6939(v_1) - 1.3007(v_2). \quad (4.5.1)$$

In this equation, v_1 is the turbine-entry-temperature (TET) and v_2 is the HPT core speed.

This model has been represented in the architecture shown in Figure 4.1 as $g(.)$.

Chapter V

CASE STUDY: SYSTEM-LEVEL METAMODELING OF A GAS TURBINE ENGINE

As mentioned in the previous chapters, the objective of this work is to develop a health assessment methodology for a complex system, whose structure is arranged in three layers. The component-level layer is the lowest layer in this hierarchy. It is true that most of the faults are initiated at this level. However, system operators, managers, and maintenance-related personnel, interact with the system through the system-level parameters. For many critical components in different application domains, failure prognostic models have been developed. However, a gap does exist between the component-level prognostic modules and system-level applications. To make use of these prognostic results and build a system-level health assessment methodology that can assist the decision makers, this gap needs to be filled. Therefore, a metamodeling procedure is developed in this work in such a way that this model is connected to component-level parameters on the bottom and system-level parameters at its top.

In the previous chapter, the development procedure of a subsystem-level metamodel $g(.)$ that relates the effects of subsystem-level operating conditions on the component degradation was discussed (shown in red circle in Figure 5.1). In this chapter, the development procedure of a system-level metamodel $f(.)$, using the subsystem-level metamodel $g(.)$, is explained. Figure 5.1 shows the block diagram of the system-level metamodel, in which there are two types of inputs.

1. Mission profile parameters (U) (e.g. aircraft speed, altitude).
2. Subsystems' health variables (X) representing degradation in various subsystems.

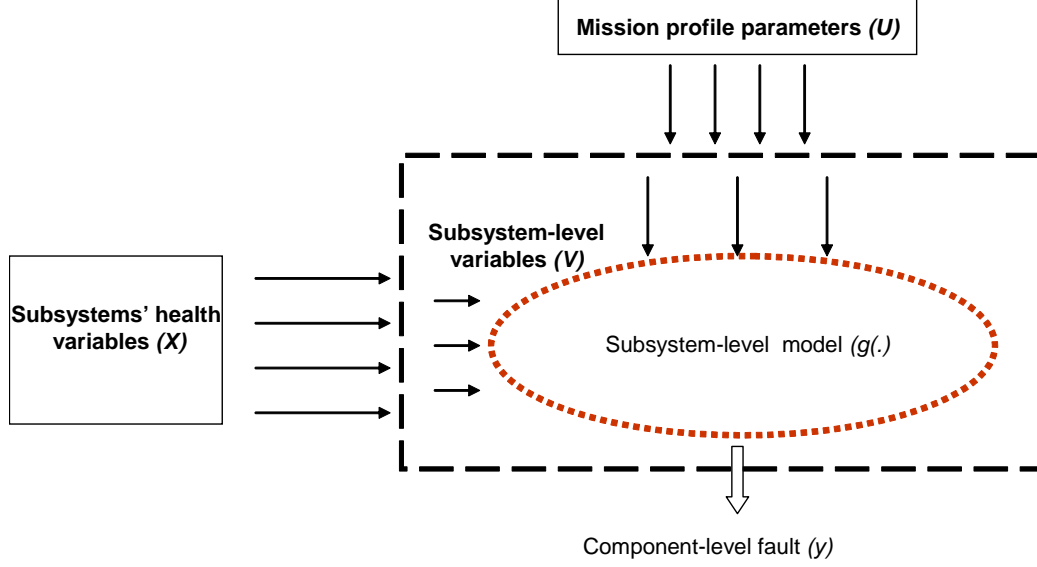


Figure 5.1: Block diagram of development procedure of system-level metamodel using subsystem level model.

Exploring the effects of the types of mission profile on the subsystem-level operating conditions require that they are represented by variables that are capable of characterizing a variety of mission profiles. These variables have been termed as mission profile parameters in this dissertation. Similarly, health of engine subsystems also needs to be represented in terms of certain parameters, which have been termed as subsystems' health parameters.

A system-level metamodel can be built using several approaches. In this dissertation, response surface methodology (RSM) is used for this purpose. In general, there are four main steps in building a response surface model.

1. Identifying the input space: In this step, first the input factors to be studied are selected and then the respective range of interest is identified. In our application, these factors are the mission profile parameters (U) and subsystems' health variables (X).
2. Sampling the input space according to the experimental plan: Sample points are chosen in the input space in order to investigate the relationship between the

input factors and the response variable, in an efficient manner. These samples are chosen based on statistical theory of design of experiments (DOE), as discussed in Chapter 3. Then, physical experiments or simulations are performed at each of the sample points, and data are generated.

3. Building the predictive model: Using the data gathered in step 2, a polynomial model is constructed using the method of least squares.
4. Analyzing results: The model is analyzed to verify that its performance meets the required criteria.

In some cases, the number of variables involved in the model are too large. Therefore, experiments are performed to “screen out” the less significant effects.

Figure 5.2 shows the sequence of steps followed in building the system-level response surface model.

The simulation platform that is used in generating the data is C-MAPSS (commercial modular aero-propulsion system simulation) [80]. The simulation output consists of subsystem-level parameters (V), which are then fed into the subsystem-level model given by Equation 4.5.1, which calculates the response variable, i.e., creep degradation.

5.1 Simulation Platform

A recently released C-MAPSS [80] (Commercial Modular Aero Propulsion System Simulation) is used as simulation platform. C-MAPSS is well suited for our application because of its following features.

- a) It is a high-fidelity model.
- b) It allows input variations of health related parameters.
- c) It allows recording of a large number of sensor measurements.

C-MAPSS is a tool for simulating a realistic large commercial turbofan engine.

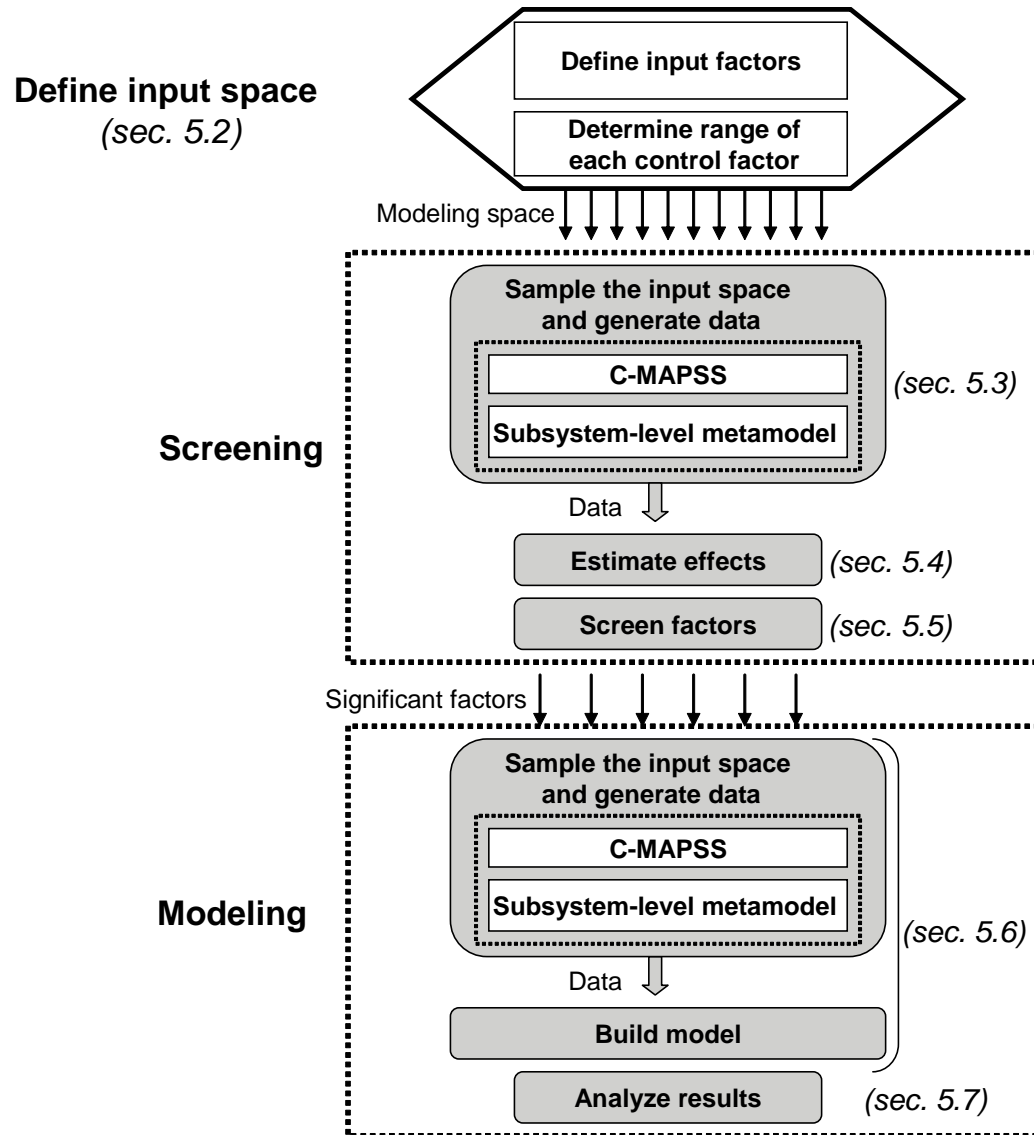


Figure 5.2: Block diagram of the system-level response surface metamodeling procedure.

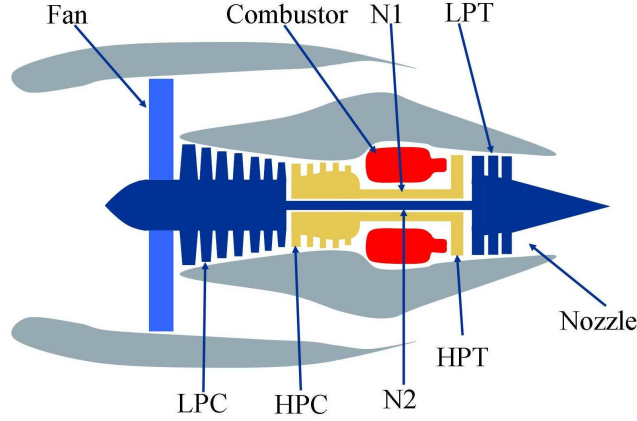


Figure 5.3: Subsystem-level construction of a turbofan engine.

The software is a combination of Matlab and Simulink (The Mathworks, Inc.) with a number of editable fields. In addition to the engine model of 90,000 lb thrust, the package includes an atmospheric model capable of operation at (i) altitudes from sea level to 40,000 ft, (ii) mach numbers from 0 to 0.90, and (iii) sea level temperatures from -60 to 103 °F, and (iv) a wide range of thrust levels throughout the full range of operating conditions. C-MAPSS has 14 inputs that include fuel flow and a set of 13 health-parameters inputs, which can be used to simulate the effects of faults and deterioration in any of the engine's five subsystems, i.e., fan, low-pressure compressor (LPC), high-pressure compressor (HPC), high- pressure turbine (HPT) and low-pressure turbine (LPT), as shown in Figure 5.3.

5.2 *Defining the input space*

The system-level metamodel aims at exploring the effects of two types of variables on the system's health.

1. Those variables that can characterize various types of mission profiles. Such variables have been termed as mission profile parameters and are denoted by U in this work.
2. The other type of variables are those that represent the degradation in the

engine's subsystems. These variables have been termed as subsystems' health parameters and are denoted by X in this work.

The input space for the system-level metamodel is constituted by these two types of variables. For each of these types, the input space is defined in two steps.

- a) Select the parameters that can represent various types of mission profiles and subsystems' health.
- b) Define the range of each parameter.

5.2.1 Parametrization of Mission Profiles

A mission profile is a detailed description of an aircraft's flight path and its in-flight activities [81]. These profiles are broken into more specialized segments known as phases, which focus on specific flight operations. Figure 5.4 shows various phases of a typical flight cycle of a commercial aircraft [82]. In Figure 5.4, the climb phase consists of an initial climb segment and a climb segment. In the case of creep, the degradation is not sensitive to instantaneous values, and depends on the average values of the parameters over the duration of the phase of a flight cycle. Hence, in this dissertation, a single climb phase is included by assuming average values of these segments. Similarly, a single descent phase is included as a representative of descent, initial approach and final approach segments. In this way, the flight cycle shown in Figure 5.4 a) is simplified as consisting of the following five phases, as shown in Figure 5.4 b).

1. Takeoff
2. Climb
3. Cruise
4. Descent

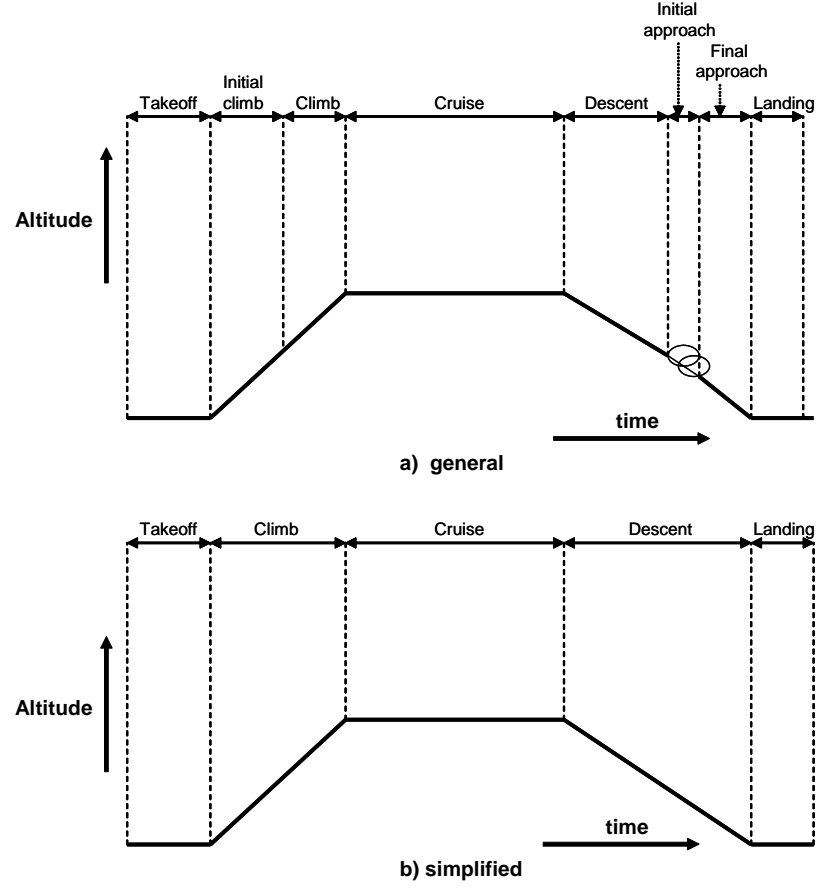


Figure 5.4: Phases of a typical flight cycle of a commercial aircraft.

5. Landing

Figure 5.5 gives an idea of the relative effect of all these phases on the creep damage, assuming nominal parameters.

From this figure, it can be readily observed that the contribution of the landing phase to the creep damage is quite insignificant, as compared to the other phases of the flight cycle. Although this figure represents creep damage during a nominal flight cycle, it can be stated that the landing phase can cause only a small creep damage even in the worst-case scenario. This is due to the short duration and extremely low power requirements of this phase of the flight cycle. Hence, the landing phase is not considered in the further modeling procedure. Therefore, the mission profile is represented as a combination of the four phases, i.e., takeoff, climb, cruise, and

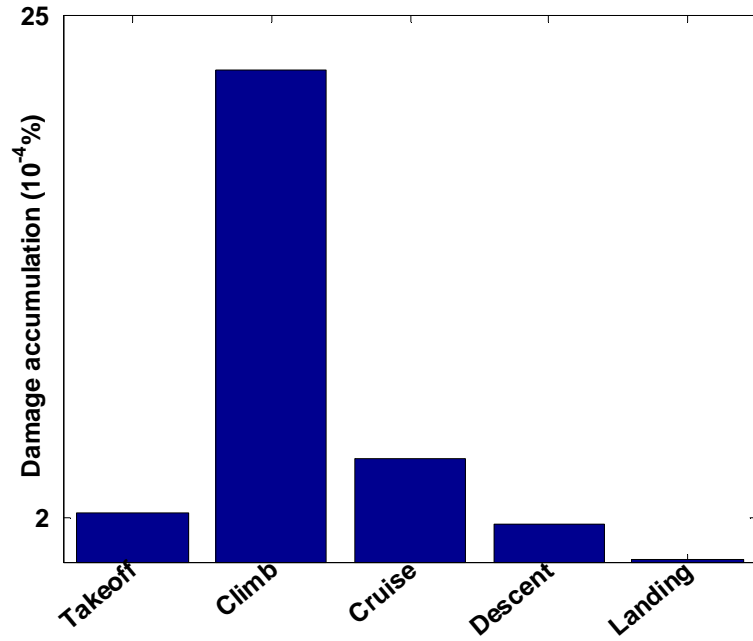


Figure 5.5: Damage accumulation during various phases of a nominal flight cycle.

descent. Each of these phases is represented as a set of parameters. It is assumed that the takeoff phase can be characterized by the sea-level temperature, takeoff speed, and throttle-resolver angle (TRA). Similarly, the climb phase is characterized by the average values of rate of climb (ROC), average mach speed of the aircraft, average TRA, and the final altitude after which aircraft enters into cruise phase. In all, the four phases are represented by 14 parameters as shown in Figure 5.6.

Once the mission profile parameters have been defined, the next step is to determine the range of each of these parameters over which the aircraft operates during its service life.

5.2.2 Defining the Range of Mission Profile Parameters

During each phase, the mission profile parameters (u_i) vary over a wide range. This range depends on the type of the aircraft, besides other factors. Although a wide range of these parameters can be used to cover a variety of aircraft types, the accuracy of the metamodel suffers as the range of the control factors increases. Table 5.1 shows

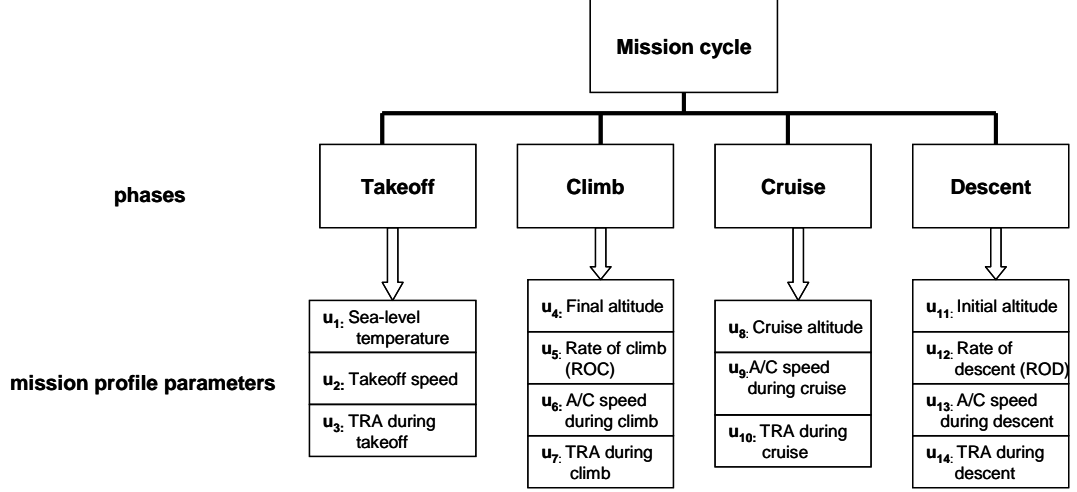


Figure 5.6: Mission profile parameters used to characterize mission of a commercial aircraft.

the range of the mission profile parameters used in this work, representing ballpark figures for a commercial aircraft.

5.2.3 Parametrization of Subsystem Health

Generally, faults and deterioration in turbine engines are modeled by adjusting independent parameters e.g. efficiency, flow, pressure-ratio, associated with each subsystem. These faults may have serious consequences [83] such as loss of throttle control, compressor stalls, aborted takeoffs, in-flight shutdowns, and the like. These adjustments tend to shift the engine performance away from nominal. As yet, precise performance parameter changes due to typical faults in the engine's components and their inter-relationships are not known [70]. However, some researchers [70, 83] have defined empirical 'fouling index' and 'erosion indicex' to describe the effects of changes in efficiencies and flow capacities of the compressors and turbines, respectively.

The fouling-index (FI) is an empirical parameter combining the adverse effects upon the engine's performance of reductions in (i) flow capacity and (ii) efficiency of any gas-path component. It is assumed that a 1% decrease of efficiency, accompanied by a 0.5% reduction in flow capacity, results in a 1 %FI. Any deterioration in the

Table 5.1: Ranges of the mission profile parameters.

Operation Phase	Factor	Symbol	Notation	Unit	Range	
					Low	High
Takeoff	Sea-level temperature	Tsl	u_1	F°	30	100
	Takeoff speed	Speed _{TO}	u_2	<i>Mach</i>	0.2	0.3
	Thrust resolver angle during takeoff	TRA _{TO}	u_3	<i>deg.</i>	80	100
Climb	Final altitude	Alt _{final}	u_4	<i>ft</i>	20000	38000
	Rate of climb	RoC	u_5	<i>ft/min</i>	1300	2200
	A/C speed during climb	Speed _{climb}	u_6	<i>Mach</i>	0.45	0.75
	Throttle resolver angle during climb	TRA _{climb}	u_7	<i>deg.</i>	72	88
Cruise	Cruise altitude	Alt _{cruise}	u_8	<i>ft</i>	20000	38000
	A/C speed during cruise	Speed _{cruise}	u_9	<i>Mach</i>	0.6	0.75
	Throttle resolver angle during cruise	TRA _{cruise}	u_{10}	<i>deg.</i>	56	73
Descent	Initial altitude	Alt _{initial}	u_{11}	<i>ft</i>	20000	38000
	rate of descent	RoD	u_{12}	<i>ft/min</i>	1000	3000
	A/C speed during descent	Speed _{descent}	u_{13}	<i>Mach</i>	0.5	0.7
	Throttle resolver angle during descent	TRA _{descent}	u_{14}	<i>deg.</i>	0	40

health of a fan, low-pressure compressor (LPC), and high-pressure compressor (HPC) is modeled by using this parameter. As these subsystems undergo deterioration in their healths, their respective *FIs* increase. It is assumed that over the entire lifespan of the engine, *FIs* increase from 0% (new engine) to 8% (old engine).

In the case of turbines, i.e., low-pressure turbine (LPT) and high-pressure turbine (HPT), deterioration is modeled by an empirical parameter which has been termed as erosion-index (EI). It combines the adverse effects upon the engine's performance of (i) an increase in flow capacity and (ii) a reduction in efficiency of the turbines [83]. It is assumed that a 1% reduction of efficiency accompanied by a 0.1% increase in flow capacity, results in a 1% EI. It is assumed that during the entire lifespan of the engine, *EIs* increase from 0% (new engine) to 8% (old engine).

Table 5.2 shows the parameters (x_1, x_2, x_3, x_4, x_5) used to model the health of the engine's subsystems. While using these parameters, it is assumed that the deteriorations in all of the engine's subsystems (fan, LPC, HPC, HPT, and LPT) can be modeled. The next step is to estimate the effects of any of these deteriorations on

Table 5.2: Subsystems' health parameters and their definitions.

Subsystem	Health parameter	Notation	Changes in performance parameters for 1% reduction in health parameter	
			efficiency	flow
Fan	Fouling index	x_1	1%	0.50%
LPC	Fouling index	x_2	1%	0.50%
HPC	Fouling index	x_3	1%	0.50%
HPT	Erosion index	x_4	1%	-0.10%
LPT	Erosion index	x_5	1%	-0.10%

the creep degradation.

5.2.4 Defining the Range of Subsystem Health Parameters

As discussed in the previous section, precise performance parameter changes due to deterioration in the engine's subsystems are not known. Therefore, empirical parameters are used to describe the effects of deterioration in the subsystems on changes in efficiencies and flow capacities. Different researchers have varied these parameters over different ranges. For example, [83] varied the range of these parameters from 1.5% to 6%, while [70] used range from 1% to 10% for the same parameters. In this work, the range of subsystems' health parameters is varied from 0 to 8% (Table 5.3).

The subsystems' health parameters (x_1, x_2, x_3, x_4, x_5) might affect creep degradation in a complex manner; for example, the mission profile parameters can interact with these parameters, the subsystems might interact with each other. In general, it is true that deterioration is a slow process. Therefore, it is reasonable to assume constant values of subsystems' health parameters during the entire cycle. Under this assumption, a single simulation can do the job for the full cycle. In this way, however, only the overall effect of the subsystems health parameters during the cycle can be

Table 5.3: Range of subsystems' health parameters.

Subsystem	Health parameter	Notation	Range	
			low (healthy)	high (deteriorated)
Fan	Fouling index	x_1	0%	8%
LPC	Fouling index	x_2	0%	8%
HPC	Fouling index	x_3	0%	8%
HPT	Erosion index	x_4	0%	8%
LPT	Erosion index	x_5	0%	8%

estimated. It is possible that these effects are strongly dependent on the flight phases. Therefore, separate experiments are carried out for each phase at its center point. The results are subsequently combined as a cycle. Hence, there are 20 subsystems' health parameters (5 for each phase) and 14 mission profile parameters, as potential input variables in the system-level metamodel. It means that, in total, there are 34 variables that need to be investigated. The number of experiments required to build a second or higher-order response surface equation (RSE) depends on the number of variables. Therefore, screening experiments are carried out before the modeling experiments to reduce the set of factors to those that are relatively significant with respect to creep degradation. Screening procedure, like the other metamodeling procedures, has two parts, i.e., experimental design and model fitting.

5.3 Design of Experiments (DOE) for Screening

In a typical screening test, a two-level fractional/factorial design is used to construct a linear RSE, in which only main effects are considered. Screening based on a linear RSE, however, can result in an erroneous elimination of a variable if the true function between response and the control variable is non-monotonic (see Figure 3.3). To avoid such an elimination, any higher-order design that can provide the quadratic effects of

the control variables need to be used.

Central composite designs (CCD) use a two-level factorial or fraction-factorial design combined with axial or star points that allow estimation of curvature. Section 3.3.1.3 discusses the well-known types of CCD used in practice. In general, choice of a CCD is not of prime importance during the screening stage, unless the same design augmented by additional sample points is to be used in the subsequent modeling stages. In the present case, the screening procedure of the mission profile parameters (u_i, s) is carried out separately from that of the subsystems' health parameters (x_i, s) . Hence, choice of a specific type of CCD is not extremely significant in this case.

5.3.1 Experimental Design for Mission Profile Parameters Screening

During the screening stage of mission profile parameters, a central composite inscribed (CCI) design, which consists of 2^k two-level full-factorial design augmented by $2k$ star-runs and one center-run, is used. Figure 5.7 shows a CCI design for 3 control factors, i.e, $k = 3$.

Tables 5.4, show the design matrix used for screening the takeoff parameters and the observed response values. Similar matrices are used for screening the climb, cruise, and descent phases, and are included in Appendix A.

5.3.2 Experimental Design for Subsystems' Health Parameters Screening

The experimental design used for screening subsystems' health parameters is given in Table 5.5. In the case of mission profile parameters screening, the factorial component was two-level full-factorial which requires 2^k factorial runs. In the case of subsystems' health screening, however, a 2^{k-1} fractional-factorial design is used to reduce the number of runs. For $k = 5$ (number of subsystems), this design requires 16 runs. A full-factorial design, if used, would have required 32 runs. Most of these runs, however, are not useful for our purpose since they are used in estimating 3rd order and 4th

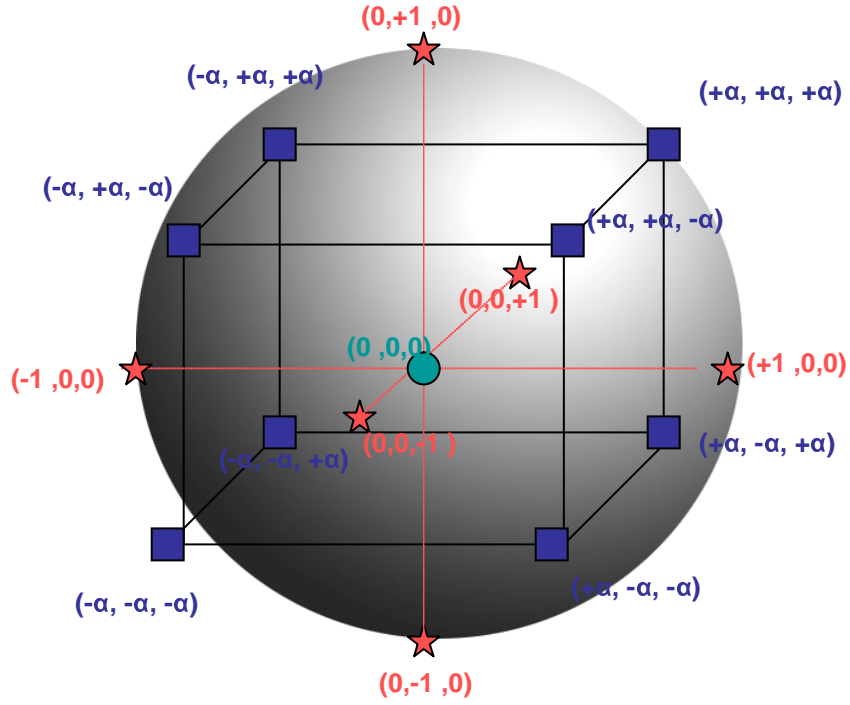


Figure 5.7: A central composite inscribed (CCI) design for 3 control factors.

Table 5.4: Design matrix for screening takeoff parameters and the observed values.

Trial No.	Levels of i/p factors (coded)			Response (creep degradation 10^{-4} %)
	Tsl	Speed _{TO}	TRA _{TO}	
	u_1	u_2	u_3	
1	-0.6	-0.6	-0.6	0.6084
2	0.6	-0.6	-0.6	3.4018
3	-0.6	0.6	-0.6	0.9333
4	0.6	0.6	-0.6	5.236
5	-0.6	-0.6	0.6	0.7806
6	0.6	-0.6	0.6	4.5093
7	-0.6	0.6	0.6	1.354
8	0.6	0.6	0.6	7.937
9	-1	0	0	0.5458
10	1	0	0	8.443
11	0	-1	0	1.302
12	0	1	0	3.144
13	0	0	-1	1.6167
14	0	0	1	2.835
15	0	0	0	2.2

order interactions. The design generator or defining relation used for this fractional-factorial part is $I = +ABCDE$, which is a resolution V design, i.e., no main effect or two-factor interaction is aliased with any other main effect or two-factor interaction, but two-factor interactions are aliased with three-factor interactions. Table 5.5 shows a design based on this defining relation, in which the first column (x_1) is written as the product of the other columns:

$$\begin{aligned} A &= AI = +A^2BCDE = BCDE \\ \Rightarrow x_1 &= (x_2)(x_3)(x_4)(x_5) \end{aligned}$$

The response values in the table are calculated at the center point of each phase. For example, the center point of the cruise phase is at,

$$\begin{aligned} Alt_{cruise} &= 30,000 ft, \\ Speed_{cruise} &= 0.67 Mach, \\ TRA_{cruise} &= 65 deg. \end{aligned}$$

Once the response values have been obtained, the next step is to estimate the effects of the mission profile parameters and subsystems health parameters by using least-square regression analysis.

5.4 Building Models for Screening

5.4.1 Effects Estimation of the Mission Profile Parameters

For each of the flight phase, least square regression analysis is used to fit a second-order model of the form given in the Equation 5.4.1.

$$y = b_0 + b_1u_1 + b_2u_2..... + b_ku_k + b_{1,1}u_1^2 + + b_{k,k}u_k^2 \quad (5.4.1)$$

In this model, y represents the creep damage ($10^{-4}\%$), while the values of the coefficients b'_i s represents the main effects of the mission profile parameters u'_i s, and the coefficients $b'_{i,i}$ s represents the quadratic effects of u_i . Table 5.6 shows the coefficient values of these parameters obtained after performing the regression analysis.

Table 5.5: Design matrix for screening subsystems' health parameters during each phase.

Trial No.	Subsystems' health deterioration (coded)					Response (creep degradation 10^{-4} %)			
	Fan	LPC	HPC	HPT	LPT	Mission Phase			
	$x_1=(x_2)(x_3)(x_4)(x_5)$	x_2	x_3	x_4	x_5	Takeoff	Climb	Cruise	Descent
1	1	-1	-1	-1	-1	2.636	24.355	5.027	1.742
2	-1	1	-1	-1	-1	2.388	22.976	4.821	1.737
3	-1	-1	1	-1	-1	4.418	42.135	7.992	1.953
4	1	1	1	-1	-1	5.710	51.735	9.343	1.997
5	-1	-1	-1	1	-1	3.908	38.114	7.574	1.933
6	1	1	-1	1	-1	5.028	46.341	8.708	1.975
7	1	-1	1	1	-1	4.581	70.775	15.771	2.390
8	-1	1	1	1	-1	4.586	68.981	15.143	2.378
9	-1	-1	-1	-1	1	3.109	29.528	5.920	1.785
10	1	1	-1	-1	1	4.145	36.449	6.919	1.814
11	1	-1	1	-1	1	7.467	67.055	11.557	2.084
12	-1	1	1	-1	1	6.708	63.004	11.050	2.074
13	1	-1	-1	1	1	5.324	58.257	10.340	2.057
14	-1	1	-1	1	1	5.315	55.323	9.889	2.046
15	-1	-1	1	1	1	4.658	77.568	17.938	2.507
16	1	1	1	1	1	4.673	84.901	20.865	2.604
17	-1	0	0	0	0	4.641	43.645	8.176	1.959
18	1	0	0	0	0	6.447	60.459	10.488	2.060
19	0	-1	0	0	0	4.995	47.853	8.698	1.991
20	0	1	0	0	0	5.993	54.763	9.626	2.020
21	0	0	-1	0	0	5.255	49.421	8.904	1.997
22	0	0	1	0	0	5.685	53.033	9.412	2.015
23	0	0	0	-1	0	3.859	36.286	6.890	1.863
24	0	0	0	1	0	6.046	70.329	12.496	2.198
25	0	0	0	0	-1	4.136	38.459	7.342	1.876
26	0	0	0	0	1	4.940	63.557	12.333	2.189
27	0	0	0	0	0	5.463	51.145	9.144	2.006

Table 5.6: Coefficient values (coded) of mission profile parameters obtained from screening experiments.

Takeoff			Climb			Cruise			Descent		
Term	Coeff.	Estimate	Term	Coeff.	Estimate	Term	Coeff.	Estimate	Term	Coeff.	Estimate
u_1	b_1	3.76	u_4	b_4	-0.47	u_8	b_8	-4.12	u_{11}	b_{11}	-0.45
u_2	b_2	1.13	u_5	b_5	-3.90	u_9	b_9	0.10	u_{12}	b_{12}	0.60
u_3	b_3	0.79	u_6	b_6	7.17	u_{10}	b_{10}	2.42	u_{13}	b_{13}	-0.34
$(u_1)^2$	$(b_{1,1})$	2.34	u_7	b_7	10.35	$(u_8)^2$	$(b_{8,8})$	1.89	u_{14}	b_{14}	0.63
$(u_2)^2$	$(b_{2,2})$	0.06	$(u_4)^2$	$(b_{4,4})$	2.82	$(u_9)^2$	$(b_{9,9})$	0.09	$(u_{11})^2$	$(b_{11,11})$	-0.30
$(u_3)^2$	$(b_{3,3})$	0.07	$(u_5)^2$	$(b_{5,5})$	2.38	$(u_{10})^2$	$(b_{10,10})$	0.41	$(u_{12})^2$	$(b_{12,12})$	-0.20
			$(u_6)^2$	$(b_{6,6})$	5.64				$(u_{13})^2$	$(b_{13,13})$	0.23
			$(u_7)^2$	$(b_{7,7})$	0.17				$(u_{14})^2$	$(b_{14,14})$	0.16

Table 5.7: Coefficient values (coded) of subsystems' health parameters obtained from screening experiments.

Takeoff			Climb			Cruise			Descent		
Term	Coeff.	Estimate	Term	Coeff.	Estimate	Term	Coeff.	Estimate	Term	Coeff.	Estimate
x_{1a}	c_{1a}	0.35	x_{1b}	c_{1b}	3.28	x_{1c}	c_{1c}	0.58	x_{1d}	c_{1d}	0.02
x_{2a}	c_{2a}	0.19	x_{2b}	c_{2b}	1.60	x_{2c}	c_{2c}	0.31	x_{2d}	c_{2d}	0.01
x_{3a}	c_{3a}	0.63	x_{3b}	c_{3b}	12.13	x_{3c}	c_{3c}	2.83	x_{3d}	c_{3d}	0.16
x_{4a}	c_{4a}	0.20	x_{4b}	c_{4b}	10.95	x_{4c}	c_{4c}	2.73	x_{4d}	c_{4d}	0.17
x_{5a}	c_{5a}	0.50	x_{5b}	c_{5b}	7.32	x_{5c}	c_{5c}	1.39	x_{5d}	c_{5d}	0.07
$(x_{1a})^2$	$(c_{1a,1a})$	0.21	$(x_{1b})^2$	$(c_{1b,1b})$	0.43	$(x_{1c})^2$	$(c_{1c,1c})$	0.18	$(x_{1d})^2$	$(c_{1d,1d})$	0.01
$(x_{2a})^2$	$(c_{2a,2a})$	0.16	$(x_{2b})^2$	$(c_{2b,2b})$	-0.32	$(x_{2c})^2$	$(c_{2c,2c})$	0.01	$(x_{2d})^2$	$(c_{2d,2d})$	0.00
$(x_{3a})^2$	$(c_{3a,3a})$	0.13	$(x_{3b})^2$	$(c_{3b,3b})$	-0.40	$(x_{3c})^2$	$(c_{3c,3c})$	0.00	$(x_{3d})^2$	$(c_{3d,3d})$	0.00
$(x_{4a})^2$	$(c_{4a,4a})$	-0.38	$(x_{4b})^2$	$(c_{4b,4b})$	1.68	$(x_{4c})^2$	$(c_{4c,4c})$	0.54	$(x_{4d})^2$	$(c_{4d,4d})$	0.03
$(x_{5a})^2$	$(c_{5a,5a})$	-0.80	$(x_{5b})^2$	$(c_{5b,5b})$	-0.62	$(x_{5c})^2$	$(c_{5c,5c})$	0.68	$(x_{5d})^2$	$(c_{5d,5d})$	0.03

The screening decision is made after the effects estimation procedure is performed for the subsystems' health parameters as well.

5.4.2 Effects Estimation of Subsystem Health Parameters

For screening purpose, the effects of subsystem health parameters on the creep degradation are estimated by least-square regression analysis. The same experimental design is used for each phase and the values of coefficients are given by the following model:

$$y_n = c_{0,n} + c_{1n}x_{1n} + \dots + c_{5n}x_{5n} + c_{1n,1n}(x_{1n})^2 + \dots + c_{5n,5n}(x_{5n})^2, \quad (5.4.2)$$

where y_n represents the creep damage during the n_{th} phase and c_{jn} represents the main effects of the health of the j_{th} subsystem during the n_{th} phase. Since there are 5 subsystems, so j can take on values from 1, 2, 3, 4, 5 for fan, LPC, HPC, HPT, and LPT respectively. Similarly n can be a, b, c, d representing takeoff, climb, cruise and descent phases, respectively. Table 5.7 shows the coefficient values of these parameters obtained after performing the regression analysis.

Once the main effects and the quadratic effects have been estimated, the insignificant parameters can be screened out.

5.5 Parameter Screening

Table 5.6 and 5.7 show that the coefficient values of most of the parameters are relatively small. Screening out these parameters simplifies the model, without having a significant loss of accuracy.

As discussed in Section 3.2, the screening decision should not be based solely on the main/linear effects. If the true relation between response and a variable is non-monotonic, a wrong decision to eliminate that variable may be made, if based only on the linear effect. Hence, the quadratic effect should also be considered before a variable is eliminated.

Figure 5.8 shows the screening process as a flowchart. The main effect estimates of all the mission profile parameters and subsystem health parameters are compared to a threshold value. One way to choose this value is based on the maximum number of parameters required to be included in the further modeling stages or the desired accuracy of the final model. If the main effect estimates are less than this threshold, their respective quadratic effects are compared to the threshold value. If the quadratic effects are also less than the threshold, then the parameter is screened out for the next modeling stage.

Figure 5.9 (top) shows the estimates of main effects as a barplot. In this plot, a coefficient value of 1.2 is set as the threshold limit. This value means that an average change of $1.2 \times 10^{-4}\%$ in creep degradation is caused by a variation in the parameter from 0 (mean value) to 1 (maximum value). There are 6 parameters ($u_1, u_5, u_6, u_7, u_8, u_{10}$) whose coefficients ($b's$) are greater than this limit, while coefficient values of 8 of these parameters ($u_2, u_3, u_4, u_9, u_{11}, u_{12}, u_{13}, u_{14}$) are lower than the limit. Before screening out these parameters, the quadratic effects of these 8 parameters (Figure 5.9 (bottom)) show that 7 of these parameters can be screened out while u_4 is included in the next modeling stage. It means that out of 14 mission profile parameters, 7 ($u_2, u_3, u_9, u_{11}, u_{12}, u_{13}, u_{14}$) are screened out, and the other 7

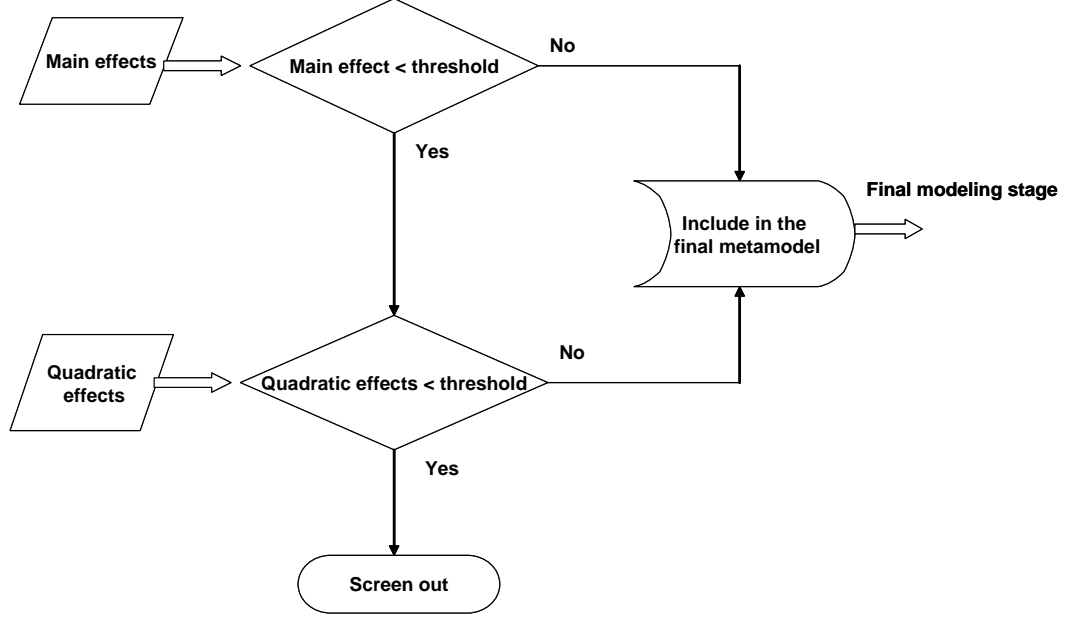


Figure 5.8: Flowchart of the screening process.

parameters ($u_1, u_4, u_5, u_6, u_7, u_8, u_{10}$) are included in the system-level metamodel.

The same procedure is used for screening of subsystem health parameters. As discussed in Section 5.4.2, coefficients of these parameters are estimated separately for each phase. Figure 5.10 shows the estimates of main effects (top) and quadratic effects (bottom). Using the same threshold limit as used for screening mission profile parameters (1.2), it is found that during phase *a* (takeoff) and *d*(descent), none of the subsystems has a significant effect on the creep damage. During phase *b* (climb), all the five subsystems are significant, while during phase *c* (cruise), only HPC, HPT, and LPT have a significant effect on creep damage. It means that out of 20 subsystems' health parameters, 12 are screened out.

Summarizing the screening results during takeoff phase (Table 5.8), only one mission profile parameter, i.e., sea-level temperature, and none of the subsystems' health parameter is significant. During climb phase, all the mission profile parameters and all the subsystems' health parameters are significant. During cruise phase, two mission profile parameters (cruise altitude and TRA) and three subsystems' health parameters

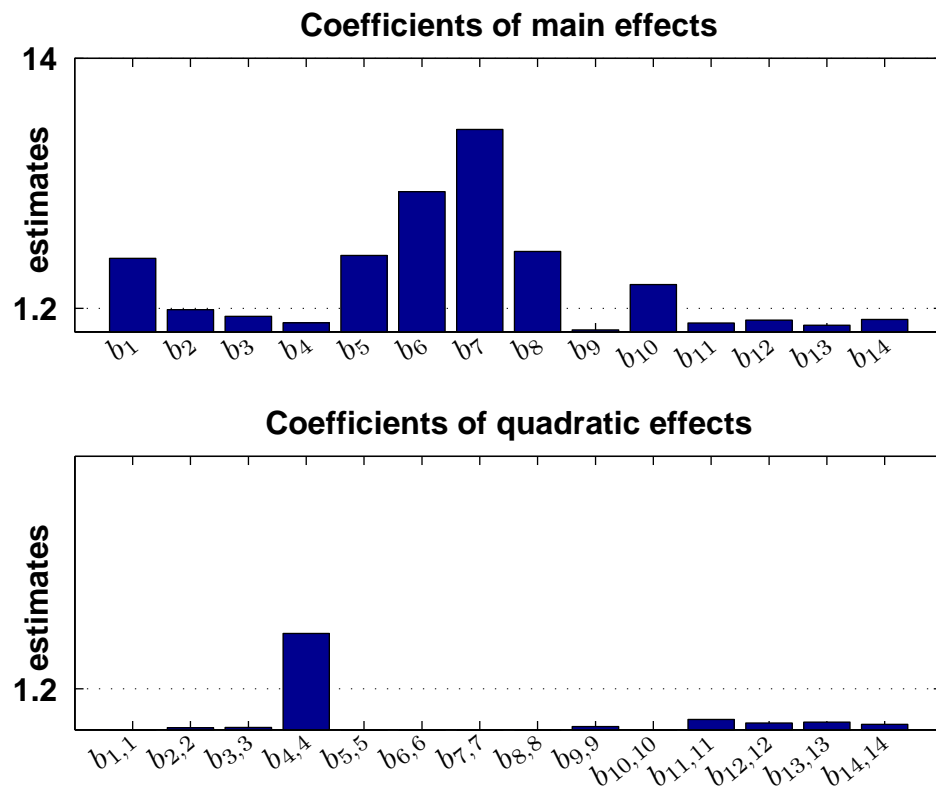


Figure 5.9: Bar plot of estimates of mission profile parameters.

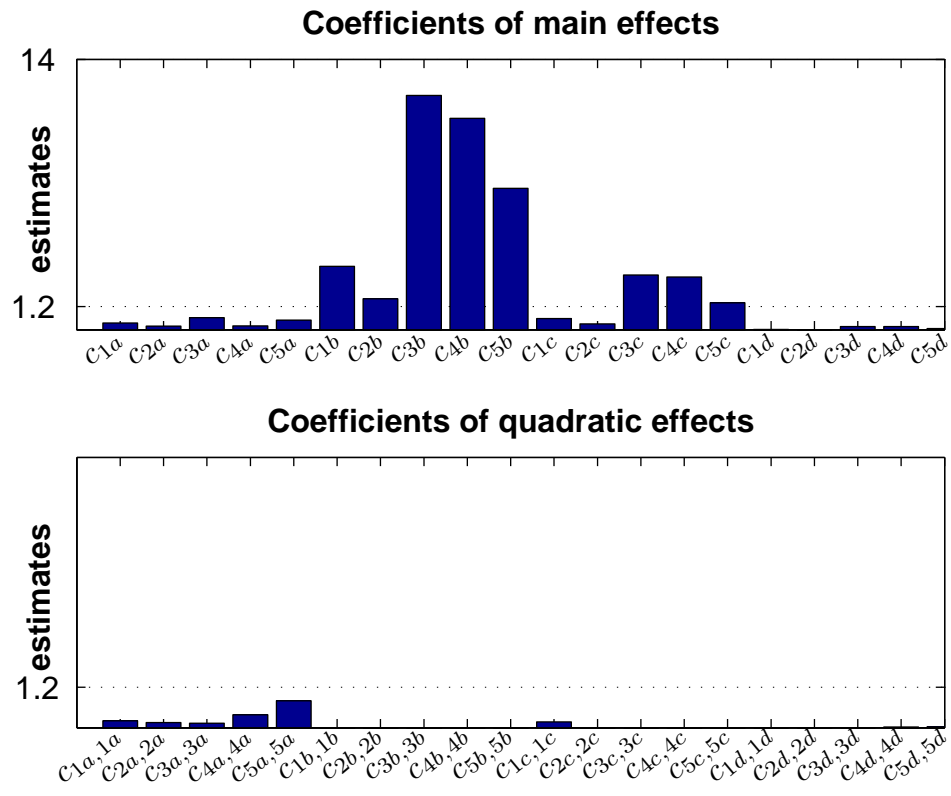


Figure 5.10: Bar plot of estimates of subsystems's health parameters.

Table 5.8: Summary of significant parameters.

Takeoff		Climb		Cruise		Descent	
Type of parameter		Type of parameter		Type of parameter		Type of parameter	
Mission profile	Subsystems' health	Mission profile	Subsystems' health	Mission profile	Subsystems' health	Mission profile	Subsystems' health
Sea-level temp. (u_1)	—	Alt _{final} (u_4)	Fan (x_{1b})	Alt _{cruise} (u_8)	HPC (x_{3c})	—	—
		ROC (u_5)	LPC (x_{2b})	TRA _{cruise} (u_{10})	HPT (x_{4c})		
		Speed _{climb} (u_6)	HPC (x_{3b})		LPT (x_{5c})		
		TRA _{climb} (u_7)	HPT (x_{4b})				
			LPT (x_{5b})				

(HPC, HPT, LPT) are significant. Descent phase is excluded from the system-level metamodeling, since none of the mission profile parameters or subsystems' health parameters is significant during this phase.

5.6 Modeling

During the screening stage, the mission profile parameters and subsystems' health parameters are considered separately to reduce the number of experiments. These parameters, however, might have a significant interaction with each other, and, therefore, are considered together in the system-level metamodeling stage. As shown in Table 5.8, there are 15 parameters that are to be included in the system-level metamodel. This metamodel can be constructed by either using a single design for the entire flight cycle using all the 15 parameters or as a combination of metamodels for each phase. Since the subsystem-level metamodel is a linear damage-summation model, the parameters in different phases do not interact with each other. Hence, the metamodel for each of the three phases (descent phase is screened out), i.e., takeoff, climb, and cruise are built separately and then added together:

$$y_{mission} = y_{takeoff} + y_{climb} + y_{cruise} \quad (5.6.1)$$

where,

$$y_{takeoff} = f_{takeoff}(u_1)$$

$$y_{climb} = f_{climb}(u_4, u_5, u_6, u_7, x_{1b}, x_{2b}, x_{3b}, x_{4b}, x_{5b})$$

$$y_{cruise} = f_{cruise}(u_8, u_{10}, x_{3c}, x_{4c}, x_{5c})$$

5.6.1 System-level Metamodel for Takeoff Phase

From Table 5.8, there is a single mission profile parameter and no subsystems' health parameter for takeoff phase. Since there is no interaction present in the system-level metamodel for this phase, the metamodel for takeoff phase can be determined by using the data from Table 5.4 as:

$$y_{takeoff} = 2.98 + 3.759u_1 \quad (5.6.2)$$

5.6.2 System-level Metamodel for Climb Phase

From Table 5.8, the system-level metamodel for the climb phase includes 9 parameters. To build this metamodel, first an experimental design is selected, and then the model is built using the data obtained from this design.

5.6.2.1 Experimental design

To generate the data for building the system-level metamodel for the climb phase, central composite circumscribed (CCC) design is used as the experimental design. CCC designs provide high quality predictions over the entire design space but require factor settings outside the range of the factors in the factorial part. In the case of mission profile parameters, setting the factors outside their ranges is not an issue but in the case of subsystems' health parameters the low range of fault indices is 0%, i.e., healthy. Hence, this requirement of CCC design cannot be met in the case of factor setting of subsystems' health parameter, and the low range of fault indices is still set at 0%. This setting will cause in a loss in the prediction accuracy of this model but only for the healthy subsystems. CCC is still chosen as the experimental design since it gives a better prediction accuracy for the remaining design space. The other types

of CCD designs have their own weak points. For example, CCI designs use only points within the factor ranges originally specified, but do not provide the same high quality prediction over the entire space compared to the CCC. CCF designs offer relatively high quality predictions over the entire design space and do not require using points outside the original factor range. However, they give poor precision for estimating the quadratic coefficients [62].

A CCD has 3 parts, i.e., a fractional/factorial design, $2k$ star-runs and one center-run. For $k = 9$, a full-factorial design has 512 runs. Most of these runs are used to estimate high-order (3rd order and higher) interactions. Generally, higher-order interactions are not significant and can be ignored. For $k = 9$, a simpler design that consist of a small number of runs (64 or fewer) can be used, but it will be of resolution *IV*. In such designs, two-factor interactions will be aliased with other two-factor interactions. Therefore, a one-fourth fractional factorial design 2_{VI}^{9-2} that consists of 128 runs is used. It is a resolution *VI* design and can estimate all the main effects and second-order interactions without any aliasing with any other two-factor interaction. In this design, the first 7 columns make a full-factorial design and the last two columns have the generators,

$$H = +ACDFG \text{ and } J = +BCEFG.$$

The design has $2k = 18$ axial-runs for the quadratic effects 1 center-run. The design and the response values are given in Appendix B.

5.6.2.2 *Model building*

After gathering the data from the CCC design, least-square regression analysis is used to build a response surface model (RSM). This model includes the main effects, interaction effects and quadratic effects.. Since there are 9 factors, the model has 9 main effects, 36 two-factor interactions, and 9 quadratic effects. In total, there are 55 estimates (given in Appendix C) in this model. It is observed that estimate values

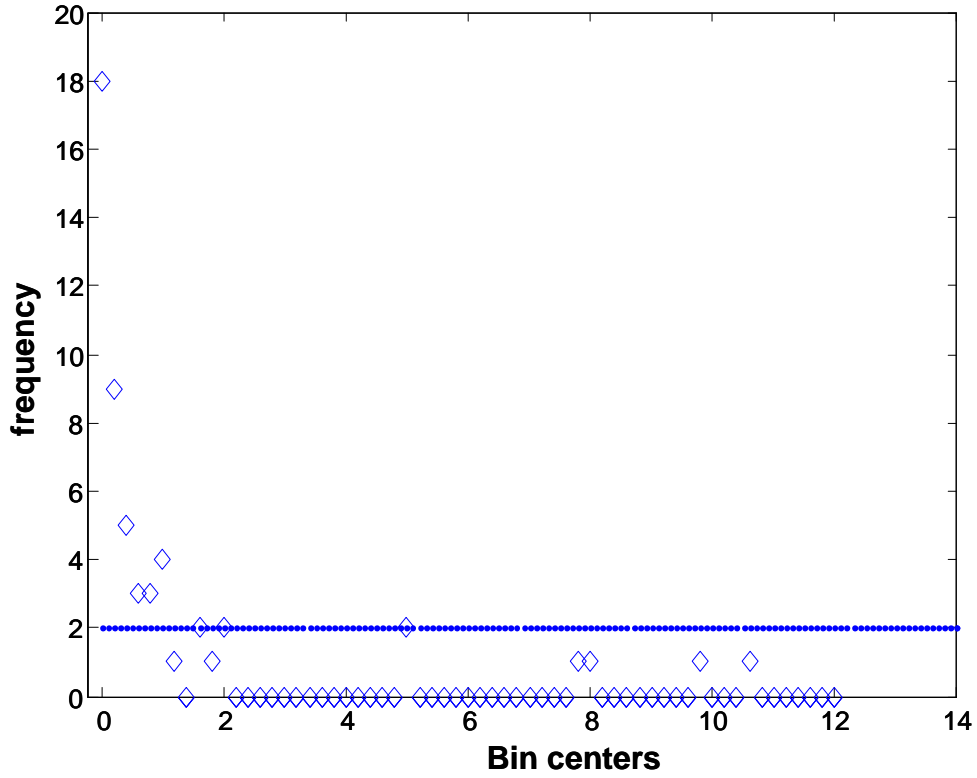


Figure 5.11: Histogram of estimates of RSM model for climb phase

of most of the terms are relatively small. Therefore, the model can be simplified by excluding these terms without a significant loss in the prediction accuracy. To decide the threshold value, histogram of the estimates is plotted as shown in Figure 5.11. From this histogram, a threshold value of 1 is selected. Hence, the model is pruned by excluding the effects whose estimate is lesser than 1. After excluding these effects, coefficients of the pruned model are recalculated and are shown in Figure 5.12.

To verify the performance of the pruned model, the prediction results of both the models, i.e. full model (with 54 effects) and pruned model (with 16 effects) are compared against the actual values given by the C-MAPSS simulation platform. Figure 5.13 shows this comparison for 30 samples chosen from the data given in Appendix B.

This figure shows that the results predicted by the pruned model are almost as

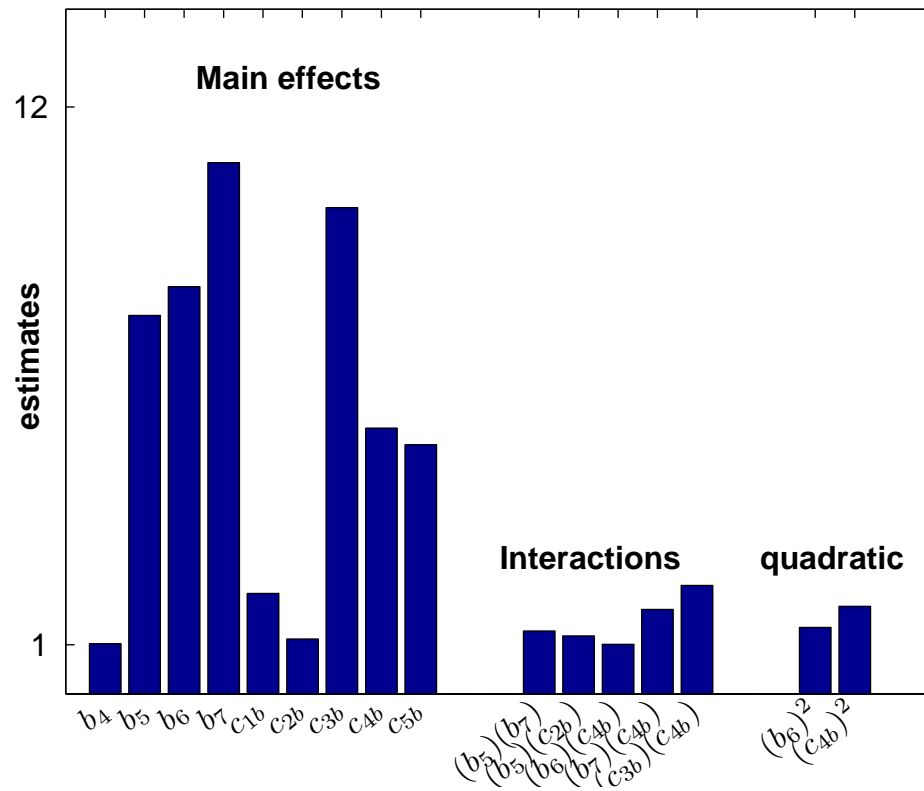


Figure 5.12: Coefficients of pruned system-level RSM model for climb phase.

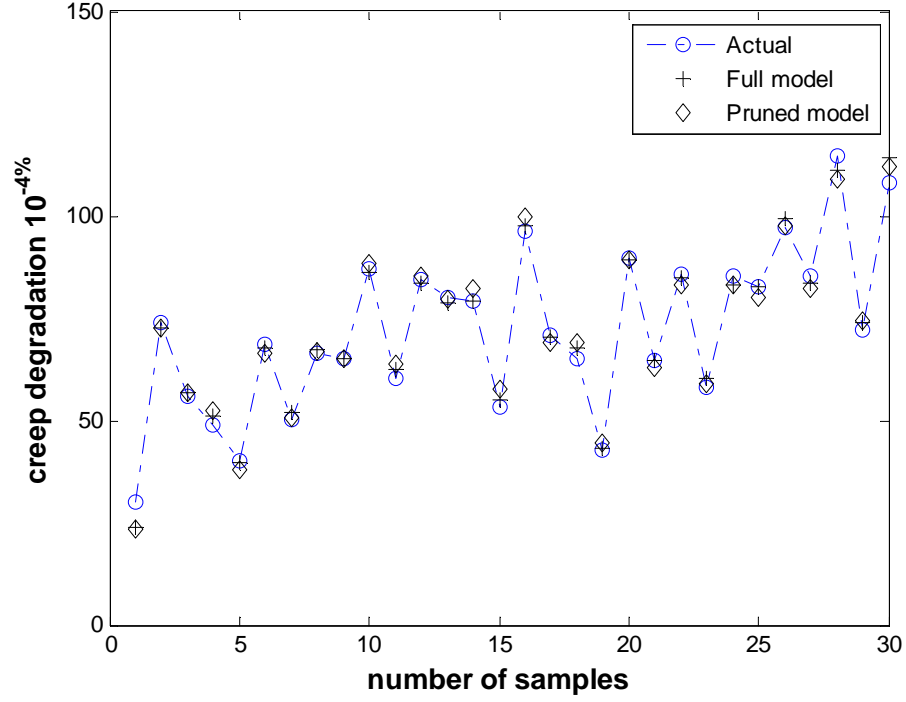


Figure 5.13: Comparison of prediction accuracy of full model and pruned model: climb phase.

accurate as the full model. Therefore, this pruned model is used in the overall system-level metamodel and is given as follows:

$$\begin{aligned}
y_{climb} = & 67.80 + 1.02u_4 - 7.74u_5 + 8.322u_6 + 10.857u_7 + 2.05x_{1b} \\
& + 1.12x_{2b} + 9.93x_{3b} + 5.43x_{4b} + 5.09x_{5b} - 1.28u_5u_7 - 1.18u_5x_{2b} \\
& - 1u_6x_{4b} - 1.72u_7x_{4b} - 2.21x_{3b}x_{4b} + 1.35u_6^2 - 1.78x_{4b}^2 \quad (5.6.3)
\end{aligned}$$

5.6.3 System-level Metamodel for Cruise Phase

From Table 5.8, the system-level metamodel for the cruise phase has 5 parameters. This model is built using the same procedure as used when building the climb phase metamodel.

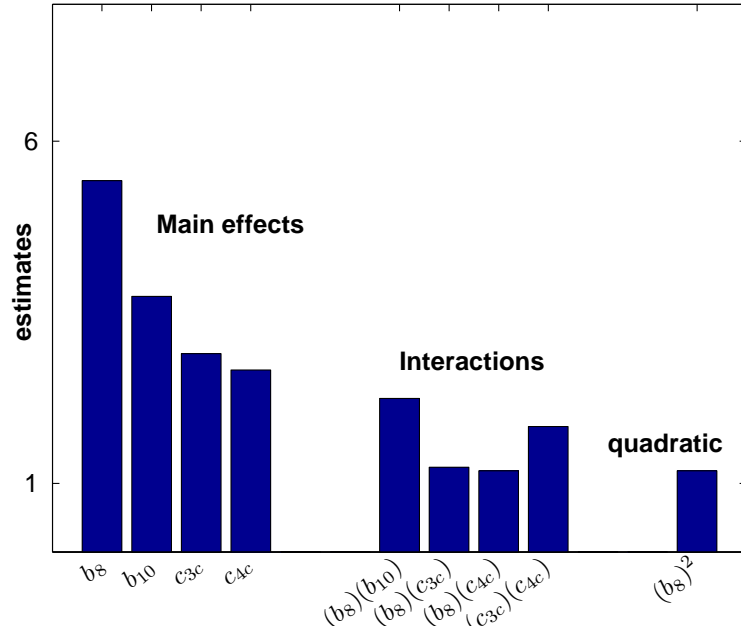


Figure 5.14: Coefficients of pruned system-level RSM model for cruise phase.

5.6.3.1 Experimental design

In the case of cruise phase metamodel, a CCC design is used for the same reasons as in the case of climb phase metamodel. Like the climb phase metamodel, the factorial part of the design is a one-half fractional factorial 2_V^{5-1} . This design consists of 16 factorial-runs, 10 axial-runs and 1 center run. The design consists of 5 columns, one for each factor. The first 4 columns make a full factorial design and the fifth column use the generator $E = +ABCD$. The design and the response values are given in the Appendix D.

5.6.3.2 Model building

The data generated from the experimental design discussed in the previous section are used to build a second-order response surface model (RSM). This model has 5 main effects, 10 interaction effects, and 5 quadratic effects. Only 9 out of these 20 effects (Figure 5.14) have estimate values greater than 1 .

Therefore, this model is pruned by keeping only these terms, and the rest are

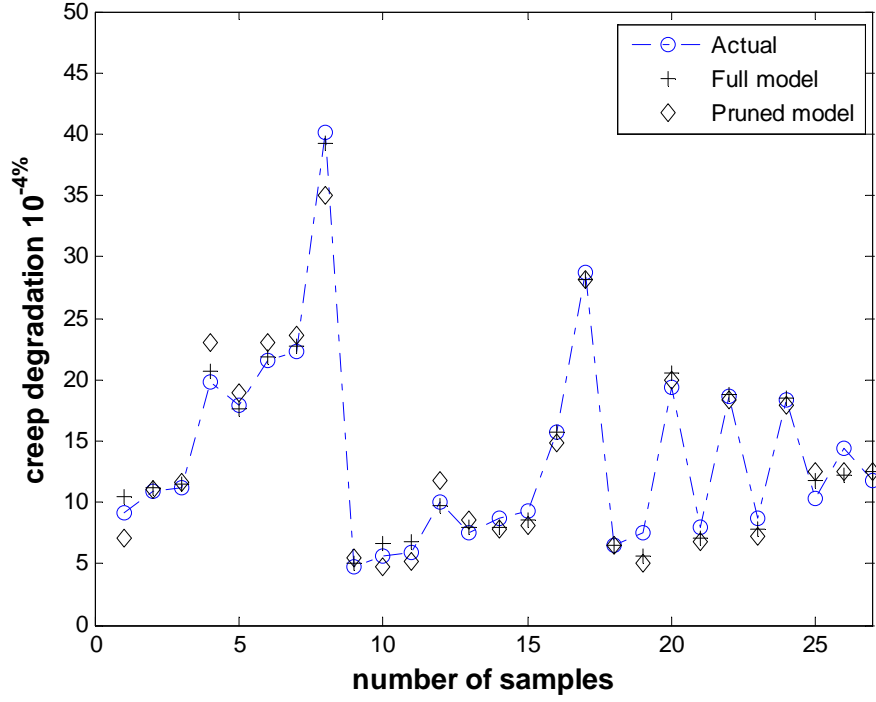


Figure 5.15: Comparison of prediction accuracy of full model and pruned model: cruise phase.

excluded. Figure 5.15 shows that the estimates given by the pruned model are slightly less accurate than those given by the full model. However, the loss in accuracy is acceptable, considering that the number of terms are reduced by more than half. Therefore, this pruned model is used in the overall system-level metamodel and is given as follows:

$$\begin{aligned}
 y_{cruise} = & 12.79 - 5.42u_8 + 3.732u_{10} + 2.896x_{3c} + 2.65x_{4c} \\
 & - 2.24u_8u_{10} - 1.23u_8x_{3c} - 1.18u_8x_{4c} + 1.83x_{3c}x_{4c} + 1.18u_8^2
 \end{aligned} \quad (5.6.4)$$

5.6.4 The Overall System-level Metamodel

The overall system-level metamodel for a single mission is built by linear summation of the three phase-metamodels, i.e.,

$$y_{mission} = y_{takeoff} + y_{climb} + y_{cruise}$$

Using the results from Equations 5.6.2, 5.6.3, and 5.6.4, in the above equation, the overall system-level metamodel is written as shown in Figure 5.16, in which the terms are arranged in order to depict main effects, interaction effects, and quadratic effects, separately.

$$Y_{mission} = \left\{ \begin{array}{l} 83.5 \quad \text{intercept term} \\ +3.8 u_1 + 1.0 u_4 - 7.7 u_5 + 8.3 u_6 + 10.8 u_7 - 5.4 u_8 + 3.7 u_{10} \\ +2.0 x_{1b} + 1.1 x_{2b} + 9.9 x_{3b} + 5.4 x_{4b} + 5.1 x_{5b} + 2.9 x_{3c} + 2.6 x_{4c} \\ -1.3 u_5 u_7 - 1.2 u_5 x_{2b} - 1.0 u_6 x_{4b} - 1.7 u_7 x_{4b} - 2.2 x_{3b} x_{4b} - 2.2 u_8 u_{10} - 1.2 u_8 x_{3c} - 1.2 u_8 x_{4c} \\ +1.3 u_6^2 + 1.2 u_8^2 - 1.8 x_{4b}^2 \end{array} \right\} \begin{array}{l} \\ \text{main effects} \\ \\ \text{quadratic effects} \end{array}$$

Figure 5.16: System level metamodel of the gas turbine engine represented as polynomial.

The model given in Figure 5.16 shows the creep damage during a single mission. The overall damage accumulated over the entire life can be calculated by keeping track of $y_{mission}$ for all the missions, i.e.,

$$y_{present} = \sum_{i=1}^k (y_{mission})_i$$

where k = total number of missions since in service and $(y_{mission})_i$ = creep damage accumulated during the i_{th} mission.

The same results are reproduced in the graphical form in Figure 5.17, in which the width of the connector is varied to represent the relative strength of the effect on the system-level health. It means that any degradation in high-pressure compressor (HPC) causes largest effect on the system's health. Interaction effects are also depicted in this figure, which shows that the effects of degradation in HPC and HPT on the system's health are somewhat dependent on each other.

Representing the system metamodel in this form offers a better understanding of

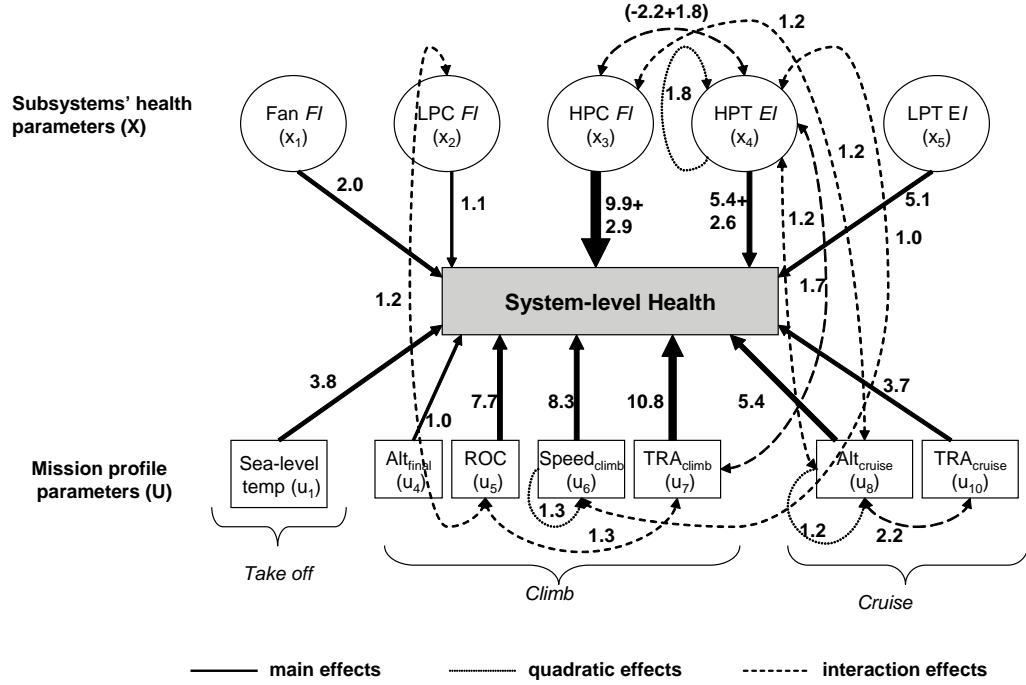


Figure 5.17: Graphical representation of the system-level metamodel.

the relative effects of subsystem health parameters ($x's$) and mission profile parameters ($u's$) on the system's health, without performing any further analysis. Other metamodeling methods, like ANN, lack this feature. Moreover, considering the dimensionality of the input space, the amount of data used to develop this model is quite small as compared to the other metamodeling methods.

The system-level metamodel developed in this work can be applied for multiple purposes. The most straightforward application of this metamodel is to calculate the damage accumulated during a flight cycle, as shown in Figure 5.18, in which the subsystem-level diagnostic information and mission profile parameters are provided as inputs to the metamodel. Although the other metamodeling methods like artificial neural networks (ANNs) can also be used to estimate the damage accumulation in the system, the following features of response surface models (RSM) make them a preferable choice.

- This model can be developed during the design stage of the system when the

designers are interested in investigating the sensor locations and quality. This model can help the designers in making such decisions. For example, if a variable is highly significant, it should be measured with high precision.

b) Response surface methodology allows us to screen out the insignificant variables. Therefore, the time-historical data can be reduced into a set of parameters, thereby reducing the storage space requirements by orders of magnitude without a significant loss of accuracy.

b) An RSM model reveals the quantitative effect of each input factor on the system-level health. This knowledge can be very helpful in mission planning. For example, the mission profile parameters can be chosen in such a way that the damage accumulated during the mission stays below a certain threshold.

c) The effect of health degradation of a subsystem on the overall system's health is provided by the RSM model. It can help in optimizing maintenance scheduling. For example, if the model shows that compressor fouling has a significant effect on the system's health, compressor wash, which is a costly procedure, can be scheduled. On the other hand, if the coefficient of the term representing the compressor health's coefficient is relatively insignificant, maintenance actions can be delayed or avoided altogether.

5.7 Model Verification and Performance Comparison

Prediction results of the system-level RSM model are verified using a new set of data, which is generated based on randomly picked combinations of these variables. For each combination, C-MAPSS is invoked to determine the creep damage, which is compared with the predicted value given by the metamodel shown in Figure 5.16. Figure 5.19 shows that the metamodel gives good results for most of the cases.

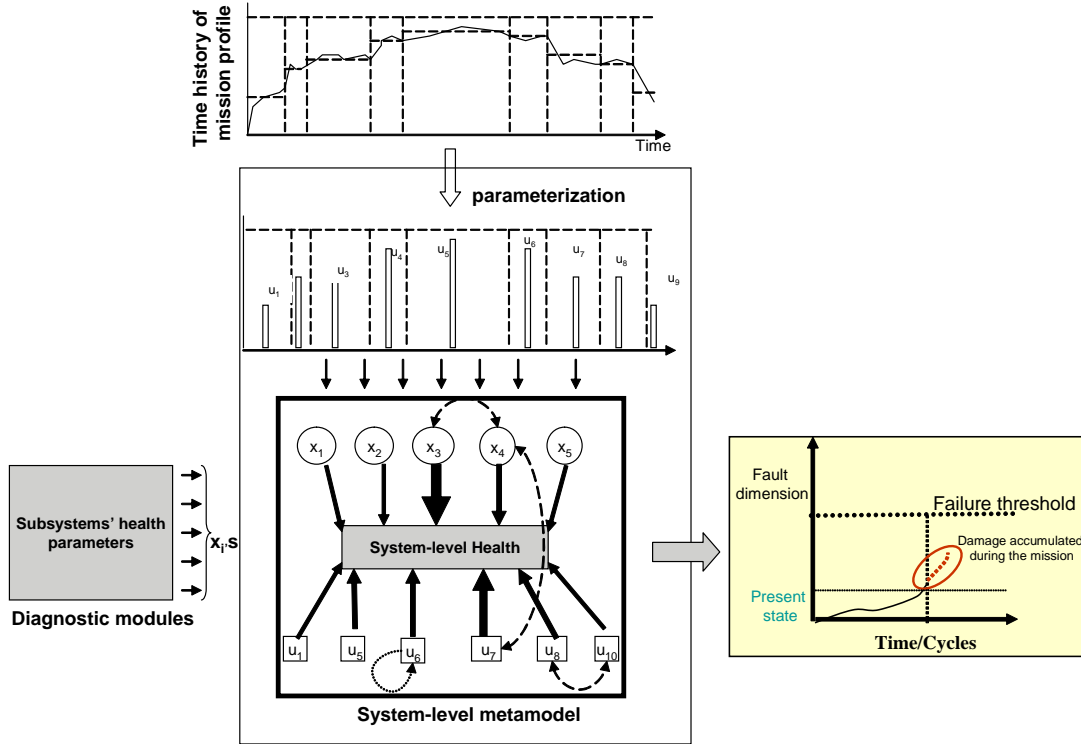


Figure 5.18: System-level metamodel applied to estimate the damage accumulated during a flight cycle.

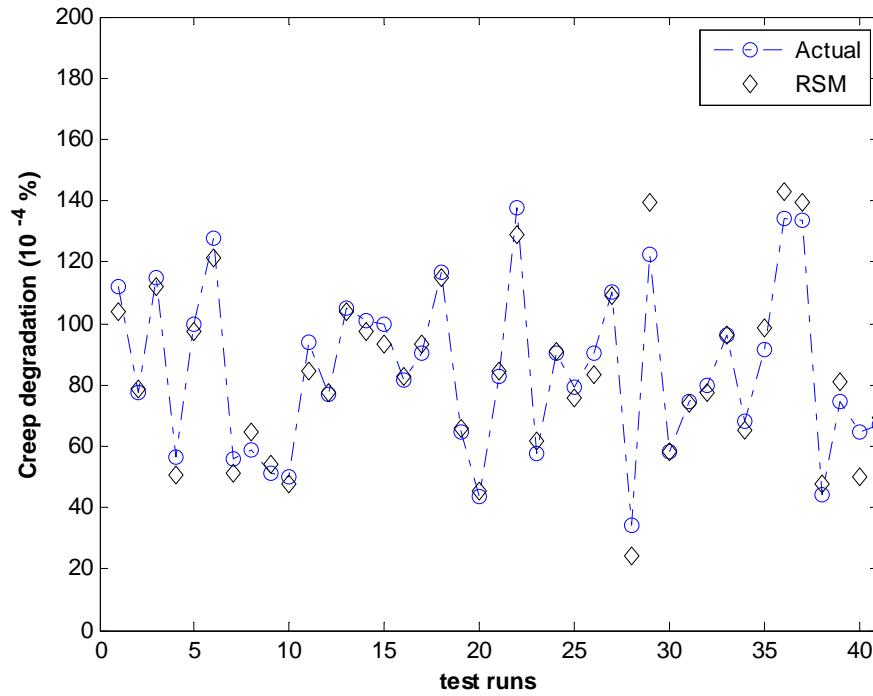


Figure 5.19: Performance of the system-level RSM model for the test cases.

5.7.1 Artificial Neural Network (ANN) Models

A multilayer feed forward ANN architecture, trained using an error back propagation algorithm (EBPA), is employed to develop the creep damage predictive models for each phase, i.e., takeoff, climb, and cruise (descent phase was excluded after the screening stage). The training algorithm being used is Levenberg-Marquardt algorithm, which appears to be the fastest method for training moderate-sized feedforward neural networks (up to several hundred weights) [84]. Transfer function being used is 'tansig' for hidden layers and 'purelin' for output layer. The number of neurons in the input layer correspond to the number of input variables (significant variables left after the screening stage) and a single neuron in the output layer (corresponding to the output).

For the takeoff phase, there is only a single parameter (u_1). It is observed that an ANN with 2 neurons in the hidden layer gives highly accurate results for this phase.

For climb and cruise phases, the number of input variables are 9 and 5, respectively (see Table 5.8). The data used to train and test the ANN models are generated by random sampling of the design space. The number of samples used to build the models are chosen to be approximately the same as used to build the RSM models. For the climb phase, 150 samples used for training and validation while for the climb phase 27 are used for training and validation. For each of these phases, the number of neurons in the hidden layer are varied from 1 to 30. The suitable number of hidden-layer neurons is determined by running 100 runs of ANN training simulation for each case and then calculating the performance of the ANN model for the test cases using the average root-mean-square error (RMSE) as the following:

$$RMSE = \sqrt{\text{mean}(y_{\text{actual}} - y_{\text{ANN}})^2}$$

In general, it is observed that too few neurons lead to underfitting. On the other hand, too many neurons can contribute to overfitting (an ANN will memorize the training sets), in which all training points are well fit, but prediction error in the case

of test points increases. Figure 5.20 shows the average values of test errors plotted vs. the number of neurons in the hidden layer. It is found that for the climb phase, the best performance is achieved by using 6 neurons in the hidden layer. For the cruise phase the best performance is achieved by using 4 neurons in the hidden layer (Figure 5.21). Thus, three ANN models are developed for each of the takeoff phase (having 2 neurons in the hidden layer), climb phase (having 6 neurons in the hidden layer), and cruise phase (having 4 neurons in the hidden layer). The overall ANN model is obtained by adding the three models. The performance of this model is then compared with that of the RSM model. Figure 5.22 shows the comparison between these two types of models, i.e., RSM and ANN for 40 test cases. From this figure, it can be observed that the RSM accuracy is quite close to that of the ANN model. It should be emphasized here that the model accuracy is not the most important criterion in preferring the model choice in this case. Here the objective is to develop a model that can assist in decision making at the system-level. Therefore, the most important criteria are the model representation that is useful in achieving this objective. Secondly, ANN does not offer any systematic procedure to reduce the number of input factors. In a complex system, the amount of data required to build the model should be feasible. In contrast to the ANN approach, the RSM methodology is useful in formulating the problem of parameter screening.

5.8 Summary of Results

5.8.1 Data Requirement vs. Number of Variables

The response surface model is built in two stages: screening and modeling. During the screening stage, mission profile parameters (u_i s) and subsystems' health parameters (x_i s) are treated separately to reduce the number of runs. Total number of parameters that are considered as input factors during the screening stage are 34 (14 of these are u_i s, and 20 are x_i s), and number of experiments performed are 188 (80 of these are

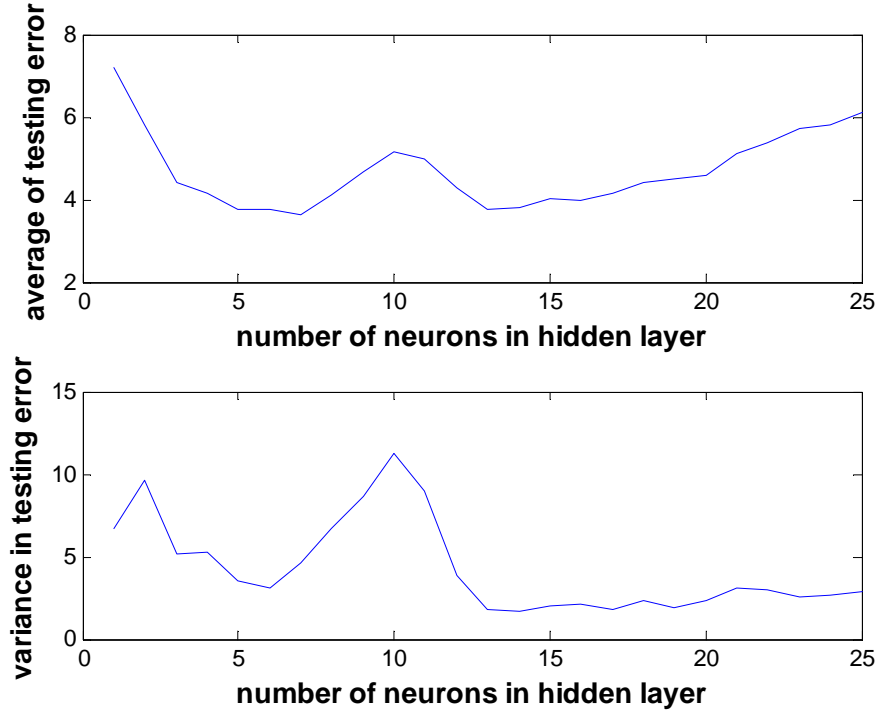


Figure 5.20: Testing error vs. number of neurons in the hidden layer: Climb phase ANN model.

used to screen out $u_i s$, and 108 are to screen out $x_i s$). Of the 34 potential factors, 19 are screened out. The remaining 15 factors are included in the next stage, i.e., modeling.

During the modeling stage, mission profile parameters and subsystems' health parameters are considered together, to investigate any possible interactions between these two types of parameters. The number of experiments performed during this stage is 174, out of which 147 are used to build the metamodel for the climb phase, and 27 are used to build cruise phase. It can be recalled that separate model for each phase was developed and subsequently added to keep the number of experiments small. The overall model contains 78 terms (2 for takeoff phase, 55 for climb phase, and 21 for cruise phase), including intercept terms, main effects, interaction effects, and quadratic effects. This model is then pruned to ignore the insignificant terms. The pruned model consists of 27 terms, as given in Figure 5.16.

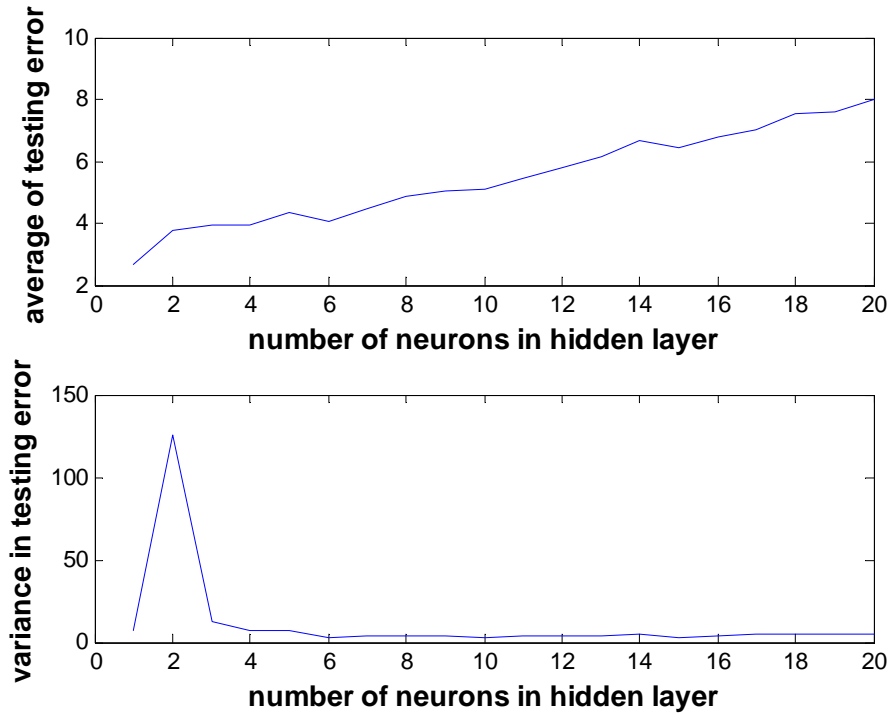


Figure 5.21: Testing error vs. number of neurons in the hidden layer: Cruise phase ANN model.

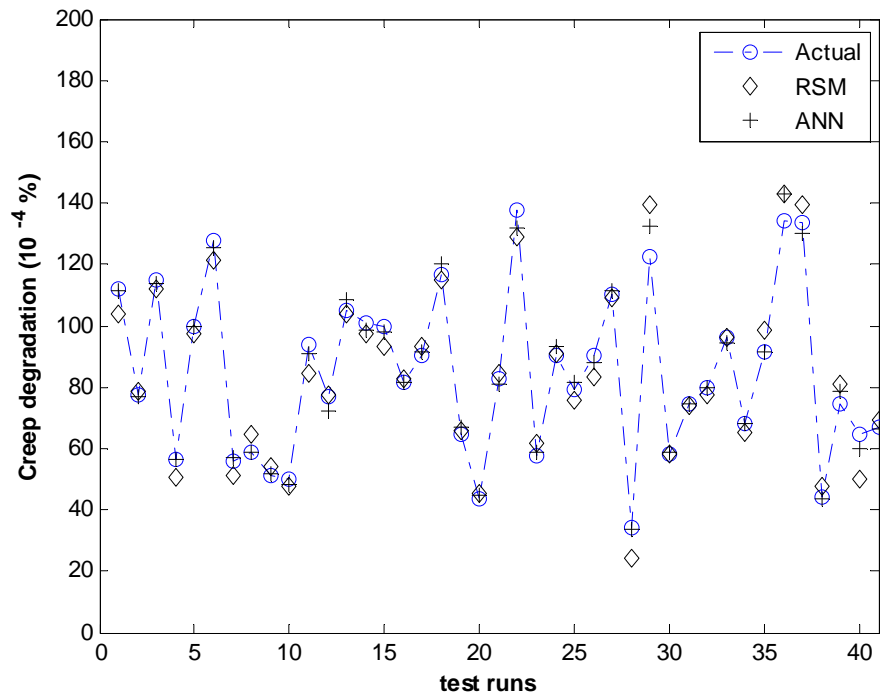


Figure 5.22: Accuracy comparison of RSM model vs. ANN model: Test cases

Summarizing the above discussion, 188 (screening) + 174 (modeling)= 362 samples are used to build a second-order polynomial model (RSM) that represents a 34-dimensional input space.

5.8.2 Model Building Procedure and Model Accuracy: RSM vs. ANN

In the case of ANN models, the performance depends on the model structure and initial weights, the optimization of which needs experience and time. In the case of RSM, a systematic procedure based on theoretical foundation has been developed over the years.

The prediction accuracy of system-level response surface model (RSM) is compared with that of artificial neural network (ANNs) models, in Section 5.7. In the case of ANN modeling, it was observed that the best performance is achieved by using five neurons in the hidden layer. Figure 5.20 shows RMSE of 100 different ANN networks, all with a five-neuron hidden layer. It can be observed from this figure that for ANN models RMSE fluctuates from 3 to 18, while the RSM model has RMSE value of less than 2.

5.8.3 Model Representation

As shown in Figures 5.16 and 5.17, the RSM can be represented in polynomial or graphical form. The representation in either form reveals the relative significance of model inputs (mission profile parameters and subsystems' health) on the system's health, considering creep life of the turbine blades as the health criterion. An ANN model, on the other hand, is a black box model, which maps the inputs to the outputs without revealing any such information.

Chapter VI

CONCLUSIONS AND SUGGESTED FUTURE WORK

This thesis presents a methodology that is developed in an effort to build prognostic models that include the effects of subsystems' health degradation and system-level operating conditions on the life of critical components. The motivation for building this model was that it should assist the system managers, operators, and maintenance-related people in making appropriate decisions. Therefore, the model was required to be simple enough so that the decision makers can readily benefit from the model representation. Theoretical framework of the methodology is presented in the first three chapters of this thesis. The methodology is developed for a general case, and an aircraft engine of turbofan type is used as a case study to test and verify the functionality of this approach. To a large extent, the methodology can be generalized to the other applications but a few details are specific to the system on which the methodology is being implemented.

The system-level health assessment methodology consists of four types of models that are arranged in a hierarchical architecture. At the lowest level, the failure mechanism model $h_1(.)$ explains the effects of stress factors s on the failure of the component. The failure mechanism model is then combined with the load-stress model $h_2(.)$, which is based on the material properties and geometry of the component. These two models are combined with each other and the resultant model is a component-level model that explains the effects of component-level loads to the rate of fault growth.

These types of models come from the physics-of-failure (PoF) knowledge and the information about the material properties and geometry of the component. If these

pieces of information are available, such component-level models can be built for any type of system.

Moving up in the hierarchy, the relation between the subsystem-level variables V and the load variables W is modeled by a subsystem-level model $g(\cdot)$. In the thesis, a turbofan engine is used as the application domain and the HPT turbine blade is considered as the critical component. The turbine blade (component) is a part of high-pressure turbine (subsystem). The relation between the load variables and the subsystem-level variables is simplified by making certain assumptions that are specific to the system. While applying the methodology on the other types of systems, these assumptions might not be valid and the relation between the subsystem-level variables and the component-level variables should be derived using the expert knowledge.

At the highest level, the system-level variables that are typically recorded as time-historical data, are reduced into a set of parameters by dividing the data into segments and subsequently, using their average values. For some of the failure mechanisms, like creep, this approach to data reduction is valid, since creep is not sensitive to high frequency patterns in the stress variables. When accounting for those failure mechanisms that are affected by the fluctuations in the stress variables, an appropriate data reduction method should be used. The next step in this direction can be the development of a method that can account for both types of failure mechanisms.

At the lowest level, the methodology is built on component level failure models. In this work, it is assumed that the system always fail due to a single type of component. For example, it is assumed in the case study that the RUL of the HPT blades is representative of the RUL of the whole engine. In practical systems, multiple types of components are responsible for the system failures. A system-level metamodeling methodology that can take care of multiple failing components, can be useful in many applications.

The methodology developed in thesis is based on damage accumulation models, which do not need the real-time diagnostic information about the failing component. For some of the components, damage summation models are useful since the deterioration is hard to be detected. In other cases, incipient faults are relatively easy to be detected and measured. In such cases, a better approach is to make the model adaptive, i.e., the coefficients are estimated based on the online diagnostic information.

Appendix A

SCREENING STAGE: DESIGN MATRICES AND RESPONSE VALUES

A.1 Takeoff Phase

Trial No.	Levels of i/p factors (coded)			Response (creep degradation 10^{-4} %)
	Tsl	Speed _{To}	TRA _{To}	
	u ₁	u ₂	u ₃	
1	-0.6	-0.6	-0.6	0.6084
2	0.6	-0.6	-0.6	3.4018
3	-0.6	0.6	-0.6	0.9333
4	0.6	0.6	-0.6	5.236
5	-0.6	-0.6	0.6	0.7806
6	0.6	-0.6	0.6	4.5093
7	-0.6	0.6	0.6	1.354
8	0.6	0.6	0.6	7.937
9	-1	0	0	0.5458
10	1	0	0	8.443
11	0	-1	0	1.302
12	0	1	0	3.144
13	0	0	-1	1.6167
14	0	0	1	2.835
15	0	0	0	2.2

A.2 Climb Phase

Trial No.	Levels of i/p factors (coded)				Response (creep degradation 10^{-4} %)
	Alt _{final}	RoC	Speed _{climb}	TRA _{climb}	
	u ₄	u ₅	u ₆	u ₇	
1	-0.6	-0.6	-0.6	-0.6	16.945
2	0.6	-0.6	-0.6	-0.6	16.6178
3	-0.6	0.6	-0.6	-0.6	13.457
4	0.6	0.6	-0.6	-0.6	12.762
5	-0.6	-0.6	0.6	-0.6	23.55
6	0.6	-0.6	0.6	-0.6	23.765
7	-0.6	0.6	0.6	-0.6	18.38
8	0.6	0.6	0.6	-0.6	18.535
9	-0.6	-0.6	-0.6	0.6	28.935
10	0.6	-0.6	-0.6	0.6	28.2
11	-0.6	0.6	-0.6	0.6	22.928
12	0.6	0.6	-0.6	0.6	27.93
13	-0.6	-0.6	0.6	0.6	42.433
14	0.6	-0.6	0.6	0.6	42.71
15	-0.6	0.6	0.6	0.6	31.617
16	0.6	0.6	0.6	0.6	32.735
17	-1	0	0	0	26.383
18	1	0	0	0	19.765
19	0	-1	0	0	24.336
20	0	1	0	0	20.937
21	0	0	-1	0	17.86
22	0	0	1	0	33.93
23	0	0	0	-1	14.3
24	0	0	0	1	26.561
25	0	0	0	0	22.495

A.3 Cruise Phase

Trial No.	Levels of i/p factors (coded)			Response (creep degradation 10^{-4} %)
	Alt _{cruise}	Speed _{cruise}	TRA _{cruise}	
	u ₈	u ₉	u ₁₀	
1	-0.6	-0.6	-0.6	5.75
2	0.6	-0.6	-0.6	2.25
3	-0.6	0.6	-0.6	5.915
4	0.6	0.6	-0.6	2.237
5	-0.6	-0.6	0.6	10.1475
6	0.6	-0.6	0.6	3.7925
7	-0.6	0.6	0.6	10.5725
8	0.6	0.6	0.6	3.886
9	-1	0	0	10.64
10	1	0	0	2.68875
11	0	-1	0	4.83125
12	0	1	0	4.898625
13	0	0	-1	2.955
14	0	0	1	7.41125
15	0	0	0	4.714

A.4 Descent Phase

Trial No.	Levels of i/p factors (coded)				Response (creep degradation 10^{-4} %)
	Alt _{initial}	RoD	Speed _{descent}	TRA _{descent}	
	u ₁₁	u ₁₂	u ₁₃	u ₁₄	
1	-0.6	-0.6	-0.6	-0.6	1.65602
2	0.6	-0.6	-0.6	-0.6	1.0137
3	-0.6	0.6	-0.6	-0.6	2.2702
4	0.6	0.6	-0.6	-0.6	1.9343
5	-0.6	-0.6	0.6	-0.6	1.09737
6	0.6	-0.6	0.6	-0.6	0
7	-0.6	0.6	0.6	-0.6	1.97227
8	0.6	0.6	0.6	-0.6	1.4897
9	-0.6	-0.6	-0.6	0.6	2.5442
10	0.6	-0.6	-0.6	0.6	1.983
11	-0.6	0.6	-0.6	0.6	2.7496
12	0.6	0.6	-0.6	0.6	2.4578
13	-0.6	-0.6	0.6	0.6	2.003
14	0.6	-0.6	0.6	0.6	1.194
15	-0.6	0.6	0.6	0.6	2.4607
16	0.6	0.6	0.6	0.6	2.0246
17	-1	0	0	0	2.15751
18	1	0	0	0	1.29089
19	0	-1	0	0	1.346
20	0	1	0	0	2.4599
21	0	0	-1	0	2.32878
22	0	0	1	0	2.0365
23	0	0	0	-1	1.5915
24	0	0	0	1	2.9279
25	0	0	0	0	1.72719

Appendix B

CLIMB PHASE METAMODELING: DESIGN MATRIX AND RESPONSE VALUES

Trial No.	A u ₄	B u ₅	C u ₆	D u ₇	E x ₁	F x ₂	G x ₃	H=ACDFG x ₄	J=BCEFG x ₅	Response (creep degradation 10 ⁻⁴ %)
1	-1	-1	-1	-1	-1	-1	-1	-1	-1	29.93556609
2	-1	-1	-1	-1	-1	-1	1	1	1	73.87543885
3	-1	-1	-1	-1	-1	1	-1	1	1	55.9260706
4	-1	-1	-1	-1	-1	1	1	-1	-1	49.02605833
5	-1	-1	-1	-1	1	-1	-1	-1	1	40.21523082
6	-1	-1	-1	-1	1	-1	1	1	-1	68.66593115
7	-1	-1	-1	-1	1	1	-1	1	-1	50.04946638
8	-1	-1	-1	-1	1	1	1	-1	1	66.50320833
9	-1	-1	-1	1	-1	-1	-1	1	-1	65.34565795
10	-1	-1	-1	1	-1	-1	1	-1	1	87.05509194
11	-1	-1	-1	1	-1	1	-1	-1	1	60.15975312
12	-1	-1	-1	1	-1	1	1	1	-1	84.50216779
13	-1	-1	-1	1	1	-1	-1	1	1	80.0623662
14	-1	-1	-1	1	1	-1	1	-1	-1	79.15508552
15	-1	-1	-1	1	1	1	-1	-1	-1	53.20300885
16	-1	-1	-1	1	1	1	1	1	1	96.18589856
17	-1	-1	1	-1	-1	-1	-1	1	1	70.98959772
18	-1	-1	1	-1	-1	-1	1	-1	-1	65.22377614
19	-1	-1	1	-1	-1	1	-1	-1	-1	42.78787042
20	-1	-1	1	-1	-1	1	1	1	1	89.86983936
21	-1	-1	1	-1	1	-1	-1	1	-1	64.78981689
22	-1	-1	1	-1	1	-1	1	-1	1	85.86660962
23	-1	-1	1	-1	1	1	-1	-1	1	58.0650697
24	-1	-1	1	-1	1	1	1	1	-1	85.08786395
25	-1	-1	1	1	-1	-1	-1	-1	1	82.8482232
26	-1	-1	1	1	-1	-1	1	1	-1	97.00325466
27	-1	-1	1	1	-1	1	-1	1	-1	85.38160131
28	-1	-1	1	1	-1	1	1	-1	1	114.5477041
29	-1	-1	1	1	1	-1	-1	-1	-1	72.2141816
30	-1	-1	1	1	1	-1	1	1	1	107.9803874
31	-1	-1	1	1	1	1	-1	1	1	99.52201181
32	-1	-1	1	1	1	1	1	-1	-1	105.6867833
33	-1	1	-1	-1	-1	-1	-1	-1	1	29.4960106
34	-1	1	-1	-1	-1	-1	1	1	-1	51.73079026
35	-1	1	-1	-1	-1	1	-1	1	-1	36.52434013
36	-1	1	-1	-1	-1	1	1	-1	1	48.16140754
37	-1	1	-1	-1	1	-1	-1	-1	-1	26.33873441
38	-1	1	-1	-1	1	-1	1	1	1	61.75555881
39	-1	1	-1	-1	1	1	-1	1	1	48.29516509
40	-1	1	-1	-1	1	1	1	-1	-1	42.85526255
41	-1	1	-1	1	-1	-1	-1	1	1	60.08913103
42	-1	1	-1	1	-1	-1	1	-1	-1	57.92178832
43	-1	1	-1	1	-1	1	-1	-1	-1	38.6443931
44	-1	1	-1	1	-1	1	1	1	1	73.63253304
45	-1	1	-1	1	1	-1	-1	1	-1	55.54026619

Trial No.	A u ₄	B u ₅	C u ₆	D u ₇	E x ₁	F x ₂	G x ₃	H=ACDFG x ₄	J=BCEFG x ₅	Response (creep degradation 10 ⁻⁴ %)
46	-1	1	-1	1	1	-1	1	-1	1	73.85141549
47	-1	1	-1	1	1	1	-1	-1	1	52.69083333
48	-1	1	-1	1	1	1	1	1	-1	69.84898715
49	-1	1	1	-1	-1	-1	-1	1	-1	48.67754998
50	-1	1	1	-1	-1	-1	1	-1	1	63.65588488
51	-1	1	1	-1	-1	1	-1	-1	1	42.56466056
52	-1	1	1	-1	-1	1	1	1	-1	65.2698578
53	-1	1	1	-1	1	-1	-1	1	1	59.65961246
54	-1	1	1	-1	1	-1	1	-1	-1	56.37834605
55	-1	1	1	-1	1	1	-1	-1	-1	37.17428344
56	-1	1	1	-1	1	1	1	1	1	73.86174106
57	-1	1	1	1	-1	-1	-1	-1	-1	52.57821778
58	-1	1	1	1	-1	-1	1	1	1	83.34699182
59	-1	1	1	1	-1	1	-1	1	1	75.93567233
60	-1	1	1	1	-1	1	1	-1	-1	79.43853902
61	-1	1	1	1	1	-1	-1	-1	1	70.89834271
62	-1	1	1	1	1	-1	1	1	-1	79.18737227
63	-1	1	1	1	1	1	-1	1	-1	70.83054361
64	-1	1	1	1	1	1	1	-1	1	94.80084329
65	1	-1	-1	-1	-1	-1	-1	1	-1	44.62106306
66	1	-1	-1	-1	-1	-1	1	-1	1	59.02103744
67	1	-1	-1	-1	-1	1	-1	-1	1	39.25384855
68	1	-1	-1	-1	-1	1	1	1	-1	68.74789698
69	1	-1	-1	-1	1	-1	-1	1	1	59.65622268
70	1	-1	-1	-1	1	-1	1	-1	-1	52.54332502
71	1	-1	-1	-1	1	1	-1	-1	-1	34.92542495
72	1	-1	-1	-1	1	1	1	1	1	82.36095671
73	1	-1	-1	1	-1	-1	-1	-1	-1	47.04683836
74	1	-1	-1	1	-1	-1	1	1	1	94.91434128
75	1	-1	-1	1	-1	1	-1	1	1	80.2357797
76	1	-1	-1	1	-1	1	1	-1	-1	77.92211188
77	1	-1	-1	1	1	-1	-1	-1	1	64.52530331
78	1	-1	-1	1	1	-1	1	1	-1	89.55563279
79	1	-1	-1	1	1	1	-1	1	-1	74.09859617
80	1	-1	-1	1	1	1	1	-1	1	99.09751991
81	1	-1	1	-1	-1	-1	-1	-1	1	51.80569648
82	1	-1	1	-1	-1	-1	1	1	-1	82.93559645
83	1	-1	1	-1	-1	1	-1	1	-1	64.64120418
84	1	-1	1	-1	-1	1	1	-1	1	85.2441033
85	1	-1	1	-1	1	-1	-1	-1	-1	45.18477959
86	1	-1	1	-1	1	-1	1	1	1	94.8151342
87	1	-1	1	-1	1	1	-1	1	1	79.35392781
88	1	-1	1	-1	1	1	1	-1	-1	75.74757098
89	1	-1	1	1	-1	-1	-1	1	1	96.79045448
90	1	-1	1	1	-1	-1	1	-1	-1	99.67781127
91	1	-1	1	1	-1	1	-1	-1	-1	70.92443035
92	1	-1	1	1	-1	1	1	1	1	112.9353144
93	1	-1	1	1	1	-1	-1	1	-1	89.74722238
94	1	-1	1	1	1	-1	1	-1	1	120.504448
95	1	-1	1	1	1	1	-1	-1	1	95.36391414
96	1	-1	1	1	1	1	1	1	-1	106.8116958
97	1	1	-1	-1	-1	-1	-1	1	1	43.70758251
98	1	1	-1	-1	-1	-1	1	-1	-1	38.30638905
99	1	1	-1	-1	-1	1	-1	-1	-1	25.73665063
100	1	1	-1	-1	-1	1	1	1	1	62.17472462

Trial No.	A u ₄	B u ₅	C u ₆	D u ₇	E x ₁	F x ₂	G x ₃	H=ACDFG x ₄	J=BCEFG x ₅	Response (creep degradation 10 ⁻⁴ %)
101	1	1	-1	-1	1	-1	-1	1	-1	39.09062305
102	1	1	-1	-1	1	-1	1	-1	1	51.6095016
103	1	1	-1	-1	1	1	-1	-1	1	34.46321587
104	1	1	-1	-1	1	1	1	1	-1	57.84690336
105	1	1	-1	1	-1	-1	-1	-1	1	46.69055738
106	1	1	-1	1	-1	-1	1	1	-1	67.96853926
107	1	1	-1	1	-1	1	-1	1	-1	55.20948727
108	1	1	-1	1	-1	1	1	-1	1	73.55228068
109	1	1	-1	1	1	-1	-1	-1	-1	41.33411621
110	1	1	-1	1	1	-1	1	1	1	78.35556584
111	1	1	-1	1	1	1	-1	1	1	67.44933903
112	1	1	-1	1	1	1	1	-1	-1	66.94558329
113	1	1	1	-1	-1	-1	-1	-1	-1	33.34807947
114	1	1	1	-1	-1	-1	1	1	1	72.36858243
115	1	1	1	-1	-1	1	-1	1	1	59.68743352
116	1	1	1	-1	-1	1	1	-1	-1	55.53150653
117	1	1	1	-1	1	-1	-1	-1	1	44.99966562
118	1	1	1	-1	1	-1	1	1	-1	68.35498823
119	1	1	1	-1	1	1	-1	1	-1	54.45333119
120	1	1	1	-1	1	1	1	-1	1	72.28079502
121	1	1	1	1	-1	-1	-1	1	-1	67.87589189
122	1	1	1	1	-1	-1	1	-1	1	91.01223278
123	1	1	1	1	-1	1	-1	-1	1	70.32913223
124	1	1	1	1	-1	1	1	1	-1	81.95916009
125	1	1	1	1	1	-1	-1	1	1	80.0604314
126	1	1	1	1	1	-1	1	-1	-1	83.59286776
127	1	1	1	1	1	1	-1	-1	-1	61.63908724
128	1	1	1	1	1	1	1	1	1	92.16081018
129	-2	0	0	0	0	0	0	0	0	64.43975338
130	2	0	0	0	0	0	0	0	0	68.30319671
131	0	-2	0	0	0	0	0	0	0	87.74852433
132	0	2	0	0	0	0	0	0	0	55.07280062
133	0	0	-2	0	0	0	0	0	0	54.55937702
134	0	0	2	0	0	0	0	0	0	91.45779077
135	0	0	0	-2	0	0	0	0	0	42.91913616
136	0	0	0	2	0	0	0	0	0	90.10453306
137	0	0	0	0	-2	0	0	0	0	63.50273432
138	0	0	0	0	2	0	0	0	0	72.24149
139	0	0	0	0	0	-2	0	0	0	65.21426591
140	0	0	0	0	0	2	0	0	0	70.49709158
141	0	0	0	0	0	0	-2	0	0	44.80406365
142	0	0	0	0	0	0	2	0	0	87.07691383
143	0	0	0	0	0	0	0	-2	0	48.34221645
144	0	0	0	0	0	0	0	2	0	72.51799732
145	0	0	0	0	0	0	0	0	-2	57.13445005
146	0	0	0	0	0	0	0	0	2	79.35291237
147	0	0	0	0	0	0	0	0	0	67.53472287

Appendix C

CLIMB PHASE METAMODELING: COEFFICIENT VALUES OF ALL TERMS

Term	Coefficient value	Term	Coefficient value
Intercept	67.40724874	b_6C_{2b}	-0.107065691
b_4	1.041509751	b_6C_{3b}	0.215366264
b_5	-7.888781279	b_6C_{4b}	-1.021302963
b_6	8.060744591	b_6C_{5b}	0.293715758
b_7	10.77427	b_7C_{1b}	0.47842646
C_{1b}	2.094536792	b_7C_{2b}	-0.019610871
C_{2b}	1.170088286	b_7C_{3b}	0.371086581
C_{3b}	9.828781103	b_7C_{4b}	-1.915487035
C_{4b}	5.014066596	b_7C_{5b}	-0.116408702
C_{5b}	5.015891037	$C_{1b}C_{2b}$	-0.242468631
b_4b_5	0.029307615	$C_{1b}C_{3b}$	0.025553035
b_4b_6	0.133715806	$C_{1b}C_{4b}$	-0.381093377
b_4b_7	0.594611532	$C_{1b}C_{5b}$	-0.103211574
b_4C_{1b}	-0.02969991	$C_{2b}C_{3b}$	0.09595933
b_4C_{2b}	0.283695312	$C_{2b}C_{4b}$	0.080902901
b_4C_{3b}	0.459640758	$C_{2b}C_{5b}$	-0.296907675
b_4C_{4b}	0.079782243	$C_{3b}C_{4b}$	-2.008037816
b_4C_{5b}	0.059947529	$C_{3b}C_{5b}$	-0.415656628
b_5b_6	-0.991900422	$C_{4b}C_{5b}$	-0.765932806
b_5b_7	-1.064924333	b_4^2	-0.814137703
b_5C_{1b}	-0.332699014	b_5^2	0.651793473
b_5C_{2b}	0.121062878	b_6^2	1.674273027
b_5C_{3b}	-1.347942616	b_7^2	0.001559837
b_5C_{4b}	-0.632315284	C_{1b}^2	0.01404092
b_5C_{5b}	-0.268730432	C_{2b}^2	0.037781282
b_6b_7	0.892757557	C_{3b}^2	-0.132392119
b_6C_{1b}	-0.023477642	C_{4b}^2	-1.650366937
		C_{5b}^2	0.479219167

Appendix D

CRUISE PHASE METAMODELING: DESIGN MATRIX AND RESPONSE VALUES

Trial No.	A u_8	B u_{10}	x_{1c}	x_{2c}	C x_{3c}	D x_{4c}	E=ABCD x_{5c}	Response (creep degradation $10^{-4}\%$)
1	-1	-1	0	0	-1	-1	1	10.78845368
2	-1	-1	0	0	-1	1	-1	13.57167361
3	-1	-1	0	0	1	-1	-1	13.82085935
4	-1	-1	0	0	1	1	1	24.73218046
5	-1	1	0	0	-1	-1	-1	14.78704172
6	-1	1	0	0	-1	1	1	25.42937872
7	-1	1	0	0	1	-1	1	27.13155484
8	-1	1	0	0	1	1	-1	34.12210376
9	1	-1	0	0	-1	-1	-1	3.97323955
10	1	-1	0	0	-1	1	1	6.549305919
11	1	-1	0	0	1	-1	1	6.938918722
12	1	-1	0	0	1	1	-1	8.649338984
13	1	1	0	0	-1	-1	1	7.511881586
14	1	1	0	0	-1	1	-1	8.704865268
15	1	1	0	0	1	-1	-1	9.251817494
16	1	1	0	0	1	1	1	15.76247636
17	-2	0	0	0	0	0	0	28.79643873
18	2	0	0	0	0	0	0	6.500071239
19	0	-2	0	0	0	0	0	7.519182328
20	0	2	0	0	0	0	0	19.40779943
21	0	0	0	0	-2	0	0	8.000469131
22	0	0	0	0	2	0	0	18.58592856
23	0	0	0	0	0	-2	0	8.675294605
24	0	0	0	0	0	2	0	18.27899561
25	0	0	0	0	0	0	-2	10.24480039
26	0	0	0	0	0	0	2	14.36372316
27	0	0	0	0	0	0	0	11.83939369

VITA

Manzar Abbas received his early education at Sacred Heart School, Jhang and Government F. C. College Lahore in Pakistan. Then, he completed his Bachelors in Avionics Engineering from National University of Sciences and Technology, Pakistan. He then moved to Georgia Institute of Technology where he received his MS in Electrical and Computer Engineering. Currently, he is part of Intelligent Control Systems Laboratory, in Georgia Tech. His current research interests focus on developing prognostic methodologies for complex systems, using surrogate modeling approaches.

REFERENCES

- [1] *Machinery Information Management Open Standards Alliance (MIMOSA)*, Open Systems Architecture for Condition Based Maintenance (OSA-CBM) primer, Aug. 2006.
- [2] A. Cundy, “Use of response surface metamodels in damage identification of dynamic structures,” 2003.
- [3] J. D. Inman, C. R. Farrar, V. L. Junior, and V. S. Junior, *Damage Prognosis*. John Wiley and Sons, 2005.
- [4] G. Vachtsevanos, F. Lewis, M. Roemer, A. Hess, and B. Wu, *Intelligent Fault Diagnosis and Prognosis for Engineering Systems*. John Wiley & Sons, 2006.
- [5] P. C. Paris and F. Erdogan, “A critical analysis of crack propagation laws,” *Journal of Basic Engineering*, vol. 85, no. 4, pp. 528–534, 1963.
- [6] Y. Harris, “A new stress-based fatigue life model for ball bearings,” *Tribology Transactions*, vol. 44, pp. 11–18, 2001.
- [7] A. R. Scott, “Characterizing system health using modal analysis,” *IEEE Transactions on Instrumentation and Control*, vol. 58, pp. 297 – 302, Feb. 2009.
- [8] B. Saha, K. Goebel, S. Poll, and J. Christophersen, “An integrated approach to battery health monitoring using bayesian regression and state estimation,” *IEEE Autotestcon Proc.*, 2007.
- [9] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*. New York Academic, 1970.

- [10] S. Arulampalam, S. Maskell, N. J. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Processing*, vol. 50, no. 2, pp. 174–188, Feb. 2002.
- [11] M. E. Orchard, B. Wu, and G. Vachtsevanos, "A particle filtering framework for failure prognosis," *Proceedings of WTC2005 World Tribology Congress III*, September 2005.
- [12] B. Saha, K. Goebel, S. Poll, and J. Christophersen, "Prognostics methods for battery health monitoring using a Bayesian framework," *Instrumentation and Measurement, IEEE Transactions on*, vol. 58, no. 2, pp. 291–296, 2009.
- [13] J. Shiroishi, Y. Li, S. Liang, T. Kurfess, and S. Danyluk, "Bearing condition diagnostics via vibration and acoustic emission measurements," *Mechanical Systems and Signal Processing*, vol. 1, no. 5, pp. 693–705, 1997.
- [14] F. O. Heimes, "Recurrent neural networks for remaining useful life estimation," *International Conference on Prognostics and Health Management*, 2008.
- [15] L. Peel, "Data Driven Prognostics using a Kalman Filter Ensemble of Neural Network Models," *International Conference on Prognostics and Health Management*, 2008.
- [16] T. Wang, J. Yu, D. Siegel, and J. Lee, "A similarity-based prognostics approach for remaining useful life estimation of engineered systems," *International Conference on Prognostics and Health Management*, 2008.
- [17] L. Yao, "Robust design goal formulations and metamodeling techniques," School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, Tech. Rep., 2004.

- [18] W. E. Biles, “Design of simulation experiments,” *Proceedings of the 1984 Winter Simulation Conference (WSC)*, pp. 99–104, 1984.
- [19] G. P. Box and N. R. Draper, *Empirical Model Building and Response Surfaces*. John Wiley and Sons, New York, 1987.
- [20] R. H. Myers, A. I. Khuri, and W. H. Carter, “Response surface methodology: 1966-1988,” *Technometrics*, vol. 31, no. 2, pp. 137–157, 1989.
- [21] G. Matheron, “Principles of geostatistics,” *Economic Geology*, vol. 58, pp. 1246 – 1266, 1963.
- [22] N. A. C. Cressie, *Statistics for Spatial Data*. John Wiley and Sons, New York, 1993.
- [23] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, “Design and analysis of computer experiments,” *Statistical Science*, vol. 4, pp. 409–435, 1989.
- [24] G. G. Wang and S. Shan, “Review of metamodeling techniques in support of engineering design optimization,” *Journal of Mechanical Design*, vol. 129, no. 4, pp. 370– 380, 2007.
- [25] V. C. P. Chen, “Application of orthogonal arrays and mars to inventory forecasting stochastic dynamic programs,” *Computational Statistics and Data Analysis*, vol. 30, pp. 317–341, 1999.
- [26] D. E. Rumelhart, B. Widrow, and M. A. Lehr, “The basic ideas in neural networks,” *Communications of the ACM*, vol. 37, no. 3, pp. 87–92, 1994.
- [27] H. Siirtola, “Interactive visualization of multidimensional data,” Department of Computer Sciences, University of Tampere, Finland, Tech. Rep., 2007.

- [28] R. Jin, W. Chen, and T. W. Simpson, “Comparative studies of metamodeling techniques under multiple modelling criteria,” *Journal of Structural and Multidisciplinary Optimization*, vol. 23, no. 1, pp. 1–13, 2001.
- [29] S. R. Karnik, V. N. Gaitonde, and J. P. Davim, “A comparative study of the ANN and RSM modeling approaches for predicting burr size in drilling,” *The International Journal of Advanced Manufacturing Technology*, vol. 38, no. 9, pp. 868–883, 2008.
- [30] S. Sadjadi, H. Habibian, and V. Khaledi, “A multi-objective decision making approach for solving quadratic multiple response surface problems,” *International Journal of Contemporary Mathematical Sciences*, vol. 3, pp. 1595–1606, 2009.
- [31] A. Zaknich, *Neural Networks for Intelligent Signal Processing*. World Scientific Pub Co Inc, 2003.
- [32] A. Ramakrishnan and M. G. Pecht, “A life consumption monitoring methodology for electronic systems,” *IEEE Transactions on Components and Packaging Technologies*, vol. 26, no. 3, pp. 625–634, 2003.
- [33] Answers.com, “system,” <http://www.answers.com/topic/system>, 2010.
- [34] P. Soderholm, “Maintenance and continuous improvement of complex systems: Linking stakeholder requirements to the use of built-in test systems,” Department of Civil and Environmental Engineering, Lule University of Technology, Sweeden, Tech. Rep., 2005.
- [35] G. K. Hobbs, *Accelerated reliability engineering: HALT and HASS*. John Wiley & Sons, 2000.
- [36] WIKIPEDIA, “Stress (mechanics),” [http://en.wikipedia.org/wiki/Stress\(mechanics\)](http://en.wikipedia.org/wiki/Stress(mechanics)), Oct. 2010.

- [37] S. Nandi and H. A. Toliyat, "Condition monitoring and fault diagnosis of electrical machines -A review," in *Industry Applications Conference, 1999. Thirty-Fourth IAS Annual Meeting. Conference Record of the 1999 IEEE*, vol. 1, 1999.
- [38] "IEC Standard 60505," *Evaluation and Qualification of Electrical Insulation Systems*.
- [39] M. Kaufhold, G. Borner, M. Eberhardt, and J. Speck, "Failure mechanism of the interturn insulation of low voltage electric machines fed by pulse-controlled inverters," *IEEE Electrical insulation magazine*, vol. 12, no. 5, pp. 9–16, 1996.
- [40] J. H. Hoover and D. P. Boden, "Failure mechanisms of lead/acid automotive batteries in service in the USA," *Journal of Power Sources*, vol. 33, no. 1-4, pp. 257–273, 1991.
- [41] H. Bindner, T. Cronin, P. Lundsager, J. Manwell, U. Abdulwahid, and I. Baring-Gould, "Lifetime modelling of lead acid batteries," *Benchmarking*, vol. 12, p. 82, 2005.
- [42] A. Dasgupta and M. Pecht, "Material failure mechanisms and damage models," *IEEE Transactions on Reliability*, vol. 40, pp. 531 – 536, Dec. 1991.
- [43] A. Dasgupta, "Failure mechanism models for cyclic fatigue," *IEEE Transactions on Reliability*, vol. 42, pp. 548 – 555, Dec. 1993.
- [44] A. Dasgupta and J. Li, "Failure mechanism models for creep and creep rupture," *IEEE Transactions on Reliability*, vol. 42, pp. 339 – 353, Dec. 1993.
- [45] C. S. Byington, M. J. Roemer, G. J. Kacprzynski, T. Galie, and I. T. L. R. NY, "Prognostic enhancements to diagnostic systems for improved condition-based maintenance," 2002.

- [46] M. A. Miner, "Cumulative fatigue damage," *ASME Journal of Applied Materials*, vol. 12, pp. A159–A164, 1945.
- [47] Z. Liu, "A methodology for probabilistic remaining creep life assessment of gas turbine components," School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, Tech. Rep., 2002.
- [48] D. S. Steinberg, *Preventing Thermal Cycling and Vibration Failures in Electronic Equipment*. New York: Wiley, 2001.
- [49] A. E. Perkins, "Investigation and prediction of solder joint reliability for ceramic area array packages under thermal cycling, power cycling, and vibration environments," School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, Tech. Rep., 2007.
- [50] D. Barker, "Combined Vibrational and Thermal Solder Joint Fatigue - A Generalized Strain versus Life Approach ," *ASME Journal of Electronic Packaging*, vol. 112, no. 2, p. 129, 1990.
- [51] C. K. Lin and H. Y. Teng, "Creep properties of Sn-3.5 Ag-0.5 Cu lead-free solder under step-loading," *Journal of Materials Science: Materials in Electronics*, vol. 17, no. 8, pp. 577–586, 2006.
- [52] A. Singh, "Development and validation of an S-N based two phase bending fatigue life prediction model," *Journal of Mechanical Design, Transactions of the ASME*, vol. 125, no. 3, p. 540, 2003.
- [53] O. Jin, H. Lee, and S. Mall, "Investigation into cumulative damage rules to predict fretting fatigue life of Ti-6Al-4V under two-level block loading condition ," *Journal of Engineering Materials and Technology, Transactions of the ASME*, vol. 125, no. 3, p. 315, 2003.

- [54] S. S. Manson and G. R. Halford, “Practical implementation of the double linear damage rule and damage curve approach for treating cumulative fatigue damage,” *International Journal of Fracture*, vol. 17, no. 2, pp. 169–192, 1981.
- [55] H. Azzam, “A practical approach for the indirect prediction of structural fatigue from measured flight parameters,” *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 211, no. 1, pp. 29–38, 1997.
- [56] H. C. Frey and D. E. Burmaster, “Methods for characterizing variability and uncertainty: Comparison of bootstrap simulation and likelihood-based approaches,” *Risk Analysis*, vol. 19, no. 1, pp. 109–130, 1999.
- [57] Y. Wen and C. Basaran, “An analytical model for thermal stress analysis of multi-layered microelectronic packaging,” *Mechanics of Materials*, vol. 36, no. 4, pp. 369–385, 2004.
- [58] J. B. Bowles, “Fundamentals of failure modes and effects analysis,” *Annual Reliability and Maintainability Symposium Tutorial Notes*, 2003.
- [59] M. Pecht, *Product reliability, maintainability, and supportability handbook*. CRC, 1995.
- [60] H. O. Fuchs, D. V. Nelson, M. A. Burke, and T. L. Toomay, “Shortcuts in cumulative damage analysis,” in *SAE National Automobile Engineering Meeting*.
- [61] J. A. Collins, *Failure of materials in mechanical design: Analysis, prediction, prevention*. Wiley-interscience, 1993.
- [62] NIST, *e-Handbook of Statistical Methods*. <http://www.itl.nist.gov/div898/handbook/>: NIST/SEMATECH, 2003.

- [63] K. T. Fang, D. K. J. Lin, P. Winker, and Y. Zhang, “Uniform design: Theory and application,” *Technometrics*, vol. 39, no. 3, pp. 237–248, 2000.
- [64] R. Jin, W. Chen, and A. Sudjianto, “On sequential sampling for global meta-modeling in engineering design,” in *Proceedings of DETC*, vol. 2. Citeseer, 2002.
- [65] M. J. Sasena, P. Papalambros, and P. Goovaerts, “Exploration of metamodeling sampling criteria for constrained global optimization,” *Engineering Optimization*, vol. 34, no. 3, pp. 263–278, 2002.
- [66] R. Myers, D. C. Montgomery, and C. M. Anderson-Cook, *Response Surface Methodology; Process and Product Optimization Using Designed Experiments*. Wiley Series in Probability and Statistics, 2009.
- [67] H. Kohen, G. F. C. Rogers, and H. I. H. Saravanamuttoo, *Gas Turbine Theory*. T. J. Press, 1996.
- [68] A. W. A. Arebi, “The benefit of compressor cleaning on power output for oil and gas field applications,” 2005.
- [69] M. E. B. Smith, “A parametric physics based creep life prediction approach to gas turbine blade conceptual design,” School of Aerospace Engineering, Georgia Institute of Technology, Tech. Rep., 2008.
- [70] M. Naeem, R. Singh, and D. Probert, “Implications of engine deterioration for a high-pressure turbineblades low-cycle fatigue (LCF) life-consumption,” *International Journal of Fatigue*, vol. 21, pp. 831–847, 1999.
- [71] T. Sourmail, *Coatings for Turbine Blades*. <http://www.msm.cam.ac.uk/phase-trans/2003/Superalloys/coatings/index.html>, 2003.

- [72] A. Hamed and W. Tabakoff, "Erosion and deposition in turbomachinery," *Journal of Propulsion and Power*, vol. 22, no. 2, pp. 350–360, 2006.
- [73] R. Viswanathan, *Damage Mechanisms and Life Assessment of High-Temperature Components*. Metals Park, Ohio 44073: ASM International, 1989.
- [74] G. F. Harrision and T. Homewood, "The Application of the Graham and Walles Creep Equation to Aeroengine Superalloys," *Journal of Strain Analysis*, vol. 29, no. 3, 1994.
- [75] D. Woodford, "Creep Analysis of Directionally Solidified GTD111 based on stress relaxation testing," *Materials at High Temperatures*, vol. 14, no. 4, pp. 413–420, 1997.
- [76] N. E. Dowling, *Mechanical Behavior of Materials*. Upper Saddle River, NJ: Prentice-Hall Inc., 1993.
- [77] R. Danzer, "The influence of creep on the fatigue life of nickel based superalloys," *Proceedings of the 5th International Conference on Creep of Materials*, 1992.
- [78] R. A. Ainsworth, P. J. Budden, and R. Ohtani, "Design and assessment of components subjected to creep," *Journal of Strain Analysis*, vol. 29, no. 3, 1994.
- [79] Gas Path Analysis Ltd , "Turbine Creep Life Cycle analysis," <http://www.gpal.co.uk/xcreep.htm>, Jun. 2010.
- [80] D. Frederick, J. DeCastro, and J. Litt, *User's Guide for the Commercial Modular Aero-Propulsion System Simulation (CMAPSS)*, NASA/ARL, 2007.
- [81] S. Jayaram and F. Rivera, "An object-oriented method for the definition of mission profiles for aircraft design," *Proceedings of AIAA 32nd Aerospace Sciences Meeting*, 1994.

- [82] T. M. Murray, *Improved Fire- and Smoke-Resistant Materials for Commercial Aircraft Interiors: A Proceedings*. Washington, D.C.: National Academy Press, 1995.
- [83] K. Goebel, N. Eklund, and B. Brunell, “Rapid detection of faults for safety critical aircraft operation,” in *Aerospace Conference, 2004. Proceedings. 2004 IEEE*, vol. 5. IEEE, 2004, pp. 3372–3383.
- [84] H. Demuth, M. Beale, and M. Hagan, *Neural Network Toolbox User’s Guide*. The Mathworks, 2010.