Reliability assessment of TMD-based control structures : A Statistical Learning Perspective

Weizhen You

Numéro d'ordre NNT: 2020LYSEC01 Année: 2020

THESE de DOCTORAT DE L'UNIVERSITÉ DE LYON
OPÉRÉE AU SEIN DE L'ÉCOLE CENTRALE DE LYON

ÉCOLE DOCTORALE MEGA

Mécanique, Enegétique, Génie civil et Acoustique

Soutenue le 28 janvier 2020 par

Weizhen YOU

Reliability Assessment of TMD-based Control Structures: A Statistical Learning Perspective

---

en vue d'obtenir le titre de

Docteur de l'École Centrale de Lyon

Devant le jury composé de:

Abdelkhalak Elhami	Professeur, Normandy University	Rapporteur
Imad Tawfiq	Professeur, Supméca-Institut supérieur de mécanique de Paris	Rapporteur
Michelle Salvia	Maître de Conférence, ECL	Examinateur
Sergio De Rosa	Professeur, Universita' degli Studi di Napoli "Federico II"	Examinateur
Michel massenzio	Professeur, IUT Lyon 1	Président
Mohamed Ichchou	Professeur, ECL	Directeur
Alexandre Saidi	Maître de Conférence, ECL	Co-directeur
Abdel-malek Zine	Maître de Conférence, ECL	Co-directeur

Acknowlegements

During my three-year PhD research, I received help and encouragement from different persons in various ways. Thanks to my parents. They always care so much about me and my life in France. When I had academic difficulties and felt stressful, they always tried to encourage me. Their encouragement is just like a beacon in the darkness. I would like to thank my thesis directors, colleagues and other collaborate researchers for their contribution of time and intelligence into this project, they are:

Prof. Mohamed ICHCHOU,

Prof. Alexandre SAIDI,

Prof. Abdel-Malek ZINE

and Prof. ZHONG Xiaopin; Dr. Elyes Mrabet, Dr. YU Hang, Dr. YIN Xiangnan, Dr. LV Ying, Dr. Christophe DROZ, Dr. Pascal FOSSAT and Mr. Nassardine GUENFOUD. After countless discussions and meetings with everyone can we eventually accomplish such a complex task of structural reliability research. Together I shall note the names of jury members who attended my thesis defence: Prof. Abdelkhalak elhami, Prof. Tawfiq Imad, Prof. Sergio De Rosa and Prof. Michelle Salvia . Thank you for the efforts made during these months that finally make my research approved. Secondly, as an international student whose hometown is ten thousand kilometers away from Ecully, I would like to express my gratitude to those who lead me to integrate into my student life in Centrale Lyon: Ms M. Belhadi, Ms M. Badalan, Ms S. Stamkulov, Ms Chaouki, Dr. Ravi, Dr. CHAI Wenqi, Dr. LOU Mingyang, Dr. LUO Lingkun, Dr. HUANG Xingrong, Dr. YANG Yifan, Ms. SUN Cheng, Dr. YI Kaijun, Dr. WANG Xiaofang, Dr. G. Mazzeo and Dr. Fabien. You opened up my window towards the vast land of France and Europe continent and taught me how to live a meaningful life during these years.

YOU Weizhen

Ecole Centrale de Lyon, January 28, 2020

Abstract

The study of structural reliability mainly concerns the evaluation and prediction of the risk of limit state violation for an engineering structure at any stage of its life. Reliability evaluation helps improve structure design and product quality, which is of great significance for companies and consumers. It is also the basis of reliability modeling and prediction. Vibration control is a technique to reduce the energy of a vibrating structure when it is excited by external forces. This technique is widely used in various systems, such as buildings, bridges, machine tools and vehicles. Reliability prediction helps companies make production planning and implement preventive maintenance.

To do the reliability assessment, a theoretical analysis model is firstly developed. Due to complex interior and exterior factors, the uncertainties in structural properties as well as those in the stochastic excitation have made reliability analysis more difficult to apply. Traditional reliability analysis is generally based on professional knowledge or explicit performance functions, which has been unrealistic for today's systems that are more complex and nonlinear due to advanced design methodologies. In fact, reliability analysis involves estimations of the so-called conditional failure probability (CFP) that can be seen as a regression problem taking the structural uncertainties as input and the CFPs as output. In this situation, growing attention has been paid to non-parametric statistical learning theories. The prediction of CFP can be realized by machine learning models, such as decision trees, Support vector machines, ensemble learning methods, etc. These models are attracting more and more attention in recent published researches. This research aims to build a theoretic framework that integrates machine learning theories in structural reliability analysis and explore their performances on different structures. This framework is of high efficiency and accuracy for structural systems that have uncertainties in both structural properties and excitation.

Keywords: Structural reliability, Uncertainties, Statistical learning

Résumé

L'étude de la fiabilité structurelle concerne principalement l'évaluation et la prédiction du risque de violation de l'état limite d'une structure d'ingénierie à n'importe quelle étape de sa vie. L'évaluation de la fiabilité permet d'améliorer la conception de la structure et la qualité des produits, ce qui est d'une grande importance pour les entreprises et les consommateurs. Le contrôle des vibrations est une technique pour réduire l'énergie d'une structure vibrante lorsqu'elle est excitée par des forces externes. Cette technique est largement utilisée dans divers systèmes, tels que les bâtiments, les ponts et les véhicules. La prévision de fiabilité aide les entreprises à planifier la production et à mettre en une maintenance préventive.

Pour faire l'évaluation de la fiabilité, un modèle d'analyse théorique est d'abord développé. En raison de facteurs intérieurs et extérieurs complexes, les incertitudes des propriétés structurelles ainsi que celles de l'excitation stochastique ont rendu l'analyse de la fiabilité plus difficile à appliquer. L'analyse de fiabilité traditionnelle est généralement basée sur des connaissances professionnelles ou des fonctions de performance explicites, ce qui était irréaliste pour les systèmes d'aujourd'hui qui sont plus complexes et non linéaires. En fait, l'analyse de fiabilité implique des estimations de la soi-disant probabilité de défaillance conditionnelle (CFP) qui peut être considérée comme un problème de régression prenant les incertitudes structurelles en entrée et les CFP en sortie. Dans cette situation, une attention croissante a été accordée aux théories de l'apprentissage statistique non paramétrique. La prédiction de la CFP peut être réalisée par des modèles d'apprentissage automatique, tels que les arbres de décision, les machines à vecteurs de support, les méthodes d'apprentissage d'ensemble, etc. Cette recherche vise à construire un cadre théorique qui intègre les théories du machine learning dans l'analyse de la fiabilité structurale et à explorer leurs performances sur différentes structures. Ce cadre est d'une efficacité et d'une précision élevées pour les systèmes structuraux qui présentent des incertitudes à la fois en termes de propriétés structurelles et d'excitation.

Mots-clés: La fiabilité structurelle, Incertitudes, Apprentissage statistique

Contents

General Introduction

Reliability evaluation helps improve structure design and product quality, which is of great significance for companies and consumers. It is also the basis of reliability modeling and prediction. Vibration control is a technique to reduce the energy of a vibrating structure when it is excited by external forces. This technique is widely used in various systems, such as buildings, bridges, machine tools and vehicles. Reliability prediction helps companies make production planning and implement preventive maintenance. To do the predictions, a reliability model is firstly determined. Due to complex interior and exterior factors, the structure properties always deviate their design values. The structural uncertainties play an important role in reliability modeling. Traditional reliability models are commonly based on a priori information and professional knowledge, which has been unrealistic for today's systems that are more complex and nonlinear due to advanced design methodologies. In this situation, growing attention has been paid to non-parametric statistical learning approaches. Seen as a classification/ regression procedure, the prediction task can be realized by machine learning models, such as decision trees, Support vector machines, ensemble learning methods, etc. These models are attracting more and more attention in recent published researches. In this research, we aim to investigate the machine learning theories and explore the application of these models in structural reliability analysis. Besides, we try to develop a new system reliability assessment method that is of high efficiency and high accuracy for structural systems that have uncertainties in the structural properties and that is subjected to stochastic excitations.

Chapter 1

Introduction

Contents

1.1 General

Reliability describes the ability of a system or component to function under stated conditions for a specified period of time [1]. Benefiting from more than half a century's development, 'reliability' has become one of the key roles that determine the product performance. Nowadays, increasingly competitive markets drive companies to produce more advanced systems to meet various requirements of the customers. Meanwhile, the complexity of the products increase drastically and the failure mechanisms become more difficult to analyze. We notice that various 'smart' products are becoming even closer to human life, making larger influence on their comfort

and safety. These conditions make reliability evaluations more and more important to apply. Structural reliability aims at quantifying the risk of failure of systems due to uncertainties in their design, manufacturing and environmental conditions. Vibration control is a technique to reduce the energy of a vibrating structure when it is excited by external forces. This technology is widely used in various systems, such as buildings, bridges, machine tools and vehicles. Reliability research in structural systems remains an open problem due to uncertainties and different failure mechanisms. This dissertation presents contributions to structural reliability analysis, modeling and prediction in passive control mode.

1.2 Background

1.2.1 Vibration control of structural systems

Depending on the locations, certain civil structured facilities can be subjected to dynamic loads due to gusty wind fronts or strong ground motion associated with earthquake events of different intensity/severity during their life service. At high levels of intensity these naturally occurring dynamic loads may induce permanent structural damage and, in extreme cases, total structural failure/collapse. During the last three decades the incorporation of various devices such as energy dissipation equipment (e.g. viscous dampers, friction dampers, etc.), and tuned-mass dampers have been considered by various researchers and has been applied in practice to passively control the vibratory motion of structures maintaining its amplitude below certain acceptable thresholds ([2], [3], [4], [5]). Typically, such "non-conventional" means of mitigating the hazard posed to structures due to the action of winds and earthquakes are applied to protect critical civil infrastructure such as high-rise buildings, hospitals, and long-span bridges. Furthermore, the employment of such passive devices is commonly considered to reinforce existing structures to meet the contemporary safety criteria and to retrofit damaged structures in the aftermath of severe seismic events. These practical applications have sustained the important and active research field of passive vibration control for various structures. Admittedly, it is noted that the improved structural performance can also be achieved by using

active/semi-active control solutions relying on the integration of sensors, controllers and real-time data processing ([3], [6], [7]). However, due to reliability issues and the installation cost of such solutions, the use of active control systems is not as wide spread as the passive control solutions. In the context of passive vibration control, the concept of the dynamic vibration absorber is historically one of the first and most widely used strategies for passive vibration mitigation of dynamically excited mechanical and civil engineering structures and structural components ([8], [9]). It relies on attaching an additional free-to-vibrate mass to the structural system (primary or host structure) whose motion is to be suppressed via certain mechanical devices. These devices are appropriately designed (or "tuned") such that a resonant out-of-phase motion of the attached mass is achieved compared to the primary structure. Arguably, the most commonly used dynamic vibration absorber is the so-called "tuned mass-damper" (TMD). In its simplest form, the TMD considers a linear spring and a viscous damper to link the additional mass to the primary structure. The effectiveness of this classical TMD relies on tuning its stiffness and damping properties such that significant kinetic energy is transferred from the vibrating primary structure to the TMD mass and is absorbed through the viscous damper. Especially during the recent one decade, most of the researchers have been focusing on passive control using tuned mass dampers (TMD) in randomly driven linear structures ([10], [11], [12], [13], [14], [15], [16]). The use of the TMDs in passive linear structural control is increasingly demanded in the field of engineering because of its efficiency and low cost. Indeed, the TMDs, belonging to the general class of vibration dampers, are very economical and efficient devices to mitigate the structural vibrations caused by external dynamic loadings.

1.2.2 Uncertainties in structures

The incorporation of uncertainty into systems analysis is an on-going topic of significance. For realistic systems, uncertainties can be involved in the design stage, in the manufacturing progress and during the service. In the design stage, uncertainties are derived from mathematical modeling process or incomplete knowledge about the system. Uncertainties in the manufacturing progress are reflected by the manufac-

turing tolerance, material scatter on account of the limited precision in tools and processes or the lack of advanced technologies. During the service, uncertainties in the excitations (such as seismic loads, waves, temperature changes and any other kind of environmental loads) and boundary conditions as well as human factors are the major concerns. As any structure has a limited age, uncertainty is usually induced from the deterioration of material properties. In this aspect, uncertainties may be imposed upon geometry tolerances, material properties or excitations. These uncertainties can influence the performances of the system during its lifetime. Uncertainty analysis, or in other words, how to mathematically quantify uncertainties, has become an indispensable task that should be carefully accomplished in different stages of the structural system. During the last two decades, engineers and scientists have paid much attention to uncertainties of various types in the system. Various representations and theories for uncertainties exist, each of which is characterized by a distinct mathematical model based on the information on hand. Primarily, uncertainties can be classified as three variants: aleatory uncertainty, epistemic uncertainty and error. The variants as well as their explanations are illustrated in Figure 1.1.

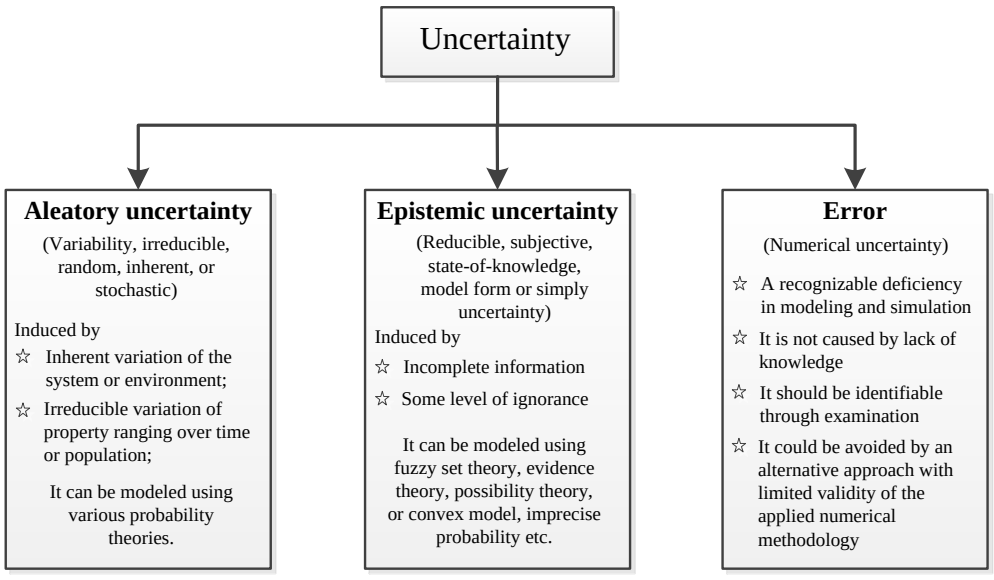

Figure 1.1: Classification of uncertainties [[17], [18]]

As is shown in Figure 1.1, aleatory uncertainty can also be understood as variability, irreducible uncertainty, inherent uncertainty or stochastic uncertainty. The term 'aleatory' is used to specialize the inherent characteristics of the variations related to the physical system or the environment of interest. Aleatory uncertainty is a kind of uncertainty that generally has priori-known statistical properties. Accordingly, the probability theories can be applied. In contrast, epistemic uncertainty is also termed as reducible uncertainty, subjective uncertainty or model form uncertainty. Epistemic uncertainty is generally induced by incomplete information, some level of ignorance of the system or the environment. The associated uncertainty analysis (UA) method depends mainly on non-probabilistic theories that are commonly known as generalized information theory. It mainly includes possibility theory, fuzzy set theory and evidence theory. As to the last term error, it could be reduced or even avoided by careful examinations or advanced methods.

1.2.3 Failure mechanisms

Failure mechanism is a physical process in which stresses cause damage to the elements comprising the system, ultimately leading to system failure [19]. According to [19] and [20], the failure mechanisms can be classified into two subgroups: overstress failures (such as large elastic deformation, yield and fracture) and wear-out failures (such as wear, corrosion and fatigue). In the first case, a failure event occurs when the stress to which the structure is subjected exceeds the allowable maximum strength. If the stress is below the threshold value, the stress has no permanent effect on the structure. In the second case, the stress causes damage that generally accumulates irreversibly. The accumulated damage does not reduce when the stress is removed, although sometimes annealing is possible. The cumulative damage does not cause any performance degradation when it is below the endurance limit. Once this limit is reached, the structure fails. As to random vibration, similar failure mechanisms can also be found in [21]: (1) Failure may occur at the first time that the random response exceeds a certain level; (2) Failure occurs when the accumulated damages reaches a fixed amount. The estimation of first-passage (or first excursion) probability is usually in conjunction with the former case; while the

aging engineering belongs to the latter one. In this work, we concentrate on the over-stress or the first-passage failure mechanism. Simply, when the structure is subject to static loads, the over-stress mechanism is considered; when the structure is under dynamic loads, the first passage problem is regarded.

1.2.4 Structural reliability

Structural reliability is concerned with the rational treatment of uncertainties in structural engineering and with the methods for assessing the safety and serviceability of civil engineering and other structures [22]. It is a subject that has grown rapidly during the last decade and has evolved from being a topic for academic research to a set of well-developed or developing methodologies with a wide range of practical applications. Reliability is commonly defined as [22] "the probability of a device performing its purpose adequately for the period of time intended under the operating conditions encountered". There are four elements to the definition that must be considered. First, probability refers to the likelihood that a device or structural component will work properly. These terms imply acceptance of some degree of uncertainty. The second element refers to adequate performance. In order to determine whether a component has performed adequately, a standard is needed to define what is meant by adequate performance. The third element is the intended period of time. This is the mission endurance or lifetime of the structure under consideration. The final element of the definition is the operating conditions. Environmental conditions play a significant role in determining structural reliability. Simply stated, structural reliability is a yardstick of the capability of a structure to operate without failure when put into service. As implied in the definition, structural failure and, hence, reliability, is influenced by many factors. In its simplest form, the measure of reliability is made by comparing a component's stress to its strength. Failure occurs when the stress exceeds the strength. The larger this gap, the higher the reliability and the heavier the structure. Conversely, the smaller the gap, the lower the reliability, but the lighter the structure. The gap between stress and strength, enforced by the factor-of-safety, generally produces adequate although unmeasured reliability. Generally, the structural reliability changes in different peri-

ods of its life, based on various interior and exterior factors. There are mainly three factors listed as below.

- Static strength A structure's capability to sustain operational loads is commonly assessed by comparing material performance parameters to limit or ultimate loads. Limit loads are generally defined as the maximum load expected during the life of the structure.

- Environmental effects Environmental factors of major importance include a combination of humidity and temperature. Many studies have been conducted to investigate moisture absorption as well as the reduction of mechanical properties due to temperature and moisture exposure. Some of the approaches used to account for environmental factors is to define exposures that are extremes and selectively evaluate by test the effects on material properties. These extremes are then considered to be invariant during the lifetime of the structure. Strength values are reduced to coincide with the environmental extremes.

- Fatigue is the gradual deterioration or progressive structural damage of a material that occurs when the material is exposed to repeated loads.

1.3 Basic contents of structural reliability analysis

One of the problems in stochastic mechanics is the estimation of the probability density of one or several structural responses. This kind of problem can be termed full probabilistic analysis. A derived, more practical problem is to find a specific probability of exceeding a response threshold that can be considered critical for the serviceability or the safety of the system. This is the basic task of reliability analysis. Notice that, in principle the second problem could be solved after the first, as the probability of failure can eventually be obtained by integration of the probability density function of the observed response. However, most reliability methods developed in the last decades attempt to estimate directly the failure probability or related reliability indices. Actually, the failure probability is calculated from an

integration within the failure region that is defined by a particular performance function.

Performance function In the reliability problem, the input random variables are collected in a vector X, whose deterministic counterpart is the vector $\mathbf{x}$. This defines the input space. The $performance function$ $G(\mathbf{x})$ (also named limit-state function, LSF) is defined in such a way that $G(\mathbf{x}) > 0$ means that the sample $\mathbf{x}$ is in the safe domain $S = \mathbf{x} : G(\mathbf{x}) > 0$; $G(\mathbf{x}) < 0$ implies that $\mathbf{x}$ is in the failure domain $F = \mathbf{x} : G(\mathbf{x}) < 0$. Specially, $G(\mathbf{x}) = 0$ is the boundary between the two domains. For example, the performance function of a structural component can be defined as

$$G = R - S \tag{1.1}$$

where S is a load effect on the component; R is the strength capacity to withstand it. The safety limit state will be violated whenever $G(\mathbf{x}) < 0$. The probability that this occurs for any single load application is the probability of limit state violation, or simply the probability of failure P_f. Generally, P_f is represented by the overlap area of the probability distributions of the two variables (see Figure 1.2). Theoretically,

$$P_f = P(G(\mathbf{x})) = \iint_{G(\mathbf{x})<0} f_{RS}(r,s)\mathrm{d}r\mathrm{d}s \tag{1.2}$$

Since this overlap may vary with the time, so P_f may be a function of time. More generally, assume that $\mathbf{x} = [x_1, x_2, ..., x_n]$ is the vector of n-dimension variables. $\mathbf{x}$ consists of the fundamental variables that define and characterize the behavior and safety of a structure. Accordingly, the probability of failure is denoted as [23]

$$P_f = \int \cdots \int_{G(X)\leq 0} f_X(x_1, x_1, ..., x_n)\mathrm{d}_{x_1}...\mathrm{d}_{x_n} \tag{1.3}$$

where $f_X(.)$ denotes the joint probability density function (pdf) of input vector . If the basic variables are all independent, $f_X(.)$ is simplified as:

$$f_X(\mathbf{x}) = \prod_{i=1}^{n} f_X(x_i) = f_X(x_1) * f_X(x_2)... * f_X(x_3) \tag{1.4}$$

Here $f_X(x_i)$ is the marginal probability density function for the basic variable x_i. The region of integrand $G(X) \leq 0$ in (1.3) denotes the hyper-space where the limit

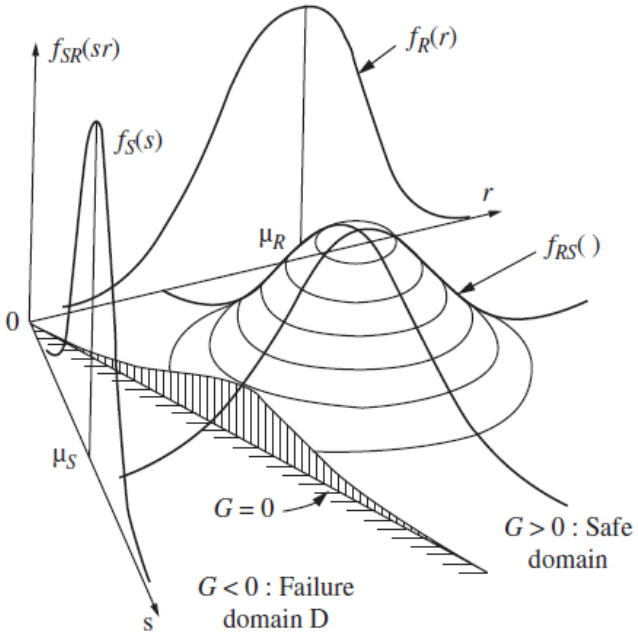

Figure 1.2: Space of two random variable (r, s) and their joint density function $f_{RS}(r, s)$; their marginal density functions f_R and f_S; the failure domain D. [24]

state violation occurs. It is directly analogous to the failure domain D. The integral of (1.3) over the failure domain $G(X) \leq 0$ cannot be performed analytically. Therefore, section 1.4 is devoted to various reliability analysis methods to approximate this integral.

1.4 Methods for reliability analysis

The methods based on the limit-state function can be grouped into two families named as analytic and synthetic, according to whether the random variable set and its effects are treated with the tools of probability theory or with those of statistics (see Figure 1.3). In the first case we have two kinds of methods based on Taylor-series expansion of the limit state function, which are known as FORM (First-Order Reliability method) and SORM (Second-Order Reliability Method). As is well known, the FORM method requires information on the value of the performance function and its gradient in the vicinity of the so-called design point and the SORM method needs in addition the knowledge of the second-order derivatives.

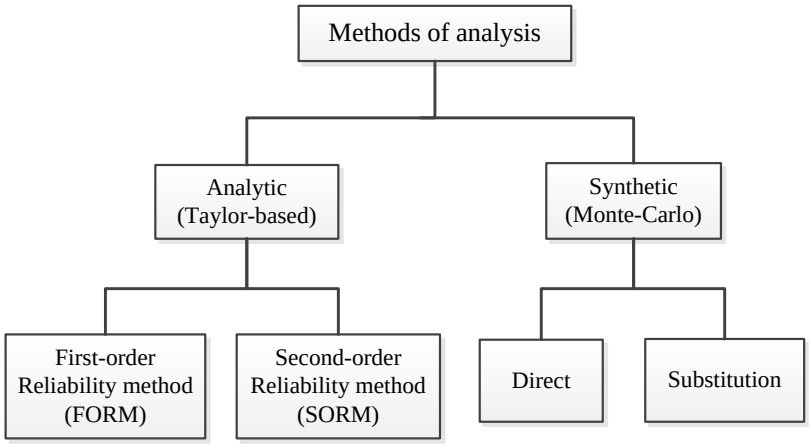

Figure 1.3: Reliability methods based on limit state functions [25]

In the second case, sample sets of the basic variables are generated and processed by a numerical solver (generally a finite element code) of the structural model in order to obtain a population of structural responses over which statistical analyses can be conducted. This is called Monte Carlo simulation (MCS). For the reliability assessment, MCS approximates the reliability by the frequency that the generated samples drop into the safe region ([26], [27], [28]).The Monte Carlo method is distinguished by its entire generality.

Nonetheless, it is a well-known fact that it requires a large computational labor. With respect to FORM and SORM it requires random samples instead of deterministic ones and its scaling with dimensionality is rather weak. Indeed, together with its generality, this later characteristic is a positive advantage. However, its computational cost remains a problem. This is not only due to the fact that it requires the many calls of the numerical solver for the assessment of any statistical quantity but also that in using it this quantity becomes a random variable. In order to diminish the computing labor of Monte Carlo methods, research efforts oriented to a reduction of the number of samples and the spread of the estimates obtained at each simulation epoch have been exerted in the last years. Sometimes, however, these developments are accompanied by a sacrifice of one or more of the attractive properties of the basic Monte Carlo method. For instance, some of these variance-

reduction methods (for example, importance sampling) require an exploration for locating the design point or the estimation of an optimal density function, which is a task greatly affected by the curse of dimensionality. Anyhow, these research efforts evidence the need of reducing the very high cost of MCS especially for the case of large structures. As the complexity increases drastically and the failure mechanisms became more difficult to recognize, new theories were derived by the researchers.

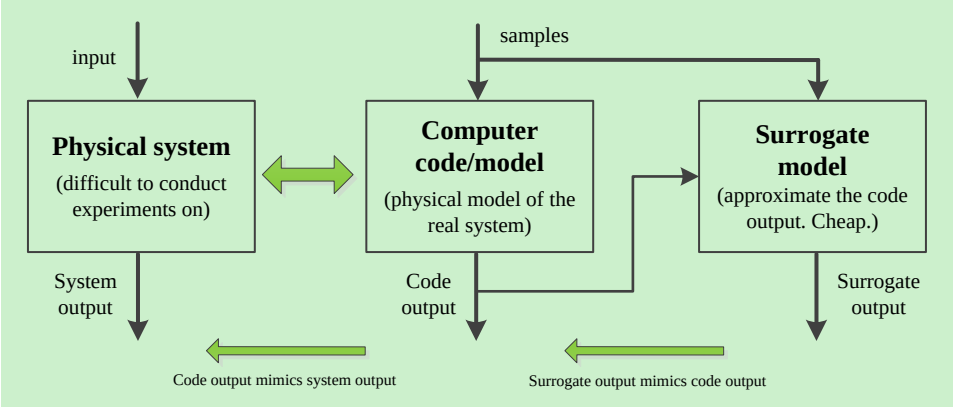

Figure 1.4: Evolution from physical model to surrogate model

As an alternative, the non-parametric models have been the focus of structural reliability research. An important step in this direction was made in the eighties with the application of the Response Surface Method (RSM) developed in the field of Design of Experiments. RSM is a collection of mathematical and statistical techniques for exploring the relationships between independent variables and response variables [29]. Its basic idea is to find a surrogate polynomial function of the basic variables that approximates the implicit performance function. The surrogate is then employed for most of the samples needed in the entire MCS. For this purpose, an experimental design is generated for calculating the undetermined coefficients of the surface so as to minimize the error of approximation, particularly in the region around the design point. The surrogate so obtained can also be connected to FORM and SORM methods. The development of RSM ([30], [31]) opened a new area of structural reliability research that can be labeled as the investigation on solver surrogate methods. The evolution from the physical models to surrogate models are

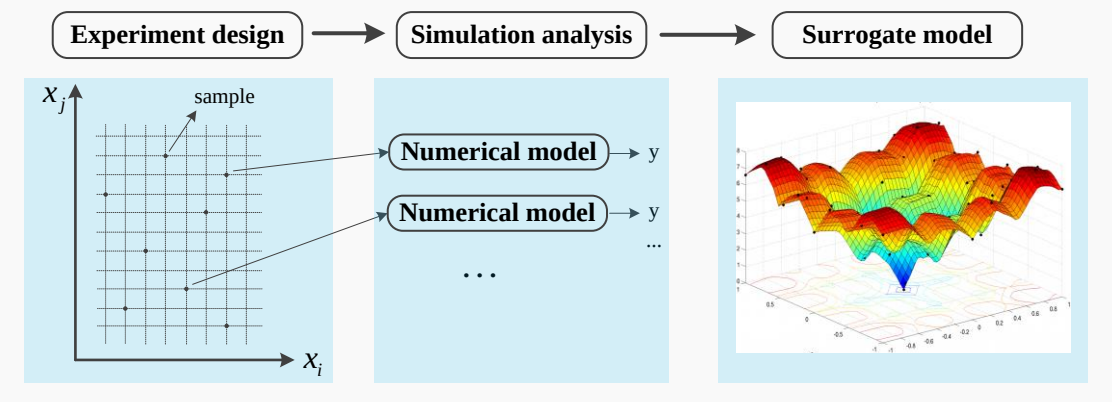

Figure 1.5: Basic procedures to build a surrogate model

depicted in Figure 1.4, and the basic procedures to build the surrogate models are shown in Figure 1.5. Other analogous models include polynomial chaos expansions ([32], [33]) and Kriging model [34]. Note that taking into account the refinement and sophistication of finite element models, the practical application of MCS could be greatly favored by the availability of such a substitute to the actual deterministic solution, since only some finite element solver calls would be needed for building the surrogate. In recent years, however, several important findings in this direction have been achieved. For example, some serious deficiencies of the RSM that make it unreliable for practical use have been highlighted. Fortunately, the development of Statistical Learning Theory and algorithms, such as Random Forests (RF) [6], Artificial Neural Networks (ANN) ([35], [36]) and Support Vector Machines (SVM) ([37], [38], [39]), afforded new ways of obtaining the surrogate of the performance function. Attention has been paid by researchers to the fact that structural reliability problems can be treated as a statistical classification task determined by the LSF. For this purpose, the modern pattern recognition methods become useful.

RF is an ensemble learning method that use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone. RF is comprised of many individual trees called classification and regression tree (CART). Each tree is built from a random sample set (i.e. bootstrap sample) generated from the training set. A tree learning process is

actually a recursive data partitioning process based on the principle of minimal sum of squared errors (SSE). Other representative tree-based methods include Gradient Boosting (GB) [6] and Extremely Randomized trees (ETs) [40]. Stacked Generalization (Stacking) involves training a meta-learner from the predictions of several base learners. First, all of the base learners are trained using the available data, then a meta-learner is trained from the predictions of the base learners. Stacking typically yields performance better than any single trained models [41]. It has been successfully applied on both regression [42] and classification. An ANN consists of some artificial neurons and connections between them. An artificial neuron is a mathematical function that receives one or more inputs and sums them to produce an output. Usually each input is separately weighted, and the sum is passed through a non-linear function known as an activation function. Training of ANN involves adjusting the parameters, or the weights and biases, of the model in order to minimize error. Back propagation is used to make those weigh and bias adjustments relative to the error, and the error itself can be measured in a variety of ways, including by root mean squared error (RMSE).

Similar to ANN, support vector machine (SVM) is an intelligent learning method for pattern recognition. However, the theoretical bases of SVM and ANN are different. ANN is based on empirical risk minimization (ERM) principle, while SVM's foundation is structural risk minimization (SRM) principle. ANN can obtain the minimization risk for the training samples due to the application of the ERM principle, but for the un-trained samples ANN may give unreliable estimation. SRM principle can improve generalization ability of learning machine by minimizing the total of empirical risk and confidence bound. Theoretically, other machine learning tools such as k-nearest neighbors (kNN) are also an alternative. According to the current research state, we believe that the explorations are far from enough, but more and more researchers are trying to introduce these methods into structural reliability domain. This dissertation devotes itself totally to this topic.

1.5 Structural reliability analysis process

Generally, the reliability analysis of an engineering structure includes three parts. Determine the basic variables that construct the structural model; express the probability distributions of these variables; define the performance function for the object structure. Based on these three tasks, the basic procedure of engineering structural reliability analysis is shown in Figure 1.6.

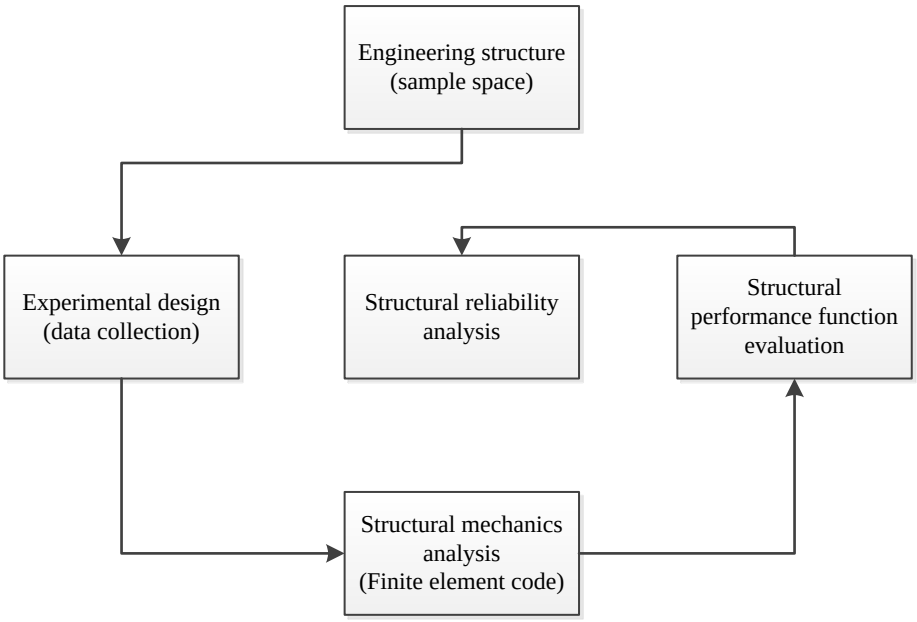

Figure 1.6: Basic procedure of engineering structural reliability analysis

1.6 Research scope and objectives

Uncertainties are inherent in structural systems, such as stochastic excitation, stochastic loading and epistemic model uncertainties. In this study, we intend to contribute in reliability analysis and prediction of structures in passive control mode.Derive a high-accurate, computationally efficient reliability evaluation method for passive control structures;Explore machine learning methods for structural reliability prediction, and compare their performances from different aspects;

1.7 Dissertation organization

In chapter 1, we present the background of reliability research on vibration control structures, investigate failure modes of structural systems and give a literature review on methods of reliability assessment. The objectives and scope of our research are pointed out as well.

In chapter 2, the fundamental knowledge of reliability analysis and reliability modeling techniques for structural systems are introduced. In this chapter, we begin with the classical reliability analysis theories that provide the basic knowledge of reliability analysis. Uncertainty analysis is subsequently followed, in which the basic probability theories are introduced and the associated methods are reviewed. They constitute the foundation of this dissertation.

In Chapter 3, the theory of uncertainty quantification is introduced and analyzed in detail.The concept of conditional failure probability is developed. Machine learning models are introduced as surrogates to model conditional failure probabilities.

In Chapter 4, to deal with passive controlled structures subject to earthquake induced vibrations, we propose a novel reliability modeling method based Random Forest and Monte-Carlo Simulation . Random Forest is distinguished due to its robustness and high accuracy in modeling and prediction work. Therefore, this part devotes to exploring the feasibility of RF and examining its performance in modeling and prediction of structural reliability. In this part, different complexity of structures are studied. We also test other representative machine learning methods, and compare them with RF in different aspects.

In Chapter 5, Tree-based ensemble methods, such as RF, always behave well, but further studies are needed to improve the prediction performance. Stacking method tries to build the prediction model in a hierarchical way, resulting in a meta-learner induced from the predictions of various base learners. Recent research shows that, with a relatively small price of CPU time, the Stacking model can largely improve reliability predictions compared with individual base learners. The prediction model is learned on a training set that is provided by Monte-Carlo simulations. In numerical simulations, different Stacking models are explored and their performances are

compared and analyzed from different perspectives. Bias-variance analysis of the prediction error is performed to evaluate the model in a more general way. Time complexity of the prediction model is also introduced and analyzed in detail.

In Chapter 6, some conclusions are drawn on this dissertation research and provide perspectives for future work.

Chapter 2

Reliability assessment of structural system: the fundamentals

Contents

2.1 General

In the chapter of introduction, we have emphasized the necessary and the importance of reliability engineering in the structural system. In this chapter, we begin with

fundamental knowledge of reliability analysis, including classic analysis methods as well as modern statistical learning models. Among the structural reliability analysis methods, we introduce the mostly-used ones, such as first/second order reliability method (FORM/SORM), Moment-based methods, Response surface methods, Simulation-based methods, Classification-based methods and Time-variant reliability methods.

The methods based on statistical learning theories are presented, including Ensemble learning methods (Random Forest, Adaptive Boosting), k-nearest neighbor method, Support Vector machines and Artificial Neural Networks.

2.2 General principle of structural reliability analysis

A structural system is defined as a structure that is required to provide specific functionality under well-defined safety constraints. Such constraints need to be taken into account during the system design phase in view of the expected environmental/ operating loads it will be subject to. In the presence of uncertainties in the physical properties of the system (*e.g.* due to tolerances in the manufacturing), in the environmental loads (*e.g.* due to exceptional weather conditions), or in the operating conditions (*e.g.* traffic), it can occur that the structure operates outside of its nominal range. In such cases, the system encounters a failure. Structural reliability analysis deals with the quantitative assessment of the probability of occurrence of such failures (probability of failure), given a model of the uncertainty in the structural, environmental and load parameters. Following the formalism introduced in [43], this part is intended as a theoretical introduction and of the available tools in structural reliability analysis. All the algorithms presented follow a black-box philosophy, *e.g.* they rely on the point-by-point evaluation of a computational model, without knowledge about its inner structure. Let us denote by X the vector of basic random variables. When considering models of mechanical systems, these variables usually describe the randomness in the geometry, material properties and loading [43]. They can also represent model uncertainties. This set also includes the variables used in the discretization of random fields, if any. The model of the system

yields a vector of response quantities $Y = M(X)$. In a mechanical context, these response quantities are *e.g.* displacements, strain or stress components, or quantities computed from the latter.

2.2.1 Limit-state function

A *limit state* can be defined as a state beyond which a system no longer satisfies some performance measure. Regardless on the choice of the specific criterion, a state beyond the limit state is classified as a *failure* of the system. Consider a system whose state is represented by a random vector of variables $X \in D_X \subset R^M$. One can define two domains D_s; $D_f \subset D_X$ that correspond to the $safe$ and $failure$ regions of the state space D_X, respectively. In other words, the system is failing if the current state $\mathbf{x} \in D_f$ and it is operating safely if $\mathbf{x} \in D_s$. This classification makes it possible to construct a $limit - statefunction$ $g(X)$ that assumes positive values in the safe domain and negative values in the failure domain:

$$\begin{aligned} \mathbf{x} \in D_s &\Leftrightarrow g(\mathbf{x}) > 0 \\ \mathbf{x} \in D_f &\Leftrightarrow g(\mathbf{x}) < 0 \end{aligned} \tag{2.1}$$

The hyper-surface in M dimensions defined by $g(\mathbf{x}) = 0$ is known as the limit-state surface, and it represents the boundary between safe and failure domains. A graphical representation of D_s, D_f and the corresponding limit-state surface $g(\mathbf{x}) = 0$ is given in Figure 2.1.

2.2.2 Failure Probability

If the random vector of state variables X is described by a joint probability density function (PDF) $f_X(x)$, then one can define the failure probability P_f as:

$$P_f = P(G(X) < 0) \tag{2.2}$$

This is the probability that the system is in a failed state given the uncertainties of the state parameters. The failure probability is then calculated as follows:

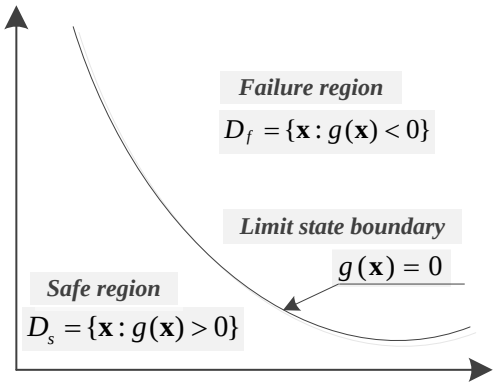

Figure 2.1: Safe and failure regions

$$P_f = \int_{D_f} f_X(\mathbf{x}) d\mathbf{x} = \int_{x:g(x)<0} \mathrm{d}x \tag{2.3}$$

Note that the integration domain in Eq. (2.3) is only implicitly defined by Eq. (2.2), hence making its direct estimation practically impossible in the general case. This limitation can be circumvented by introducing the indicator function of the failure domain, a simple classifier given by:

$$1_{D_f}(x) = \begin{cases} 1, & g_(x) < 0 \\ 0, & g_(x) > 0 \end{cases} \tag{2.4}$$

In other words, $1_{D_f}(x) = 1$ when the input parameters x cause the system to fail and $1_{D_f}(x) = 0$ otherwise. This function allows one to cast Eq. (2.3) as follows:

$$P_f = \int_D 1_{D_f}(x) f_X(x) \mathrm{d}x = E[1_{D_f}(x)] \tag{2.5}$$

where $E[\cdot]$ is the expectation operator with respect to the PDF $f_X(x)$. This reduces the calculation of P_f to the estimation of the expectation value of $1_{D_f}(x)$.

In this section, we investigate various methods to cope with the calculation of Eq. (2.5). The methods are classified into three main categories: analytical approximation techniques and simulation-based methods. Besides these two categories we can

also find some other not commonly-used methods out of our scope, such as stochastic finite element method (SFEM) for structural reliability which is time-consuming. Furthermore, the time-variant or time-dependent reliability attracts more and more attention. We therefore also review time-variant modeling methods. On the other hand, when no performance function but test data exists, reliability analysis has to rely on statistical methods, such as Bayesian reliability.

2.3 Strategies for the estimation of P_f

From the definition of $1_{D_f}(x)$ in Section 2.2 it is clear that determining whether a certain state vector $\mathbf{x} \in D_X$ belongs to D_s or D_f requires the evaluation of the limit-state function $g(\mathbf{x})$. In the general case this operation can be computationally expensive, *e.g.* when it entails the evaluation of a computational model on the vector x. For a detailed overview of standard structural reliability methods and applications, see e.g. Ditlevsen and Madsen [44]; Melchers [24]; Lemaire [45]. In the following, three strategies are discussed for the evaluation of P_f, namely approximation, simulation and adaptive surrogate-model based methods, see the classification in Figure 2.2.

2.3.1 Analytical approximation methods

Approximation methods are based on approximating the limit-state function locally at a reference point (e.g. with a linear or quadratic Taylor expansion). This class of methods can be very efficient (in that only a relatively small number of model evaluations is needed to calculate), but it tends to become unreliable in the presence of complex, non-linear limit state functions. Two approximation methods are introduced in this dissertation, i.e. FORM and SORM.

2.3.1.1 First Order Reliability Method (FORM)

The first order reliability method aims at the approximation of the integral in eq.(1.3) with a three-step approach: 1) An iso-probabilistic transform of the input random vector $\mathbf{x} \sim f_X(\mathbf{x})$ into a standard normal vector $\mathbf{u} \sim N(0, I_n)$; 2) A

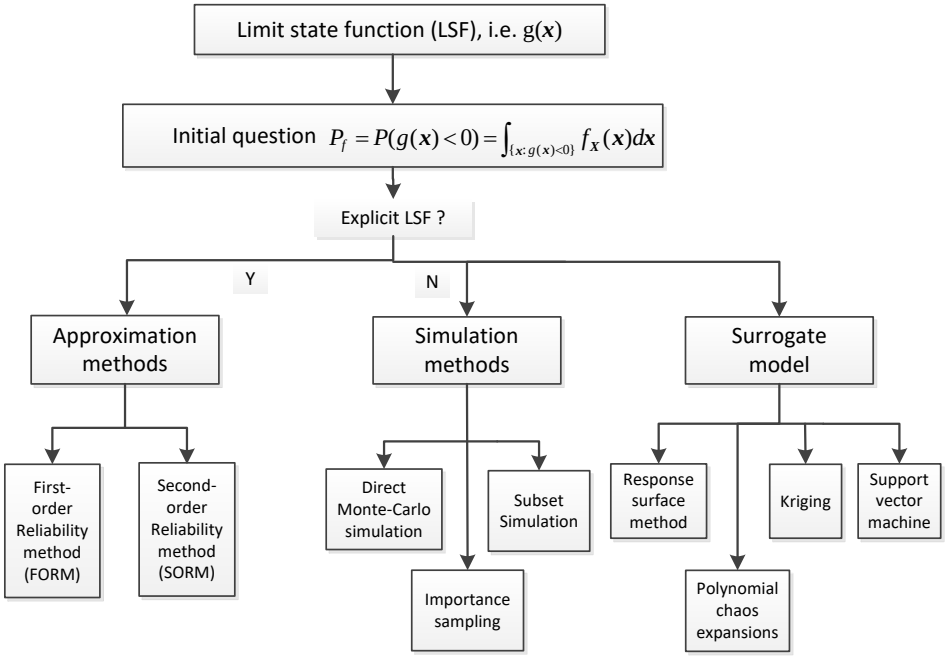

Figure 2.2: Classification of strategies to estimate probability of failure ([46], [47], [48])

search for the most likely failure point in the standard normal space (SNS), known as the design point $\mathbf{u}^*$; 3) A linearization of the limit-state surface at the design point $\mathbf{u}^*$ and the analytical computation of the resulting approximation of P_f.

Iso-probabilistic transform

The first step of the FORM method is to transform the input random vector $\mathbf{x} \sim f_X(x)$ into a standard normal vector $T(\mathbf{x}) \sim N(0, I_n)$. This transform can be used to map the integral in eq.(1.3) from the physical space of X to the standard normal space of U:

$$P_f = \int_{D_f} f_X(\mathbf{x})\mathrm{d}\mathbf{x} = \int_{u \in R^n : G(u) < 0} \phi_n(u)\mathrm{d}\mathbf{u} \tag{2.6}$$

where $G(u) = g(T^{-1}(u))$ is the limit-state function evaluated in the standard normal space and $\phi_n(u)$ is the standard multivariate normal PDF given by:

$$\phi_n(u) = (2\pi)^{M/2} exp(-\frac{1}{2}(u_1^2 + u_2^2 + ... + u_n^2)) \tag{2.7}$$

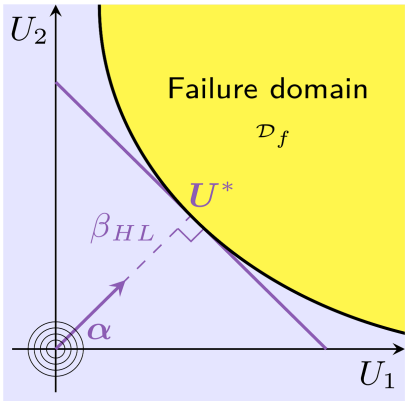

Figure 2.3: Graphical representation of the linearization of the limit-state function around the design point at the basis of the FORM estimation of P_f. ([43])

This PDF is maximal at the origin and decreases exponentially with $||u||^2$. Thus the points that contribute at most to the integral in eq.(2.6) are those of the failure domain that are closest to the origin of the space. Thus the second step in FORM consists in determining the so-called design point , i.e. the point of the failure domain closest to the origin in the standard normal space. This point is obtained by solving an optimization problem:

$$u^* = argmin(||u||^2), s.t. G(u) < 0, u \in R^n. \tag{2.8}$$

Due to the probability measure in Eq.(2.6), u^* can be interpreted as the most likely failure point in the standard normal space. The norm of the design point $i.e.||u||$, is an important quantity in structural reliability known as the Hasofer-Lind reliability index: $\beta_{HL} = ||u^*||$. An important property of the index is that it is directly related to the exact failure probability Pf in the case of linear LSF in the standard normal space: $P_f = \Phi(-\beta_{HL})$, where Φ is the standard normal cumulative density function. The estimation of P_f in the FORM algorithm is based on approximating the limit-state function as the hyperplane tangent to the limit-state function at the design point. Figure 2.3 illustrates this approximation graphically for the two-dimensional case.

To search for the design point in the general non-linear case, eq.(2.8) may be cast as a constrained optimization problem with Lagrangian:

$$L(u,\lambda) = \frac{1}{2}||u||^2 + \lambda G(u) \tag{2.9}$$

where λ is the Lagrange multiplier. The related optimality conditions read:

$$\begin{aligned} \bigtriangledown_u L(u^*,\lambda^*) &= 0 \\ \frac{\partial L}{\partial \lambda}(u^*,\lambda^*) &= 0 \end{aligned} \tag{2.10}$$

The standard iterative approach to solve this nonlinear constrained optimization problem is given by the Rackwitz-Fiessler algorithm [49].

2.3.1.2 Second Order Reliability Method (SORM)

The SORM is an improved version of the FORM for P_f estimate. After the design point U^* is identified by FORM, the failure probability is approximated by a tangent hyper-paraboloid defined by the second order Taylor expansion of $G(U^*)$ given by:

$$G(U) \approx \bigtriangledown^T_{|u^*} \cdot (u-u^*) + \frac{1}{2}(u-u^*)^T H (u-u^*) \tag{2.11}$$

where H is the Hessian matrix of the second derivatives of $G(U)$ evaluated at U^*. The failure probability in the SORM approximation can be written as a correction factor of the FORM estimate that depends on the curvatures of the hyper-hyperboloid in eq.(2.11). To estimate the curvatures, the hyper-paraboloid is firstly cast into canonical form by rotating the coordinates system such that one of its axes is the vector. Usually the last coordinate is chosen arbitrarily for this purpose. A rotation matrix Q can be built by setting α as its last row and by using the Gram-Schmidt procedure to orthogonalize the remaining components of the basis. Q is a square matrix such that $Q^T Q = I$. The resulting vector V satisfies:

$$U = QV \tag{2.12}$$

In the new coordinates system and after some basic algebra (see e.g. [50]), one can rewrite eq.(2.11) as:

$$G(V) \approx || \bigtriangledown G(u^*)|| \cdot (\beta - V_n) + \frac{1}{2}(V - V^*)^T Q H Q^T (V - V^*) \tag{2.13}$$

where β is the Hasofer-Lind reliability index calculated by FORM, $V_n = \alpha^T (Q^T V)$ and $V^* = 0, ..., \beta^T$ is the design point in the new coordinates system. By dividing eq.(2.13) by the gradient norm $|| \bigtriangledown G(U*)||$ and introducing the matrix $A = QHQ^T / || \bigtriangledown G(U*)||$. One obtains:

$$\tilde{G}(V) \approx \beta - V_n + \frac{1}{2}(V - V^*)^T A (V - V^*) \tag{2.14}$$

where $\tilde{G}(V) = G(V) / || \bigtriangledown G(U*)||$. After neglecting second-order terms in V_n and diagonalizating the A matrix via eigenvalue decomposition one can rewrite eq.(2.14) explicitly in terms of the curvatures κ_i of an hyper-paraboloid with axis α:

$$\tilde{G}(V) \approx \beta - V_n + \frac{1}{2} \sum_{i=1}^{M-1} \kappa_i V_i \tag{2.15}$$

For small curvatures $\kappa_i < 1$, the failure probability P_f can be approximated by the Breitung formula [50]:

$$P_{f,SORM} \approx \Phi(\beta_H L) \prod_{i=1}^{M-1} (1 + \beta_H L \kappa_i)^{-1/2}, \kappa_i < 1 \tag{2.16}$$

Note that for small curvatures the Breitung formula approaches the FORM linear limit. The accuracy of eq.(2.16) decreases for larger values of κ_i, sometimes even if $\kappa_i < 1$. A more accurate formula is given by the Hohenbichler formula [51].

2.3.2 Moment-based methods

Obviously, the FORM and SORM need to search for the design point by iteration which is time consuming. The moment-based methods [44] do not need the design point not to mention the iteration to find the design point. Using moments is the simplest way to describe a distribution. After transformed from the variable space to the standardized normal space, the simplified representation of the distribution facilitates the determination of failure probability, i.e. no design point needs to be found any more.

Second moment is a very popular one due to its simplicity. It considers only the first two moments (mean and variance) but not higher order moments, such as skewness and flatness. However only the normal distribution can be represented exactly by second moment. Hence the second moment methods work under the assumption that the random variable obeys a normal distribution. When the second moment density approximation and the 'first-order' approximation are brought together, we have the so-called first-order second moment (FOSM) reliability method [52]. When considering the non-linear performance, the advanced first-order second moment (AFOSM) [53] may be an appropriate method. The basic random variable is transformed into a reduced coordinate system and the Taylor's expansion is performed at design point rather than the mean value in FOSM.

As stated, the FOSM is based on the assumption of normal distribution. The higher order moments are suggested for other cases of more-complicated distribution. For example, the first-order third moment method (FOTM) was developed by Tichy [54] and the higher-order moment methods are reviewed by Zhao [55]. Those methods involves the skewness, flatness or higher order moment that has been ignored by the standard FOSM. However, it is obviously a drawback for the moment based methods that the process of obtaining statistical moments is very inefficient because that involves also numerical integrations.

2.3.3 Simulation based methods

Simulation methods are based on sampling the joint distribution of the state variables X and using sample-based estimates of the integral in eq.(2.5). At the cost of being computationally very expensive, they generally have a well-characterized convergence behavior that can be exploited to calculate confidence bounds on the resulting Pf estimates. Three sampling-based algorithms are introduced here, *i.e.* Monte Carlo simulation, Importance Sampling and Subset Simulation.

2.3.3.1 Monte Carlo Simulation

Monte Carlo (MC) simulation is used to directly compute the integral in eq. (1.3) by sampling the probabilistic input model. Given a sample of size N of the input random vector X, $X = x^{(1)}, ..., x^{(N)}$, the unbiased MCS estimator of the expectation value in Eq. (1.4) is given:

$$P_{f,MC} \approx \hat{P}_f = \frac{1}{N} \sum_{k=1}^{N} 1_{D_f}(x^{(k)}) = \frac{N_f}{N} \tag{2.17}$$

where N_f is the number of independent samples such that $g(x) < 0$. In other words, the Monte Carlo estimate of the failure probability is the fraction of samples that belong to the failure domain over the total number of samples. An advantage of Monte Carlo simulation is that it provides an error estimate. Indeed the indicator function $1_{D_f}(x)$ follows by construction a Bernoulli distribution with mean $\mu_{1_{D_f}} = P_f$ and variance $\sigma^2_{1_{D_f}} = P_f(1 - P_f)$. Hence, $\hat{P}_f$ has the variance $\hat{\sigma}^2_{P_f}$ given by

$$var(\hat{P}_f) = \frac{1}{N^2} var(\sum_{k=1}^{N} 1_{D_f}(x^{(k)})) = \frac{1}{N^2} \sum_{k=1}^{N} var(1_{D_f}(x^{(k)}) = \frac{N \cdot \sigma^2_{1_{D_f}}}{N^2} = \frac{\sigma^2_{1_{D_f}}}{N} \tag{2.18}$$

Typically such plots will show that these measures are reduced as the number of samples increases and that a degree of stability is reached at a sufficiently high number of samples. The confidence intervals on $\hat{P}_f$ can be given as follows [56]:

$$\hat{P}_f \in [\hat{P}_f + \hat{\sigma}_{P_f} \Phi^{-1}(\alpha/2), \hat{P}_f + \hat{\sigma}_{P_f} \Phi^{-1}(1 - \alpha/2)] \tag{2.19}$$

where $\Phi(x)$ is the standard normal cumulative distribution function (CDF) and $\alpha \in [0, 1]$ (significance level) is a scalar such that the calculated bounds correspond to a confidence level of $1 - \alpha$. An important measure for assessing the convergence of a MCS estimator is given by the coefficient of variation (COV) defined as:

$$COV = \frac{\sigma_{P_f}}{\hat{P}_f} = \sqrt{\frac{(1 - \hat{P}_f)}{N \cdot \hat{P}_f}} \tag{2.20}$$

The coefficient of variation of the MCS estimate of a failure probability therefore decreases with $\sqrt{N}$ and increases with decreasing P_f. To give an example, to es-

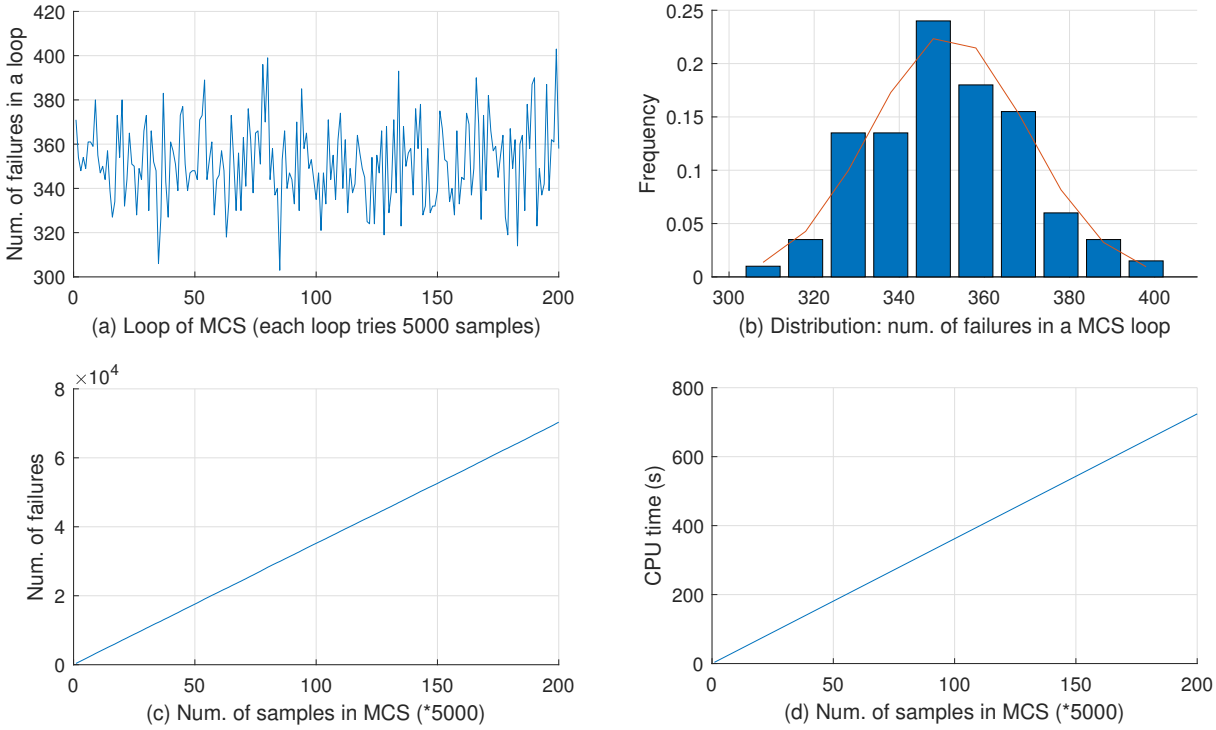

Figure 2.4: Some descriptions of failure probability convergence by MCS

timate a $P_f = 10^{-3}$ with 90% accuracy, $N = 10^5$ samples are needed. The COV is often used as a convergence criterion to adaptively increase the MC sample size until some desired COV is reached. In Figure 2.4 and Figure 2.5, some aspects of the P_f in its convergence process are illustrated. 200 times of simulations are implemented, each of which analyzes 5000 samples. In Figure 2.4, the subplots (a) and (b) shows how the number of failures in each loop fluctuates, but it follows a normal distribution; the subplot (c) shows the number of failures with respect to different size of sample set in the MCS, and the subplot (d) shows the corresponding CPU time. Figure 2.5 shows the convergence (with 95% confidence interval, or CI in short) of $\hat{P}_f$ with respect to different size of sample set in the MCS, and Figure 2.6 shows the corresponding coefficient of variation of $\hat{P}_f$.

The MCS method is powerful, when applicable, due to its statistically sound formulation and global convergence. However, its main drawback is the relatively slow

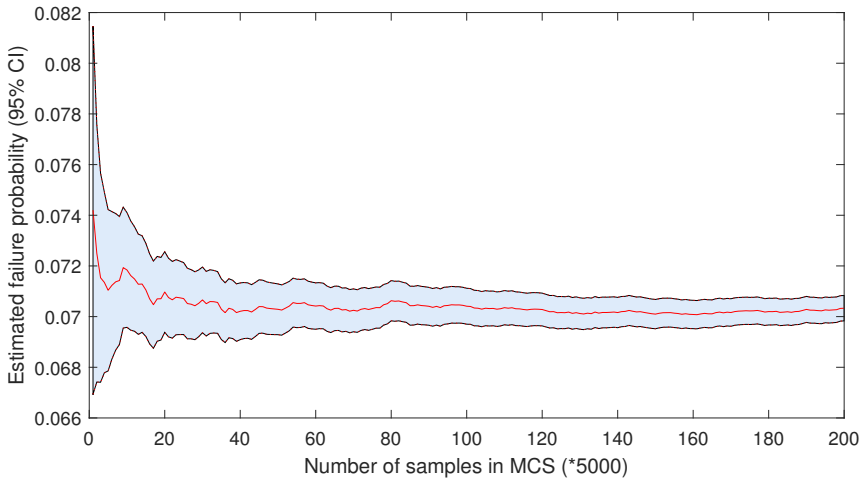

Figure 2.5: Estimated P_f with 95% CI

converge rate that depends strongly on the probability of failure. The variance reduction techniques, such as importance sampling [57] and subset simulations [58], have been proposed.

2.3.3.2 Importance Sampling

Importance sampling (IS) is an extension of the FORM and MCS methods that combines the fast convergence of FORM with the robustness of MCS. The basic idea is to recast eq. (1.3) as:

$$P_f = \int_{D_x} 1_{D_f}(x)\frac{f_X(x)}{\phi(x)}\phi(x)\mathrm{d}x = E_\phi[1_{D_f}(x)\frac{f_X(x)}{\phi(x)}] \tag{2.21}$$

where $\phi(x)$ is an M-dimensional importance distribution (also called instrumental /proposal distribution) and E_ϕ denotes the expectation value with respect to the same distribution. The estimate of P_f given a sample set $X = \mathbf{x}^{(1)}, ..., \mathbf{x}^{(n)}$ drawn from ϕ is therefore given by:

$$\hat{P}_f = \hat{E}_\phi[1_{D_f}(X)\frac{f_X(X}{\phi(X)}] = \frac{1}{N}\sum_{i=1}^{N}[1_{D_f}(\mathbf{x}^{(k)})w(\mathbf{x}^{(k)})] \tag{2.22}$$

where the weight $w(\mathbf{x}^{(k)}) = f_X(\mathbf{x}^{(k)})/\phi(\mathbf{x}^{(k)})$. Briefly speaking, the following procedures are executed to estimate $\hat{P}_f$: 1) Take N samples i.e. $X = \mathbf{x}^{(1)}, ..., \mathbf{x}^{(n)}$ from

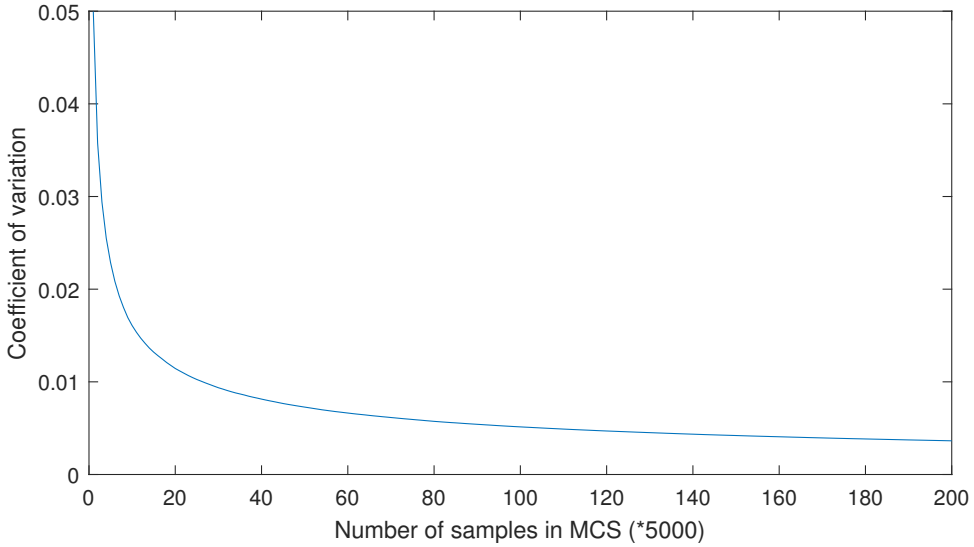

Figure 2.6: Coefficient of variation of the estimated P_f

the proposal distribution $\phi(x)$; 2) Calculate the target density $f_X(\mathbf{x}^{(k)})$ and proposal density $\phi(\mathbf{x}^{(k)})$; $f_X(\mathbf{x}^{(k)})$ is the joint PDF value of the uncertain parameters; 3) Calculate the weight $w(\mathbf{x}^{(k)})$ and the indicator value $1_{D_f}(\mathbf{x}^{(k)})$ for each sample $\mathbf{x}^{(k)}$; 4) Estimate the value $\hat{P}_f$ according to eq.(2.22). The most important task is to find a proposal distribution $\phi(X)$. The proposal distribution can be constructed from the design point u^*. In standard normal space, eq.(2.22) can be rewritten as:

$$\hat{P}_f = \hat{E}_\phi[1_{D_f}(T^{-1}(u))\frac{\phi_M(u)}{\Phi(u)}] \tag{2.23}$$

When the results from a previous FORM analysis are available, a particularly efficient sampling distribution in the standard normal space is given by [57]:

$$\Phi(u) = \phi_M(u - u^*) \tag{2.24}$$

where u^* is the estimated design point. Given a sample set $U = \mathbf{u}^{(1)}, ..., \mathbf{u}^{(N)}$ of $\phi(u)$, the estimate of P_f becomes:

$$\hat{P}_f = \frac{1}{N} exp(-\beta_{HL}^2/2) \sum_{i=1}^{N} 1_{D_f}(T^{-1}\mathbf{u}^i) exp(-\mathbf{u}^{(k)} \cdot U^*) \tag{2.25}$$

with corresponding variance:

$$\hat{\sigma}^2_{\hat{P}_f} = \frac{1}{N}\frac{1}{N-1}\sum_{i=1}^{N} 1_{D_f}(T^{-1}(\mathbf{u}^{(k)}))\frac{\phi(\mathbf{u}^{(k)})}{\Phi(\mathbf{u}^{(k)})} \tag{2.26}$$

The coefficient of variation and the confidence bounds can be calculated analogously to eqs (2.19) and (2.20), respectively, and can be used as a convergence criterion to adaptively improve the estimation of $\hat{P}_f$.

2.3.3.3 Subset Simulation

Monte Carlo simulation requires a large number of limit-state function evaluations to converge with an acceptable level of accuracy when P_f is small. Subset simulation is a technique introduced by Au and Beck [58] that aims at offsetting this limitation by solving a series of simpler reliability problems with intermediate failure thresholds. Consider a sequence of failure domains $D_1 \supset D_2 \supset ... \supset D_m = D_f$ such that $D_f = \cap_k = 1^k D_k$. With the conventional definition of LSF in Section 1.3, such sequence can be built with a series of decreasing failure thresholds $t_1 \supset t_2 \supset ... \supset t_m = 0$ and the corresponding intermediate failure domains $D_k = x : g(x) \leq t_k$. One can then combine the probability mass of each intermediate failure region by means of conditional probability. By introducing the notation $P(D_x) = P(x \in D_X)$ one can write [58]:

$$\hat{P}_f = P(D_m) = P(\cap_k = 1^m)P(D_k) = P(D_1)\prod_{i=1}^{m-1} P(D_{i+1}|D_i) \tag{2.27}$$

With an appropriate choice of the intermediate thresholds $t_1, t_2, ..., t_m$, eq.(2.27) can be evaluated as a series of structural reliability problems with relatively high probabilities of failure that are then solved with MC simulation. In practice the intermediate probability thresholds t_i are chosen on-the-fly such that they correspond to intermediate values $P(D_i) \approx 0.1$. The convergence of each intermediate estimation is therefore much faster than the direct search for P_f.

2.3.4 Surrogate model based methods

Surrogate model based methods are designed for large complex structures that generally do not have explicit LSF. For such structures, the probability of failure cannot be obtained by analytical methods; besides, MCS demand large numbers of samples and consume a lot of time. The basic principle of surrogate models is to find an alternative function that is equivalent to the real implicit LSF. The surrogate model is then treated as an explicit LSF such that the failure probability is evaluated. Obviously, the reliability calculation result depends largely on how closely the surrogate function approximates to the real limit state surface. Some representative models are Response surface method, Polynomial chaos expansion, Kriging and Support vector machine. Besides, the author find that, the machine learning models such as ensemble learning models and ANNs are attracting more and more attention recently. Therefore, these methods have large potentials. In the following, a detailed description of each of the methods is given.

2.3.4.1 Response surface methods

For complicated components, it is hard to find the close-form limit state functions. The moment based methods mentioned above cannot be applied directly. Some researchers ([59], [31]) proposed to use response surface to determine a substitute of the implicit performance functions. The basic idea is to suppose a simple parametric expression with respect to random variables then interpolate within some running results to estimate those parameters such that the interpolation error is minimized. The approximated performance function $\hat{g}(x)$ is called response surface function which is usually in polynomial form. For example, a second-order polynomial takes the form

$$\hat{g}(x) = a_0 + \sum_{i=1}^{n} a_i x_i + \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_i x_j \tag{2.28}$$

or

$$\hat{g}(x) = a_0 + \sum_{i=1}^{n} a_i x_i + \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ii} x_i^2 \tag{2.29}$$

where a_i, a_{ii} and a_{ij} are the coefficients that need to be determined; n is the number of random variables. For a complete second-order polynomial (see eq.(2.28)), the number of coefficients to be estimated is $K = 2n + 1 + n(n-1)/2$. Therefore, the size of the training data should be larger than K. Based on this approximation the FORM/SORM and all of the moment-based methods can be used after then.

2.3.4.2 Polynomial chaos expansion

The original PCE, also termed as the homogenous chaos expansion and Hermite polynomial chaos expansion (HPCE), was developed by Wiener [32]. With this expansion, any second-order random variable or stochastic process, i.e. the quantities with finite variance, may be expanded as follows:

$$u = \hat{u}_0 H_0 + \sum_{i_1=1}^{\infty} \hat{u}_{i_1} H_1(\xi_{i_1}) + \sum_{i_1=1}^{\infty}\sum_{i_2=1}^{i_1} \hat{u}_{i_1 i_2} H_2(\xi_{i_1}, \xi_{i_2}) + \sum_{i_1=1}^{\infty}\sum_{i_2=1}^{i_1}\sum_{i_3=1}^{i_2} \hat{u}_{i_1 i_2 i_3} H_3(\xi_{i_1}, \xi_{i_2}, \xi_{i_3}) + ... \tag{2.30}$$

where $H_p(\xi_{i_1}, ..., \xi_{i_p})$ denotes the multivariate Hermite polynomial chaos of order p in terms of standard normal vector $\xi = [\xi_{i_1}, ..., \xi_{i_p}]$ and $\hat{u}_{i_1 ... i_p}$ is the associated coefficient. For notational convenience, eq.(2.30) can be rewritten as

$$u = \sum_{i=1}^{\infty} u_i \Phi_i(\xi) \tag{2.31}$$

In this expression, there is a one-to-one correspondence between the polynomial basis functions $\Phi_i(\xi)$ and $H_p(\xi_{i_1}, ..., \xi_{i_p})$, also the deterministic coefficients u_i and $\hat{u}_{i_1,...,i_p}$. Two categories of methods are distinguished to determine the coefficients the intrusive method [58] and non-intrusive method [60]. When the random inputs are standard normal, a possible Hilbertian basis is the multivariate Hermite polynomial chaos basis.

2.3.5 Time-variant reliability

Most mechanical components experience changing the material properties and the running environments over time. In this case the performance is also a function of time or a stochastic process, denoted by $g_t(x,t)$. So does the reliability. Similarly the failure probability function of time is defined as

$$P_f(t) = P_r[g_t(\mathbf{x},t) \leq 0] = \int_{g_t(\mathbf{x},t)\leq 0} f(\mathbf{x},t)\mathrm{d}\mathbf{x} \tag{2.32}$$

where $f(\mathbf{x},t)$ is the density function of random variable and time. It is almost impossible to exploit the whole failure function of time. However the time variant failure probability is reduced to a time-invariant problem for a fixed time. All methods mentioned for time-invariant reliability can be applied. Sometimes and in some applications, the failure probability in a time interval is of more interest than the general time-variant failure probability. Suppose the time interval is denoted by $T_{in} = [t_1, t_2]$, then the time-interval failure probability is defined as

$$P_f(T_{in}) = P_r[g_t(\mathbf{x},t) \leq 0], t \in T_m \tag{2.33}$$

It is still difficult to compute the probability due to the continuity of time. This can be solved by various techniques, such as the work of Breitung [50], Li [61], Kuschel [49] and of Andrieu-Renaud [62]. The basic idea is to use the outcrossing rate over the limit state of the structure. The outcrossing event at time t is defined as the event that at time t the performance $g_t(\mathbf{x},t) > 0$ and at time $t+\mathrm{d}t$, $g_t(\mathbf{x},t+\mathrm{d}t) \leq 0$, where $\mathrm{d}t$) is a small time increment. Thus the mean value of the outcrossing rate at time t is estimated as follows

$$v_c(t) = \lim_{\delta t \to 0} \frac{P_r[g_t(\mathbf{x},t) > 0 \cap g_t(\mathbf{x},t) \leq 0]}{\delta t} \tag{2.34}$$

Usually it is a small probability that the random process $g_t(\mathbf{x},t)$ out-crosses the bound 0. In this case we can assume that two arbitrary outcrossing events are independent of each other. In other words, we assume that the total number of the outcrossing events in T_{in} obeys a Poisson distribution [63]. Finally we can get an approximation of the failure probability by using Poisson process model,

$$P_f(T_{in}) \approx exp\{-\int_{t_1}^{t_2} v_c(t)\mathrm{d}\mathbf{t}\} \tag{2.35}$$

For a component composed of multiple response processes, its failure probability is a joint probability of the individual events, i.e.

$$P_f(T_{in}) = P_r[\bigcup_{i\in I_{lim}} \min_{t\in T_{in}} g_t^{(i)}(\mathbf{x},t) \leq 0] \tag{2.36}$$

where $g_t^{(i)}(\mathbf{x},t)$ is ith limit state function under consideration and I_{lim} is the index set of the limit state functions. If the response processes are independent, the joint probability can be evaluated straightforwardly. Further, Song [64] investigated how to consider the reliability of a component who consists of multiple interdependent response processes or a vector random process.

2.3.6 Statistical learning models

When the performance functions are implicit as often encountered in real world, the analytical approximation techniques cannot do anything to find them and the Monte Carlo simulations require a million times of calculation on LSF. This is impractical in the complex cases because of the large consumption of time. One of the strategy to facilitate this difficulty is the use of polynomial function which approximates the implicit performance and is also known as response surface method. However it may be time-consuming to use the polynomial function when the components are complex and the number of random variables is large [65]. Moreover the approximate performance function is lack of adaptivity and flexibility and we cannot guarantee that it is sufficiently accurate to the true one. To address this issue, researchers proposed to use the statistical classifiers as an additional treatment or an approximation of performance. These methods are called classification-based methods. The mostly-used ones for reliability analysis are the methods using neural networks ([66], [67]) and those using support vector machine [68]. The readers are also referred to [25] in which Hurtado has investigated detailedly the structural reliability in statistical perspective and [69] the detailed aspects of the classifiers.

2.3.6.1 Support Vector Machine

Support vector machines [70] are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as a binary target value corresponding to one of the two categories, a SVM model is built that will mark a new example as one target value or the other. In this algorithm, each example is plotted as a point in the n-dimension space (n is the number of features in each example). Then a hyper-plane is found that differentiate the two classes optimally, meanwhile a margin is determined (see Figure 2.7). New examples are then mapped into that same space and predicted to belong to a class based on which side of the margin they fall. This hyper-plane generally takes the form $f(x) = w \cdot x + b$. It is constructed by solving the optimization problem below,

$$\min_{w,b} \frac{1}{2}||w||^2, s.t. y_i(w \cdot x_i + b) \geq 1, \forall i \tag{2.37}$$

In regression case, SVM is called Support Vector Regression, or SVR. As the output is a continuous value, it becomes difficult to predict as there are infinite possibilities. To handle this, a margin of tolerance ε is preset such that the optimization problem is formulated as (see Figure 2.8),

$$\min_{w,b} \frac{1}{2}||w||^2, s.t. |y_i - (w \cdot x_i + b)| > \varepsilon, \forall i \tag{2.38}$$

2.3.6.2 Artificial Neural Networks

Artificial neural networks (ANN) [71] are one of the main tools used in machine learning. An ANN is an interconnected group of nodes, inspired by a simplification of neurons in a brain. It generally consist of input and output layers, as well as a hidden layer consisting of units that transform the input into something that the output layer can use, see Figure 2.9. The central idea of ANNs is to extract linear combinations of the inputs as derived features, and then model the target as a non-linear function of these features. A basic unit of an ANN (i.e. a neuron) is shown

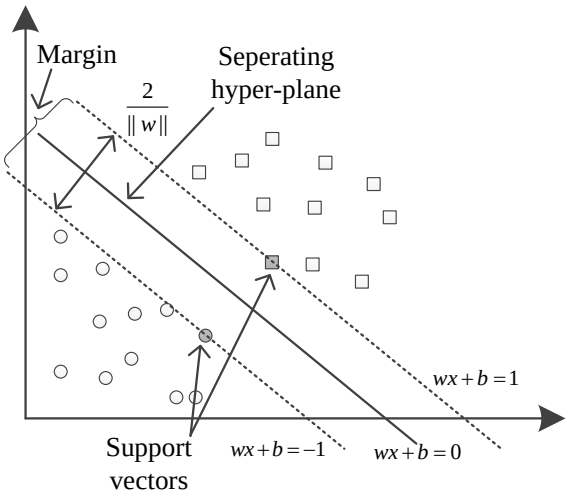

Figure 2.7: SVM for classification

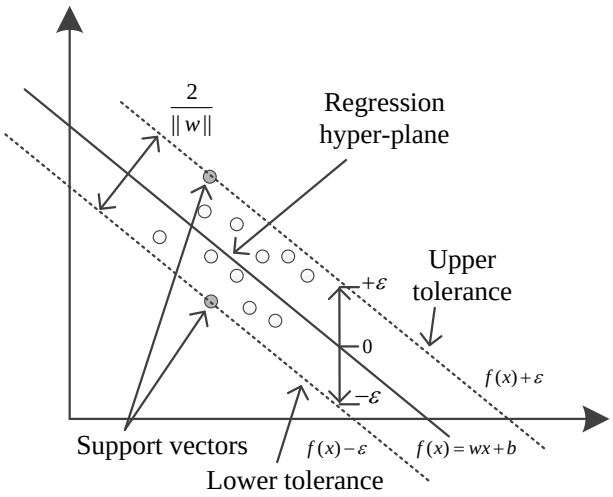

Figure 2.8: SVM for regression

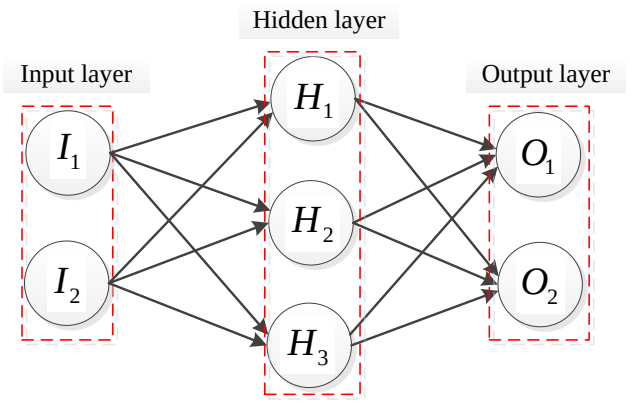

Figure 2.9: General structure of ANN

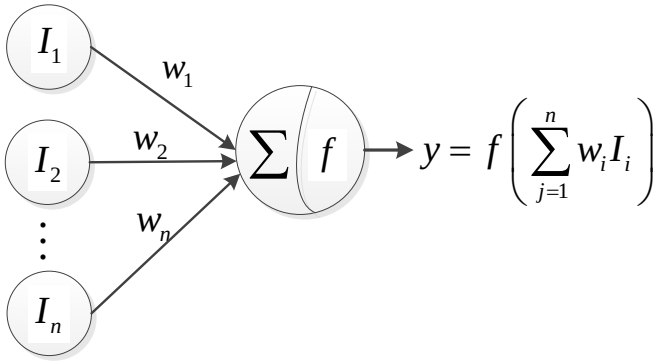

Figure 2.10: A neuron in ANN

in Figure 2.10. It consists of an artificial neuron and a number of input values $I_1, I-2, ..., I_n$.

Usually each input is separately weighted, and the sum is passed through a non-linear function (see Figure 2.10) known as an activation function or transfer function. The transfer functions usually have a sigmoid shape (see Figure 2.11), but they may also take the form of other non-linear functions. To train an ANN, one of the most popular algorithms is back propagation technique. After initially choosing the weights of the network randomly, the back propagation algorithm is applied to compute the necessary corrections. The algorithm is decomposed into the following

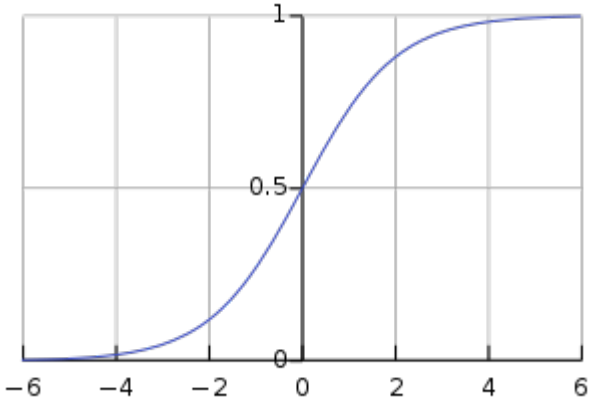

Figure 2.11: Sigmoid function

four steps [72]: a) Feed-forward computation; b) Back propagation to the output layer; c) Back propagation to the hidden layer; d) Weight updates. The algorithm is stopped when the value of the error function has become sufficiently small.

2.3.6.3 k-Nearest Neighbor (k-NN) method

The idea behind the k-NN algorithm is to build a classification method using no assumptions about the form of the function $y = f(x_1, x_2, ..., x_p)$ that relates the dependent variable, y to the independent variables $x_1, x_2, ..., x_p$. The only assumption we make is that it is a 'smooth' function. This is a non-parametric method because it does not involve estimation of parameters in an assumed function form. This algorithm is used for classification and regression. There are two important problems to solve in the process of learning. The value k is generally a positive, small integer which can directly affect the prediction results. Cross-validations can be applied to determine the optimal k. For two points $\mathbf{x}^{(1)} = (\mathbf{x}^{(1)}(1), ..., \mathbf{x}^{(1)}(n))$ and $\mathbf{x}^{(2)} = (\mathbf{x}^{(2)}(1), ..., \mathbf{x}^{(2)}(n))$, the distance is calculated using one of the following measures:

- Euclidean Distance: $d(x, y) = \sqrt{\sum_{k=1}^{n} (\mathbf{x}^{(1)}(k) - \mathbf{x}^{(2)}(k))^2}$;
- Minkowski Distance: $d(x, y) = (\sum_{k=1}^{n} |\mathbf{x}^{(1)}(k) - \mathbf{x}^{(2)}(k)|^p)^{1/p}$;
- Mahalanobis Distance: $d(x, y) = \sqrt{\sum_{k=1}^{n} |\mathbf{x}^{(1)}(k) - \mathbf{x}^{(2)}(k)|}$.

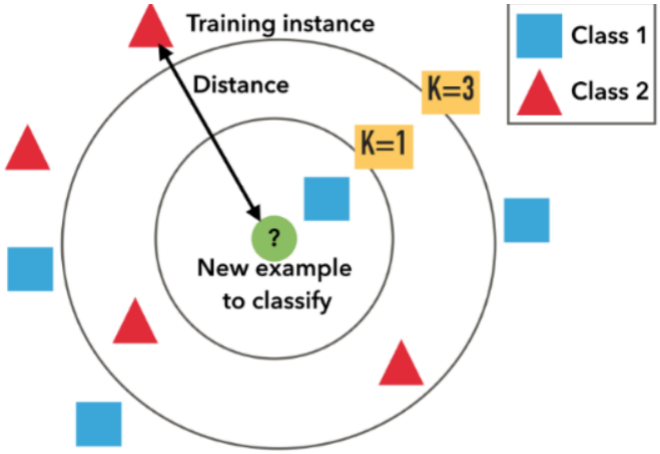

Figure 2.12: k-NN for predictions

In classification, an object is classified by a majority vote of its neighbors, with the object being assigned to the class the most common among its k nearest neighbors. Figure 2.12 illustrates how k-NN is employed to make predictions in classification case. Similarly, in regression case, the output is the average of the values of its k nearest neighbors. Both for classification and regression, a useful technique can be to assign weight to the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones.

2.3.6.4 Bayesian inference method

Bayesian inference is based on Bayes' Theorem given by [75] $p(\theta|\mathbf{D}) = c^{-1}L(\theta;\mathbf{D})p(\theta)$, where θ is the vector of unknown parameters; $\mathbf{D}$ is a set of available data; $L(\theta;\mathbf{D})$ is the likelihood of θ when we consider $\mathbf{D}$, it is calculated by $L(\theta;\mathbf{D}) = p(\mathbf{D}|\theta)$, $p(\theta)$is the prior probability density function (PDF) of θ which can be learned from the historical data or from experts' experience. The product of these two terms determines the shape of the posterior PDF $p(\theta|\mathbf{D})$ which reflects the updated shape of the model after incorporating the information contained in $\mathbf{D}$. The normalizing constant c is given by $c = p(\mathbf{D}) = \int p(\mathbf{D}|\theta)p(\theta)\mathrm{d}\theta$. Here, c is called the evidence. The evidence is used for performing model class comparison and

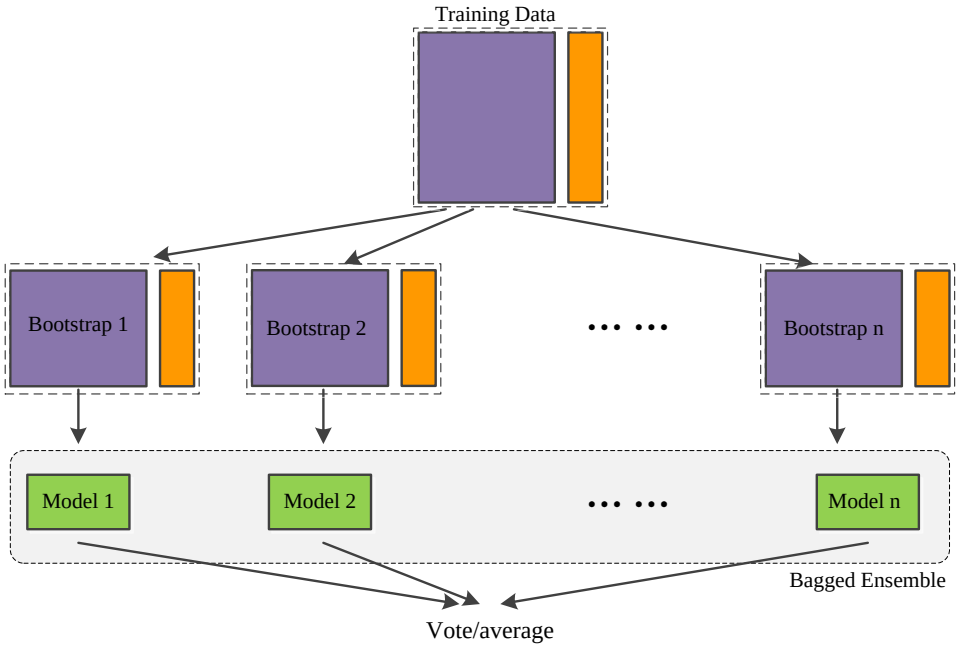

Figure 2.13: Process of bootstrap aggregating

selection. In this way, the uncertainty in selecting the best approximating model, among a set of possibilities, can be treated quantitatively.

2.3.6.5 Tree methods: Random Forest

Random Forest is an ensemble learning method that uses multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone. RF is comprised of many individual trees called classification and regression tree (CART), each of which is induced from a bootstrap sample. Then the CARTs are aggregated to make predictions for a future input. Refer to Figure 2.13. An underlying assumption is that the base learners are independent. As the trees become more correlated (less independent), the model error tends to increase. Randomization helps to reduce the correlation among decision trees so that the model accuracy is improved. Two kinds of randomness exist in tree learning process: Bootstrap sampling and node splitting.

A CART has a binary recursive structure. The first node is called root node. The terminal nodes are called leaves. The connection between two nodes is called a

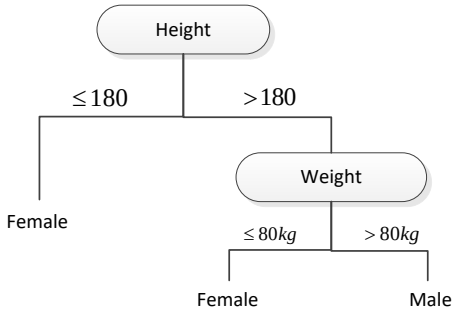

Figure 2.14: Example regression tree

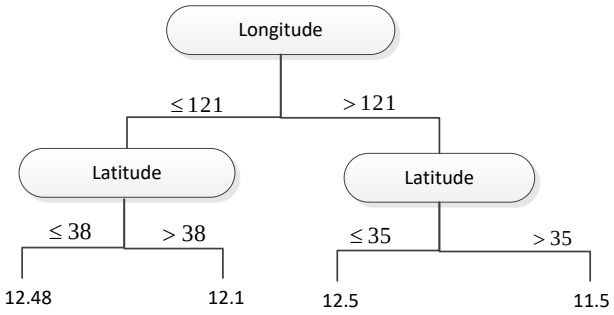

Figure 2.15: Example classification tree

branch. The nodes that have branches are called non-leaf nodes. Each non-leaf node specifies an attribute and each branch emanating from a node specifies the possible values of that attribute. The tree learning process is actually a recursive data partitioning process. The root node corresponds to the whole learning set. When a node is split into two child nodes, the data set is simultaneously divided into two subsets according to the splitting point. In continuous decision case, the CART is called regression tree; in categorical decision case, the CART is called classification tree. See examples in Figure 2.14 and Figure 2.15. Figure 2.14 shows a regression tree induced from a house data. The average house price is determined by the longitude and latitude of the place, for example, the average price is $12.1k\$/m^2$ when the condition $Longitude \leq 121, Latitude > 38$ is satisfied.

Figure 2.15 shows a classification tree induced from a gender data. The gender is judged by the height and weight of the person, for example, the gender is male when

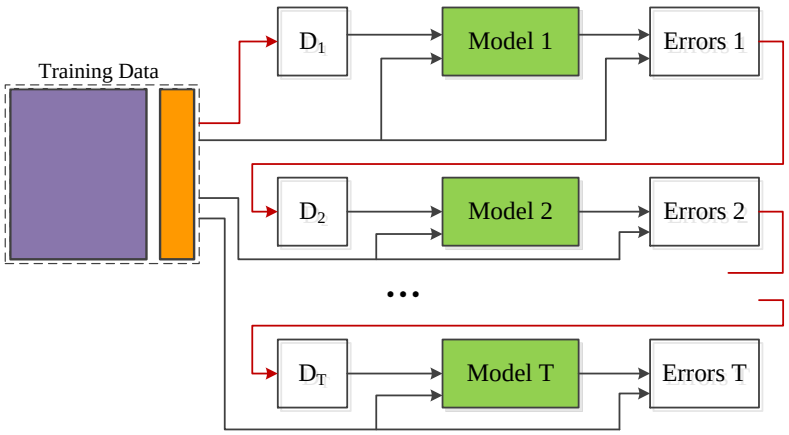

Figure 2.16: Inducing process of AdaBoost

the condition $height > 180, weight > 80$ is satisfied. For a new sample **x'**, we make the prediction by majority voting (classification case) or averaging all predicted values of all trees (regression case). More details will be provided in Chapter 3.

2.3.6.6 Tree methods: Gradient Boosting

Gradient Boosting is a machine learning meta-algorithm. It can be used in conjunction with many other types of learning algorithms to improve their performance. The output of the other learning algorithms ('weak learners') is combined into a weighted sum that represents the final output of the boosted classifier. AdaBoost is adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers. The individual learners can be weak, but as long as the performance of each one is slightly better than random guessing, the final model can be proven to converge to a strong learner. Weak classifiers can be Decision tree/Decision stump, Neural Network, Logistic regression or even SVM. To determine the coefficient for each classifier, we initialize the weight distribution of the samples in the dataset by a uniform distribution, and update this distribution by considering the misclassified samples in each loop. Meanwhile we calculate the coefficient for the current classifier.

Finally, we have a certain number of weak classifiers as well as their weights, then the strong classifier is obtained as the weighted sum of these classifiers,

$$f(x) = \sum_{i=1}^{M} \alpha_t h_t(x) \tag{2.39}$$

where α_t is the coefficient for the 'weak' classifier $h_t(x)$. For GB, we use all training data, instead of bootstrapping, to build each tree.

The first tree is built from an initialized constant model (i.e. a constant value or a one-node tree) [71]. From the second tree, to build each tree, we need to calculate the errors of the former trees (as a weighted sum) on all training points until the stopping criteria (number of trees, error threshold, $\cdots$) are satisfied. See Figure 2.16.

2.3.6.7 REPTree and Random Tree

REPTree (Reduced Error Pruning Tree) is a fast decision tree learner that builds a classification/regression tree by information gain/variance as the splitting criterion, and prunes it using reduced error pruning method [73]. We assume there is a training set and a test set. REPTree generates a single decision tree from the training set. The tree is grown in the standard top-down manner, by finding a split for the current node (if entropy/square error can be reduced by splitting the data further), dividing the data based on the split, recursively generating corresponding successor nodes for each node. Once the tree has been grown, it is pruned using Quinlan's reduced-error pruning method with back-fitting [73], making pruning decisions by measuring classification error/mean squared error on the hold-out set.

Random tree is a method for constructing a tree that performs no pruning [23]. For classification problem, Random Tree grows a tree by selecting splits based on information gain, the attribute with the maximum information gain is chosen to split the current node. For regression problem, it selects the split that minimizes squared error (locally, for the data at the node being split, consider the mean as the predictor for each candidate subset to calculate squared error). Different from REPTree that chooses an attribute among all to split the node, a Random Tree considers a subset

of randomly chosen attributes for splitting at each node. It performs multi-way splits on nominal attributes, with one branch for each value.

Table 2.1: characteristics of different types of single trees (regression case)

Item	CART	RandomTree	REPTree
`Bootstrap`	yes	no	no
`Split type`	binary	binary	binary
`Split principle`	square error	square error	square error
`Pruned`	no	no	yes
`Split attribute`	the best one from a random subset	the best one from a random subset	the best one among all attributes

A comparison of the different types of trees CART, RandomTree and REPTree is shown in Table 2.1. We assume the problem that we are facing is a regression problem.

2.3.6.8 Extra-Trees

Extremely randomized trees, Extra-Trees or ETs, is so named because the ETs have further randomness in node splitting compared with RF. In principle, RF and ETs are similar. Both of them consist of multiple decision trees. In the tree inferring process, both of them use a random subset of the features as the candidate splitting features. The mainly different operations are listed below. (a). the way to prepare the training set for each tree. RF uses a bootstrap replica to train each tree. In contrast, ETs employ the whole training set to train each tree;
(b). the way to find the splitting point. In RF, a decision tree firstly finds the best splitting point for each candidate feature, then chooses the best one; however, in an ET, a random splitting point is chosen for each candidate feature, then the best one

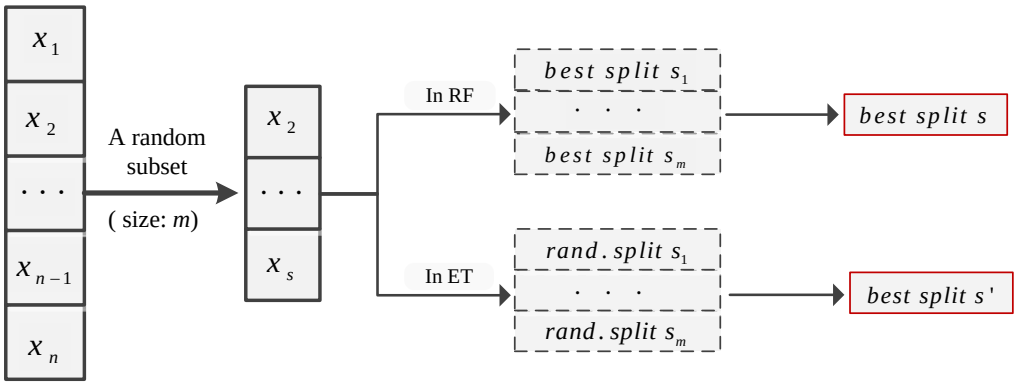

Figure 2.17: Comparison of RF and ETs in node splitting

is chosen. See Figure 2.17.

2.4 Chapter summary

In this chapter, we have presented the fundamental knowledge of reliability. Traditional methods such as FORM, SORM and response surface methods are introduced in detail. Simulation-based methods including MCS and advanced MCS are described. Besides, the statistical machine learning methods are extensively introduced that include SVM, ANN, k-NN and trees-based methods. The tree-based machine learning methods are the research focus in this dissertation, and they are going to be explored and applied in structural reliability analysis in the later chapters.

Chapter 3

Uncertainty quantification of structural systems

Contents

3.1 Introduction

The design and optimization of industrial products (e.g. automobiles, bridges, airplanes$\cdots$) have made the computing tools inevitable. Despite the increasingly accurate models and the fast calculation of computers, the complexity of the real world problems is difficult to handle comprehensively. Generally, the calculations rely upon some simplified assumptions about the system such that the computational models are possible to build and analyzed. Moreover, model responses are induced by feeding the values of the input parameters into the model. However, due to complex environmental factors as well as the design defects, the input values are always difficult determine. In other words, they are characterized with uncertainties. The input uncertainties finally result in the response uncertainties of the system. This part mainly introduces the general framework to deal with the uncertainty propagations within the system.

3.2 General framework

Probabilistic engineering aims at describing the uncertainties existing in the physical systems and studying the influence of these uncertainties on the system response. In this thesis, methods for applying uncertainty analysis in physical systems are taken into account. The main steps for such analysis are summarized in Figure 3.1.

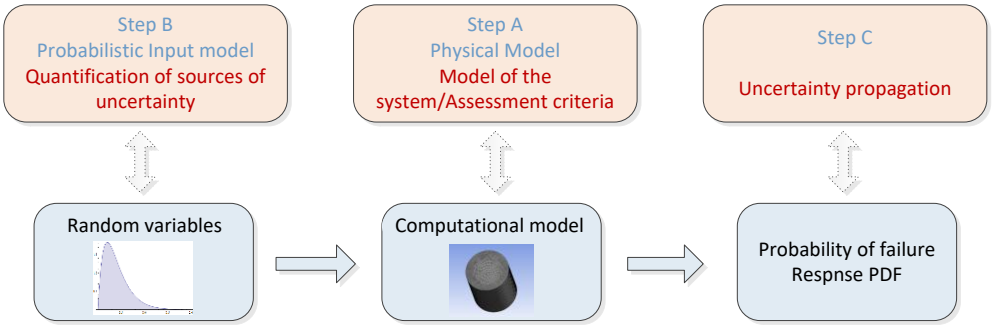

Figure 3.1: Illustration of the general framework for uncertainty quantification [43]

3.2.1 Models of physical system

Step A consists of building the mathematical model from the physical process analysis of the system and setting the assessment criteria (e.g. failure criteria) that will be used to evaluate the physical system of interest. This step gathers all the ingredients used for a classical deterministic analysis of the physical system to be analyzed. Let X denote the vector containing all basic random input variables. When considering models of mechanical systems, these variables usually describe the randomness in the structural, environmental and load parameters. The physical model can be seen as a black-box function, i.e. as an unknown map from the space of input parameters to that of output quantities [74]:

$$Y = F(X) \tag{3.1}$$

where X is a random vector that parametrizes the variability of the input parameters (typically through a joint probability density function, PDF) and Y is the vector of model responses quantities. In a mechanical context, these response quantities can be displacements, stress components, *etal*. The computational model can be deterministic in the sense that evaluating it repeatedly for a given input realization $\mathbf{x}_0$ will always give the same result $y_0 = F(\mathbf{x}_0)$. However, the uncertainty in the input variables causes Y to be a random variable/vector as well. In this case, for each realization of X, the response vector can be determined through pointwise evaluations $y^{(i)} = F(\mathbf{x}^{(i)})$, where the input vector $\mathbf{x}^{(i)}$ is a realization of X. Indeed, one of the main applications of uncertainty quantification is that of propagating the randomness in the input X to the output Y. Commonly used models in advanced application often comprise both random and deterministic parameters. When this is the case, the following notation is introduced to clarify such distinction:

$$Y = F(X, P) \tag{3.2}$$

where P is a set of deterministic parameters that are used to properly configure a model (e.g. configuration options, fixed values for parametric functions, etc.). Analytic functions can be considered as a computational model that is normally

known in their closed-forms. However, the vast majority of physical phenomena cannot be approximated by closed-form equations. In the domain of stochastic dynamics with random vibrations, numerical models, such as finite element schemes, are always adopted to calculate the model response.

3.2.2 Quantification of sources of uncertainty

A designed system generally provides a theoretical framework where the input parameters are selected in order that design criteria related to the purpose of the system are satisfied. In contrast, a practical system is a physical object that is built according to a given design. In fact, a real system cannot fully match the initial design. On one hand, the material properties of the real system may differ slightly from the designed properties; on the other hand, the loadings of the designed system are always idealized such that they can only roughly describe the characteristics of those of the real system.

For all the reasons above, a probabilistic model of the input parameters becomes necessary. The uncertainties that are related to the input parameters may be of different kinds, but they are mainly classified into two classes, i.e. aleatoric uncertainty and epistemic uncertainty. The details of the two classes are introduced in Figure 1.2 of Section 1.2.2. In many cases, it is difficult to distinguish clearly between aleatoric and epistemic uncertainty, since both types may be present in the same system. In this thesis, no distinction is made between the two kinds of uncertainties in their modelling, and the probabilistic model of the latter will be built to characterize uncertainties in the input parameters. It is direct to estimate a probabilistic model that represents the scattering of the available data since it is merely a problem of statistics. Note that in some cases (e.g. in the design stage of the object system) where no data is available, a prior probabilistic model can be pre-assumed by expert experience. Then Bayesian statistics can be applied to combine the priori model and the scarce data. In general, step B results in a specification of a random vector of input parameters X in terms of its joint PDF. When spatial variability is modelled by random fields, the discretization of the random fields will result in some basic random variables as an addition to the input random vector.

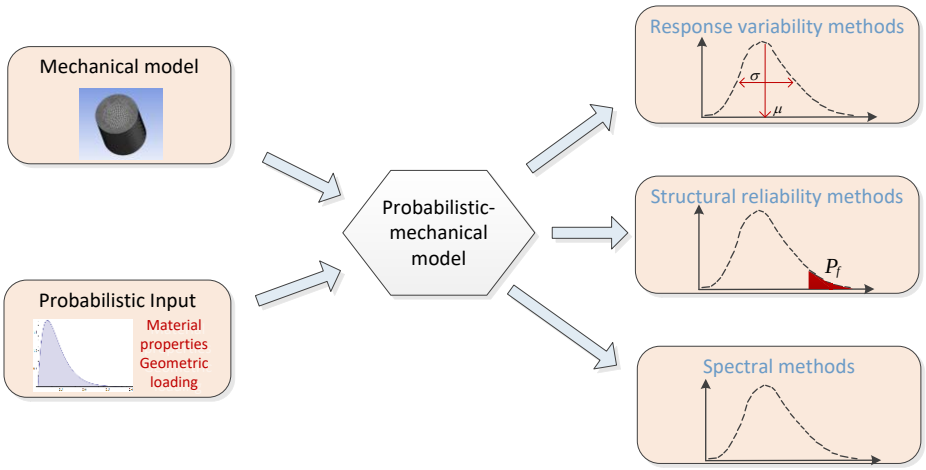

Figure 3.2: Classification of uncertainty propagation methods [43]

3.2.3 Uncertainty propagation

Let us consider a random response $Y = F(X)$, where the input X is random vector. The most important content in the field of probabilistic engineering mechanics is to study the probabilistic content of Y, namely its joint PDF $f_Y(y)$. Practically, this function is not directly computable except in special cases. Therefore, techniques for uncertainty propagation have to be developed. Based on the specific information on the random response that is studied, these techniques can be classified into three categories, see Figure 3.2.

Response variability methods are mainly about computing the mean and variance values of the model response. These indices only give some information on the central part of the response PDF, so higher order moment methods can be investigated. Structural reliability methods [24] essentially investigate the tails of the response PDF and compute the probability of exceeding a prescribed threshold (i.e. probability of failure). Spectral methods characterize the complete randomness of the response in an intrinsic way by using suitable tools of functional analysis. Also known as stochastic finite element methods in the context of computational mechanics [60], spectral methods have been a popular research subject during recent years. These methods aim at constructing an expansion from a suitable basis of functions

so that researchers are allowed to solve problems of second moment or structural reliability by a straightforward post-processing of the expansion coefficients. In this thesis, structural reliability methods are the main concern.

3.3 Conditional failure probability

3.3.1 Basic concepts and definition

The object structure studied in this research is a linear dynamic system with uncertainties existing in the structural properties, such as mass, damping and stiffness. Besides, the structure is subject to stochastic loading modelled as a Gaussian process. Hence, the assessment of failure probability involves the uncertainties in both structural parameters and excitations. As introduced in section 1.2.3, for the first excursion probability problem, the performance function is equal to or smaller than zero whenever the response exceeds the prescribed threshold. Taking into account the definition of performance function in Section 2.2.1, the first excursion probability is defined by means of a multi-dimensional integral:

$$P_f = \int_{g(\mathbf{x},\mathbf{z})\leq 0} p_X(x)p_Z(z)\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{z} = \int_{\mathbf{x}\in\Omega_X,\mathbf{z}\in\Omega_Z} I_f(\mathbf{x},\mathbf{z})p_X(x)p_Z(z)\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{z} \tag{3.3}$$

where $I_f(\mathbf{x},\mathbf{z})$ is the indicator function which equals to 1 when the performance function $g(\mathbf{x},\mathbf{z}) \leq 0$ and 0 otherwise. It is noted that the calculation in Eq. (3.3) considers uncertainty in both structural parameters and excitations. In fact, it is possible to define the first excursion probability conditioned on a particular realization of the uncertain structural parameters. The latter probability is denoted as $P_f^C(\mathbf{x}^{(i)})$ and is defined as

$$P_f^C(\mathbf{x}^{(i)}) = \int_{g(\mathbf{x},\mathbf{z})\leq 0} p_Z(z)\mathrm{d}\mathbf{z} = \int_{\mathbf{z}\in\Omega_Z} I_f(\mathbf{x}^{(i)},\mathbf{z})p_Z(\mathbf{z})\mathrm{d}\mathbf{z} \tag{3.4}$$

where the superscript C in $P_f^C(\mathbf{x}^{(i)})$ denotes a conditional failure probability (CFP) with respect to a realization of the uncertain structural parameters. It is noted that the characterization of the integrals in Eqs. (3.3) and (3.4) may involve hundreds

of random variables in the context of dynamical systems under stochastic loading. Therefore, the current reliability problem forms actually a high dimensional problem. Considering the relative independence between the uncertainties of the structural properties and those of the excitations, the integral in Eq.(3.3) can be reformed as follows,

$$P_f = \int_{\mathbf{x}\in\Omega_X} [\int_{\mathbf{z}\in\Omega_Z} I_f(\mathbf{x},\mathbf{z}) p_Z(\mathbf{z}) \mathrm{d}\mathbf{z}] p_X(\mathbf{x}) \mathrm{d}\mathbf{x} = \int_{\mathbf{x}\in\Omega_X} P_f^C(\mathbf{x}) p_X(\mathbf{x}) \mathrm{d}\mathbf{x} = E[P_f^C(\mathbf{x})] \tag{3.5}$$

where $E[\cdot]$ denotes the mathematical expectation.

3.3.2 Estimation of Conditional failure probability

Practically, Eq.(3.5) can be approximated by analyzing a certain number of realizations of the random vector X, say $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(N_x)}$. For each realization, a conditional failure probability is estimated. The Monte-Carlo estimator of P_f is given by the empirical mean,

$$\hat{P}_{f,MCS} = \frac{1}{N_x} \sum_{i=1}^{N_x} P_f^C(\mathbf{x}^{(i)}) \tag{3.6}$$

Besides, the conditional failure probability in Eq.(3.4) is denoted as

$$P_f^C(\mathbf{x}^{(i)}) = E[I_f(\mathbf{x}^{(i)}, \mathbf{z})] \tag{3.7}$$

Eq.(3.7) can be approximated by analyzing a certain number of realizations of the random vector Z, say $\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, ..., \mathbf{z}^{(N_z)}$. The Monte-Carlo estimator of $P_f^C(\mathbf{x}^{(i)})$ is given by

$$P_{f,MCS}^C(\mathbf{x}^{(i)}) = \frac{1}{N_z} \sum_{j=1}^{N_z} I_f(\mathbf{x}^{(i)}, \mathbf{z}^{(j)}) \tag{3.8}$$

While in principle both approximations of probabilities P_f and $P_f^C(\mathbf{x})$ can be carried out through MCS, the demanded computational burden would become quite heavy for the practical engineering problems. This is because the aforementioned probabilities are generally quite small and to ensure the accuracy, a large number of samples

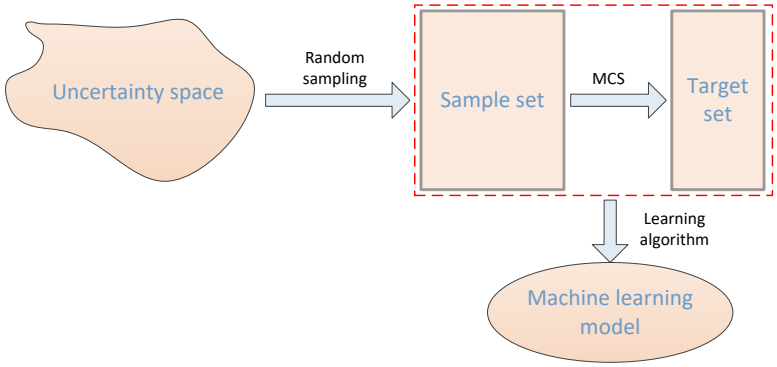

Figure 3.3: Theoretical framework to approximate $P_f^C(X)$ by statistical learning model

are needed (refer to Section 2.3.3.1). It is noted from Eq.(3.8) that each realization of the random vector X results in a specific conditional failure probability.

3.3.3 Machine learning methods as surrogates

From Section 2.3.2.1, it is clear that there is a mapping from the space of random vector X to the space of conditional failure probability $P_f^C(X)$. If there is a model that explicitly provide us this mapping relationship, most of the MCS procedures can be avoided so that the efficiency of the estimation could be largely improved. Based on this idea, surrogate models have been explored this thesis, and the statistical learning models have been extensively studied and employed to accomplish such a task. Figure 3.3 illustrates the theoretical framework to approximate the conditional failure probability $P_f^C(X)$ by the statistical learning models.

In Figure 3.3, a sample set consists of a certain number of cases each of which is a realization of the random vector X. Then by MCS a conditional failure probability $P_f^C(\mathbf{x}^{(i)})$ is estimated. From Eq.(3.8) it is known that in each loop of the MCS, a random realization of the stochastic excitation process is obtained as the input of the structural system and the response is then analyzed, see Figure 3.4 for the details. As is shown in Eq.(3.8), in practical simulations, the excitations are denoted by a discretized form, i.e. a random vector Z. Thus, the structural responses are obtained also in a discretized form, refer to Figure 3.5. $\mathbf{e}_i$ and $\mathbf{y}_i$ is the ith realization of the

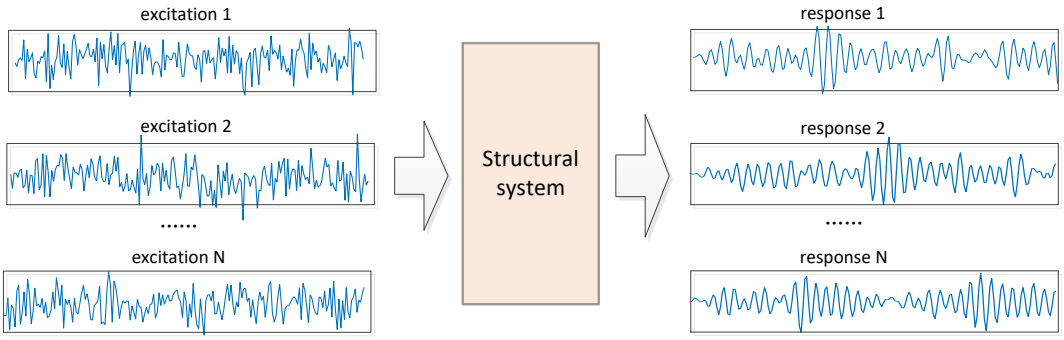

Figure 3.4: Structural responses induced by different random excitations

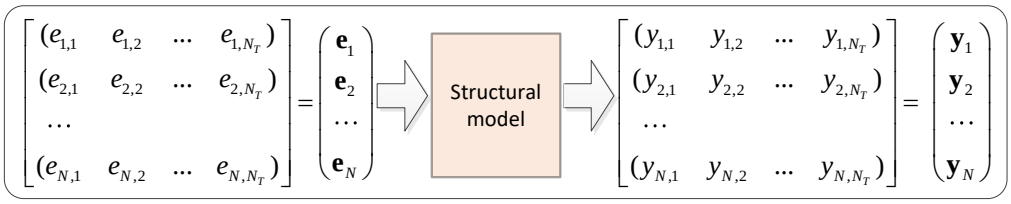

Figure 3.5: Numerical representations of the process in Figure 3.4

random vector Z and its corresponding response; N_T is the dimension of discretized excitation process.

From the framework illustrated in Figure 3.3, just a limited number of samples are needed from the uncertainty space so that a statistical learning model is built. With the induced model the conditional failure probabilities for other samples will be directly estimated. Therefore, a large amount of computation time can be saved. Chapter 4 and Chapter 5 are devoted to building statistical learning models to approximate the conditional failure probabilities. The performance of these models are also analyzed in various aspects.

3.4 Evaluation of very small conditional failure probabilities

As is mentioned in chapter 2, the general MCS can be treated as a standard reference. However, it is not computationally efficient for estimating very small failure probabilities (for example, $< 1\%$) since the number of samples required to achieve a given accuracy is inversely proportional to the scale of Pf. In other words, estimating small probabilities requires information from rare samples that induce system failures. On average it requires many samples before one such failure sample occurs. In view of this, the importance sampling method should be implemented to compute P_f. In principle, an IS method tries to adjust the sampling density so that more samples from the failure region F can be obtained. The efficiency of the method relies on the construction of the importance sampling density (ISD), for which the knowledge about the failure region is inevitably required.

In the framework of MCS, each sample taken as input of the system is analyzed and the system response is obtained. The classic finite element codes is experimentally proved low efficient. Besides, the input excitations are generally discretized according to the time sampling step. Therefore, the response process can be calculated by a convolution integral based on the structural impulse response and the excitations. In this article, we assume the stochastic excitation process for the structure is modelled as a Gaussian process. In this aspect, some representation methods in the random field domain can be applied to describe discretized Gaussian process. This subsection is devoted to the theories about discrete representation of random processes, the convolution integral based structural response calculation and an IS method to improve the estimation efficiency of conditional failure probabilities.

3.4.1 K-L decomposition of Gaussian excitation process

Let $f(t)$ be a scalar function that represents the excitations acting over the object structure during a time period $t \in [0, T]$. It is assumed this excitation process can be approximated by a Gaussian process, *i.e.* for each finite set of time realizations $t_1, t_2, ..., t_{n_T} \in [0, T]$, the n_T random variables $f(t_1)$, $f(t_2)$, ..., $f(t_{n_T})$ follow a

joint Gaussian distribution. Several discrete representation methods are available to describe such randomness in terms of a finite set of standard normal random variables. For a second-order Gaussian process $f(t)$ with mean function $\mu(t)$, the existing discrete representation methods all result in the following form,

$$f(t) = \mu(t) + \sum_{i=1}^{n} u_i s_i(t) = \mu(t) + s(t)^T u \tag{3.9}$$

where $u = [u_1, u_2, ..., u_n]^T$ is a vector of standard normal variables, $s(t) = [s_1(t), ..., s_n(t)]^T$ is a vector of deterministic basis functions dependent on the covariance structure of the excitation process, and n is a measure of resolution of the representation. To carry out this task, several discretization methods have been developed since the $1980s$, a comprehensive review and comparison of these methods is presented in [75]. Series expansion methods are mostly considered. In this article, K-L expansion is employed for Gaussian process representation.

3.4.1.1 K-L expansion method

In the K-L expansion method, the random process can be decomposed as,

$$f(x) = \mu(x) + \sum_{i=1}^{\infty} \sqrt{\lambda_i} u_i \phi_i(x) \tag{3.10}$$

where $\mu(x)$ is the mean function and $u_i, i \in N^+$, are zero-mean orthogonal variables. The eigen-pair $\{\lambda_i, \phi_i\}$ are the solutions of the eigenvalue problem:

$$\int_B C_{HH}(x, x')\phi_i(x')dx' = \lambda_i \phi_i \tag{3.11}$$

which is a Fredholm integral equation of second kind. The kernel $C_{HH}(x, x')$ being an autocovariance function, is symmetric and positive definite. The set of eigenvalues is moreover real, positive and numerable. It is possible to order the eigenvalues λ_i in a descending series converging to zero. Truncating the ordered series in eq.(3.10) after the M-th term gives the K-L approximation:

$$f(x) \approx \mu(x) + \sum_{i=1}^{M} \sqrt{\lambda_i} u_i \phi_i(x) \tag{3.12}$$

when the random field under consideration is Gaussian, the set of u_i are independent standard normal variables. Eq.(3.11) can be solved analytically only for few auto-covariance functions and geometries of B. Detailed closed form solutions for triangular and exponential covariance functions for one-dimensional homogeneous fields can be found in [60]. Consider a discrete representation of time t such that $\Delta t = T/(n_T - 1)$ where n_T is the number of time points such that the time instants of analysis are $T_k = (k-1)\Delta t, k = 1, ..., n_T$. Thus, the discrete representation of $f(t)$ at time t_k is:

$$f(t_k) \approx f_0(t_k) + \sum_{i=1}^{n_{KL}} \sqrt{\lambda_i} u_i \phi_i(t_k) \quad = f_0(t_k) + \sum_{i=1}^{n_{KL}} f_i^{KL}(t_k) u_i \tag{3.13}$$

where u_i, $i = 1, ..., n_{KL}$ are independent, identically distributed standard Gaussian random variables with joint PDF $p_u(u)$, $f_0(t_k)$ and $f_i^{KL}(t_k)$ denote the mean function and the sth component of the K-L vector at time t, and n_{KL}, is the order of truncation of the series expansion. Note that according to Eq.(3.13) the force at the kth time instant is actually a function of the vector of uncertain variables $\mathbf{u} = (u_1, u_2, ..., u_{n_{KL}})^T$. The K-L vector $f_1^{KL}(t_k), ..., f_{n_{KL}}^{KL}(t_k)$ is determined by solving an eigen-problem of the corresponding covariance matrix of the discrete stochastic process. In this dissertation, it is assume, without loss of generality, that the excitation is a zero-mean Gaussian process, *i.e.* $f_0(t_k) = 0$, $k = 1, ..., n_T$.

3.4.1.2 An exemplary implementation of K-L expansion

This part introduces a simple implementation of K-L expansion. Assume that the output of a system, according to time t, is denoted as

$$f(t) = 1.4118 + [b * c/(\sqrt{\pi}(t-s)] * \exp[-c^2 * (log|t-s| - d)^2] \tag{3.14}$$

where the coefficients s, b, c and d are characterized with uncertainties below,

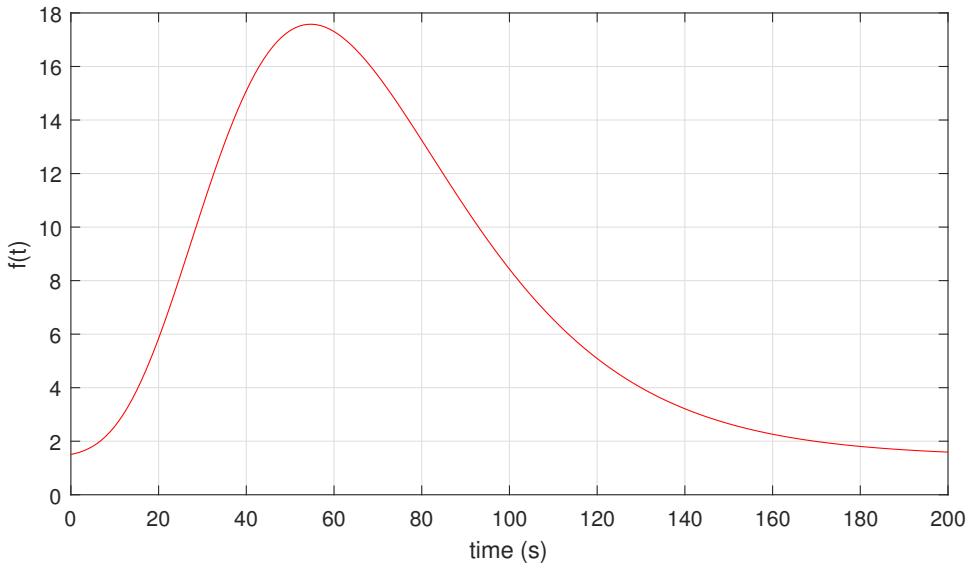

Figure 3.6: The nominal output

$$
\begin{aligned}
&s \sim unif(-30.53, -24.98), E(s) = -27.76;\\
&b \sim unif(1193.22, 1207.64), E(b) = 1200;\\
&c \sim unif(2.02, 2.15), E(c) = 2.09;\\
&d \sim unif(4.50, 4.56), E(d) = 4.53.
\end{aligned}
\tag{3.15}
$$

According to the uncertainties of the coefficients, ten samples of the output process $f(t)$ are generated, see Figure 3.7. To make comparisons, Figure 3.6 shows the nominal output when the coefficients take their nominal values. The ten sampled process in are then used to calculate the eigenvalues and eigenfunctions of the Fredholm integral equation of the second kind.

In this simulation, the five largest eigenvalues and their corresponding eigenvectors are calculated. In Figure 3.8, it is found that the eigenvalues are largely different, the largest is about 35.0 while the smallest one is close to 0. Therefore, the choice of $n_{KL} = 5$ ensures the accuracy of K-L expansion. Figure 3.9 displays the five eigenfunctions related to the five largest eigenvalues. Figure 3.10 shows the five scaled eigenfunctions, refer to eq.(3.13). Notice that the value of the ith scaled eigenfunction at time t_k is calculated as $\sqrt{\lambda_i}\phi_i(t_k) = f_i^{KL}(t_k)$. Figure 3.10 also

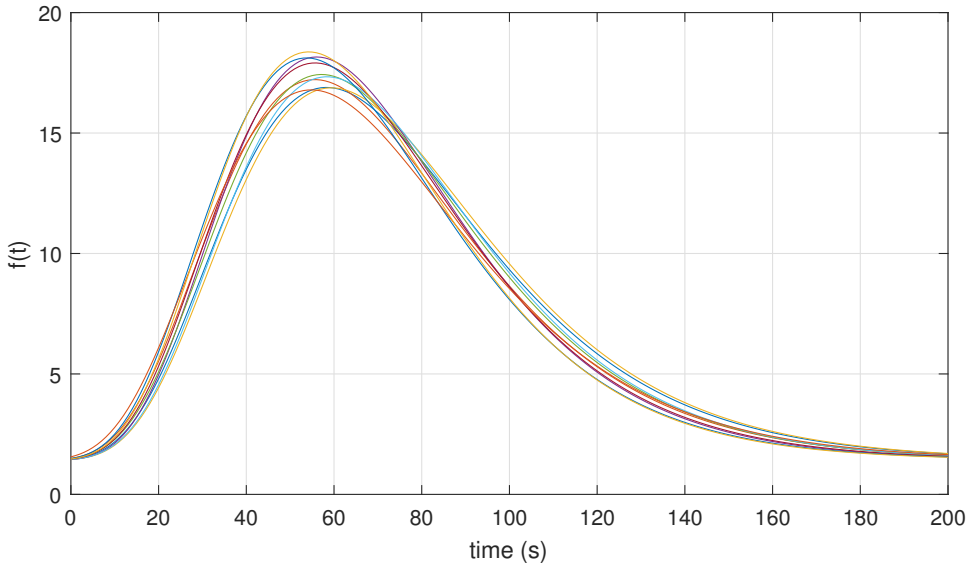

Figure 3.7: Ten sampled outputs according to the uncertainties of the coefficients

shows the mean function, *i.e.* $f_0(t_k)$ in eq.(3.13), that is calculated by averaging the ten sampled functions shown in Figure 3.7.

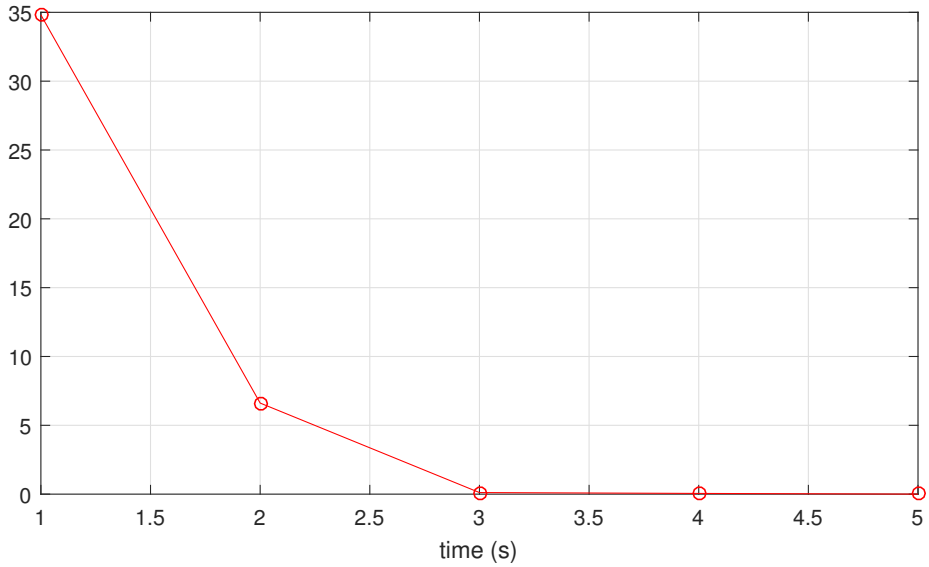

Figure 3.8: The first five eigenvalues of the Fredholm integral equation of the second kind

In Figure 3.11, 20 sampled functions are generated according to the already calculated five scaled eigenfunctions and the mean function, refer to eq.(3.13). Notice that at each sampled time point, $n_{KL} = 5$ standard normal variables (*i.e.* u_i in

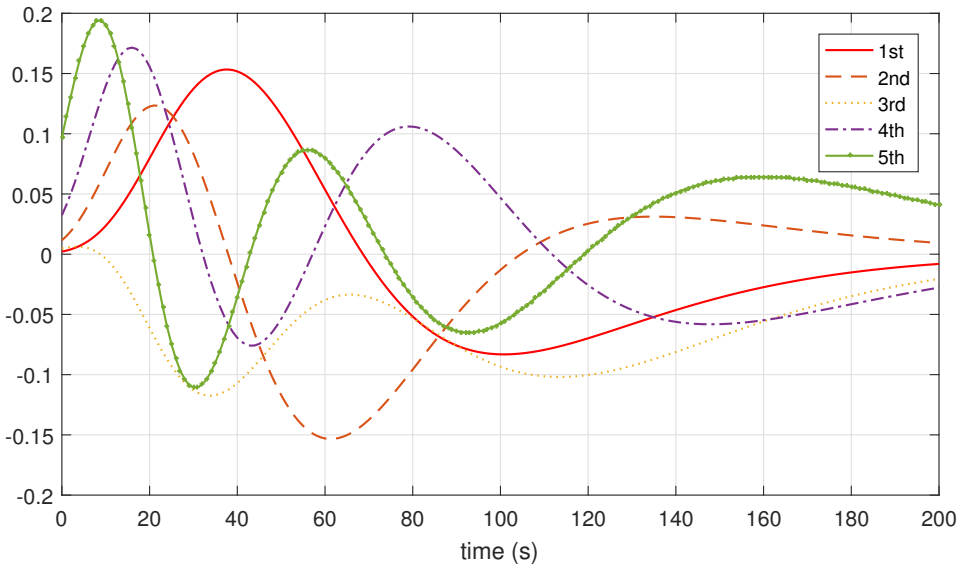

Figure 3.9: The first five eigenfunctions of the Fredholm integral equation of the 2nd kind

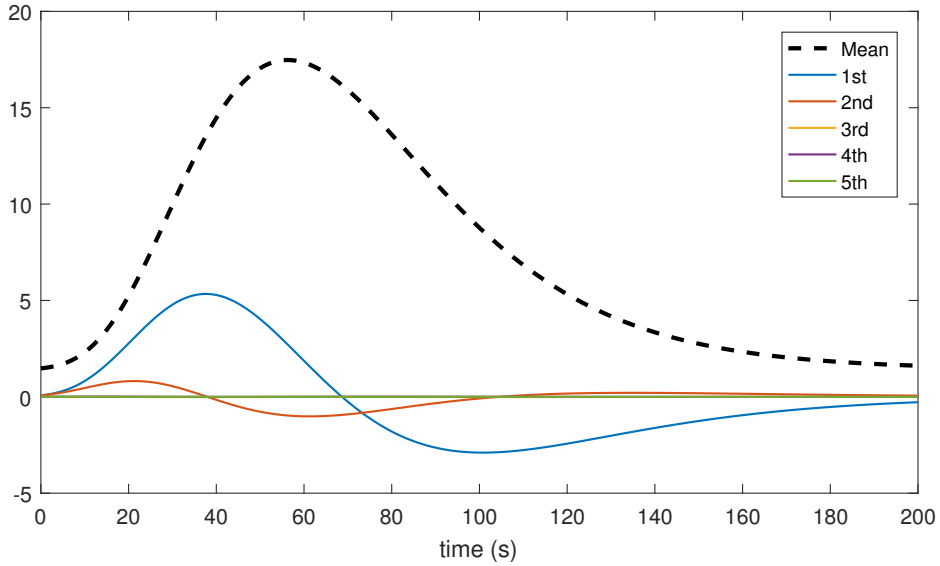

Figure 3.10: The five scaled functions corresponding to Figure 3.23

eq.(3.13)) are generated as the 'weights' of the five scaled eigenfunctions. It is found that the 20 sampled functions in figure 3.11 vary around the mean function and they are very similar to the ten functions in Figure 3.7. These results tell that the

K-L expansion works well in representing the random process.

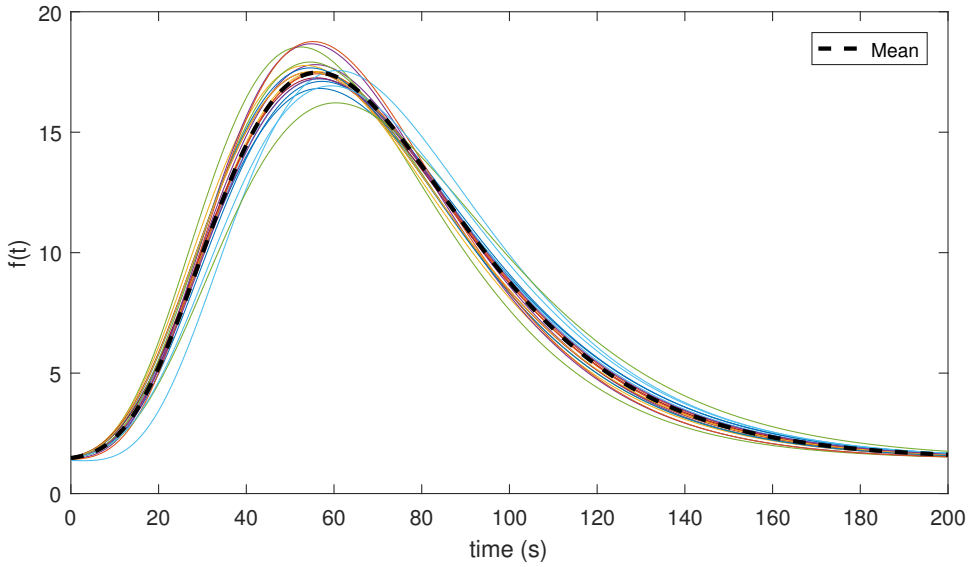

Figure 3.11: Twenty sampled realizations with $n_{KL} = 5$

3.4.2 Convolution integral to calculate structural responses

Consider a classically-damped linear elastic structural system represented by an appropriate model (e.g. a finite element model) comprising a total of n DOFs. In addition, consider a vector of random variables of dimension grouping the uncertain structural parameters. The system is subjected to a Gaussian process excitation with zero mean, $f(t)$, which is represented in the discrete form in Eq. (3.13). Then the response of the system can be evaluated by the convolution integral below [76],

$$y_i(t, \mathbf{x}) = \int_0^t h_i(t - \tau, \mathbf{x}) f(\tau) d\tau \tag{3.16}$$

where $h_i(t - \tau, \mathbf{x})$ is the unit impulse response function for the ith DOF of the structure at time t due to a unit impulse applied at time τ. $\mathbf{x}$ is a vector that consists of the structural properties. Without loss of generality, the zero initial conditions at $t = 0$ is assumed. As a result of linearity, the response of interest is actually a linear combination of the contributions from each input $f(\tau)$. Substituting for $f(\tau)$

in eq.(3.13) and then changing the orders of summation and integration, we obtain

$$\begin{aligned} y_i(t,\mathbf{x}) &= \int_0^t h_i(t-\tau,\mathbf{x}) \sum_{l=1}^{n_{KL}} u_l f_l^{KL}(\tau) d\tau \\ &= \sum_{l=1}^{n_{KL}} u_l a_l(t,\mathbf{x}) \\ &= \mathbf{a}(t,\mathbf{x})^T \mathbf{u} \end{aligned} \tag{3.17}$$

where $\mathbf{a}(t,\mathbf{x}) = [a_1(t,\mathbf{x}), ..., a_{n_{KL}}(t,\mathbf{x})]^T$ and

$$a_l(t,\mathbf{x}) = \int_0^t h_i(t-\tau,\mathbf{x}) f_l^{KL}(\tau) d\tau, l = 1, ..., n_{KL} \tag{3.18}$$

The response process is found to be the scalar product of two vectors: the random vector $\mathbf{u}$ and the deterministic basis function vector $\mathbf{a}(t,\mathbf{x})$, whose elements are convolutions of the basis functions $f_l^{KL}(t)$ of the excitation and the unit impulse response function $h_i(t-\tau,\theta)$ of the system. The geometric interpretations made earlier for the Gaussian process $f(t)$ apply to the response process $y_i(t,\theta)$ as well. For the simulation purpose, the convolution integral in eq.(3.18) is re-written in its discrete form, *i.e.*

$$a_l(t,\mathbf{x}) = \Delta t \sum_{j=1}^{k} h_i(t_k - t_j, \mathbf{x}) f_l^{KL}(t_j) \tag{3.19}$$

3.4.3 Failure probability at any time t

Consider the set of realizations of $f(t)$ that lead to the event $y(t_0) \geq y_0$ at time $t = t_0$, where y_0 is a selected threshold. These correspond to realizations of u that satisfy the condition,

$$y_0 - \mathbf{a}(t_0)^T \mathbf{u} \leq 0 \tag{3.20}$$

In the space of $\mathbf{u}$, these lie in a half space bounded by the hyper-plane $y_0 - \mathbf{a}(t_0)^T \mathbf{u} = 0$ having the unit normal $\hat{\alpha}(t_0) = \mathbf{a}(t_0)/||\mathbf{a}(t_0)||$ and distance $\beta(y_0, t_0) = y_0/||\mathbf{a}(t_0)||$ from the origin. This is illustrated in Figure 3.12 in the plane formed by the co-ordinate u_1 and the vector $\hat{\alpha}(t_0)$. According to the theory of structural reliability

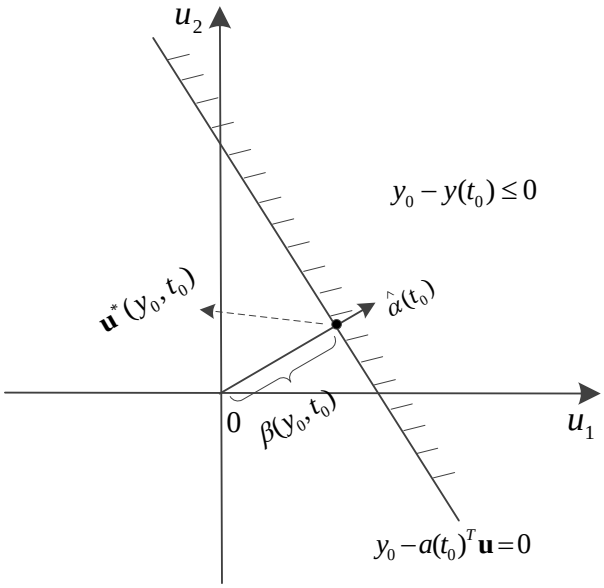

Figure 3.12: Illustration of response at time t_0 in a geometric way [76]

[44], $g_0(u) = y_0 - \mathbf{a}(t_0)^T\mathbf{u}$ is the limit-state function, $g_0(u) = y_0 - \mathbf{a}(t_0)^T\mathbf{u} = 0$ is the limit-state surface, $\hat{\alpha}(t_0)$ is the unit outbound normal vector which towards the failure region, and $\beta(y_0, t_0)$ is the reliability index for the event $g_0(u) \le 0$. From a well known result for the probability content of a half space in the standard normal space, the probability of failure is denoted as

$$P(g_0(u) \le 0) = \Phi(-\beta(y_0, t_0)), \tag{3.21}$$

where $\Phi(.)$ denotes the standard normal cumulative probability function.

Among all realizations of u that give rise to the event $y(t_0) \ge y_0$, the one that has the highest likelihood is the one nearest to the origin. This point, known as the 'design point' in the theory of structural reliability, is given by

$$\mathbf{u}^*(y_0, t_0) = \beta(y_0, t_0)\alpha(t_0) = y_0 \frac{\mathbf{a}(t_0)}{||\mathbf{a}(t_0)||} \tag{3.22}$$

The corresponding 'design point' excitation and response, according to eq.(3.9), eq.(3.17) and eq.(3.22), are

$$f^*(t) = \mathbf{s}(t)^T \mathbf{u}^*(y_0, t_0) = y_0 \frac{\mathbf{s}(t)^T \mathbf{a}(t_0)}{||\mathbf{a}(t_0)||^2} = y_0 \frac{f^{KL}(t)\mathbf{a}(t_0)}{||\mathbf{a}(t_0)||^2} \tag{3.23}$$

$$y^*(t) = \mathbf{a}(t)^T \mathbf{u}^*(y_0, t_0) = y_0 \frac{\mathbf{a}(t)^T \mathbf{a}(t_0)}{||\mathbf{a}(t_0)||^2} \tag{3.24}$$

It is noticed that $y^*(t_0) = y_0$ corresponds to the critical state at time t_0. For a linear system subject to Gaussian excitations, the design point realization is proportional to the threshold y_0. That means, if the threshold y_0 changes, then the design point realization will be changed with the same proportion. Of most interest is the design point excitation $f^*(t)$, which is the specific realization of $f(t)$ that has the highest likelihood to give rise to the event $y(t_0) \leq y_0$. This realization is of particular interest from the viewpoint of design, as one can assure safety by providing adequate capacity against this particular deterministic excitation [76].

3.4.4 Formulation of conditional failure probability

In the perspective of first-passage probability, it is general to compare the responses of interest $y_i(t), i = 1, ..., n_r$ against acceptable threshold levels y_i^* within the time duration T of the stochastic excitation. A failure takes place whenever the response y_i exceeds its corresponding threshold. From this point of view, failure implies not meeting the predefined conditions (this does not necessarily imply collapse). According to subsection 3.3, the failure event F can be defined in terms of $\mathbf{x}$ and $\mathbf{u}$:

$$F = \{\mathbf{x} \in R^{n_\mathbf{x}}, \mathbf{u} \in R^{n_{KL}} : g(\mathbf{x}, \mathbf{u}) \leq 0\} \tag{3.25}$$

where $g(\mathbf{x}, \mathbf{u})$ is the so-called performance function. The value of $g(.)$ is equal or smaller than zero whenever a response exceeds its prescribed threshold. Considering the different DOFs of the structure as well as the sampled time points, the performance function $g(\mathbf{x}, \mathbf{u})$ is formulated as

$$g(\mathbf{x}, \mathbf{u}) = 1 - \max_{i=1,...,n_r} (\max_{k=1,...,n_T} (\frac{|y_i(t_k, \mathbf{x}, \mathbf{u})|}{y_i^*})) \tag{3.26}$$

where $|\cdot|$ denotes absolute value. Taking into account the definition of performance function introduced in Eq.(3.26), the first excursion probability is defined by means of a multidimensional integral:

$$P_f = \int_{g(\mathbf{x},\mathbf{u})\leq 0} p_{\mathbf{x}}(\mathbf{x}) p_u(u) d\mathbf{x} du = \int_{\mathbf{x}\in\Omega_\theta, u\in\Omega_u)\leq 0} I_F(\mathbf{x}, u) p_\theta(\theta) p_u(u) d\mathbf{x} du \quad (3.27)$$

where $I_F(\cdot)$ denotes the indicator function which is equal to 1 in case the performance function $g(\cdot) \leq 0$ and 0 otherwise. Note the definition in eq.(3.27) allows calculating the failure probability considering uncertainty in both structural parameters and excitation. Actually, for a particular realization of the uncertain structural parameters $\mathbf{x}$, the conditional failure probability is denoted as

$$P_f^C(\mathbf{x}) = \int_{g(\mathbf{x},\mathbf{u})\leq 0} p_u(u) d\mathbf{x} du = \int_{u\in\Omega_u} I_F(u) p_u(u) du \quad (3.28)$$

It is noticed that the integrals in eq.(3.28) may involve hundreds or thousands random variables related to the stochastic loading. Therefore, the corresponding reliability problem constitute a high dimensional problem. While in principle P_f^C could be estimated using MCS, the associated computational burden would be quite considerable for problems of engineering involving very small failure probabilities. This is because, for small failure probabilities, a very large number of samples are required to ensure a reliable estimation result. In order to improve the computational efficiency, a very efficient importance sampling strategy [77] for estimating the conditional failure probability is introduced in the next subsection.

Elementary failure events

According to the definition of first excursion problem, failure occurs whenever any of the responses of interest y_i exceeds its designed threshold y_i^* within the time duration of the excitation process. The occurrence of failure at the kth time step due to the ith system response is termed *elementary failure event* [77] and is denoted as $F_{i,k}(\mathbf{x})$, $i = 1, ..., n_r, k = 1, ..., n_T$. Therefore, the failure event $F(\theta)$ is the union of these elementary failure event, *i.e.*

$$F(\mathbf{x}) = \bigcup_{i=1}^{n_r} \bigcup_{k=1}^{n_T} F_{i,k}(\mathbf{x}) \quad (3.29)$$

where

$$F_{i,k}(\mathbf{x}) = \{u \in R^{n_{KL}} : |y_i(t_k, \mathbf{x}, u)| \geq y_i^*\} \tag{3.30}$$

The elementary failure event $F_{i,k}(\mathbf{x})$ can be decomposed as the union of two disjoint events [77]: up-crossing event where the response up-crosses its threshold , *i.e.*

$$F_{i,k}^{+}(\mathbf{x}) = \{u \in R^{n_{KL}} : a_{i,k}(\mathbf{x})^T u \geq y_i^*\} \tag{3.31}$$

and down-crossing event where the response down-crosses its threshold , *i.e.*

$$F_{i,k}^{-}(\mathbf{x}) = \{u \in R^{n_{KL}} : -a_{i,k}(\mathbf{x})^T u \geq y_i^*\} \tag{3.32}$$

For the out-crossing event $F_{i,k}(\mathbf{x}) = F_{i,k}^{+}(\mathbf{x}) \cup F_{i,k}^{-}(\mathbf{x})$, since $F_{i,k}^{+}(\mathbf{x})$ and $F_{i,k}^{-}(\mathbf{x})$ are disjoint, $F_{i,k}(\mathbf{x})$ has two design points corresponding to $F_{i,k}^{+}(\mathbf{x})$ and $F_{i,k}^{-}(\mathbf{x})$. The failure probability for $F_{i,k}(\mathbf{x})$ is the sum of those of the two out-crossing cases. According to eq.(3.18), their hyper-planes have the unit normals with inverse directions. Moreover, they have the same distance from the origin, *i.e.*, $\beta_{i,k}^{+}(\mathbf{x}) = \beta_{i,k}^{-}(\mathbf{x})$ Therefore, the two regions are symmetric, the probability related to $F_{i,k}(\mathbf{x})$ is simply $2\Phi^{-1}(-\beta_{i,k}(\mathbf{x}))$. Figure 3.13 provides an illustration of an elementary failure region $F_{i,k}(\mathbf{x})$ with $n_r = 1$, $n_T = 1$, $n_{KL} = 2$.

3.4.5 IS technique to evaluate P_f^C

To apply IS technique to evaluate the P_f^C, the integral in eq.(3.28) is re-written as

$$\begin{aligned} \hat{P}_f^C(\mathbf{x}) &= \int_{u \in \Omega_u} I_F(u) \frac{p_u(u)}{p_{IS,u}(u)} p_{IS,u}(u) du \\ &= \frac{1}{N} \sum_{v=1}^{N} I_f(\mathbf{x}, z^v) \frac{p_u(u^{(v)})}{p_{IS,u}(u^{(v)})} \end{aligned} \tag{3.33}$$

where $p_{IS,u}(u)$ is the Importance Sampling density (ISD) function and $u^{(v)}, v = 1, ..., N$ are samples of uncertain vector U obtained via the ISD $p_{IS,u}(u)$. The most important part for implementing the IS procedures described in eq.(3.33) is the design of the ISD function that is able to obtain more samples in the failure region, meanwhile ensuring a low variability of the estimated probability, i.e. failure samples are drawn frequently while the variability of the ratio involving $p_u(u)$ and $p_{IS,u}(u)$

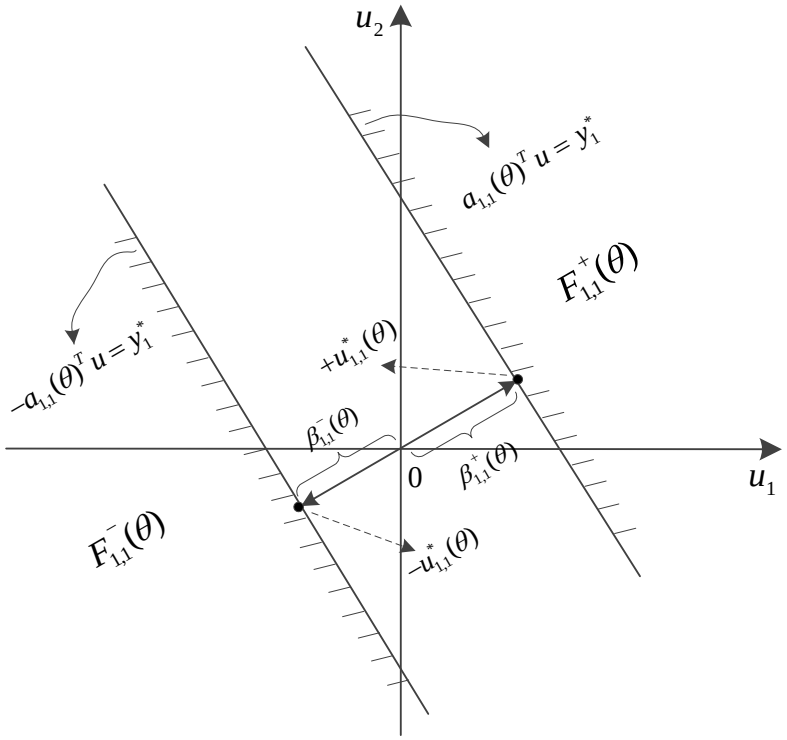

Figure 3.13: Illustration of an symmetric elementary failure region

is low [78]. For the particular case of linear structures subject to Gaussian loading, [77] proposes a very efficient ISD which is defined as a weighted sum of probability density functions conditioned on the elementary failure events described previously:

$$p_{IS,u}(u) = \sum_{i=1}^{n_r} \sum_{k=1}^{n_T} w_{i,k}(\mathbf{x}) p_u(u|F_{i,k}(\mathbf{x})) \tag{3.34}$$

where $p_u(u|F_{i,k}(\mathbf{x}))$ is the probability distribution of U conditioned on the elementary failure event $F_{i,k}$. $w_{i,k}$ is the weight associated with the elementary failure domain $F_{i,k}(\mathbf{x})$ and is defined such that [34]:

$$w_{i,k}(\mathbf{x}) = \frac{\Phi(-\beta_{i,k}(\mathbf{x}))}{\sum_{i=1}^{n_r} \sum_{k=1}^{n_T} \Phi(-\beta_{i,k}(\mathbf{x}))} \tag{3.35}$$

Inserting eq.(3.34) and eq.(3.35) into eq.(3.33) and representing $p_u(u|F_{i,k}(\theta))$ by the Bayes' theorem yields the following estimator for the failure probability [77]:

$$\hat{P}_f^C(\mathbf{x}) = \frac{1}{N}\sum_{v=1}^{N}\frac{\hat{P}_{f,s}(\mathbf{x})}{\sum_{i=1}^{n_r}\sum_{k=1}^{n_T} I_{F_{i,k}}(\mathbf{x}, z^{(v)})} \tag{3.36}$$

where $I_{F_{i,k}}(\mathbf{x}, z)$ is an indicator function equal to one in case the (i,k)th failure event takes place and zero, otherwise; and $\hat{P}_{f,s}(\mathbf{x})$ is the sum of the probabilities of occurrence of the elementary failure events and is defined as:

$$\hat{P}_{f,s}(\mathbf{x}) = 2(\sum_{i=1}^{n_r}\sum_{k=1}^{n_T}\Phi(-\beta_{i,k}(\mathbf{x}))) \tag{3.37}$$

As a summary, the estimation of the failure probability using the estimator in eq.(3.33) requires the characterization of the elementary failure domains through the design points and reliability indices. Samples of the vector U distributed according to the ISD in eq.(3.34) are required as well. For details on how to generate these samples, see Table 3.1. We called this K-L expansion based IS method KL-IS method.

3.4.6 Comparisons between standard MCS and KL-IS

To compare the efficiency of standard MCS and KL-IS method, the CPU time is firstly employed as the index. The object structure is a ten-story shear building shown in Figure 3.14, excited by stochastic ground excitations. This system has been studied by Hadi and Arfiadi in [79] and by Lee et al.in [80] and, for the sake of comparison, the structural parameters used in the present work are the same as those used by the afore-mentioned authors.

The nominal values of structural parameters are assumed to be $m_i = 360 \times 10^3 kg$, $k_i = 650 \times 10^6 N/m$, and $c_i = 6.2 \times 10^6 Ns/m$ [81], where $i = 1, ..., 10$. The natural frequency and damping ratio of the TMD are $2\pi rad/s$ and 3% respectively. The observation time period is $[0, 20s]$, with time step $T_s = 0.05s$. The threshold values for each DOF are set the same, *i.e.* $y_{lim} = 0.025m$. Noticed that the threshold here denotes the relative response between two consecutive DOFs.

The Gaussian white noise process with the standard deviation $\sigma = \sqrt{2\pi S_0/T_s}$ is assumed as the excitation process for the structure, where the power spectral density $S_0 = 0.031m^2/s^3$. In standard MCS, we record the average CPU time consumed

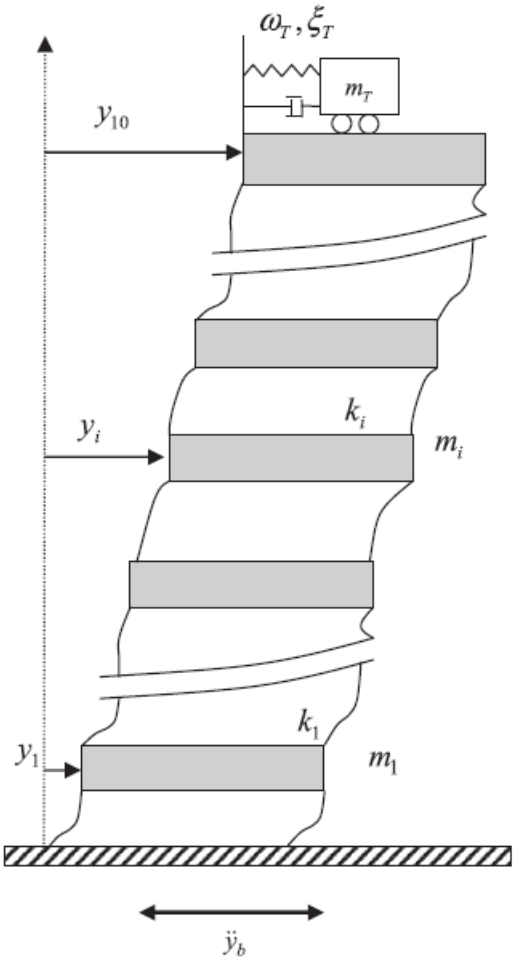

Figure 3.14: Ten storey shear building under earthquake excitations [81]

Table 3.1: Sampling scheme according to ISD proposed in eq.(3.34) [77]

Input: Deterministic basis function vectors $\mathbf{a}_{i,k}(\mathbf{x})$, the set of weights $\{w_{i,k}(\mathbf{x})\}_{1,1}^{n_r,n_T}$.

Output: N samples, *i.e.* $\{z^{(v)}\}_{v=1}^{N}$.

Step1: Set $v = 1$. Draw a pair of indices (I, K) from the set $\{(i,k)\}$, $i \in \{1, ..., n_r\}$, $k \in \{1, ..., n_T\}$. The pair (i,k) is sampled according to its weight $w_{i,k}(\mathbf{x})$.

Step2: Generate a sample u from the distribution $p_u(u)$.

Step3: Draw randomly two numbers c_1, c_2 from $[0,1]$ uniform distribution.

Step4: Calculate $\alpha = -\Phi^{-1}((1-u_1)\Phi(-\beta_{I,K}(\mathbf{x})))$, where $\Phi^{-1}(\cdot)$ is the inverse CDF of the standard normal distribution.

Step5: Compute the vector $\gamma_{I,K} = u^*_{I,K}(\mathbf{x})/||u^*_{I,K}(\mathbf{x})||$.

Step6: Set the sample $z^{(v)}$ as $z^{(v)} = sign(0.5 - c_2) * [u + (\alpha - u^T\gamma_{I,K})\gamma_{I,K}]$, where $sign(0.5 - c_2)$ equals 1 if $c_2 \leq 0.5$ and -1 otherwise.

Step7: Judge the value v. If $v < N$, $v = v + 1$, return to *step1*; otherwise, stop and output the IS samples $z^{(v)}, v = 1, ..., N$.

to analyze one sample of the excitation process, then, by eq.(2.22) in chapter 2, we can calculate the number of samples needed to calculate a particular value of P_f. It is assumed the COV is 10%, which means that the range scope of the estimated P_f around the mean is 10%. To apply the KL-IS method, the number of terms in the K-L expansion is set as $n_{KL} = 300$, and the number of importance samples generated is set as $N = 2000$. Figure 3.15 and Figure 3.16 show the evolution of CPU time of the two methods, according to different values of P_f. In this simulation, due to the reason that the P_f values range extensively from 10^{-5} to 5×10^{-1}, the logarithm (10 as the base) of P_f is used to denote the x-coordinate.

From Figure 3.15 it is known that, for larger values of P_f the number of excitation samples is small. Actually we can calculate the CPU time by the expression $T =$

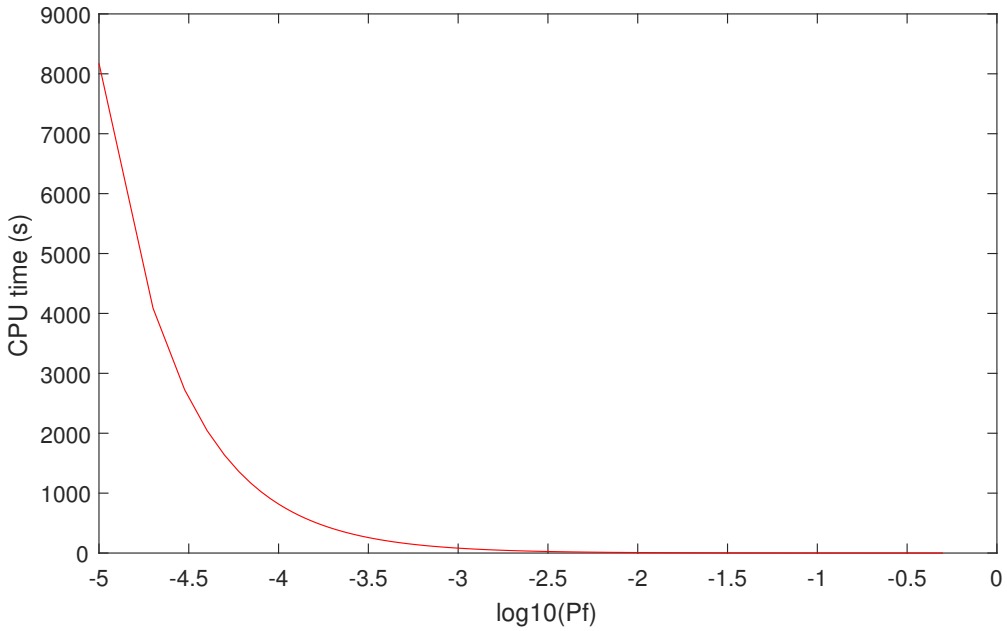

Figure 3.15: Evolution of CPU time according to different values of P_f (by standard MCS)

$\bar{t} * \frac{100}{P_f}$ after some simple algebra from eq.(2.22). Here $\bar{t}$ is the average CPU time spent to analyze one sample of the excitation process. According to the simulation results, $\bar{t}$ is calculated as $7.55 \times 10^{-4}s$. As an example, if the actual P_f value is 10^{-4}, then at least $\frac{100}{10^{-4}} = 10^6$ samples are needed to analyze, resulting in $T = 7.55 \times 10^{-4} * \frac{100}{10^{-4}} = 755s$ spent according to the accuracy demand. However, if the actual P_f value is 10^{-1}, then only $\frac{100}{10^{-1}} = 10^3$ samples are needed, which consumes $T = 7.55 \times 10^{-4} * \frac{100}{10^{-1}} = 0.755s$. Therefore, the standard MCS is only practically suitable to calculate relatively larger P_f (for example larger than 1%).

In contrast, Figure 3.16 shows a short but almost constant value of CPU time. The reasons are as follows. Firstly, the KL-IS method employs discrete representation method, *i.e.* K-L expansions, to represent the excitations for the structure. The K-L expansion decomposes the Gaussian random process into deterministic part and uncertain part. The deterministic part, together with the structural impulse responses (see Figure 3.17). Because the failure probabilities are calculated from the relative response of different DOFs, these responses are also calculated, see Figure 3.18, is used to calculate the convolution integrals so that a coefficient matrix is

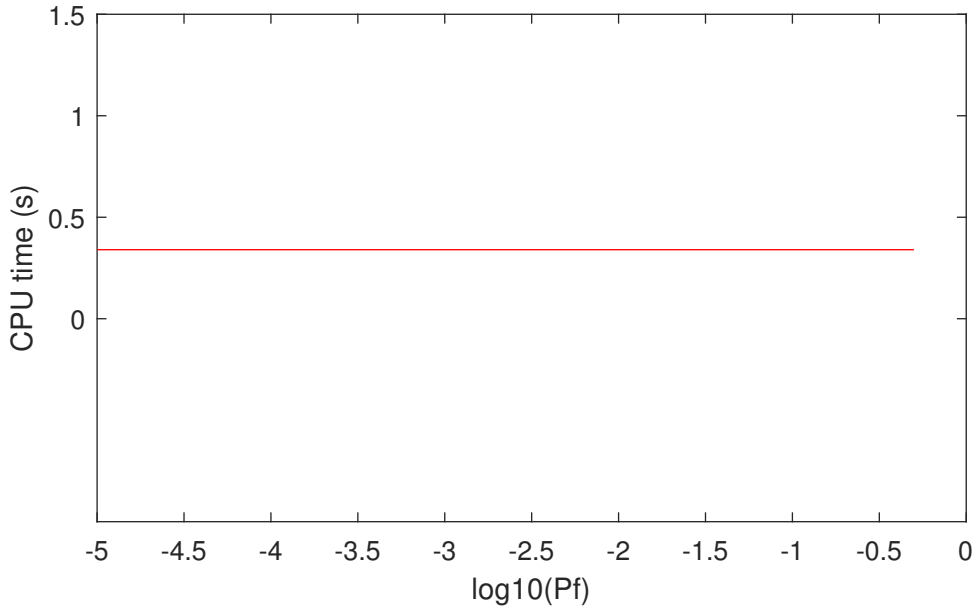

Figure 3.16: Evolution of CPU time according to different values of P_f (by KL-IS)

obtained that comprises the vectors $\mathbf{a}_{i,k}(\mathbf{x})$, where $i = n_r$, $k = 1, ..., n_T$ (see eq. (3.17)). It is noticed that once this coefficient matrix is determined, the failure probability of any DOF at any sampled time point can be directly calculated via eq.(3.19) (see results in Figure 3.19).

Afterwards, we just need to generate the standard Gaussian variables (with dimension n_{KL}) and feed them into the coefficient matrix to calculate the structural responses. According to the simulation records, these procedures is not time consuming (about $4s$). Secondly, in the generating process of importance samples, with the aide of the already calculated coefficients (see eq. (3.17)) the importance samples are very efficient to obtain. This part consumes less than $0.5s$.

Even though the number of importance samples are increased, the time consumed does not change so much. The special mechanism of KL-IS method makes it very powerful to calculate very small values of P_f without large sacrifice of time. Therefore, the KL-IS method is very suitable to calculate small P_f (for example less than 1%). Moreover, the instant failure probabilities of different DOFs within the observed time interval are calculated from their coefficient matrices, see Figure 3.19. It is found that, from the 1st DOF to the top DOF the failure probabilities decrease

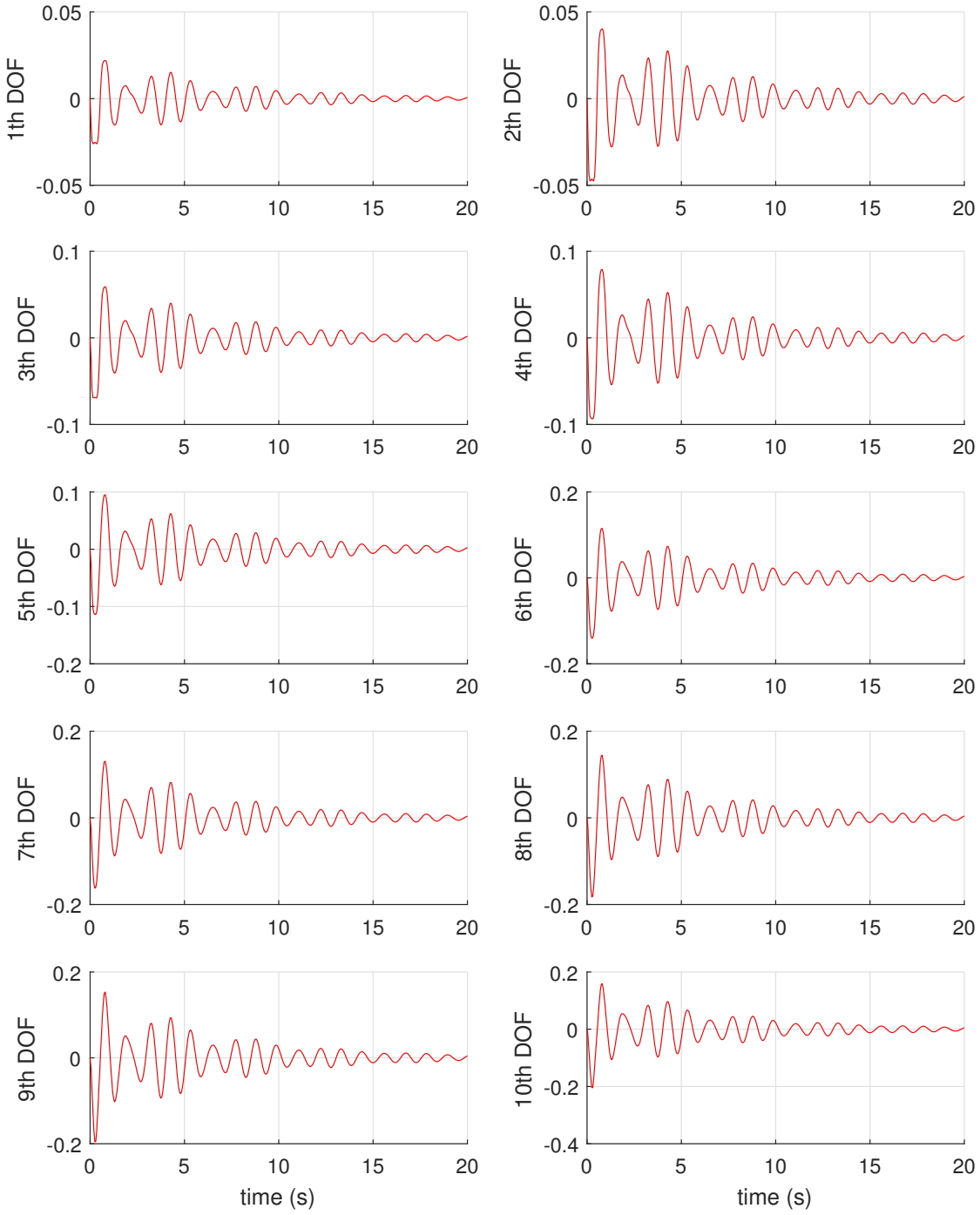

Figure 3.17: Impulse (absolute) response of different DOFs

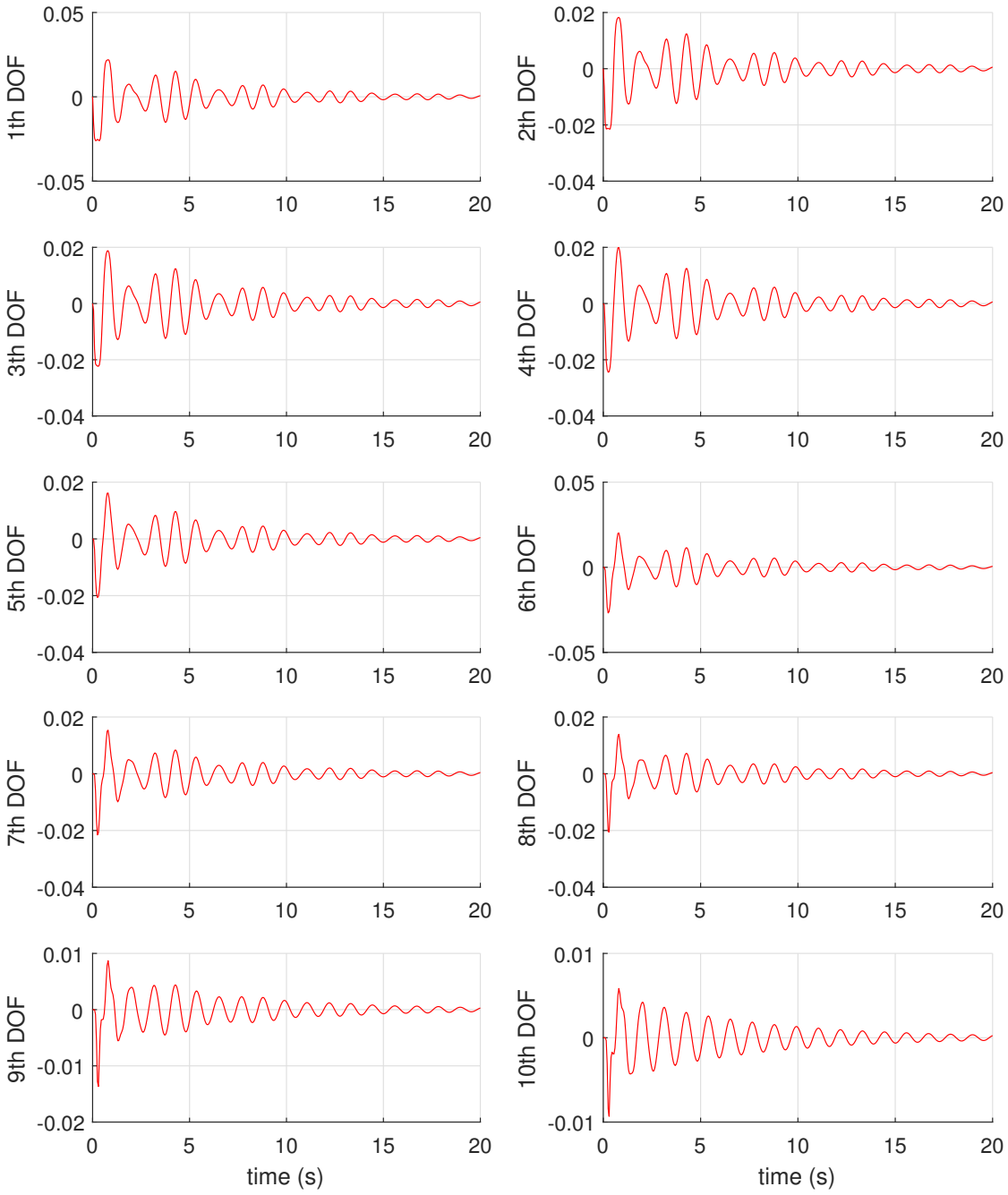

Figure 3.18: Impulse (relative) response of different DOFs

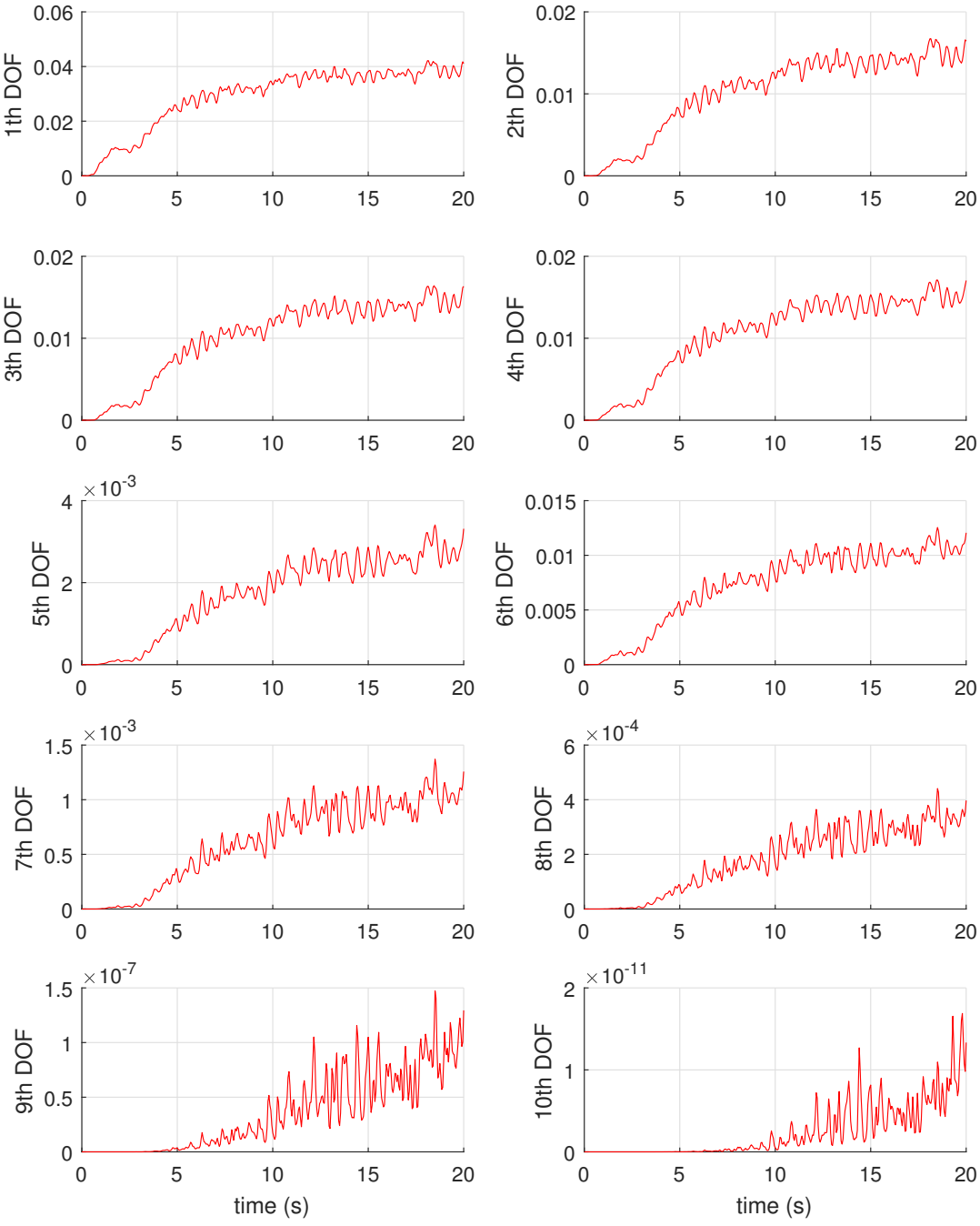

Figure 3.19: Evolution of failure probabilities of different DOFs

evidently. This result tells that the relative responses of these DOFs decrease. This phenomenon can be reflected in the relative responses due to unit impulse excitation, see Figure 3.18.

3.5 Chapter summary

In this chapter, we have presented the fundamental knowledge of uncertainty quantification. The concept of conditional failure probability is also introduced. This chapter describes the basic procedures to handle the structural uncertainties via machine learning models. Besides, the KL-IS method has been presented in detail due to its clear advantage over the standard MCS method in evaluating very small failure probabilities. In chapter 4, the machine learning theories will be extensively explored and studied in conditional failure probability modeling and prediction.

Chapter 4

Structural Reliability assessment: the Random Forest approach

Contents

4.1 Introduction

Theoretically, evaluation of the probability of failure can be accomplished by solving a multivariate integral analytically or numerically within the failure domain [82].

But this is only computationally practical for low-dimension (generally less than five) problems. Thus, much of the efforts have been devoted to developing alternative approaches, such as surrogate models and simulation methods.

Surrogate models were developed to deal with highly non-linear or implicit LSFs. They are introduced to reduce computational burden in reliability analysis. Response surface method (RSM) is among the most popular surrogate models. Generally, researchers combine RSM with other approaches to refine the model parameters [83]. Support vector machine (SVM) has also gained much attention recently. Pan and Dias [84] combined an adaptive SVM and Monte-Carlo simulation to solve non-linear and high-dimensional problems in reliability analysis. According to Dai et. al [85], a new multi-wavelet linear programming SVM for the reliability analysis. Other surrogate models have been proposed by the other researchers [86], [87].

In terms of simulation methods, Monte-Carlo simulation (MCS) technique is representative to solve high-dimensional and non-linear complex problems. It is dimension-independent, and often regarded as a benchmark to evaluate the other probabilistic methods due to its versatility and robustness. Zio [88] described in detail the basic theories about MCS techniques used for reliability analysis in his recent book. Jose et al. [89] proposed a MCS methodology to estimate the reliability of a multi-state network, and was able to obtain accurate approximations of the reliability. Daoud and Mahmoud [90] proposed to use MCS to obtain the task route reliability of the mobile agent based systems (MABS), and the simulation results showed its robustness. Padmanabhan et. al [26] used Monte Carlo Simulation techniques for Reliability-Based Optimization (RBO), they combined limit state approximations with MCS to improve the efficiency of reliability analysis. Some researchers tried to improve the efficiency of the analysis by combining MCS with other methods, especially surrogate models such as SVM [68], artificial neural networks (ANN) [91] and [92]Kriging.

Seen as a classification/ regression procedure, the prediction task can be realized by machine learning models among which SVM [37] and [35]ANN were widely considered in recent articles. However, they cannot avoid shortcomings in all situations. The disadvantages in ANNs mainly include the complex architecture optimization,

low robustness and enormous training time [93]. SVM is time consuming for large-scale applications and sometimes shows large error in sensitivity calculation of reliability index. Naive Bayes model learns the a priori distribution from the observed data and build the posterior distribution for new observations. But it is based on the conditional independence assumption, and for the continuous variables it needs to discretize them [94]. Bayesian networks (BNs) employ conditional probability tables (CPT) to deal with inter-dependent variables, but the results highly rely on assumptions about the target system behavior [95]. Besides, the rules encoded by BNs are difficult to interpret efficiently [96].

Random Forest is a tree method distinguished for its robustness and high accuracy in modeling and predictions [97]. However, its application to reliability prediction is still limited in current literatures. This part therefore aims to explore the feasibility of RF as well as examine its performance in reliability modeling and prediction of passive controlled structures. The remainder of the paper is organized as follows. Section 4.2 provides the response analysis of multi-TMD structures subject to seismic excitations. Section 4.3 introduces in detail the framework of the proposed method. Section 4.4 provides the numerical simulations and performance analysis. Section 4.5 concludes this article with comments.

4.2 Structural response analysis

An ideal mechanical TMD system (see Figure 4.1) is composed of a main structure represented by a N degree-of-freedom (N-DOF) system, and a TMD structure with n degree-of-freedoms (n-DOF). All the DOFs of the TMDs are arranged in parallel to absorb the vibration energy of the main structure.

In case of the TMD system excited by a base acceleration $\ddot{y}_b$, the first $(n-1)$ degrees of freedom (DOFs) of the structural response is determined by the following motion equation,

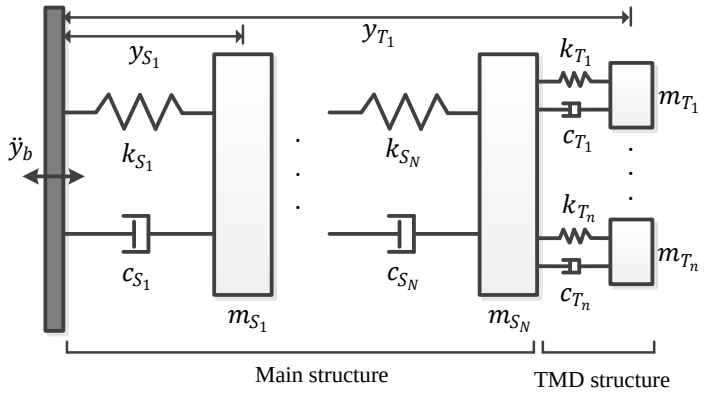

Figure 4.1: Multi-DOF TMD system

$$\begin{aligned} m_{S_i} \cdot \ddot{y}_{S_i} + c_{S_i} \cdot (\dot{y}_{S_i} - \dot{y}_{S_{i-1}}) + k_{S_i} \cdot (y_{S_i} - y_{S_{i-1}}) \\ + c_{S_{i+1}} \cdot (\dot{y}_{S_i} - \dot{y}_{S_{i+1}}) + k_{S_{i+1}} \cdot (y_{S_i} - y_{S_{i+1}}) \\ = -m_{S_i} \cdot \ddot{y}_b \end{aligned} \tag{4.1}$$

where $i = 1, 2, ..., n-1, y_{S_0} = 0$. After some algebra, it is equivalent to

$$\begin{aligned} m_{S_i} \cdot \ddot{y}_{S_i} + (c_{S_i} + c_{S_{i+1}}) \cdot \dot{y}_{S_i} - c_{S_i} \cdot \dot{y}_{S_{i-1}} - c_{S_{i+1}} \cdot \dot{y}_{S_{i+1}} \\ - (k_{S_i} + k_{S_{i+1}}) \cdot y_{S_i} - k_{S_i} \cdot y_{S_{i-1}} \\ - k_{S_{i+1}} \cdot y_{S_{i+1}} \\ = -m_{S_i} \cdot \ddot{y}_b \end{aligned} \tag{4.2}$$

For the nth degree of freedom, the motion equation is

$$\begin{aligned} m_{S_n} \cdot \ddot{y}_{S_n} + c_{S_n} \cdot (\dot{y}_{S_n} - \dot{y}_{S_{n-1}}) + k_{S_n} \cdot (y_{S_n} - y_{S_{n-1}}) \\ + c_{T_1} \cdot (y_{S_n} - y_{T_1}) + k_{T_1} \cdot (y_{S_n} - y_{T_1}) \\ + ... + c_{T_m} \cdot (\dot{y}_{S_n} - \dot{y}_{T_m}) + k_{T_m} \cdot (y_{S_n} - y_{T_m}) \\ = -m_{S_n} \cdot \ddot{y}_b \end{aligned} \tag{4.3}$$

Eq.(4.3) is equivalent to

$$\begin{aligned} m_{S_n}\cdot\ddot{y}_{S_n} + c_{S_n}\cdot\dot{y}_{S_n} + k_{S_n}\cdot y_{S_n} - c_{S_n}\cdot\dot{y}_{S_{n-1}} - k_{S_n}\cdot y_{S_{n-1}} & \\ + \sum_{i=1}^{m} c_{T_i}\cdot(\dot{y}_{S_n} - \dot{y}_{T_i}) & \\ + \sum_{i=1}^{m} k_{T_i}\cdot(y_{S_n} - y_{T_i}) & \\ = -m_{S_n}\cdot\ddot{y}_b & \end{aligned} \tag{4.4}$$

Therefore, we obtain the motion equations for all DOFs. For all the TMDs, the motion equations are

$$\begin{aligned} & m_{T_1}\cdot\ddot{y}_{T_1} + c_{T_1}\cdot(\dot{y}_{T_1} - \dot{y}_{S_n}) + k_{T_1}\cdot(y_{T_1} - y_{S_n}) = -m_{T_1}\cdot\ddot{y}_b \\ & m_{T_2}\cdot\ddot{y}_{T_2} + c_{T_2}\cdot(\dot{y}_{T_2} - \dot{y}_{S_n}) + k_{T_2}\cdot(y_{T_2} - y_{S_n}) = -m_{T_2}\cdot\ddot{y}_b \\ & ... \\ & m_{T_m}\cdot\ddot{y}_{T_m} + c_{T_m}\cdot(\dot{y}_{T_m} - \dot{y}_{S_n}) + k_{T_m}\cdot(y_{T_m} - y_{S_n}) = -m_{T_m}\cdot\ddot{y}_b \end{aligned} \tag{4.5}$$

By assuming the state space $\mathbf{y} = [y_{S_1}, ..., y_{S_n}, y_{T_1}, ..., y_{T_n}]$, we express the motion of the system as

$$\mathbf{M}\ddot{\mathbf{y}}(t) + \mathbf{C}\dot{\mathbf{y}}(t) + \mathbf{K}\mathbf{y}(t) = -\mathbf{M}\mathbf{r}\ddot{y}_b \tag{4.6}$$

where $\mathbf{r} = [1, 1, ..., 1]^T$ consists of $(n + m)$ elements. The mass matrix is $\mathbf{M} = diag[m_{S_1}, ..., m_{S_n}, m_{T_1}, ..., m_{T_m}] = diag[\mathbf{M}_S, \mathbf{M}_T]$. Here, $\mathbf{M}_S = diag[m_{S_1}, ..., m_{S_n}]$, $\mathbf{M}_T = diag[m_{T_1}, ..., m_{T_n}]$. The damping matrix $\mathbf{C}$ is a square matrix whose non-zero elements are

$$\mathbf{C}_{ii} = \begin{cases} c_{S_i} + c_{S_{i+1}}, & i = 1, ..., n-1 \\ c_{S_i} + \sum_{i=1}^{m} c_{T_i}, & i = n \\ c_{T_{i-n}}, & i = n+1, ..., n+m \end{cases} \tag{4.7}$$

Similar to $\mathbf{M}$, $\mathbf{C} = diag[\mathbf{C}_S, \mathbf{C}_T]$. Here, $\mathbf{C}_S$ consists of the first n rows and the first n columns of $\mathbf{C}$; $\mathbf{C}_T$ consists of the last m rows and the last m columns of $\mathbf{C}$. The stiffness matrix $\mathbf{K}$ is a square matrix whose non-zero elements are

$$\mathbf{K}_{ii} = \begin{cases} k_{S_i} + k_{S_{i+1}}, & i = 1, ..., n-1 \\ k_{S_i} + \sum_{i=1}^{m} k_{T_i}, & i = n \\ k_{T_{i-n}}, & i = n+1, ..., n+m \end{cases} \tag{4.8}$$

In a similar way, $\mathbf{K} = diag[\mathbf{K}_S, \mathbf{K}_T]$. Here, $\mathbf{K}_S$ consists of the first n rows and the first n columns of $\mathbf{K}$; $\mathbf{K}_T$ consists of the last m rows and the last m columns of $\mathbf{K}$. Assume $\mathbf{z}_0 = [y_{S_1}, \ldots, y_{S_n}, y_{T_1}, \ldots, y_{T_m}, \dot{y}_{s1}, \ldots, \dot{y}_{sn}, \dot{y}_{T_1}, \ldots, \dot{y}_{T_n}]^T$, we obtain

$$\dot{\mathbf{z}}_0 = \mathbf{A}_0 \mathbf{z}_0 + \mathbf{r}_0 \ddot{y}_b, \tag{4.9}$$

where $\mathbf{A}_0 = [\mathbf{O}_{n+m}, \Lambda_{n+m}; \mathbf{K}_U, \mathbf{K}_D]$. $\mathbf{O}_{n+m}$ and Λ_{n+m} are zero matrix and identity matrix respectively, the sizes of which are all $(n+m) \times (n+m)$. $\mathbf{r}_0 = [0, ..., 0, -1, ..., -1]^T_{1\times(2n+2m)}$. $\mathbf{K}_U = [\mathbf{K}_0, \mathbf{A}_1, \mathbf{C}_0, \mathbf{A}_2]$, where

$$\mathbf{K}_0 = -\mathbf{M}_S^{-1} \times \mathbf{K}_S \tag{4.10}$$

$$\mathbf{C}_0 = -\mathbf{M}_S^{-1} \times \mathbf{C}_S \tag{4.11}$$

$$\mathbf{A}_1 = [[\mathbf{0}]_{(n-1)\times m}; [k_{T_1}, ..., k_{T_m}]/m_{S_n}] \tag{4.12}$$

$$\mathbf{A}_2 = [[\mathbf{0}]_{(n-1)\times m}; [c_{T_1}, ..., c_{T_m}]/m_{S_n}] \tag{4.13}$$

$\mathbf{K}_D = [\mathbf{B}_1, \mathbf{K}_1, \mathbf{B}_2, \mathbf{C}_1]$, where

$$\mathbf{K}_1 = -\mathbf{M}_T^{-1} \times \mathbf{K}_T \tag{4.14}$$

$$\mathbf{C}_1 = -\mathbf{M}_T^{-1} \times \mathbf{C}_T \tag{4.15}$$

$$\mathbf{B}_1 = [[\mathbf{0}]_{m\times(n-1)}, [\frac{k_{T_1}}{m_{T_1}}, ..., \frac{k_{T_m}}{m_{T_m}}]^T] \tag{4.16}$$

$$\mathbf{B}_2 = [[\mathbf{0}]_{m\times(n-1)}, [\frac{c_{T_1}}{m_{T_1}}, ..., \frac{c_{T_m}}{m_{T_m}}]^T] \tag{4.17}$$

Applying Kanai-Tajimi (KT) model [98] to simulate the base acceleration $\ddot{y}_b$,

$$\begin{cases} \ddot{y}_f + 2\varepsilon_f\omega_f\dot{y}_f + \omega_f^2 y_f = -w(t) \\ \ddot{y}_b = \ddot{y}_f + w(t) = -(2\varepsilon_f\omega_f\dot{y}_f + \omega_f^2 y_f) \end{cases} \tag{4.18}$$

Here $w(t)$ is zero mean Gaussian white noise, *i.e.* the excitation to the system. ω_f and ε_f are the natural frequency and damping ratio of the filter, y_f is the relative response. We build the global state space vector of the system by, $\mathbf{Z} = [y_{s_1}, \ldots, y_{s_n}, y_{T_1}, \ldots, y_{T_m}, y_f, \dot{y}_{s_1}, \ldots, \dot{y}_{s_n}, \dot{y}_{T_1}, \ldots, \dot{y}_{T_m}, \dot{y}_f]^T$.
Then the motion of the system becomes

$$\dot{\mathbf{Z}} = \mathbf{A}\mathbf{Z} + \mathbf{f}, \tag{4.19}$$

where $\mathbf{f}$ is a column vector that contains zeros except the last element $-w(t)$. The matrix $\mathbf{A}$ is extended from $\mathbf{A}_0$ by considering the state y_f. By now, the standard state space is built as

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \\ \mathbf{y} &= \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \end{aligned} \tag{4.20}$$

where the input vector $\mathbf{x}$ corresponds to $\mathbf{Z}$ in eq.(4.19); the output $\mathbf{y}$ corresponds to the response of certain DOFs. Obviously, $\mathbf{B}\mathbf{u} = \mathbf{f}$, $\mathbf{D}\mathbf{u} = 0$, $\mathbf{C}$ is a diagonal matrix to decide the output response of interest.

4.3 Basic thought of the proposed methodology

The basic thought of the proposed methodology is schematically described in Figure 4.2. In deterministic case where the structural properties are known, the risk of failure can be quantified as the failure probability.
However, the situation changes when uncertainties are associated with the material properties. In the framework of uncertainty, a number of samples are taken from

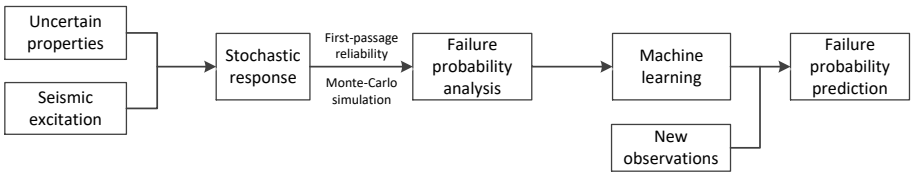

Figure 4.2: The schematic of the proposed method

the uncertainty space, and the reliability analysis is implemented via First-passage probability theory and Monte-Carlo simulations. Subsequently, based on the reliability data obtained in the first step, a machine learning (ML) model is induced. This model can be used to do reliability predictions whenever new observations of the structural properties are available.

4.3.1 Expression of uncertainties

We assume the uncertainties only exist in the mass (m), the damping factor (c) and the stiffness (k) of the primary system. In the present paper, the uncertainties of the parameters adopt the forms of normal distributions [99], $m \sim N\left(m_0, \sigma_m^2\right)$, $c \sim N\left(c_0, \sigma_c^2\right)$, $k \sim N\left(k_0, \sigma_k^2\right)$. Here m_0 , c_0 and k_0 are the nominal mass, damping factor and the stiffness, and σ_m^2 ,σ_c^2 , σ_k^2 are the corresponding variances.

4.3.2 Reliability evaluation

First-passage probability In vibration analysis of structural systems subjected to uncertain excitation modeled as stochastic process, it is generally required to estimate the probability that the system's response will stay within safe, prescribed limits, within a specified time span. The determination of such a probability is usually called 'first-passage problem' [100].

Mathematically, the first-passage problem can be expressed as: to find the probability, $P(t)$, that any one of the response states of interest, $Y_i(t)$, $i = 1, \cdots, m$ of a system exceeds in magnitude a critical barrier at least once in a defined time interval $[0, T]$ (see Figure 4.3),

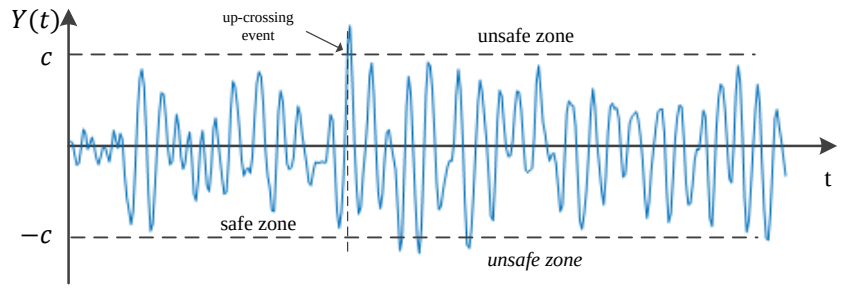

Figure 4.3: A response process containing exceedance events

$$P_f = Prob(F) = P\left(|Y_i(t)| > c,\ t \in [0,\ T]\right) \tag{4.21}$$

Where F denotes a failure event of the structure. We note that the definition of the failure criterion is of importance to perform reliability estimation. In the framework of first-passage theory, the structure failure probability can be understood as the risk that the structure responses exceed a predefined limit within a certain interval of time. For the proposed seismic model, the Gaussian white noise process is selected as the source of excitation.

MCS for reliability evaluation In simulation methods, the stochastic excitation is specified as a certain number of input random variables [77], $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ which can be simulated as a generation of the excitation. Thus, the output response will be a function of these variables. To apply MCS for reliability evaluation, N samples of the multi-dimensional random vector $\mathbf{x}$ will be obtained from a multi-variable Gaussian distribution. Each sample is seen as a white noise process. Obviously, the n variables correspond to n sample of time points in $[0, T]$. In terms of joint probability density function (PDF) $f_{\mathbf{x}}(x)$ for input random variables, the failure probability can be analytically expressed as the integral,

$$P_f = \int_F f_X(\mathbf{x})\mathrm{d}\mathbf{x} = \underset{G(X)\leq 0}{\int\int} \ldots \int f_X(x_1,\ x_2,\ \ldots,\ x_n)dx_1dx_2\ldots dx_n \tag{4.22}$$

where $F : G(\mathbf{x}) \leq 0$ is the failure region in n-dimensional space of X. It is practically impossible to carry out direct calculation of this integration. Thus, Monte-Carlo estimation is used as a numeric integration approach,

$$\hat{P}_f = E\left[I_F(\mathbf{x})\right] = \frac{1}{N}\sum_{i=1}^{N} I\left[G\left(\mathbf{x}^{(i)}\right) \leq 0\right] \tag{4.23}$$

where $I_F(\cdot)$ is the indicator function: $I_F(\mathbf{x}) = 1$ if $\mathbf{x} \in F$ and $I_F(\mathbf{x}) = 0$ otherwise; $\mathbf{x}^{(i)}$ is the ith sample of the joint probability distribution.

The failure criterion is defined as the following: $y_{max} > y_{limit}$, 'failure'; $y_{max} \leq y_{limit}$, 'function'. Here, $y_{max} = max\left(\mid y_s \mid\right)$ is the maximum displacement of the response among all the degrees of the base structure. More intuitively, the Monte Carlo estimator of the structure failure probability is written as [16],

$$\hat{P}_f = \frac{1}{N_{mc}}\sum_{i=1}^{N_{mc}} I_f\left(\mathbf{x}^{(i)}\right) \tag{4.24}$$

where $\mathbf{x}^{(i)}$ is the ith sample of the excitation process; N_{mc} is the number of samples used in MCS, and $I_f(\cdot)$ is the indicator function that satisfies

$$I_f(\cdot) = \begin{cases} 1, & y_{max} > y_{limit} \\ 0, & y_{max} \leq y_{limit} \end{cases} \tag{4.25}$$

4.3.3 Different ML methods for reliability modeling

Several machine learning models have been explored in our research. Some of them demand prior knowledges about the data while others just focus on the data itself. During the research, we tested some representative methods on the data, including Bayesian inference [101], k-nearest neighbor (k-NN) [102], Gradient Boosting (GB) [103] and Random Forest. Some brief introductions of them are listed below.

- Bayesian Inference method is based on Bayes' theorem. It operates by firstly assigning a prior distribution $\pi(\theta)$ for the parameters. Then, considering the experimental data D, $\pi(\theta)$ is updated by the likelihood $\pi(D|\theta)$. The posterior distribution $\pi(\theta|D)$ is obtained by re-normalizing the updated $\pi(\theta)$, *i.e.*

$$\pi(\theta|D) = \frac{\pi(D|\theta)\pi(\theta)}{\int_\theta \pi(D|\theta)\pi(\theta)} \propto \pi(D|\theta)\pi(\theta) \tag{4.26}$$

- k-NN is a type of instance-based learning where the function is only approximated locally. In regression case, the input consists of the k closest training examples in the feature space, and the output is the average of target values of the k neighbors. The value k can be determined by cross-validation, and the 'closeness' can be measured by Euclidean distance. We can also use weighted k-NN where the neighbors are weighted, according to their distances, to determine the output.

- Gradient Boosting (GB) is a generalization of Adaptive-Boosting (AdaBoost). It is a kind of ensemble learning method that combines multiple 'weak learners' as a weighted sum so that the final output is determined. The weak learners can be decision trees, decision stumps *et.al.*

- Random Forest. Even though RF and GB belong to the same category where the decisions are made by combining the outputs of all individual trees, they are largely different in two aspects. Firstly, the way a tree is built. In GB, the trees are built one by one with each tree being built in the background of the previous one; In RF, trees are induced independently, and the randomness in the tree building process helps reduce the correlation among trees. Secondly, the way to make predictions. GB makes predictions by calculating the weighted sum of the decisions of all trees, while RF just averages the decisions.

The exploration of machine learning techniques in structural reliability research is far from enough, and we are the first to introduce RF in reliability analysis. Besides, our research shows that the RF model outperforms the other three models on our data. The following part will introduce RF model in detail.

4.3.4 RF for Reliability modeling

Basic principles of the model Random Forest is an ensemble learning method that use multiple learning algorithms to obtain better predictive performance than could

be obtained from any of the constituent learning algorithms alone. RF is comprised of many individual trees called classification and regression tree (CART). Each tree is built from a random sample set (*i.e.* bootstrap sample) generated from the training set. As we use bootstrap sampling to obtain the samples, we therefore just obtain a subset of the training data which we call 'Inbag'; the remaining samples are called 'OOB (out-of-bag)' which is the complementary set of the 'Inbag'.

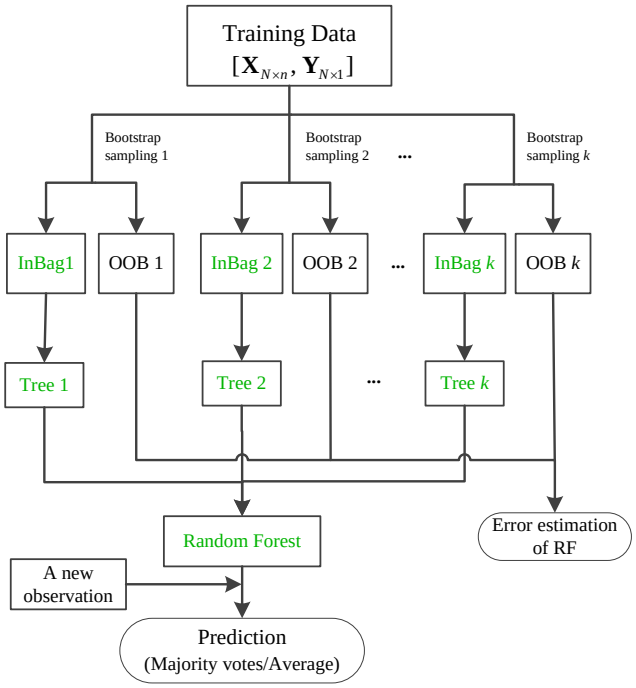

Figure 4.4: Random Forest: from modeling to prediction. D_N is the training data, $\mathbf{X}_{N\times n}$ is the input matrix that contains N observations of the input variables, n is the number of input variables, $\mathbf{Y}_{N\times 1}$ is a column vector that contains the output values

Figure 4.4 illustrates how a Random Forest is learned and then employed to do predictions. The 'OOB' data can be used to estimate the model error. When the model is built, we are able to do predictions for future observations, either by majority voting or averaging the predicted values of all trees.

An underlying assumption of the ensemble process is that the base learners are independent. As the trees become more correlated (less independent), the model error tends to increase. Randomization helps to reduce the correlation among decision trees in order that the model accuracy is improved. Two kinds of randomness exist

in the tree learning process.

- *Bootstrap sampling* Bootstrap sampling is applied to generate a sample set for each tree, which results in about one-third of the samples left out for each bootstrap training set.

- *Node splitting* For each node splitting, a subset of variables is generated by randomly choosing a certain number of variables from the variable set. Then one of these variables is selected, according to an evaluation criterion, to split the node. Generally, a subset contains $sqrt(n)$ of the attributes.

Tree growing process A tree has a binary recursive structure containing several nodes and branches. Each node specifies a variable and a branch specifies the possible values it takes. A tree learning process is actually the node splitting process (see Figure 4.5), beginning with the root node that corresponds to the whole learning set. Each split results in two subsets of the data that falls into this node so that the resulting child nodes are the 'purest'. As to a regression tree, it is grown starting from the root node by repeating the following steps on each node.

Step1 Find each variable's best split. Randomly choose $\sqrt{(n)}$ variables as the candidate variable set. For each candidate variable, sort its values in an ascending order. For the sorted variable, examine all candidate split points to find the one that maximize the goodness of split when the node is split according to it.

Step2 Find the node's best split. Among the best splits found in *step*1, choose the one that maximizes the splitting criterion.

Step3 Node splitting. Split the node using its best split found in *step*2 if the stopping rules are not satisfied, otherwise, the node is taken as a leaf whose value is determined by averaging the decisions falling into this node.

In Figure 4.5, $x_1, x_2, \cdots, x_n$ are the variables in the n-dimension uncertainty space; m is the number of samples from the uncertainty space; y is the target variable that has m target values here.

Splitting criterion At each node, we perform a split: we choose a variable x_i and its value $x_{i,j}$ that minimizes the sum of squared errors (SSE), *i.e.*

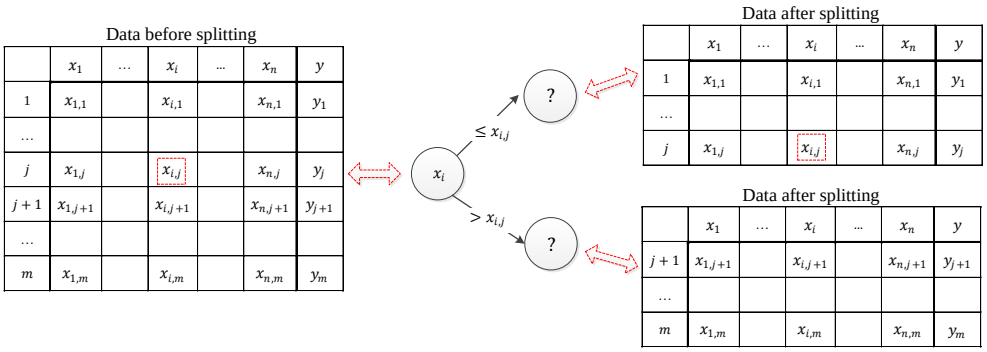

Figure 4.5: An illustrative node splitting process. The symbol '?' means that the variable used to carry out the next splitting needs to be determined

$$\min_{i,j}[\sum_{x\in R_1}(y_i - c_1)^2 + \sum_{x\in R_2}(y_i - c_2)^2] \tag{4.27}$$

where $R_1 = \{x|x_i \leq x_{i,j}\}$ is a the region that satisfies the condition $x_i \leq x_{i,j}$, $R_2 = \{x|x_i > x_{i,j}\}$ is a the region that satisfies $x_i > x_{i,j}$. Besides, c_m is the average of the target values falling into the region $c_m = E(y|x_i \in R_m), m = 1, 2$.

Definition (impurity) Assuming $h(t)$ is the sample set that falls into node t, a is a sample in $h(t)$, then at node t the impurity is defined as

$$i(t) = \frac{1}{N(t)} \sum_{a\in h(t)} (y_a - \overline{y}(t))^2 \tag{4.28}$$

where $N(t)$ is the number of samples falling into node t, y_a is the decision value for sample a , $\overline{y}(t)$ is the average of all decision values in $h(t)$. At node t, the best splitting point s will maximize

$$\Delta i(s,t) = i(t) - p_1 i(t_1) - p_2 i(t_2) \tag{4.29}$$

where t_1 and t_2 are the two child nodes resulting from the splitting point s. p_1 and p_2 are the portions of samples that fall into t_1 and t_2 respectively. We notice that minimizing the SSE is equivalent to maximizing the decrease of impurity (DI) $\Delta i(s,t)$.

Stopping rules The splitting will be stopped if one or more of the following conditions is met: a). the impurity decrement is smaller than the threshold value; b). the current tree depth reaches the preset limit; c). the size of a node is less than the preset minimum size; d). all cases in a node have identical values for each attribute.

Training data preparation As RF belongs to supervised learning category, the training data should contain the input values and output values within each case. The reliability data is obtained as the training data. According to section 4.3.1, a set of samples of the uncertain variables are generated within their predefined uncertainty frameworks. Then, based on the section 4.3.2, the Monte-Carlo simulations are applied to estimated the failure probabilities for these samples.

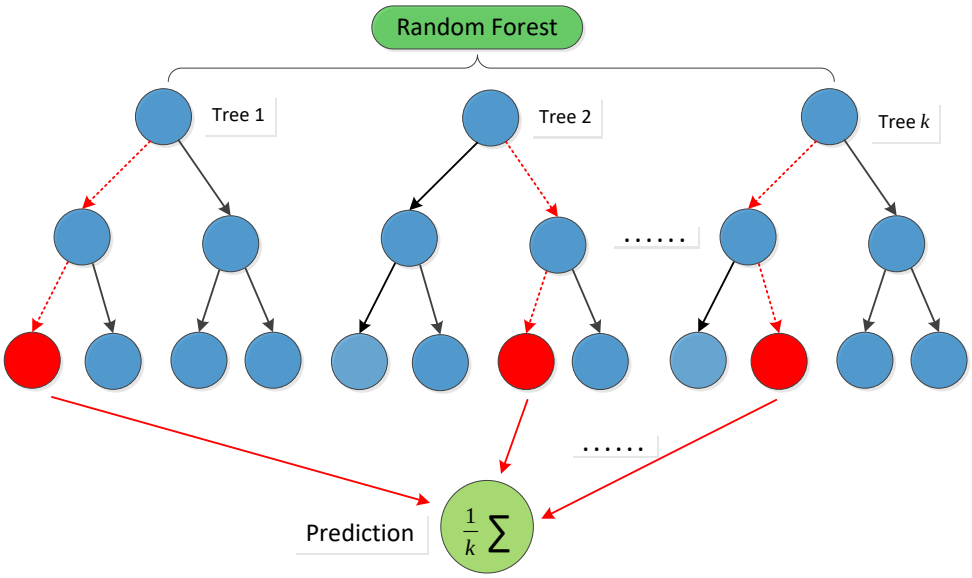

Figure 4.6: Prediction by RF

Predictions by RF When the RF model is built, for a test sample $\mathbf{x}_{new}$, its features values are analyzed according to the rules of each randomly created decision tree sch that the outcomes are predicted and stored. The prediction by RF is carried out by averaging the predictions of all individual trees, *i.e.*

$$\hat{f} = \frac{1}{k}\sum_{i=1}^{k} f_i\left(\mathbf{x}_{new}\right) \tag{4.30}$$

where k is the number of trees within RF, $f_i\left(\mathbf{x}_{new}\right)$ is the prediction by the ith

tree, refer to Figure 4.6. Specifically, $f_i(\mathbf{x}_{new}) = T_{D_N,\theta_i}(\mathbf{x}_{new})$ where T_{D_N,θ_i} is the random tree learned from the ith bootstrap sample. θ_i is an independent random seed that helps generate a bootstrap sample from the training set D_N. In the process of prediction by an individual tree, $\mathbf{x}_{new}$ is pushed through the tree (starting from the root), according to the splitting rules, until it reaches a certain leaf. Then the prediction value is obtained as the leaf, see the illustration in Figure 4.7. In Figure 4.7, it is a regression tree learned from the training data. Each observation in the training data consists of four input values $[x_1, x_2, x_3, x_4]$ and an output value y.

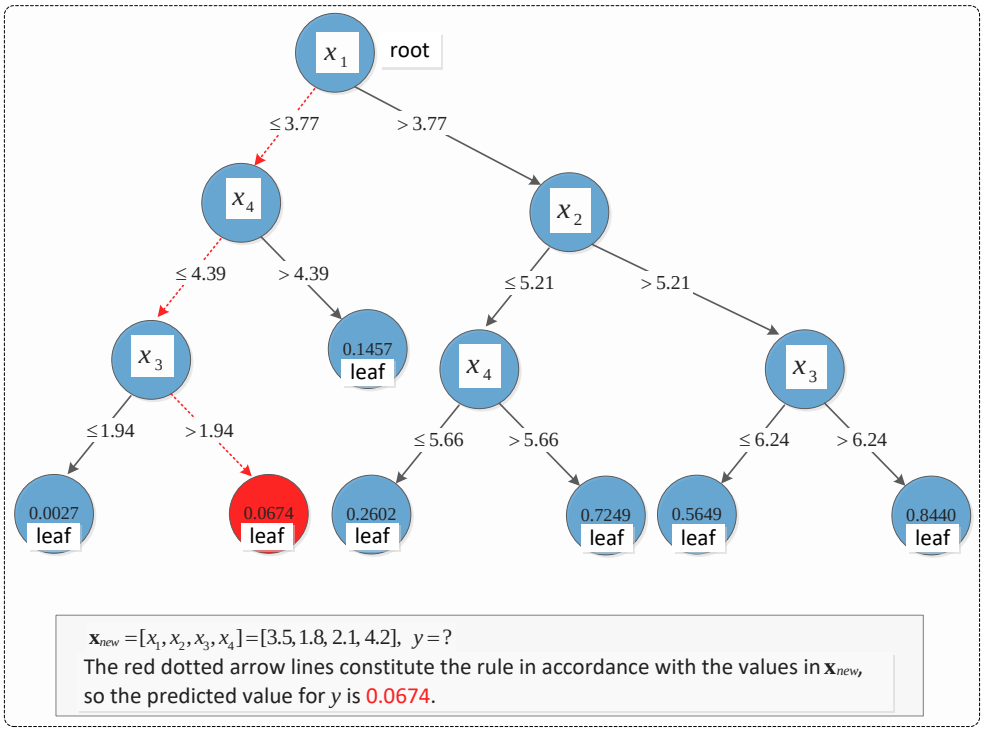

Figure 4.7: Illustrative example: prediction by a single tree

4.4 Simulation and performance analysis

This part devotes itself to exploring the RF ability in different aspects. Some numerical examples are provided to test the RF performance on different structures with single/multi-DOFs. The influence factors on model performance are proposed and

studied. Feature importance analysis is also implemented. Finally the comparison results of different methods are provided.

4.4.1 Three numerical examples

Example 1 The structure consists of 1-DOF base structure and 1-DOF TMD, its parameter values are introduced in Elyes *et.al*[104], see Table 4.1. Besides, the natural frequency and damping ratio of the seismic model are $25.224 rad/s$ and 0.4 correspondingly. The power spectral density (PSD) of the white noise process is set $0.031 m^2/s^3$. The failure criterion is $4.3 \times 10^{-2} m$. In this example, the uncertainties exist in the base structure properties and take the form of normal distributions with standard deviations $\sigma_{m_s} = 0.1$, $\sigma_{c_s} = 0.001$, $\sigma_{k_s} = 5.0$. The sample data of the uncertain properties are obtained according to their uncertainty characteristics.

Table 4.1: Nominal values of the structure

Parameters	$m(kg)$	$c(N \cdot s/m)$	$k(N/m)$
`Base structure`	1.0	0.03	696.4
`TMD`	0.02	0.0695	12.725

The estimates of the failure probabilities for these samples are applied by Monte-Carlo simulations. To evaluate the model performance in fitting the data, we apply 10-fold cross-validation on the training data. Meanwhile we range the size of the data to see its influence on cross-validation evaluation indices. The sizes of data range from 1000 cases to 19000 cases. Another 1000 cases are sampled as a test set to study the model performance in prediction ability. See results in Figure 4.8.
In Figure 4.8, the indices Co-coef, RMSE and RAE are correlation coefficient, root mean square error and relative absolute error respectively, *i.e.*

$$Co-coef = \frac{\sum_{i=1}^{N}(y_i - \overline{y})(\hat{y}_i - \tilde{y})}{\sqrt{\sum_{i=1}^{N}(y_i - \overline{y})^2 \sum_{i=1}^{N}(\hat{y}_i - \tilde{y})^2}} \tag{4.31}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2} \tag{4.32}$$

$$RAE = \frac{\sum_{i=1}^{N} |\hat{y}_i - y_i|}{\sum_{i=1}^{N} |\overline{y} - y_i|} \tag{4.33}$$

where y_i, $\hat{y}_i$ are respectively the target values and predicted values, and $\overline{y}$, $\tilde{y}$ are the corresponding mean values. N is the number of sample cases.

As can be seen, the correlation coefficient between the estimated values and observed values increase drastically when more training data are available. Meanwhile, the two error indices RMSE and RAE decrease clearly before they converge at about 10000 of the data size. The cross-validation results tell that the proposed RF model behaves very well in fitting the data. The test results show that the model gives accurate predictions, so it didn't over-fit the data. This is the evidence that the randomness in the RF model inducing process can effectively avoid over-fittings [105]. Besides, the size of the training data plays an important role in determining the accuracy of RF model.

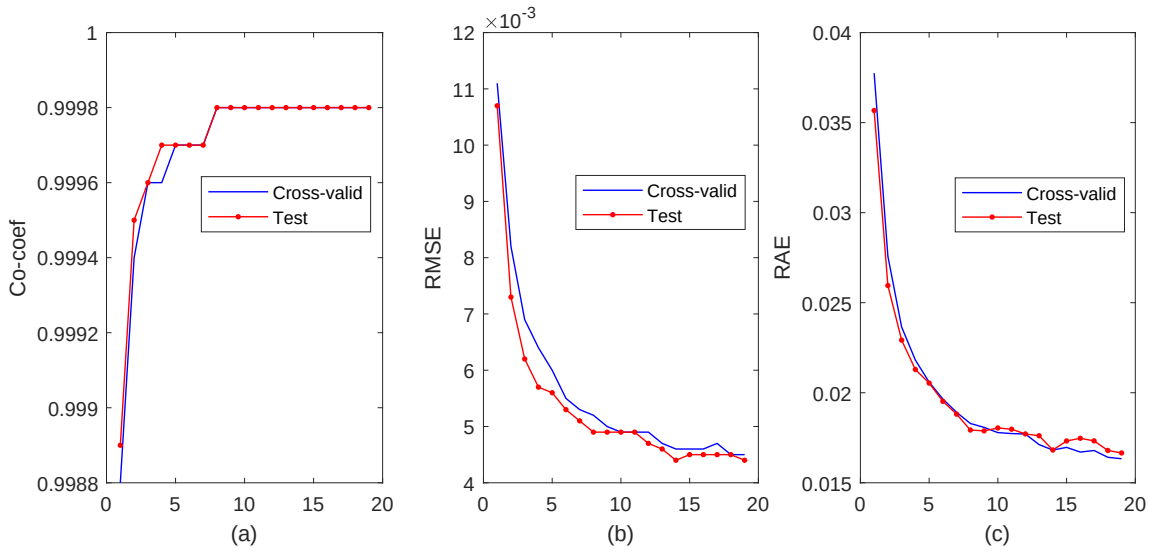

Figure 4.8: Model evaluation on different data(×1000) for 1-DOF structure

Example 2 The structure consists of 2-DOF base structure attached to a 1-DOF TMD. The natural frequency, damping ratio of the seismic model and PSD of the white noise process are the same as those in Example 1, but the nominal configura-

tions of the structure are different, see Table 4.2.

Table 4.2: Nominal values of the structure

Parameters	$m(kg)$	$c(N \cdot s/m)$	$k(N/m)$
`1st DOF`	4.6	62.0	6500
`2nd DOF`	4.6	62.0	6500
`TMD`	1.38	38.997	1.8327

The uncertainties of structure properties are normally distributed with standard deviations $\sigma_{m_{s1}} = \sigma_{m_{s2}} = 1$, $\sigma_{c_{s1}} = \sigma_{c_{s2}} = 10$, $\sigma_{k_{s1}} = \sigma_{k_{s2}} = 300$. The failure criterion is defined as, any degree of the main structure exceeds the assumed threshold value of displacement. Here the threshold is set $1.9 \times 10^{-2} m$. Considering the increase of the number of uncertain properties, the sizes of data sampled from the assumed distributions range from 1000 cases to 29000 cases as the training data. Another 1000 cases are sampled to test the model performance in prediction ability. See the simulation results in Figure 4.9.

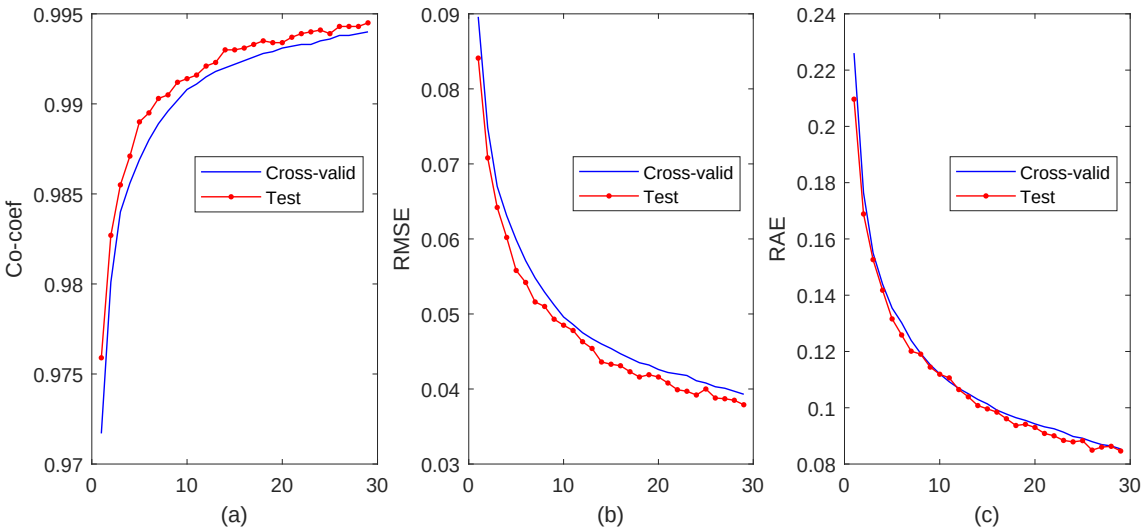

Figure 4.9: Model evaluation on different data(×1000) for 2-DOF structure

In Figure 4.9, the three evaluation indices change in a similar way to those in Example 1, even though the convergence values of the error indices are higher. This is mainly because when the number of variables increases, the space dimension of the variable becomes higher, thus the uncertainties become more difficult to handle. Undoubtedly, more observations will improve the model performance.

Example 3 The main structure is 3-DOF and the TMD structure is 2-DOF. Each degree of the structure, including the main structure and the TMD structure, has the same parameter configurations as those in the *example 2*. The threshold of the failure criterion is $2.5 \times 10^{-2} m$. The simulation results are shown in Figure 4.10.

Figure 4.10 shows that the correlation coefficients of both cross-validation and prediction are still staying in a high level. The co-efficient and the error indices change in a similar trends to those in *Example 2* except that the convergence values (especially RAE) are larger than those in the last example. This is an evidence that when the dimension of uncertain space is high, the data will be more difficult to model. Intuitively, we need more samples to train the model, but there are some other factors that may possibly affect the model performance. Therefore, Section 4.2 will explore the possible influence factors on the model.

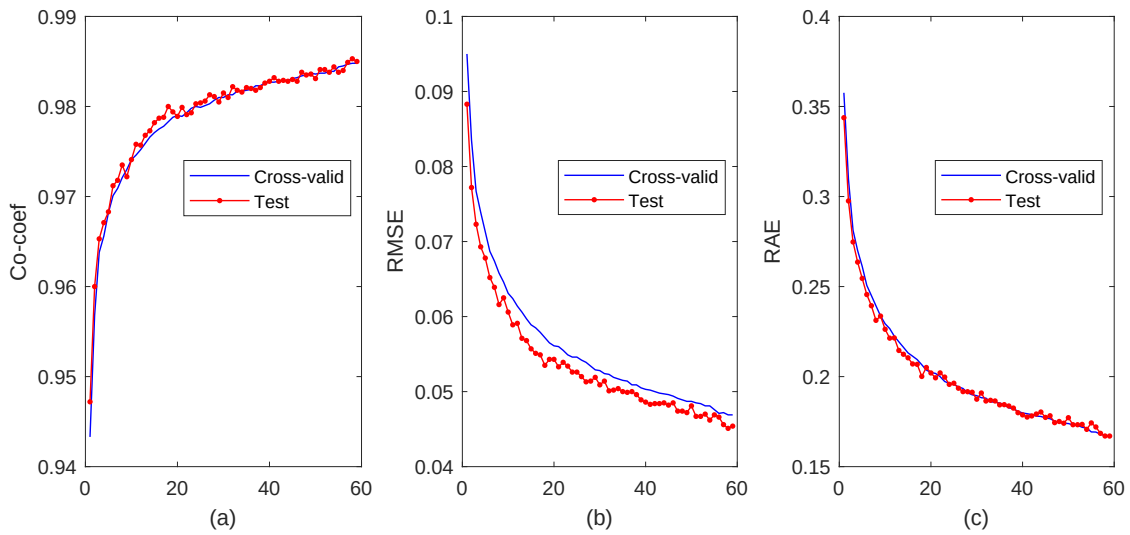

Figure 4.10: Model evaluation on different data(×1000) for 3-DOF structure

4.4.2 Parameter influence analysis

Besides the size of the sample that influences the model performance, the hyper-parameters in the proposed method may also make contributions to the evaluation results. In this part, the authors consider three factors: the variances of the random variables, the number of trees in the RF model and the number of candidate features for each splitting in tree inducing process.

4.4.2.1 Variance of the variables

The 1-DOF structure in *Example 1* is employed in this section. Considering the intuitivity of standard deviation and its direct relationship with variance, we use standard deviation instead of variance as the measure of dispersion of the samples. Knowing that there are three parameters of the main structure, i.e. m_s, c_s, k_s, we focus on one parameter and change its standard deviation in the samples. Meanwhile we fix the other two parameters, i.e. keep their standard deviations unchange. Based on the nominal values of the properties of the structure, we range the standard deviation of m_s from 1×10^{-2} to 1×10^{-1}, with step 1×10^{-2}. In a similar way, we range the standard deviation of c_s from 3×10^{-4} to 3×10^{-3}, with step 3×10^{-4}; for k_s, the standard deviation ranges from 5×10^{-1} to 5, with step 5×10^{-1}. See table 4.3. In Table 4.3, SD means 'standard deviation'; SD1 is the first value of SD of the variable. For example, SD1 for m_s is 0.01.

Table 4.3: Standard deviations of the uncertain properties

Items	SD1	SD2	SD3	SD4	SD5	SD6	SD7	SD8	SD9	SD10
m_s	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10
c_s	0.0003	0.0006	0.0009	0.0012	0.0015	0.0018	0.0021	0.0024	0.0027	0.003
k_s	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0

In each simulation for a variable that takes a specific SD in Table 3, we use a training set with 7000 samples to build the RF model, and another 1000 sample as the test

set. As before, we implement 10-fold cross-validation to evaluate the model. Two indices are employed to evaluate the model performance, i.e. RMSE and MRE. Here MRE means 'mean relative error' which is calculated as

$$MRE = \frac{1}{N}\sum_{i=1}^{N}\frac{|\hat{y}_i - y_i|}{y_i} \tag{4.34}$$

where y_i, $\hat{y}_i$ are respectively the target values and predicted values. N is the number of samples to test the model. In contrast with RMSE that describes the absolute differences between predicted values and target values, MRE is designed to measure the percentage error of the predictions with respect to the target values. Figure 4.11 and 4.12 show how the model performance is affected by the changing variances of the uncertain properties.

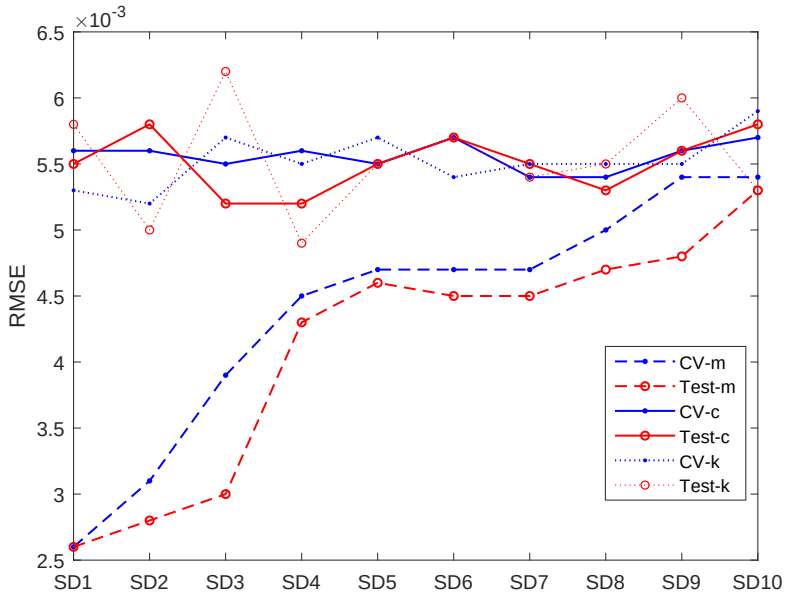

Figure 4.11: Simulation result based on different standard deviations

Figure 4.11 and Figure 4.12 show that, the variance of the uncertain property m_s makes clear influence on the RF model performance, while the variances of c_s and k_s do not. In Figure 4.11(12), 'CV-m', 'Test-m' respectively mean cross-validation results and test results when we study the variance of m_s; 'CV-c', 'Test-c', 'CV-k'

and 'Test-k' can be understood in a similar way. With the increase of the standard deviations of the variable m_s, the RMSE increases no matter for cross-validation or predictions. Meanwhile, the MRE shows the same trend. This is the evidence that, when the variance of m_s in the samples is high, the proposed model will be less accurate. By comparisons, we know that the variances of c_s or k_s can't clearly affect the model performance. Therefore, it is concluded that the RF model performance is more sensitive to the sample variance of the mass of the structure.

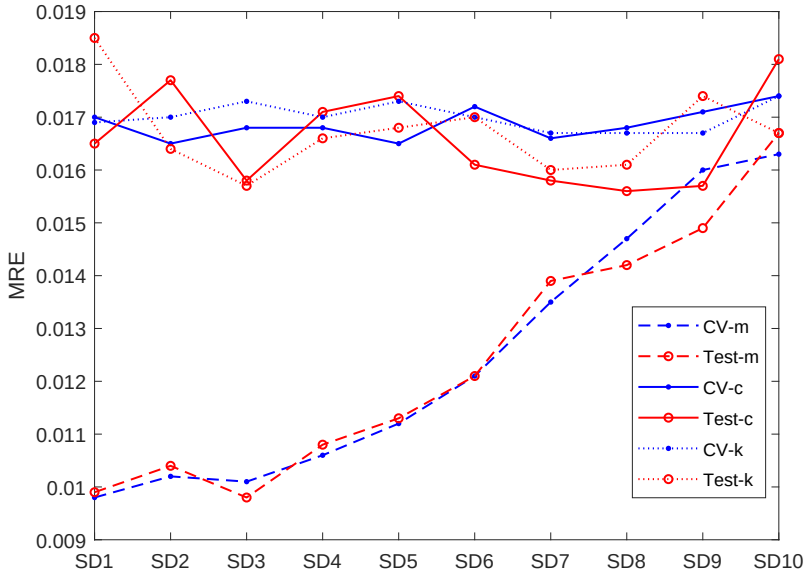

Figure 4.12: Simulation result based on different standard deviations

4.4.2.2 Hyper-parameter configuration of the RF model

The simulations are based on the 3-DOF structure in *Example*3 which has the standard deviations $\sigma_{m_s} = 0.3$, $\sigma_{c_s} = 3$, $\sigma_{k_s} = 90$. To study the influence of number of trees within RF on the model performance, 11000 cases of this dataset are randomly selected to train the model and the other 1000 cases are left to test the model in prediction. The number of trees ranges from 20 to 200, with step 20. The correlation coefficients and error indices are shown in Figure 4.13. From Figure 13, we know that with the increase of trees, the RF model tends to be more accurate.

Moreover, the RMSE begins to converge when the number of trees exceeds 120. As the computational cost increases linearly with k, $k = 120$ is a good choice to make a trade-off between computational complexity and accuracy [51]. This is a direction to determine the value of k. Therefore, based on the available data, the best k should be the minimum number achieving the accuracy level with a reasonable computational cost. It is also found that increasing the number of trees effectively reduces RAE, meanwhile increasing the correlation coefficient.

In the same simulation, we consider the influence of the number of candidate features and change it in each splitting procedure. Considering that there are 9 uncertain attributes to process, we range the number of candidate features from 2 to 5, and see the influences on different evaluation indices. See results in Figure 4.13. The simulation results show that the model accuracy reaches its highest when the number of features is taken as 4. This result is close to the conclusion in Leo Breiman [105]. This simulation shows the evidence that the number of candidate features is another key point to consider in RF model selections.

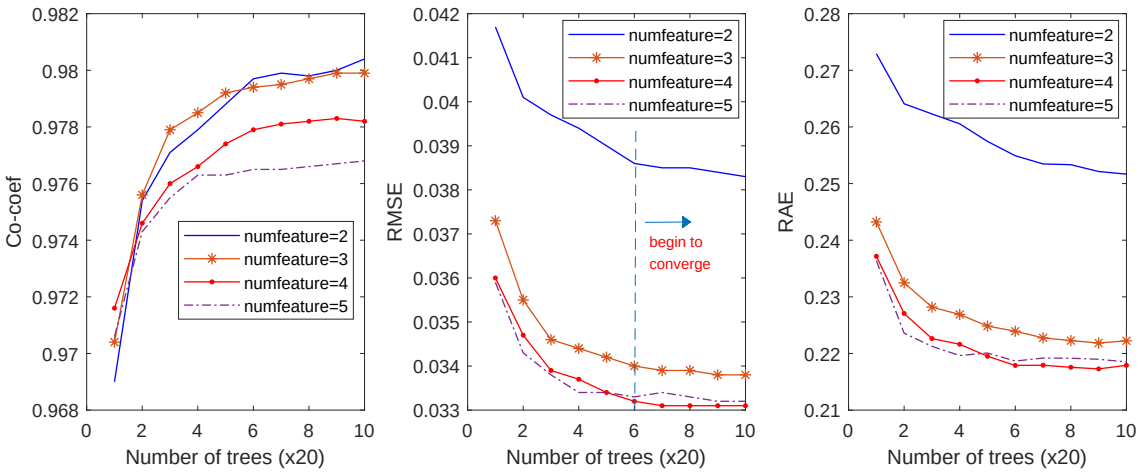

Figure 4.13: Influence of the number of features and number of trees

4.4.3 Feature importance analysis

Feature importance is a way to quantize the contributions of the features(variables) to the overall model's predictive performance. For high-dimension problems, it helps

filter the variables that have little influence on the response so that the computational efficiency is improved. The feature importance is reflected by the change in OOB error for each feature when that feature is randomly permuted in the OOB observations. The higher the difference, the more important a feature is, vice versa. This is called out-of-bag feature importance. We firstly introduce the concept of OOB error. To build a RF, bootstrap sampling is carried out to generate a certain number of bootstrap datasets, each of which has a corresponding OOB dataset. For each observation $X_i(i = 1, 2, ..., N)$ in the training set, we estimate its out-of-bag prediction by averaging over the predictions of the trees (in the ensemble) that take X_i as out-of-bag data. That is,

$$F(X_i) = \frac{1}{N_i} \sum_{X_i \notin S_i} f(X_i; T_{i,j}), \tag{4.35}$$

where $S_i = \{S_{i,j}\}_{j=1}^{N_i}$ denotes all the Bootstrap datasets that take X_i as out-of-bag data. $S_{i,j}$ is the jth$(j = 1, 2, ..., N_i)$ dataset within S_i. $T_{i,j}$ is the tree grown from $S_{i,j}$. $f(X_i; T_{i,j})$ is the predicted target value for X_i by the tree $T_{i,j}$. The out-of-bag error for X_i can be determined by comparing the predicted target value with the true target value in a square error sense, *i.e.*

$$OOBE(X_i) = (F(X_i) - y_i)^2 \tag{4.36}$$

Then the OOBE for the RF model is calculated by taking the root average of OOBEs of all the observations in the training set. In order to calculate feature importances, the procedures are carried out on each tree in the RF. We denote $\{T_i\}_{i=1}^{k}$ as the RF that consists of k random trees and S_{D_N,θ_i} as the OOB dataset corresponding to T_i. Firstly, we calculate the root average value of OOBEs of the observations within S_{D_N,θ_i}, that is $\varepsilon_{T_i} = mean[OOBE(X)]$ given that $X \in S_{D_N,\theta_i}$. Then the values of the jth variable $\mathbf{x}_j(j = 1, ..., n)$ in the OOB data are randomly permuted, meanwhile keeping other variable values unchanged. In a similar way, we use the altered OOB observations to calculate the new OOB error ε_{j,T_i}, see the procedures in Figure 4.14. Generally, ε_{j,T_i} should be larger than ε_{T_i}. We calculate the difference between the two OOB errors, i.e. $d_{j,T_i} = \varepsilon_{j,T_i} - \varepsilon_{T_i}$, which can be seen as the importance index

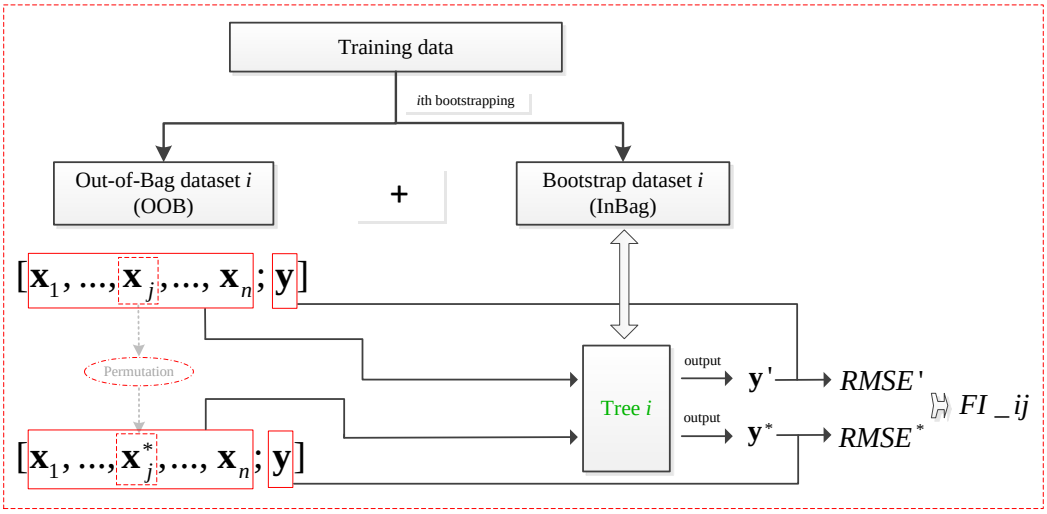

Figure 4.14: FI evaluation of jth feature via Tree i in RF

of $\mathbf{x}_j$ via the tree T_i. We repeat the procedures in Figure 4.14 by each tree in the forest, finally we obtain a matrix that consists of the feature importance indices induced by all trees in the RF, denoted below,

$$FI = \begin{pmatrix} d_{1,T_1} & d_{2,T_1} & \dots & d_{n,T_1} \\ d_{1,T_2} & d_{2,T_2} & \dots & d_{n,T_2} \\ \dots & \dots & \dots & \dots \\ d_{1,T_k} & d_{2,T_k} & \dots & d_{n,T_k} \end{pmatrix} \tag{4.37}$$

where k is the number of trees in the RF, and n is the number of features in the training set. Accordingly, the importance measure of $\mathbf{x}_i$ is obtained as the average FI value over all trees, *i.e.*,

$$FI\_i = ave\{d_{i,T_1}, d_{i,T_2}, ..., d_{i,T_k}\} \tag{4.38}$$

the standard deviation (SD) of the estimation of FI of variable $\mathbf{x}_i$ $(i = 1, ..., n)$ is computed as

$$SD(FI\_i) = \sqrt{var\{d_{i,T_1}, d_{i,T_2}, ..., d_{i,T_k}\}} \tag{4.39}$$

To make comparison more intuitive, all variable importances are scaled in order that they sum up to 1. We consider the same dataset analyzed in section 4.2.2 with the same standard deviations, and implement the feature importance analysis. The RF contains 150 trees. The number of candidate features to split a node is 3. See results in table 4.4 and Figure 4.15.

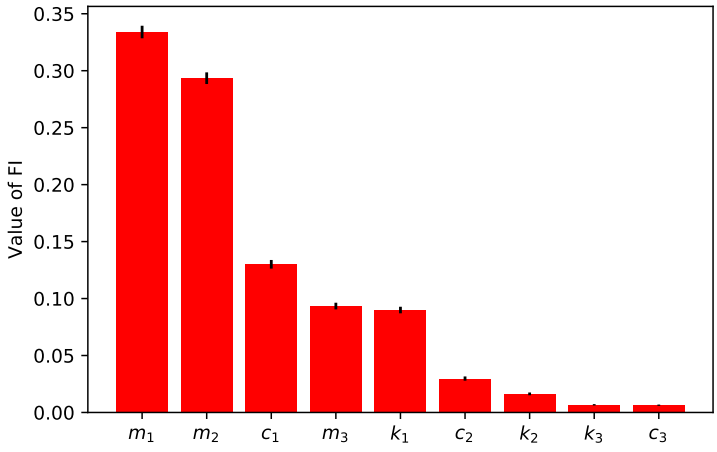

Figure 4.15: FIs of the structural properties (with SDs)

The feature importances in Figure 4.15 illustrate that the mass of each degree of the main structure are the principle elements to determine the model outputs. In contrast, the stiffness and damping coefficients are much less influential especially for the second and third degrees.

Table 4.4: List of feature importances

Feature	m_{s_1}	m_{s_2}	c_{s_1}	m_{s_3}	k_{s_1}	c_{s_2}	k_{s_2}	k_{s_3}	c_{s_3}
FI value	0.3340	0.2934	0.1301	0.0934	0.0900	0.0298	0.0163	0.0067	0.0064
SD of FI	0.0055	0.0038	0.0029	0.0051	0.0019	0.0012	0.0029	0.0006	0.0006

4.4.4 Comparisons with other machine learning methods

To compare the RF model with other models, three representative machine learning methods are selected, i.e. Bayesian inference method, k-NN method and GB method. We take the three different structures in section 4.1 as the object structures, and the same datasets are considered as the training data. The main structures have the number of DOFs ranging from 1 to 3. For the first system with single DOF main structure, 8000 samples are chosen to train the model, while for the second and third structure, 15000 and 60000 samples are respectively chosen as the training data. Within each training data, 1000 samples are randomly chosen as the test set to test the model. To reduce the influence of the randomness of the data on the models, we repeat the test procedures for 50 times and average the evaluation results. RMSE is chosen as the index to evaluate the model performance. See the simulation results in Table 4.5.

Table 4.5: RMSEs of different methods

Structural DOF	1-DOF	2-DOF	3-DOF
Bayes method	0.0208	0.0709	0.0952
k-NN method	0.0180	0.0517	0.0486
GB method	**0.0051**	0.0631	0.0462
RF method	0.0056	**0.0433**	**0.0451**

The evaluation indices in Table 4.5 tell that the RF model behaves very well in reliability prediction compared with the other three models. For the single DOF structure, even though GB model behaves the best, the RF model achieved almost the same accuracy (only 5×10^{-4} in difference). For the 2-DOF and 3-DOF structures, the RF model gives the most accurate predictions compared with the other three. Bayes method give the lowest accuracy in reliability prediction of all the three structures. Generally, Bayesian inference method requires prior knowledge or assumptions on the data distributions while k-NN does not. This makes k-NN tend

to be more accurate than Bayes method when no prior information is available.

The GB method provides close, though less accurate, predictions to those by RF. As an ensemble learning method, GB is different from RF in the aspect that it combines multiple 'weak learners' as a weighted sum to make predictions. It builds one tree at a time, and each new tree is built to reduce the errors of the previously built tree. The way each tree is built makes this model prone to overfit the data. This is one of the reasons that GB is less accurate than RF in some situations.

4.4.5 An illustrative example of the prediction results

This section provides the training and prediction results of the RF model based on the data employed in section 4.2.2. Some parameter configurations are listed in Table 4.6. Besides, the PSD takes 0.031, the displacement threshold takes $2.5 \times 10^{-2} m$, and the first-passage time span is $20s$. In order to present the simulation results more visibly, 100 samples are randomly selected from the training result and another 50 samples are obtained in the same way from the test result. The RF contains 160 trees, and the number of candidate features is set as 4.

Table 4.6: Structure parameters ($i = 1, 2, 3; j = 1, 2$)

Index	m_{s_i}	c_{s_i}	k_{s_i}	m_{T_j}	c_{T_j}	k_{T_j}
nominal	4.6	62	6500	1.38	1.8327	38.997
SD	0.5	5	150	-	-	-

The results in Figure 4.16 show that the RF model prediction values are in good accordance with the actual values. The RMSE in the simulation results is small enough, but the RAE values is a little higher than those for 1-DOF structures. The increase of the predicting error is due to the weak concept of local neighborhood as the input space dimensionality grows [106], but more samples and suitable configurations of the RF model can help reduce the error.

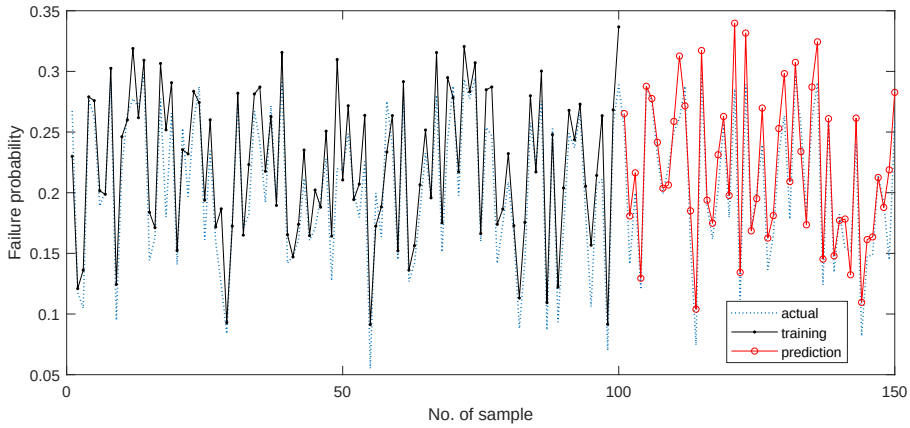

Figure 4.16: Simulation results of the object structure via RF

4.4.6 Two case studies on multi-DOF structural system

In the subsections 4.4.1, the reliability analysis of three different DOF structures (*i.e.*, 1-DOF, 2-DOF and 3-DOF) have been studied with the aide of Random Forest. It is noticed that the three structures do not have high-dimension uncertain structural parameters. Random Forest is known to be powerful in handling high-dimension uncertainties, thus in this subsection, the multi-DOF structural system is studied. Two simulations are carried out in this subsection. In the first simulation, the object structure is multi-DOF one introduced from Elyes et.al [81]. The RF performance is analyzed and compared with standard MCS results. In the second simulation, the RF model is tested on a famous benchmark problem introduced by G.I. Schueller [107]. It is also compared with standard MCS, Subset simulations (3 variants) [108], complex modal analysis (CMA) [109], spherical subset simulation (S^3) [110], auxilary domain method (AMD) [111] and line sampling [112].

4.4.6.1 TMD-based ten-DOF shear building

The system is a 10-storey shear building, excited by earthquakes, see Figure 4.17. Different from Figure 3.14 where the studied structure has deterministic properties, the structure in Figure 4.17 has uncertainties in its mass, stiffness and damping indices (except for the TMD structure). In subsection 3.4.6, this structure has been

employed to compare the efficiency of KL-IS method with standard MCS method. However, this part devotes itself to conditional failure probability modeling by Random Forest as well as the overall failure probability considering uncertainties both in structural properties and in ground excitations. The structural parameters have nominal values that are the same as those in Figure 3.8 [81]. Besides, the parameters of the TMD (also the KT model) are deterministic and the same to those in Figure 3.14. The nominal values and their uncertainties are listed in Table 4.7. Notice that, in Table 4.7, the SD values are taken as 10% of the corresponding nominal values. The structural failure takes place when a certain DOF of the structure at a certain time point has the response exceeding a prescribed threshold value. Here, for simplicity, the same threshold value is assumed for all DOFs within the observed time interval.

Table 4.7: Statistical properties of the structural parameters

Variables	Mean	SD
$m_{s_1}, ..., m_{s_{10}}$	$360 \times 10^3 kg$	$36 \times 10^3 kg$
$c_{s_1}, ..., c_{s_{10}}$	$620 \times 10^4 Ns/m$	$62 \times 10^4 Ns/m$
$k_{s_1}, ..., k_{s_{10}}$	$650 \times 10^6 N/m$	$65 \times 10^6 N/m$

In this simulation, the threshold is set as $y_{lim} = 0.28m$; the observed time interval is $[0, 20s]$ with the time step $T_s = 0.05s$ after discretization. To prepare the training data for learning the Random Forest model, 1000 samples of the structural properties are randomly obtained according to their PDF (assuming normal distributions with the statistical moments displayed in Table 4.7), then the conditional failure probability is estimated for each sample. In the framework presented in Figure 4.2, a Random Forest model is induced from the training set containing 1000 cases each of which has 30 input variables (*i.e.* structural parameters) and an output value (*i.e.* the value of conditional failure probability). Besides, a test set of 1000 cases is obtained to test the performance of the learned Random Forest model.

The parameters in the Random Forest model are shown in Table 4.8. 'nTrees' is the

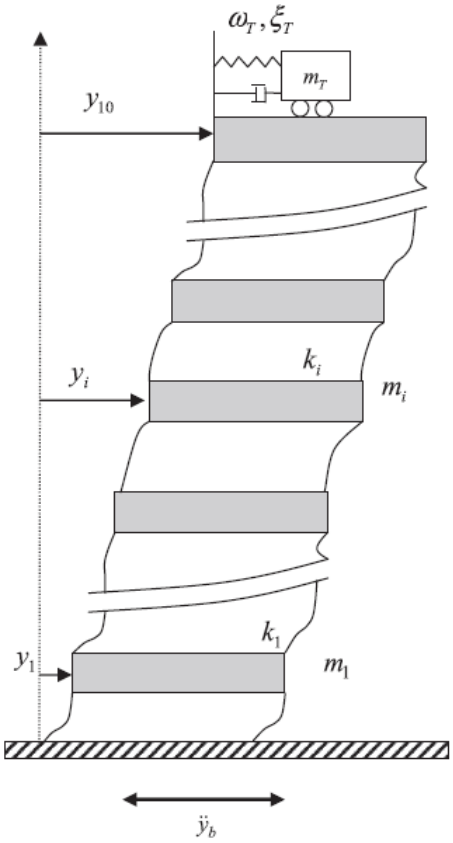

Figure 4.17: Ten-DOF uncertain structure subjected to stochatic excitations [81]

number of random trees in Random Forest; 'nFeatures' is the number of features (variables) in the training set; 'maxFeatures' is the number of sampled features at each node. The RMSE is evaluated on the test set, see Table 4.9. Then the learned RF is employed to make predictions on a new set of 2000 cases. According to eq.(3.6), the structural failure probability is estimated considering uncertainties both in structural properties and excitations, See the reults in Table 4.9.

Table 4.8: Parameters of RF

nTrees	nFeatures	maxFeatures
50	30	5

From Table 4.9, it is found that the estimated failure probability $\hat{P}_{f,RF}$ by RF is close

Table 4.9: Simulation results by RF

Train set	Test set	RMSE	$\hat{P}_{f,RF}$	$\hat{P}_{f,MCS}$
1000	1000	0.0265	0.0789	0.0701

to the reference value $\hat{P}_{f,MCS}$ obtained from standard MCS. As the value of y_{lim} has direct influence on the estimated failure probability, different y_{lim} have been studied in this simulation. The y_{lim} ranges from $0.28m$ to $0.37m$ with a discretization step $0.01m$. For each y_{lim}, the KL-IS method is implemented to prepare the training set. 200 cases are generated in each training set. Then a RF model consisting of 30 regression trees is trained. Each tree in the RF has the parameter maxFeature= 5. The estimated failure probabilities (as well as their standard deviations) resulting from different y_{lim} values are listed in Table 4.10. The reference values calculated by standard MCS are also provided to make comparisons. In principle, the failure probability of the object structure decreases when the threshold y_{lim} increases. In Table 4.10, both the simulation results of MCS and those of RF have shown this phenomenon. By comparisons of the two simulation results, it is found that the RF model achieves close estimation results to those achieved by standard MCS.

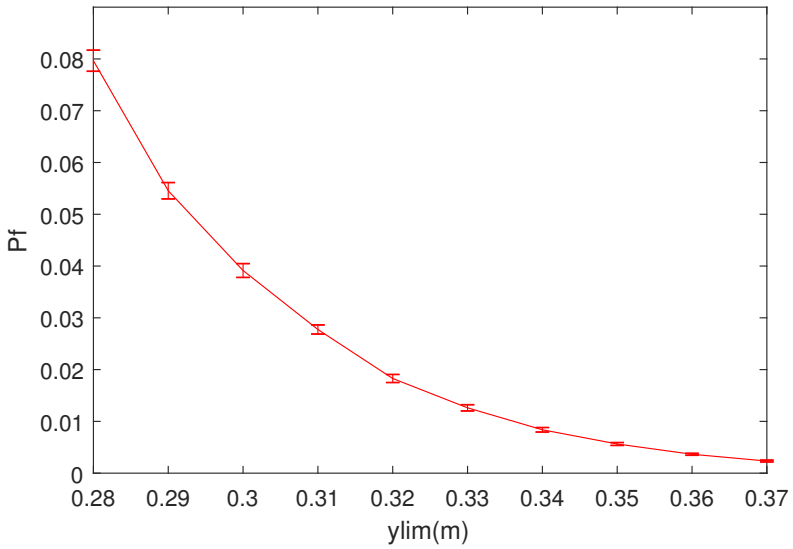

Figure 4.18: P_f estimations (with error bars) by RF

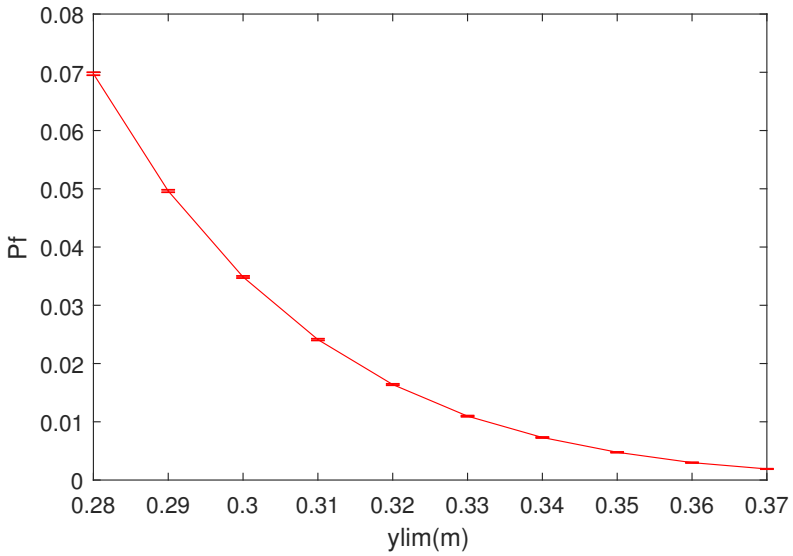

Figure 4.19: P_f estimations (with error bars) by standard MCS

4.4.6.2 Duffing type ten-DOF oscillator with bounded uncertainties

As a benchmark problem introduced in G.I. Schueller [107], the ten-DOF Duffing type oscillator has been widely used by researchers in structural reliability domain. In this study, we focus on the linear random structures under stochastic excitations. The governing equation is given by

$$\mathbf{M}\ddot{\mathbf{y}}(t) + \mathbf{C}\dot{\mathbf{y}}(t) + \mathbf{K}\mathbf{y}(t) = \mathbf{F}(t) \tag{4.40}$$

where $\mathbf{M}$, $\mathbf{C}$, $\mathbf{K}$ are the mass matrix, damping matrix and stiffness matrix respectively. $\mathbf{y}$ is the structural response; $\mathbf{F}(t) = u(t)[m_1, m_2, ..., m_{10}]^T$ is the force on the structure. For more details about the dynamic process, refer to section 4.2. The statistical properties of the structural parameters and their constraints are listed in Table 4.11. Notice that the damping ratio $c_i = 2\zeta_i\sqrt{m_i k_i}$ in this case is the proportional viscous damping. Moreover, the Gaussian random variables for structural properties are censored to avoid the deviation from the mean being larger than five times the standard deviation. In Table 4.11, $r = SD/\mu$. The stochastic excitation $p(t)$ is modeled by a modulated filtered Gaussian white noise:

Table 4.10: Comparisons between standard MCS and RF results

$y_{lim}(m)$	$\hat{P}_{f,RF}$	$SD_{RF}(*10^{-3})$	$\hat{P}_{f,MCS}$	$SD_{MCS}(*10^{-4})$
0.28	0.0789	2.036	0.0701	2.547
0.29	0.0545	1.577	0.0496	2.172
0.30	0.0391	1.334	0.0349	1.834
0.31	0.0278	0.882	0.0241	1.534
0.32	0.0183	0.778	0.0164	1.270
0.33	0.0126	0.602	0.0110	1.042
0.34	0.0084	0.436	0.0073	0.852
0.35	0.0056	0.279	0.0048	0.689
0.36	0.0036	0.188	0.0030	0.546
0.37	0.0023	0.181	0.0019	0.436

$$p(t) = \Omega_{1g}^2 v_{f1}(t) + 2\zeta_{1g}\Omega_{1g} v_{f2}(t) - \Omega_{2g}^2 v_{f3}(t) - 2\zeta_{2g}\Omega_{2g} v_{f4}(t) \tag{4.41}$$

where the state space function with respect to the state vector $v_f(t) = [v_{f1}(t), v_{f2}(t), v_{f3}(t), v_{f4}(t)]^T$ of the filter is,

$$\dot{v}_f(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\Omega_{1g}^2 & -2\zeta_{1g}\Omega_{1g} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ \Omega_{1g}^2 & 2\zeta_{1g}\Omega_{1g} & -\Omega_{2g}^2 & -2\zeta_{2g}\Omega_{2g} \end{bmatrix} v_f(t) + [0, w(t), 0, 0]^T \tag{4.42}$$

Here, $w(t)$ stands for a modulated Gaussian white noise with auto-correlation function $E(w(t)w(t+\tau)) = I\delta(\tau)h^2(t)$ and I denotes the intensity function of the white noise. $h(t)$ has the following form,

$$h(t) = \begin{cases} 0, & t = 0, \\ t/2, & t \in [0, 2s] \\ 1, & t \in [2s, 10s] \\ exp(-0.1(t-10)), & t \in [10s, 20s] \end{cases} \tag{4.43}$$

$\delta(t)$ is the Dirac delta function that equals to $+\infty$ at $t = 0$ and 0 at $t \neq 0$. The values $\Omega_{1g} = 15.0rad/s$, $\zeta_{1g} = 0.8$, $\Omega_{2g} = 0.3rad/s$, $\zeta_{2g} = 0.995$ and $I = 0.08m^2/s^3$ are used to model the filter. The input excitation of the filter is a shot noise that consists of a series of independent normally distributed impulses arranged at each time step [113]. The magnitude of the impulse at time $t_k = k\Delta t$ has mean 0 and standard deviation $h(t)\sqrt{I/\Delta t}$. It is noticed that, in G.I. Schueller [107], the expression of the standard deviation '$h(t)\sqrt{I\Delta t}$' is wrong, however, some other researchers still use this wrong expression in their published papers.

Table 4.11: Statistical properties of the structural parameters

Variables	Mean(μ)	SD	Ratio(r)	Range scope
$m_1, ..., m_{10}$	$10 \times 10^3 kg$	$1.0 \times 10^3 kg$	0.1	$\mu \pm 5\mu r$
k_1, k_2, k_3	$40 \times 10^6 N/m$	$4.0 \times 10^6 N/m$	0.1	$\mu \pm 5\mu r$
k_4, k_5, k_6	$36 \times 10^6 N/m$	$3.6 \times 10^6 N/m$	0.1	$\mu \pm 5\mu r$
k_7, k_8, k_9, k_{10}	$32 \times 10^6 N/m$	$3.2 \times 10^6 N/m$	0.1	$\mu \pm 5\mu r$
$\zeta_1, ..., \zeta_{10}$	$620 \times 10^4 Ns/m$	$62 \times 10^4 Ns/m$	0.1	$\mu \pm 5\mu r$

The failure criterion is defined by the maximum relative displacements between two consecutive DOFs over the time interval $[0.0s, 20.0s]$. The sampling time step is set as $\Delta t = 0.05s$. The failure probabilities are calculated for different threshold values listed in Table 4.12. We are interested in the first excursion probability that the relative maximum displacement of the first DOF is greater than the threshold $0.057m$ and $0.073m$; besides, the probability that the relative maximum displacement between the 9th DOF and 10th DOF exceeds the threshold $0.013m$ and $0.017m$ is

considered.

Table 4.12: Thresholds of interest to evaluate failure probability [107]

Failure defined by	Res. threshold1	Res. threshold2
First DOF	$0.057m$	$0.073m$
Tenth DOF	$0.013m$	$0.017m$

In this simulation, the estimated failure probability values are very small (below 10^{-4}), therefore KL-IS method is employed to calculate the conditional failure probabilities. In applying K-L expansions, 3000 samples of the filtered white noise processes are generated from the filter, then the eigenvalue problem is solved (refer to eq.(3.11)). The number of K-L terms kept is $n_{KL} = 300$. The relative response of each DOF can be directly calculated when the structure experiences impulse response. Figure 4.20 shows the relative responses assuming that the structure is having its nominal configurations.

In the framework of MCS, 300 samples of the structural properties are analyzed, resulting in 300 conditional failure probabilities. A RF model with 30 rgression trees is then trained from the training set that consists of 300 cases and $3 \times 10 = 30$ variables in each case. For each threshold value in Table 4.12, similar Monte-carlo procedures are carried out. Once a RF model is trained, it is employed to make predictions on a newly generated data set that contains 2000 random samples of the structural properties. By eq.(3.5), the overall failure probability is estimated that is concerned with both structural uncertainties and excitation uncertainties. As already mentioned before, the proposed method is compared with other methods (the results are provided by those authors), see the results in Table 4.13.

A general conclusion drawn from Table 4.13 is that, a number of methods and their variants exist by which high-dimensional reliability problems with respect to linear or nonlinear stochastic dynamics can be solved very efficiently when compared to standard Monte Carlo. It is noticed that the estimated probabilities are very small (e.g. $P_f = 10^{-6}$), which implies that standard Monte Carlo is practically not

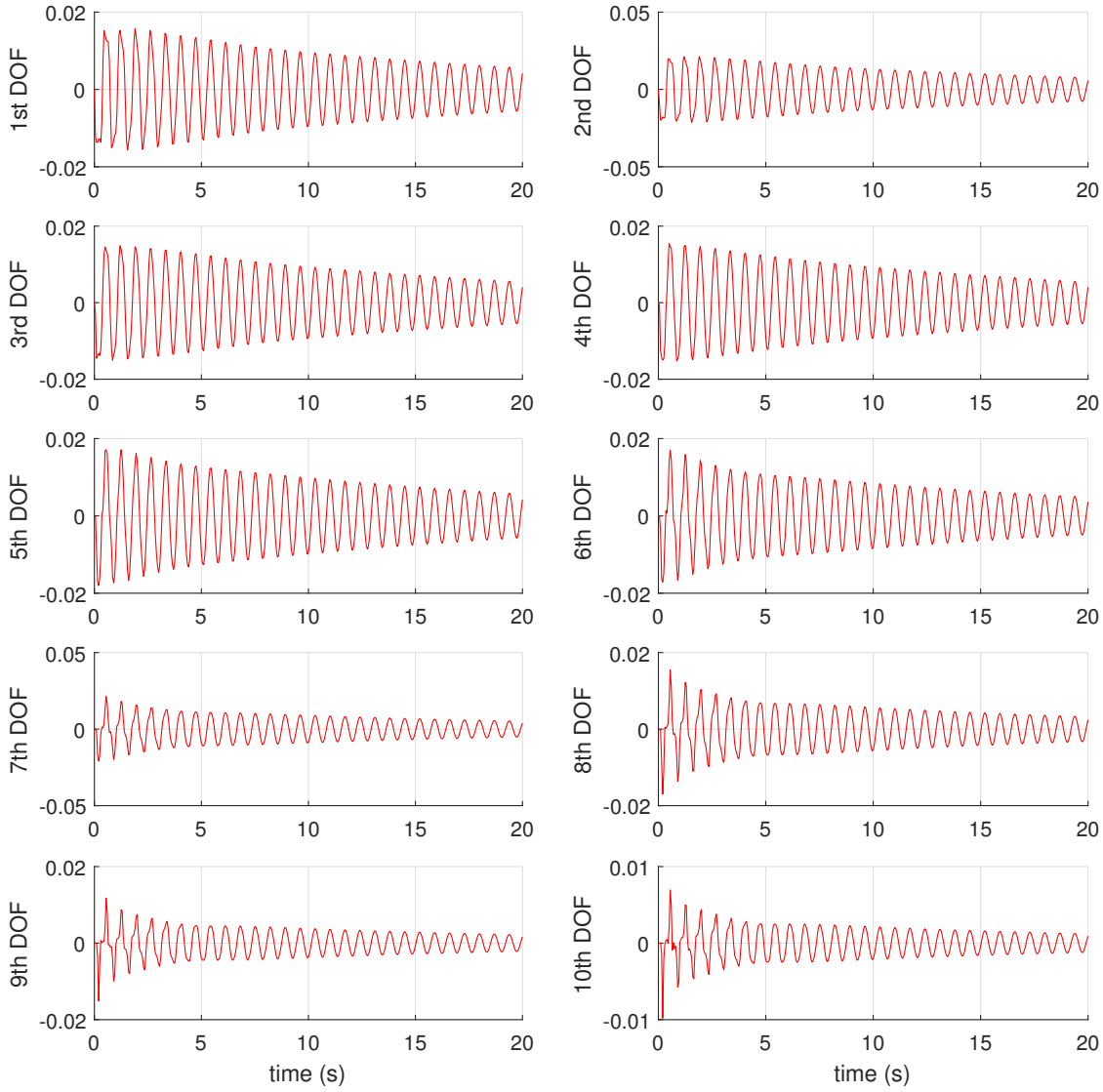

Figure 4.20: Relative response of different DOFs of the structure

applicable, because it would require a very large number of simulations. By the comparisons between RF and the other methods in Table 4.13, it is found that the number of samples analyzed in the proposed method is very small (e.g. 300), but the finally estimated failure probabilities are highly accurate. This result tells that, RF model is powerful in handling high-dimension reliability problems.

Table 4.13: Reliability estimation results from different thresholds

Method	1stDOF $0.057m$	1stDOF $0.073m$	10thDOF $0.013m$	10thDOF $0.017m$
Standard MCS	$1.06e^{-4}$	$8.07e^{-7}$	$4.88e^{-5}$	$2.52e^{-7}$
num of samples	$2.98e^{7}$	$2.98e^{7}$	$2.98e^{7}$	$2.98e^{7}$
SubsetSim/MCMC	$1.20e^{-4}$	$1.00e^{-6}$	$6.60e^{-5}$	$4.70e^{-7}$
num of samples	1850	2750	2300	2750
SubsetSim/Hybrid	$1.10e^{-4}$	$1.10e^{-6}$	$5.90e^{-5}$	$3.20e^{-7}$
num of samples	2128	3163	2645	3680
Complex Modal Ana.	$1.00e^{-4}$	$9.80e^{-7}$	$6.00e^{-5}$	$4.60e^{-7}$
num of samples	300	300	300	300
Spherical SubsetSim	$9.20e^{-5}$	$8.80e^{-7}$	$4.60e^{-5}$	$5.30e^{-7}$
num of samples	3070	4200	3250	4900
Line sampling	$9.80e^{-5}$	$9.70e^{-7}$	$6.00e^{-5}$	$4.60e^{-7}$
num of samples	360	3600	360	360
Random Forest	$7.6e^{-5}$	$1.0e^{-6}$	$4.2e^{-5}$	$1.1e^{-7}$
num of samples	800	1000	800	900

4.5 Chapter summary

In this chapter, the Random Forest algorithm is investigated in reliability modeling and prediction of passive controlled structures. A comparative study of the predictive performance of the model on different degree-of-freedom structures are presented. Results from three numerical examples showed that the use of Random Forest model in failure prediction of passive controlled structures is a meaningful exploration. The analysis of influence factors on the model performance provides some possibilities to improve the prediction results when the samples are limited. Moreover, the feature importance analysis tell that the mass is the most important

features in reliability modeling. The comparisons between RF and other machine learning methods shows the evidence that the RF model is an alternative way to study structural reliability.

Chapter 5

Structural Reliability estimation: the Stacking approach

Contents

5.1 Introduction

The ensemble learning methods mainly include Bagging, Boosting and Stacking. In principle, Bagging adopts Bootstrap sampling to learn independent base learners and

takes the majority/average as the final prediction. Boosting updates weight distribution in each round, and learns base models accordingly, then combines them according to their corresponding accuracy. Different from these two approaches, Stacking [41] learns a high-level model on top of the base models (classifier/regressor). It can be regarded as a meta-learning approach in which the base models are called first-level models and a second-level model is learnt from the outputs of the first-level models. A short description of all three methods is introduced below.

- Bagging method generally builds several instances of a black-box estimator from bootstrap replicates of the original training set and then aggregates their individual predictions to form a final prediction. This method is employed as a way to reduce the variance of a base estimator (e.g., a decision tree) by introducing randomization into its construction process.

- Boosting is a widely used ensemble approach, which can effectively boost a set of weak classifiers to a strong classifier by iteratively adjusting the weight distribution of samples in the training set and learning base classifiers from them. At each round, the weight of misclassified samples are increased and the base classifiers will focus on these more. This is equivalent to inferring classifiers from training data that are sampled from the original data set based on the weight distribution.

- Stacking (Stacked Generalization) involves training a learning algorithm to combine the predictions of several other learning algorithms. See Figure 5.1 for an illustration. First, all of the other algorithms are trained using the available data, then a combiner algorithm is trained to make a final prediction using all the predictions of the other algorithms as additional inputs. Stacking typically yields performance better than any single trained models [41]. It has been successfully applied on both regression [42] and classification. It was reported to outperform Bayesian model-averaging [114].

To understand why the generalization ability of an ensemble is usually much stronger than that of a single learner, Dietterich [115] gave three reasons by viewing the na-

ture of machine learning as searching a hypothesis space for the most accurate hypothesis. The first reason is that, the training data might not provide sufficient information for choosing a single best learner. For example, there may be many learners perform equally well on the training data set. Thus, combining these learners may be a better choice. The second reason is that, the search processes of the learning algorithms might be imperfect. For example, even if there exists a unique best hypothesis, it might be difficult to achieve since running the algorithms result in sub-optimal hypotheses. Thus, ensembles can compensate for such imperfect search processes. The third reason is that, the hypothesis space being searched might not contain the true target function, while ensembles can give some good approximation.

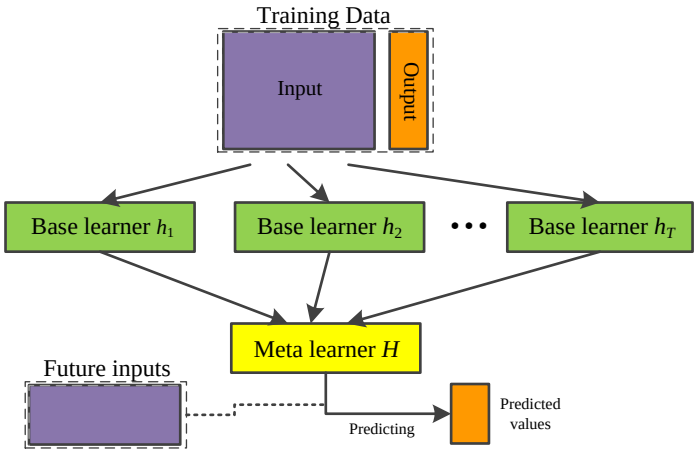

Figure 5.1: General framework of Stacking

There are many theoretical studies on ensemble methods, yet it is far from a clear understanding of their underlying mechanisms [116]. For example, empirical observations show that Boosting often does not suffer from overfitting even after a large number of rounds, and sometimes it is even able to reduce the generalization error after the training error has already reached zero. The bias-variance decomposition is often used in studying the performance of ensemble methods. It is known that Bagging can significantly reduce the variance, so it is better to be applied to learners (such as decision trees) suffered from large variance. Boosting can significantly reduce the bias in addition to reducing the variance.

5.2 Principle of Stacking

Stacking is an ensemble learning technique that learns a meta-learner based on the output of multiple base learners. In a typical implementation of Stacking, a number of first-level individual learners are generated from the training data set by employing different learning algorithms. Those individual learners are then combined by a second-level learner which is called meta-learner. See Figure 5.2.

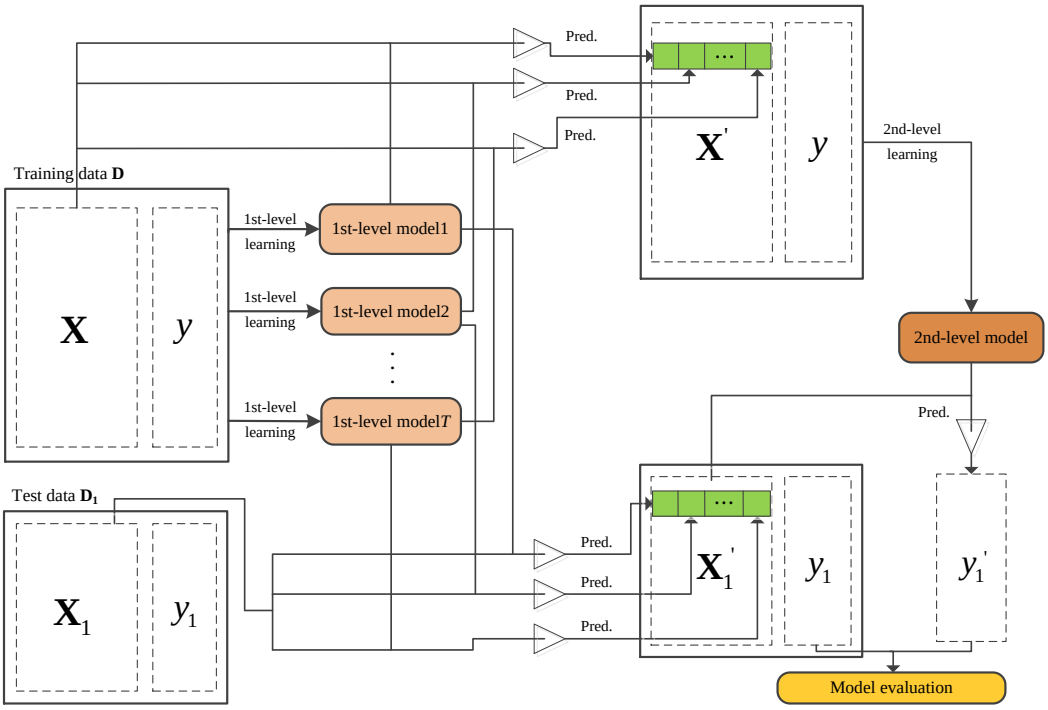

Figure 5.2: Diagram of Stacking method (include model evaluation on test data)

5.3 General Stacking procedures

Table 5.1 shows the general procedures of Stacking, which demands three steps. Firstly, learn first-level (base) learners based on the original training data. There are several ways to learn base learners. We can apply Bootstrap sampling technique to learn independent learners or adopt the strategy used in Boosting. Secondly,

construct a new data set based on the output of base learner. Assume that each example in D is $(\mathbf{x}_i, y_i)$. We construct a corresponding example $(\mathbf{x}'_i, y_i)$ in the new data set, where $\mathbf{x}'_i = h'(h_1(\mathbf{x}_i), h_2(\mathbf{x}_i), ..., h_T(\mathbf{x}_i))$. Thirdly, learn a second-level (meta) learner from the newly constructed data. Any learning method could he applied to learn the meta learner.

Table 5.1: Pseudo-code for Stacking [117]

Input: Training data $D = \{\mathbf{x}_i, y_i\}_{i=1}^m$, $\mathbf{x}_i \in \mathbf{X}$, $y_i \in y$; a Stacking algorithm.

Output: A meta learner $\boldsymbol{H}$.

Step1: Induce T base learners, *i.e.* h_1, h_2, ..., h_T, from the training set.

Step2: Construct a new dataset D', where $D' = \{(\mathbf{x}'_i, y_i)\}_{i=1}^m$. Here

$\mathbf{x}'_i = [h_1(\mathbf{x}_i), h_2(\mathbf{x}_i), ..., h_T(\mathbf{x}_i)]$.

Step3: Build a meta-learner $\boldsymbol{H}$ from D'.

Once the second-level learner is generated, for an test example $\mathbf{x}$, its predictions are $h'(h_1(\mathbf{x}), h_2(\mathbf{x}), ..., h_T(\mathbf{x}))$, where $(h_1, h_2, ..., h_T)$ are first-level learner and h' is the second-level learner. We can see that Stacking is a general framework. We can plug in different learning approaches or even ensemble approaches to generate first or second level learner. Compared with Bagging and Boosting, Stacking "learns" how to combine the base learner instead of voting.

5.4 Cross-Validations in Stacking

In Table 5.1, we use the same data set D to train first-level classifiers and prepare training data for second-level classifiers, which may lead to over-fitting. To solve this problem, the idea of cross validation is incorporated in stacking. K-fold cross validation (CV) is the most commonly used technique to evaluate classification performance. To evaluate the prediction ability of a learning algorithm, we conduct the following K-fold cross validation procedure: We partition training data into K

disjoint subsets and run the learning algorithm K times. Each time we learn a classifier from $K-1$ subsets and use the learnt model to predict on the remaining one subset and obtain the predictions for this subset. After K runs, all the K subsets obtain their predictions. See Figure 5.3. Each time a classifier is learnt, it is applied on the test set. Therefore the predictions are repeated K times on the test data. The final prediction on the test data is obtained by averaging the predictions of the K runs. The averaged predictions can be seen as a newly-created feature used to train the 2nd-level classifier.

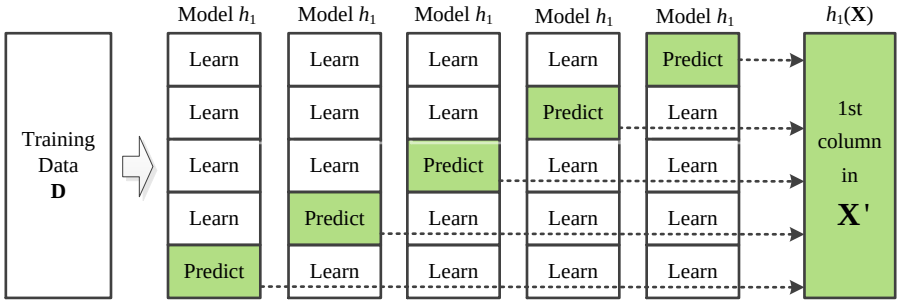

Figure 5.3: CVs (K=5) to create 2nd-level data. In practice, K=10 is used

5.5 Numerical simulations

5.5.1 A general study of Stacking model

The Stacking method is tested on an object structure that consists of a 3-DOF main structure and a 2-DOF TMD. The uncertainties in the main structure only exist in the mass (m), damping factor (c), stiffness coefficient (k). The structure parameters are listed in Table 5.2. 'SD' means standard deviation.

The other parameters are $\omega_f = 8\pi$, $\xi_f = 0.4$, $S_0 = 0.031$. S_0 is the power spectral density of the white noise process. The training set has 20000 cases; the test set has 1000 cases. To show the evaluation results in a dynamic way, the size of training set is changed from 1000 to 20000 with step 1000. For each training set, a Stacking model is learned. In this model, the base learners are chosen among tree models

Table 5.2: Structure parameters ($i = 1, 2, 3; j = 1, 2$)

Index	m_{s_i}	c_{s_i}	k_{s_i}	m_{T_j}	c_{T_j}	k_{T_j}
Nominal	4.6	62	6500	1.38	1.83	39.0
SD	1	10	300	-	-	-

such as RF, Extra-trees (ETs) [40] or GB. The meta-model is GB [6]. Root mean square error (RMSE) is used to evaluate the model.

In Figure 5.4, the accuracy of two Stacking models as well as three single models *i.e.* RF, ETs and GB, are compared. It is found that the Stacking models always outperform the individual base models when enough training data is available. The Stacking1 model has smaller RMSEs than its base learners RF and ETs. Similarly, the Stacking2 model is more ac-curate than its base learners RF, ETs and GB all the time. By comparison of the two Stacking models, it is evident that more base models will probably result in more accurate predictions.

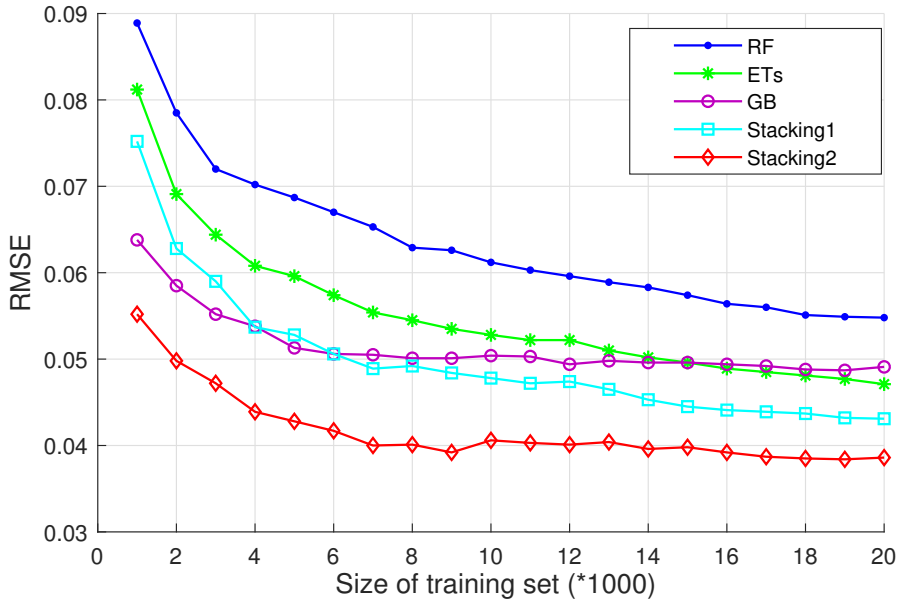

Figure 5.4: Model evaluations. Both Stacking1 and Stacking2 take GB as meta-model. Stacking1 takes RF, ETs as base models; Stacking2 takes RF, ETs, GB as base models

The models RF, ETs and GB are all tree-based methods, but they have different principles. The combination of them helps offset their weakness and improve the prediction accuracy. Interestingly, the model Stacking1 does not achieve better performance than GB when the size of training data is less than 6000. One reason is that GB is not a base model of Stacking1. If GB is added into Stacking1, resulting in Stacking2, the performance will be largely improved. This phenomenon also indicates that the choice of base models is important. In view of this, section 5.2 is devoted to the choice of base models.

5.5.2 Choice of base models and meta-model

As is shown in Section 5.1, the Stacking models generally achieve better performance than the single base learners. We now assume that the Stacking model has two base learners chosen from the set RF, GB, ETs, and the meta-learner as the remaining one from this set, for example RF, GB as the base learners and ETs as the meta-learner. Besides, we take SVR(Support Vector Regression) as a candidate of the meta-learner. The same datasets (as in Section 4.1) are employed to build the Stacking models. The results are shown in Figure 5.5.

As can be seen from Figure 5.5, the choice of base learners makes difference. The Stacking model 'RF&ETs-GB' achieves the best performance while 'RF&GB-ETs' ranks the last. Even though the base learners are the same, the Stacking performances differ due to the choice of the meta-learner. For ex-ample, the models 'RF&GB-ETs' and 'RF&GB-SVR' share the same base learners RF, GB but different meta-learners (*i.e.* ETs vs SVR). They result in largely different prediction accuracies. Similar cases are 'ETs&GB-RF' vs 'ETs&GB-SVR' and 'RF&ETs-GB' vs 'RF&ETs-SVR'. The parameters of the base models are listed in Table 5.3. 'nTrees' is the number of random trees in a base model; 'nFeatures' is the number of features in the training set; 'maxFeatures' is the number of sampled features at each node. For the meta-model(if it is not SVR), maxFeatures=2.

Figure 5.6 shows the performances of the Stacking models that commonly have three base learners. Compared with the Stacking performances in Figure 5.5, the Stacking models in Figure 5.6 result in much better performances. It is found that,

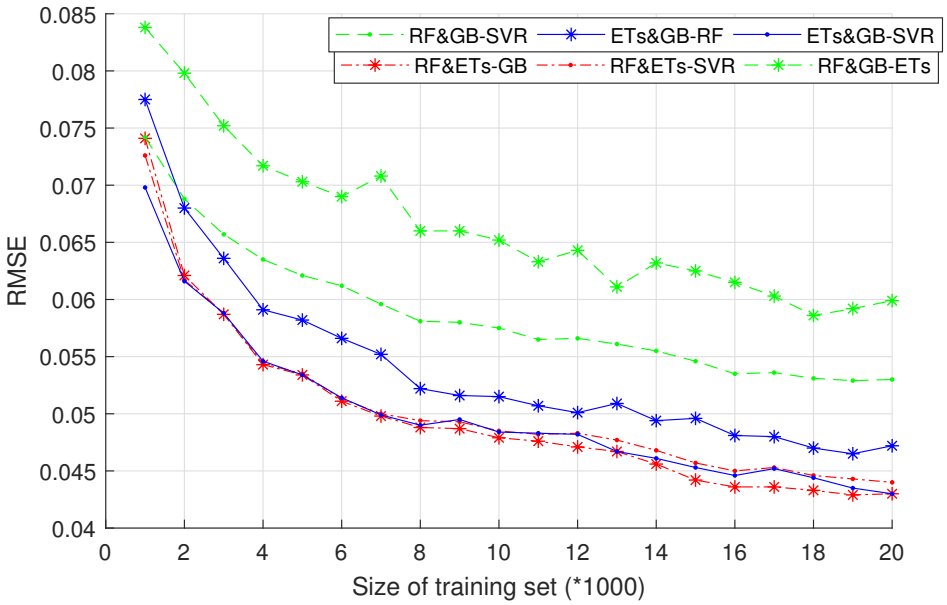

Figure 5.5: Performances of Stacking models that have two base learners and a meta-learner. The form 'A&B-C' denotes that A and B are base learners and C the meta-learner

when the SVR model is employed as the third base learner, the Stacking performance largely improves. As an example, when the SVR is added as a third base learner to 'GB&ETs-RF', the RMSE is reduced much at any point. However, this does not mean that adding more base learners always result in better predictions, see a special case in Figure 5.7.

From Figure 5.7, it is seen that the blue solid line has much larger error than the red dashed line. Actually this is not the only case, there are several other cases that have

Table 5.3: Hyper-parameters of the base models

Hyper-parameters	RF	GB	ETs
Number of random trees	50	50	50
Total num. of features	9	9	9
Num. of features to sample	3	3	3

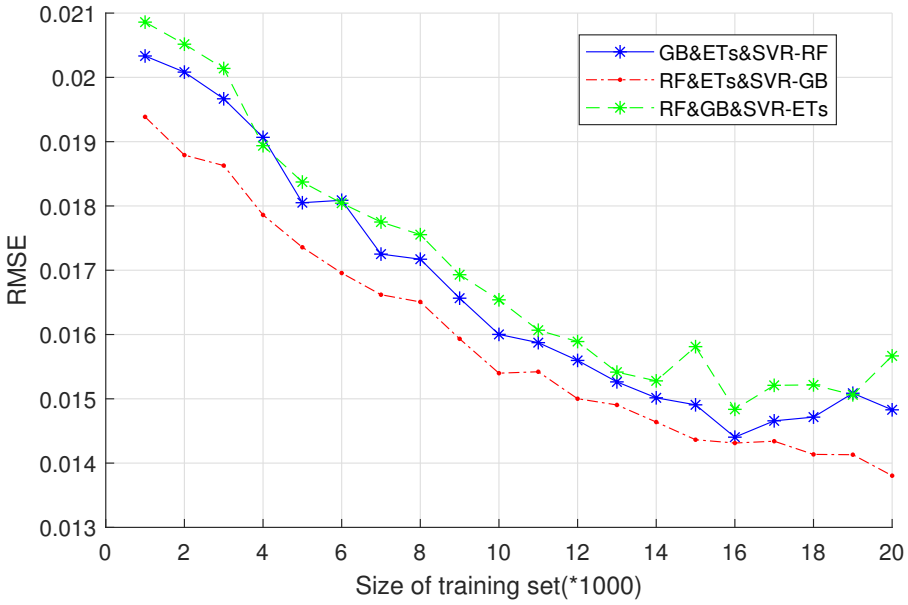

Figure 5.6: Performances of Stacking models that have three base learners and a meta-learner. The form 'A&B&C-D' denotes that A, B and C are base learners and D the meta-learner

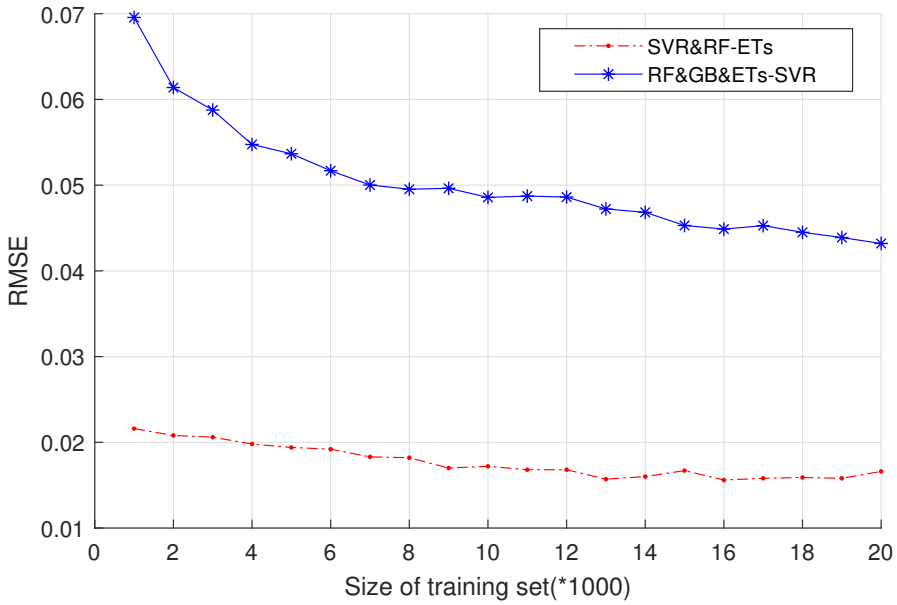

Figure 5.7: A special case: 'two' is better than 'three'

similar performances. In Figure 5.8, five Stacking models are examined that have two base learners. It is noticed that, comparing with the plots in Figure 5.5, when the SVR model is employed as one of the two base learners, the Stacking performance is largely improved. In principle, SVR totally differs from RF, GB and ETs. Besides, it is found that when GB model is taken as the meta-learner, the model performance will always be improved. For example, in Figure 5.5, the model 'RF&ETs-GB' achieves higher accuracy than 'RF&ETs-SVR'; in Figure 5.8, the model 'SVR&RF-GB' and 'SVR&ETs-GB' achieves the highest accuracy among the Stacking models explored; in Figure 5.6, the model 'RF&ETs&SVR-GB' is also among the most accurate models, but it consumes more time than the models 'SVR&RF-GB' and 'SVR&ETs-GB'. As we know, GB model is good at reducing bias. This is a reason why the Stacking models having GB as the meta-learner always result in higher accuracy.

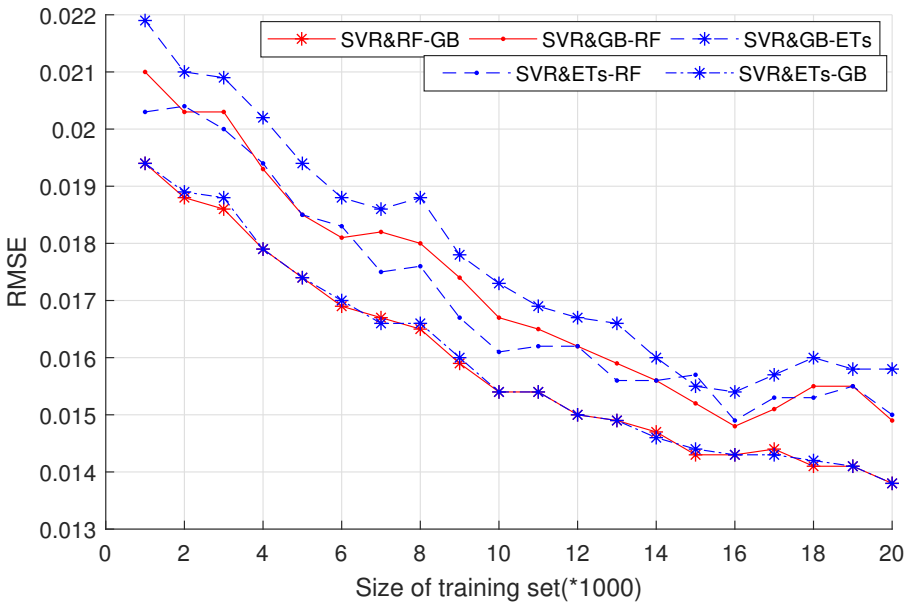

Figure 5.8: Stacking models that have two base learners

5.5.3 Hyper-parameters in the Stacking model

In Section 5.2, the base models as well as the meta-models have fixed parameters. For example, the SVR model takes radial basis function (rbf) as its kernel function; the models RF, GB and ETs have their parameters listed in Table 5.2. In this section, the roles of model parameters are explored. The training set consists of 10000 instances, each of which has 9 attributes. Firstly, we focus on the Stacking model 'SVR&RF-GB'. We change the number of trees in RF meanwhile keeping the same the settings of SVR and GB. The number of trees in RF range from 10 to 100. See the simulation results in Figure 5.9.

In Figure 5.9, it is found that the number of random trees in RF has little effect on the performance of the Stacking model. In contrast, the number of trees in GB model has great influence on the Stacking performance (see Figure 5.10). Besides, the RMSE value converges when the number of trees in GB is 50. For SVR model as one of the base learners, the kernel function is an important parameter. In this aspect, four kernel types are examined, *i.e.* 'Linear', 'Polynomial', 'Radial basis function (rbf)', 'Sigmoid'. In Figure 5.11, it is found that the Stacking model reaches its best performance when the kernel function of SVR is 'rbf'.

As is known that each random tree in RF (or GB) is trained in a recursive, binary splitting way. For each splitting, a random subset of features in the training data is sampled. The number of features is an important index that determines the model performance. In Figure 5.12, the RMSE is plotted according to different sizes of this subset. Obviously, the number of sampled features for each splitting has little effect on the prediction accuracy, because the RMSE value changes very small even though it has a trend of decrease. In contrast, for the mete-model GB, the number of sampled features for each splitting plays an important role. As the Stacking model 'SVR&RF-GB' has two base learners, the training data for the meta-model only has two features. From Figure 5.13, it is clear that the RMSE of the model reaches its lowest when the number of features sampled is $n = 2$.

Stacking models always outperforms the individual models, and their performance can be further improved by adjusting the hyper-parameters. But the intrinsic causes

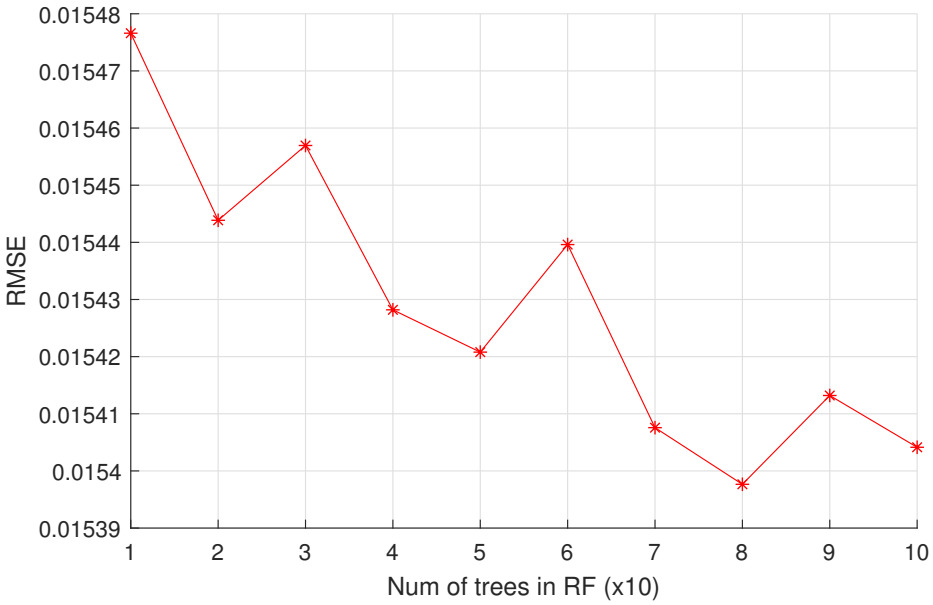

Figure 5.9: Change of RMSE with respect to the number of trees in RF

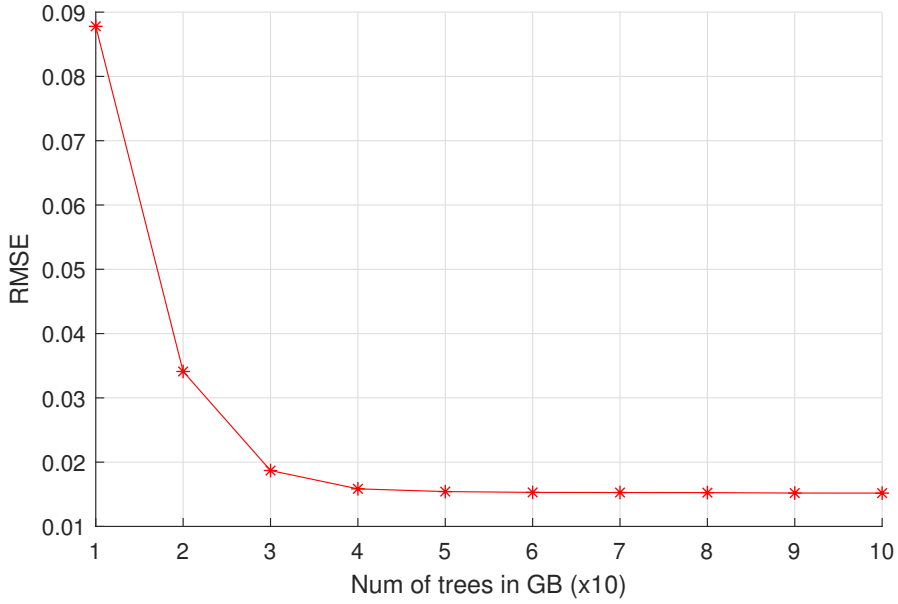

Figure 5.10: Change of RMSE with respect to the number of trees in GB

of these improvements are still not studied. Researchers always employ bias-variance analysis to see how different components of the error are changed by a design of a

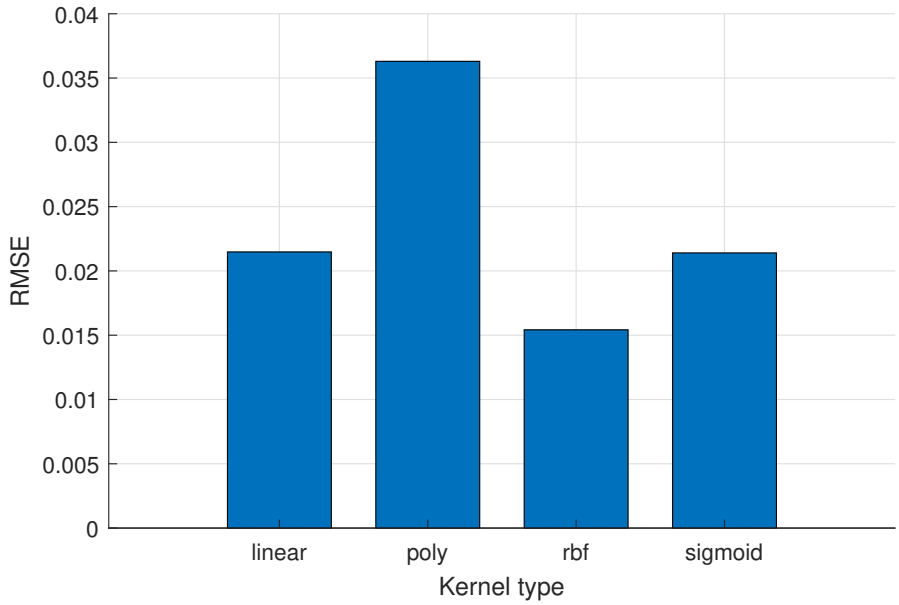

Figure 5.11: RMSE with respect to the kernel type of SVR

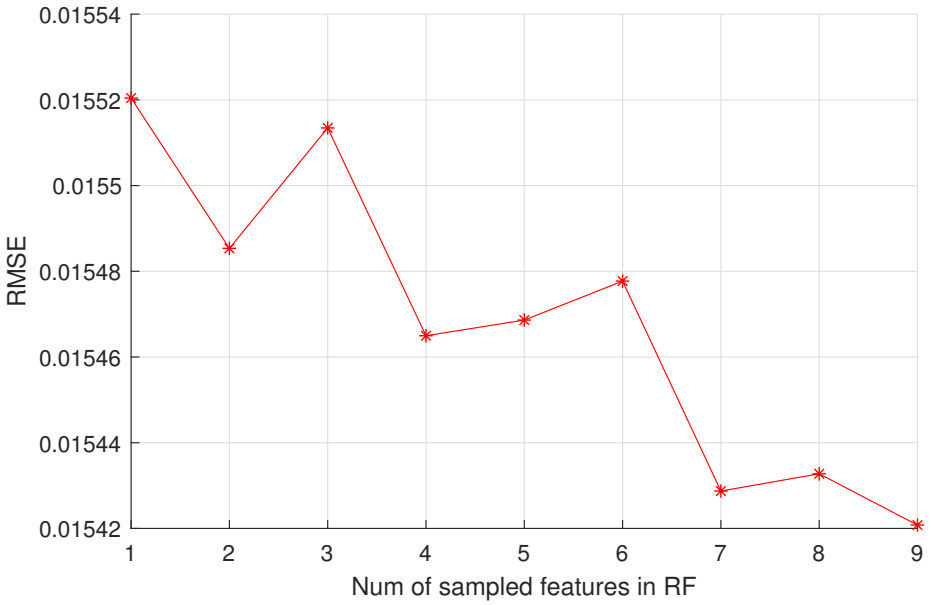

Figure 5.12: RMSE in terms of the number of sampled features in each splitting in RF

new model. Therefore, section 4.4 devotes itself to bias-variance analysis of the Stacking model.

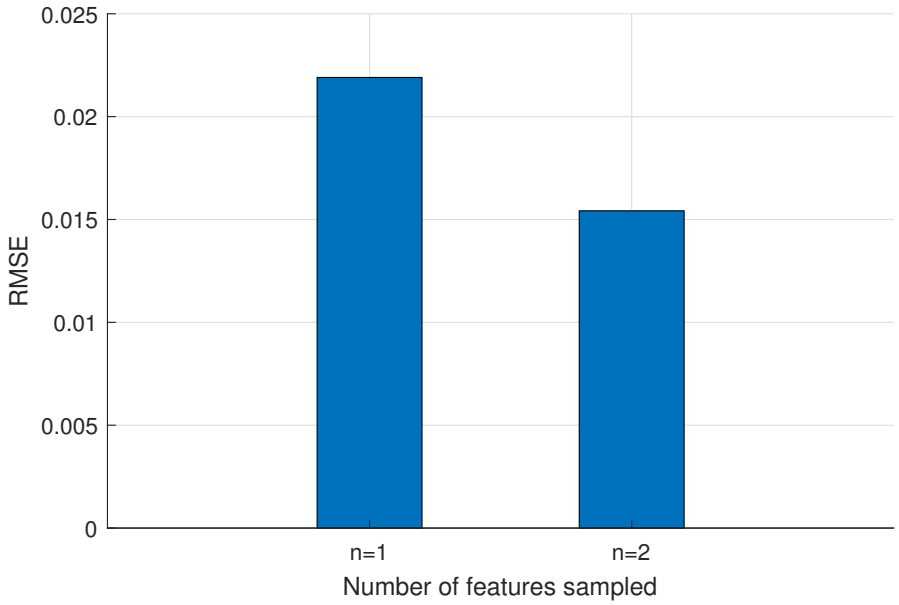

Figure 5.13: RMSE in terms of the number of sampled features for each splitting in GB

5.5.4 Bias-variance analysis of Stacking model

Bias-variance decomposition [8] is an important general tool for analyzing the performance of learning algorithms. Given a learning target and the size of training set, it divides the generalization error of a learner into three components, *i.e.*, intrinsic noise, squared bias and variance. For a regression problem where f is the unknown target function to be learned, f maps elements in an input space X into numeric values. Let D be a probability distribution over X such that a random example, $x \in X$, is drawn with probability $D(x)$. Let $L = \{(x, f(x)+\varepsilon)|x \in X\}$ be a learning set of m cases. For each case, the input x is drawn from D and the output value $y = f(x)+\varepsilon$ is corrupted by Gaussian noise with zero mean and standard deviation σ. From L, an algorithm A will output an hypothesis h_L. For a given test point $\mathbf{x}_0$, the predicted output is $\hat{y} = h_L(x_0)$.

$$
\begin{aligned}
&E_{L,\varepsilon}[(y-\hat{y})^2] = E_{L,\varepsilon}[(y-h)+(h-\hat{y})]^2 \\
&= E_{L,\varepsilon}[(y-h)^2 + (h-\hat{y})^2 + 2(y-h)(h-\hat{y})] \\
&= E_{L,\varepsilon}[(y-h)^2 + (h-\hat{y})^2] + 2E_{L,\varepsilon}[y(h-\hat{y}) - h(h-\hat{y})] \\
&= E_{L,\varepsilon}[(y-h)^2] + E_{L,\varepsilon}[(h-\hat{y})^2]
\end{aligned} \tag{5.1}
$$

The term $E_{L,\varepsilon}[y(h-\hat{y}) - h(h-\hat{y})] = 0$, because

$$
\begin{aligned}
&E_{L,\varepsilon}[y(h-\hat{y})] = E_{L,\varepsilon}[f(x)+\varepsilon]\cdot[E_L[h_L(x)] - h_L(x)] \\
&= E_{\varepsilon}[f(x)+\varepsilon]\cdot E_L E_L[h_L(x)] - h_L(x) \\
&= E_{\varepsilon}[f(x)+\varepsilon]\cdot E_L[h_L(x)] - E_L[h_L(x)] = 0
\end{aligned} \tag{5.2}
$$

$$
\begin{aligned}
&E_{L,\varepsilon}[h(h-\hat{y})] = E_{L,\varepsilon}E_L[h_L(x)]\cdot[E_L[h_L(x)] - h_L(x)] \\
&= E_L[h_L(x)]\cdot E_L E_L[h_L(x)] - h_L(x) \\
&= E_L[h_L(x)]\cdot E_L[h_L(x)] - E_L[h_L(x)] = 0
\end{aligned} \tag{5.3}
$$

The term $E_{L,\varepsilon}[(y-h)^2]$ is further decomposed as follows,

$$
\begin{aligned}
&E_{L,\varepsilon}[(y-h)^2] = E_{L,\varepsilon}(y^2 - 2yh + h^2) \\
&= E_{L,\varepsilon}(y^2) - 2E_{L,\varepsilon}(yh) + E_{L,\varepsilon}(h^2) \\
&= E_{\varepsilon}(y^2) - 2hE_{\varepsilon}(y) + h^2 \\
&= E_{\varepsilon}[y-f(x)]^2 + f^2(x) - 2hf(x) + h^2 \\
&= E_{\varepsilon}(\varepsilon^2) + [h-f(x)]^2 \\
&= \sigma^2 + [h-f(x)]^2
\end{aligned} \tag{5.4}
$$

Therefore,

$$
\begin{aligned}
&E_{L,\varepsilon}[(y-\hat{y})^2] = \sigma^2 + [h-f(x)]^2 + E_L[(h-\hat{y})^2] \\
&= noise^2 + bias^2(\hat{y}) + var(\hat{y})
\end{aligned} \tag{5.5}
$$

The squared noise describes how much y varies from $f(x)$. It is due to the problem at hand and cannot be avoided. It is the lower bound of the expected error of any learning algorithm on the target. The squared bias is the difference between the average estimate and the target. The variance describes the magnitude of the fluctuation of the estimate (for different training sets of the same size). Since the noise is intrinsic and difficult to estimate, it is often subsumed into the bias term [118]. Thus, the generalization error is broken into bias and variance terms, *i.e.*

$$\begin{aligned} & E_{L,\varepsilon}[(y-\hat{y})^2] = (h-y)^2 + E_L[(h-\hat{y})^2] \\ & = bias^2[\hat{y}] + var[\hat{y}] \end{aligned} \tag{5.6}$$

Generally, if the model is 'too global', then its bias will be high, the model is under-fitting the data; If the model is 'too local', then its variances will be high, the model is over-fitting the data. In practice, we have only one dataset to train the model. To measure bias and variance, bootstrap sampling is generally applied to simulate multiple training sets. The pseudo-code is shown in Table 5.4.

Table 5.4: Pseudo-code of bias&variance calculation

Input: Training data $D = \{\mathbf{x}_i, y_i\}_{i=1}^m$, $\mathbf{x}_i \in \mathbf{X}$, $y_i \in Y$; a Stacking algorithm.

Output: Bias and variance of the model.

Step1: Generate n bootstrap replicates of D, *i.e.* $\{D_i\}_{i=1}^n$.

Step2: Apply learning algorithm to each D_i and obtain an hypothesis h_i.

Step3: Let T_i be the OOB points for D_i, compute predicted values $h_i(\mathbf{x})$ for $\mathbf{x}_i \in T_i$, By now, each $\mathbf{x}_i \in D$ has several predictions.

Step4: For $\mathbf{x}_i$, estimate the bias and variance according to eq.(5.6).

Step5: Output the averaged results from *Step4* over all points in D.

The bias-variance analysis is applied on the model 'SVR&RF-GB' which has base models SVR, RF and meta-model GB. The training set is taken from section 5.5.1,

containing 10000 cases. Based on the procedures in Table 5.4, the bias, variance and MSE (also RMSE) is computed, see Table 5.5. The base learners SVR takes 'rbf' as its kernel function, RF has the same parameters as listed in Table 5.3. 10-fold CVs is applied in the 1st-level training of Stacking. 100 bootstrap datasets are generated to represent the variations of the training data. In Table 5.5, the analysis results of the single models, i.e. SVR, RF and GB, are also provided and compared with those of the Stacking model. Obviously, the Stacking model achieves much lower squared bias than any single models; its variance is larger than SVR, but it is smaller than RF and GB. However the sum of the two terms, i.e. MSE, is much smaller than any single model. Generally, the bias-variance decomposition results vary when a single model (RF or GB) changes its value of K, *i.e.* number of sampled features in each split. This is because the model complexity changes when K is set differently, which possibly results in overfitting or underfitting.

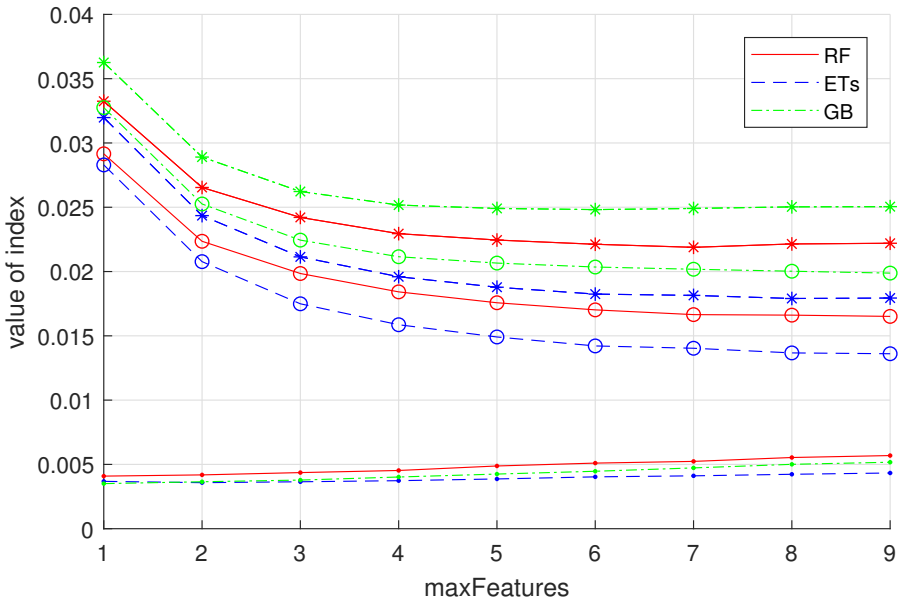

Figure 5.14: Bias-variance of single models in terms of maxFeatures in each split

Figure 5.14 shows how the bias and variance of a single model changes with respect to different K values in that model. As the plot in Figure 5.14 shows, the three models have similar trends of change. The plots with $'*'$ denote the MSE of the

models; the plots with $'o'$ denote the bias of the models; the plots with $'.'$ denote the variance of the models. For small values of K, random effects are strong, leading to high bias and low variance; when K becomes larger, random effects are weaker, resulting in low bias but high variance. For all three models, too small K leads to a too large bias, for which the variance cannot compensate. For RF, the optimal trade-off is at $K = 5$; for GB, $K = 4$; for ETs, $K = 6$.

Table 5.5: Bias-variance decomposition result

Models	$bias^2$	$variance$	MSE	$RMSE$
SVR&RF-GB	**2.82e-04**	**1.13e-05**	**2.93e-04**	**1.71e-02**
SVR	4.00e-03	3.28e-6	4.00e-03	6.32e-02
RF	4.15e-03	5.48e-4	4.69e-03	6.85e-02
GB	6.50e-03	4.01e-4	6.90e-03	8.31e-02

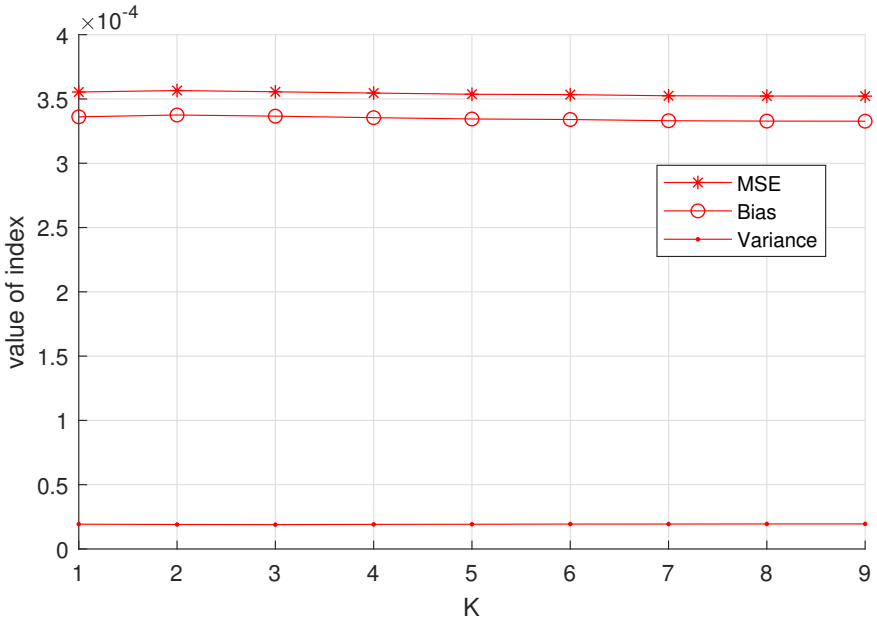

Figure 5.15: Bias-variance of Stacking model in terms of K in each split in base learner RF

However, in the Stacking model where different base models are combined to learn a meta-learner, the role of K may change. In the model 'SVR&RF-GB', the K values in RF as well as GB are seen as a variable to see its influence on the bias-variance of this Stacking model. A training set of 5000 cases with 9 features is employed, 50 random trees are learned in both RF and GB. Figure 5.15 shows how the bias and variance (also MSE) changes according to different K values in RF. It is found that the bias and variance of the Stacking model does not change so much. Therefore, K in RF does not plays an important role in the bias-variance decomposition results. In contrast, Figure 5.16 tells that the value K in the meta-model GB greatly affects the bias and variance computation results. At $K = 1$, $bias = 5.24e{-}4$, $variance = 0.72e{-}4$, $MSE = 5.96e{-}4$; at $K = 2$, $bias = 3.33e{-}4$, $variance = 0.20e - 4$, $MSE = 3.53e - 4$. Therefore, K is important in determining the Stacking performance.

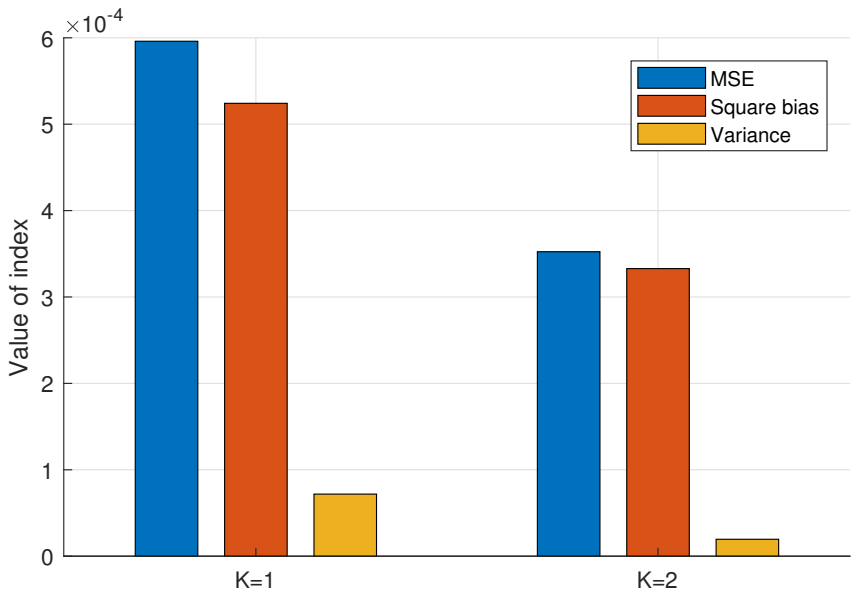

Figure 5.16: Bias-variance values in terms of K in each split in meta-model

5.5.5 Complexity analysis of Stacking models

In learning a model, the time complexity is the number of operations required for building the model from the training data. Here, the Stacking2 model is employed as an example. In this stacking model, RF, ETs and GB are base learners, GB is meta-learner. Some parameters are defined: N is the size of the training set; $K_1(K_2)$ is the number of sampled variables at each node in the 1st(2nd) level models; M_1, M_2 and M_3 are the number of random trees in the base learners RF, ETs and GB respectively; M_4 is the number of random trees in the meta-learner. 10-fold CVs are used to train the base learners. For RF, because of 10-fold CVs, 90% of the training data is used to train a RF. Besides, the bootstrapping takes 63.2% of the training data to train each tree. Therefore, the time complexity to build a RF is [41],

$$\begin{aligned} T_{RF} &= O(M_1 * K_1 * (0.632 * 0.9N)\log^2(0.632 * 0.9N)) \\ &= 0.57 * O(M_1 * K_1 * N\log^2(0.57N)) \\ &= 0.57M_1K_1 * O(N\log^2 N) \end{aligned} \tag{5.7}$$

Then, 10% of the training data L_1 is used to make predictions. The time complexity to make predictions is

$$\begin{aligned} T'_{RF} &= O(0.1N * M_1 \log(0.632 * 0.9N)) \\ &= 0.1M_1 * O(N \log N) \end{aligned} \tag{5.8}$$

For ETs, it is noted that in each loop of CV, all cases in the training data are used to train the trees in ETs. The time complexities to build the model and make predictions are

$$\begin{aligned} T_{ETs} &= O(M_2 * K_1) * (0.9N) \log(0.9N)) \\ &= 0.9M_2K_1 * O(N \log N) \end{aligned} \tag{5.9}$$

$$\begin{aligned} T'_{ETs} &= O(0.1N * M_2 \log(0.9N)) \\ &= 0.1M_2 * O(N \log N) \end{aligned} \tag{5.10}$$

For GB, each tree (from the 2nd) is built from the weighted sum of the prediction errors of the former trees. So the time complexity of training a GB model is divided into two parts: time complexity of building the trees and time complexity of making predictions by the trees. It is noted that the first and last trees are not used to make predictions in the training process of GB. The first tree is a constant which is seen as a 1-node tree [71]. It is calculated as the average value of the targets in the training set. For the first tree, the time complexity is ignored, so we just consider the other $(M_3 - 1)$ trees. In GB, the whole training set is used to train each tree. The time complexity to build the trees in GB model is

$$\begin{aligned} T_{GB}^1 &= O((M_3 - 1)K_1(0.9N)\log^2(0.9N)) \\ &= 0.9(M_3 - 1)K_1 * O(N\log^2 N) \end{aligned} \tag{5.11}$$

Let's see how the time complexity of making predictions is calculated during the inducing process of GB model. In the inducing process of GB model, to build the 3rd tree, we need the predictions of the 2nd tree, the time complexity to make predictions by the 2nd tree is $O(\log(0.9N))$; to build the 4th tree, the predictions of the 2nd and 3rd trees are needed, the time complexity is $2O(\log(0.9N))$; $\cdots$; to build the M_3th tree, the predictions of the 2nd until (M_3-1)th trees are needed, the time complexity is $(M_3 - 2) \times O(\log(0.9N))$. Therefore, the total time complexity to make predictions within GB model learning process is

$$\begin{aligned} T_{GB}^2 &= O(\log(0.9N)) \sum_{i=1}^{M_3-2} i \\ &= 0.5 * (M_3 - 2)(M_3 - 1) * O(\log N) \end{aligned} \tag{5.12}$$

In total, the time complexity needed for building a GB model is

$$\begin{aligned} T_{GB} &= T_{GB}^1 + T_{GB}^2 \\ &= 0.5 * (M_3 - 2)(M_3 - 1) * O(\log N) \\ &+0.9(M_3 - 1)K_1 * O(N \log^2 N) \end{aligned} \tag{5.13}$$

Then the GB model is used to make predictions on the 10% of the training set. The time complexity is

$$\begin{aligned} T'_{GB} &= O(0.1N(M_3-1)\log(0.9N)) \\ &= 0.1(M_3-1) * O(N\log N) \end{aligned} \tag{5.14}$$

To make it clearer, the time complexities of the models are collected in Table 5.6.

Table 5.6: Time complexities (average) of the base models in Stacking2

Models	Model building	Predicting
RF	$0.57M_1K_1 * O(N\log^2 N)$	$0.1M_1 * O(N\log N)$
ETs	$0.9M_2K_1 * O(N\log N)$	$0.1M_2 * O(N\log N)$
GB	$[\begin{array}{l} 0.9(M_3-1)K_1 * O(N\log^2 N) \\ +0.5 * (M_3-2)(M_3-1) * O(\log N) \end{array}]$	$0.1(M_3-1) * O(N\log N)$

In the 1st-level training process, RF, ETs and GB are trained respectively. 10-fold CVs are applied where 9 folds are used to train the model and 1 fold is used to make predictions. The time complexity is denoted as

$$T_1 = 10 \cdot (T_{RF} + T'_{RF} + T_{ET} + T'_{ET} + T_{GB} + T'_{GB}) \tag{5.15}$$

where T, T' are the time complexity to build the model and make predictions respectively.

$$\begin{aligned} T_1 &= 10 \cdot (T_{RF} + T'_{RF} + T_{ET} + T'_{ET} + T_{GB} + T'_{GB}) \\ &= 10 \cdot [0.57M_1K_1 * O(N\log^2 N) + 0.1M_1 * O(N\log N) \\ &+0.9M_2K_1 * O(N\log N) + 0.1M_2 * O(N\log N) \\ &+0.9(M_3-1)K_1 * O(N\log^2 N) + 0.5 * (M_3-2) \\ &*(M_3-1) * O(\log N) + 0.1(M_3-1) * O(N\log N)] \\ &= (5.7M_1 + 9M_3 - 9)K_1 * O(N\log^2 N) \end{aligned} \tag{5.16}$$

In the 2nd-level training process, a GB model is trained, using the features created by the base learners and the decisions from the original training set,

$$\begin{aligned} T_2 &= (M_4 - 1)K_2 * O(N \log^2 N) + 0.5 * (M_4 - 2)(M_4 - 1) * O(\log N) \\ &= (M_4 - 1)K_2 * O(N \log^2 N) \end{aligned} \tag{5.17}$$

Therefore, the total time complexity is denoted as (notice that $\widetilde{N} \approx 0.632 N_1$)

$$\begin{aligned} T &= T_1 + T_2 \\ &= (5.7M_1 + 9M_3 - 9)K_1 * O(N \log^2 N) + (M_4 - 1)K_2 * O(N \log^2 N) \\ &= (5.7M_1K_1 + 9M_3K_1) - 9K_1 + M_4K_2 - K_2) * O(N \log^2 N) \end{aligned} \tag{5.18}$$

To conclude, for the Stacking2 model, the total time complexity is denoted as

$$T = (2.5M_1K_1 + 4M_3K_1 - 4K_1 + M_4K_2 - K_2) \times O(N \log^2 N) \tag{5.19}$$

where N is the size of the original training set. We see that the time complexity T calculated above approximates a linear relationship with respect to N. Now we have calculated the time complexity of Stacking2 model. For Stacking1 model, the time complexity can be calculated in a similar way. Figure 5.17 shows the CPU time spent to build the models involved in Figure 5.4. In the process of learning a Stacking model, several base models need to be built, so it is reasonable that the Stacking models always take more time to build than single models do. The comparison of two Stacking models tells that the running time will increase when more base learners are added in the first-level training process. For each Stacking model, the CPU time increases in a close-to-linear way with respect to the size of training data.

The Stacking model 'SVR&RF-GB' is also analyzed as to the CPU time. As is shown in Figure 5.18, the CPU time of this Stacking model grows linearly with the increase of the training data. The single models SVR, RF and GB also have increases

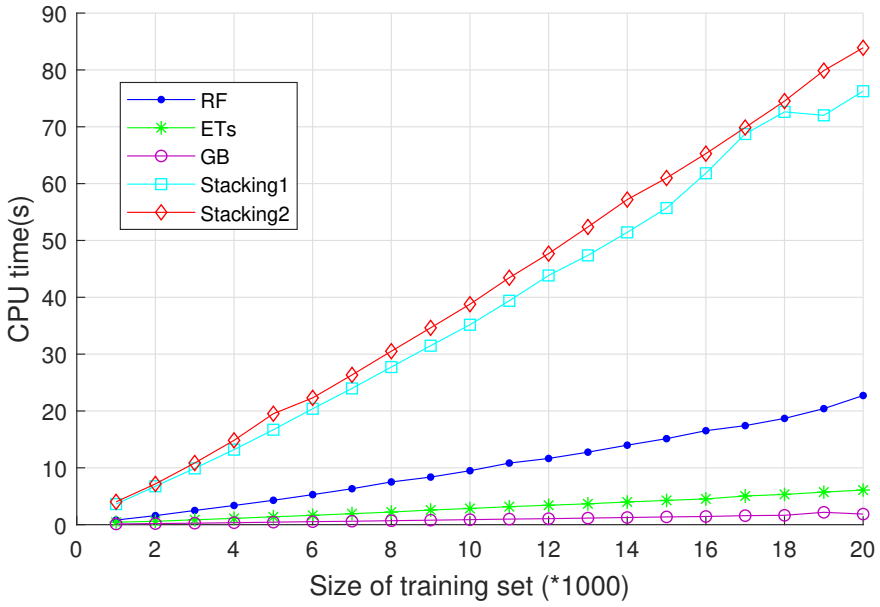

Figure 5.17: CPU time of the models employed in the simulations

of CPU time but they are much slower. The main reason is that, in the training process of the Stacking model, 10-fold CVs are applied to train each base learner as well as make them predict on the training data. Figure 5.19 compares the accuracies of the Stacking model with the single models. These plots show an attractive improvement of the prediction performance. These numerical simulations indicate that the Stacking model gives significantly better results in terms of reducing the errors in predictions. Actually, by applying the Stacking model 'SVR&RF-GB', the prediction accuracy is improved by about 80% compared with the single models.

5.6 Comparisons between Stacking and RF

5.6.1 The benchmark problem in section 4.4.6

The Stacking method is compared with RF in the benchmark problem of 10-DOF oscillator introduced in section 4.4.6.2. The base learners of the Stacking model are SVM and RF (with 30 trees), the meta-learner is ETs (with 30 trees). For each threshold examined, the two methods use the same size of training set, see Table 5.7. Notice that the failure probability values are calculated in a mean sense. Form Table

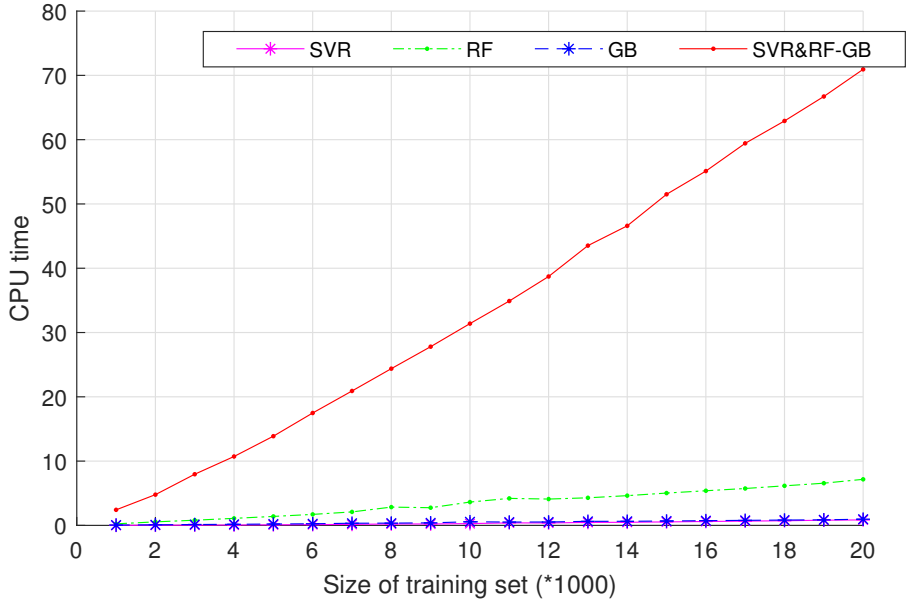

Figure 5.18: CPU time of Stacking model 'SVR&RF-GB' and single models

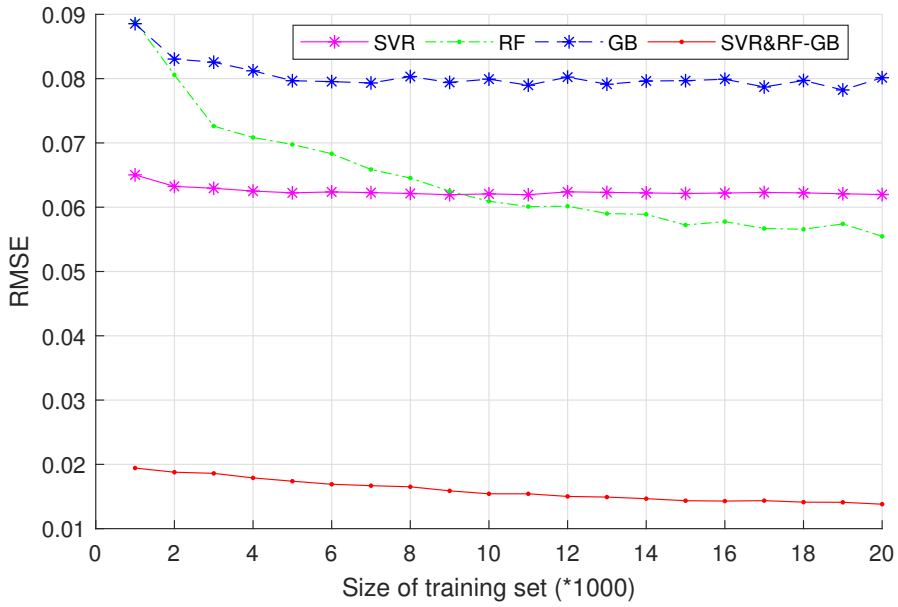

Figure 5.19: RMSE of Stacking model 'SVR&RF-GB' and single models

5.7, it is found that the Stacking model outperforms RF model when the threshold is $y^1_{lim} = 0.057m$ (*i.e.* the threshold for 1stDOF is $0.057m$), $y^1_{lim} = 0.073m$ and $y^{10}_{lim} = 0.013m$. They have equal estimations when the threshold is $y^{10}_{lim} = 0.017m$.

Table 5.7: Compare Stacking and RF in reliability estimations

Method	$y^1_{lim} = 0.057m$	$y^1_{lim} = 0.073m$	$y^{10}_{lim} = 0.013m$	$y^{10}_{lim} = 0.017m$
Standard MCS	$1.06e^{-4}$	$8.07e^{-7}$	$4.88e^{-5}$	$2.52e^{-7}$
num of samples	$2.98e^{7}$	$2.98e^{7}$	$2.98e^{7}$	$2.98e^{7}$
Random Forest	$7.6e^{-5}$	$1.0e^{-6}$	$4.2e^{-5}$	$1.1e^{-7}$
num of samples	800	1000	800	900
Stacking	$1.0e^{-4}$	$9.2e^{-7}$	$4.3e^{-5}$	$1.1e^{-7}$
num of samples	800	1000	800	900

It is noticed that only one threshold value is studied in Table 5.7. In contrast, multi-threshold values are considered here. Firstly, the failure criterion is defined as $y^1_{lim} = 0.057m \cup y^{10}_{lim} = 0.013m$, which means that the failure takes place when the 1st DOF exceeds $0.057m$ or the 10th DOF exceeds $0.013m$; Secondly, in a similar way, the failure criterion is defined as $y^{10}_{lim} = 0.073m \cup y^{10}_{lim} = 0.017m$. Then the failure estimations by the two methods are calculated, see Table 5.8. It is found that, in these two settings, the two methods result in similar failure probability estimations.

Table 5.8: Compare Stacking and RF when multi-thresholds are assumed

Method	$y^1_{lim} = 0.057m \cup y^{10}_{lim} = 0.013m$	$y^1_{lim} = 0.073m \cup y^{10}_{lim} = 0.017m$
Standard MCS	$1.12e^{-4}$	$1.00e^{-6}$
num of samples	$1.0e^{6}$	$1.0e^{6}$
Random Forest	$1.11e^{-4}$	$1.10e^{-6}$
num of samples	800	1100
Stacking	$1.20e^{-4}$	$1.18e^{-6}$
num of samples	800	1100

5.6.2 NonGaussian structural properties

As is known that all the simulations before are based on the assumption of Gaussian random structural parameters. In this part, non-Gaussian variables are studied. Three situations are discussed here. 1) Lognormal: all random parameters follow lognormal distributions; 2) Gamma: all random parameters follow Gamma distributions; 3) Mixed: the first 5 DOFs follow Lognormal distributions while the last 5 DOFs follow Gamma distributions. Table 5.9 shows the results of failure probability estimations in the first situation.

Table 5.9: Compare Stacking and RF when the structural parameters are all lognormal

Method	$y^1_{lim} = 0.057m$	$y^{10}_{lim} = 0.013m$
Standard MCS	$8.8e^{-5}$	$4.0e^{-5}$
num of samples	$1.0e^{6}$	$1.0e^{6}$
Random Forest	$7.1e^{-5}$	$4.0e^{-5}$
num of samples	1600	1600
Stacking	$8.0e^{-5}$	$4.1e^{-5}$
num of samples	1600	1600

Table 5.10: Compare Stacking and RF when the structural parameters are all Gamma

Method	$y^1_{lim} = 0.057m$	$y^{10}_{lim} = 0.013m$
Standard MCS	$9.8e^{-6}$	$3.7e^{-6}$
num of samples	$2.0e^{7}$	$2.0e^{7}$
Random Forest	$9.7e^{-6}$	$3.6e^{-6}$
num of samples	500	800
Stacking	$9.6e^{-6}$	$3.7e^{-6}$
num of samples	500	800

The 1st DOF or 10th DOF are considered to define the failures. It is found in Table 5.9 that, both methods achieve relatively high accurate estimations of failure probability. In the first case where $y^1_{lim} = 0.057cm$, the Stacking method behaves better than RF; while in the second case where $y^{10}_{lim} = 0.013cm$, both methods result in estimations very close to the reference values estimated by standard MCS. In the second situation, all parameters follow Gamma distributions. The same thresholds are studied as those in Table 5.9. We find that the two methods all give very close results to the refered values. In Table 5.11, the types of parameter distributions are mixed. The two methods still give high accurate estimation results.

Table 5.11: Compare Stacking and RF when the structural parameters are a mixture of Lognormal and Gamma

Method	$y^1_{lim} = 0.057m$	$y^{10}_{lim} = 0.013m$
Standard MCS	$3.1e^{-5}$	$4.0e^{-6}$
num of samples	$2.0e^{7}$	$2.0e^{7}$
Random Forest	$3.0e^{-5}$	$4.1e^{-6}$
num of samples	1000	900
Stacking	$3.2e^{-5}$	$4.1e^{-6}$
num of samples	1000	900

5.7 Chapter summary

In this chapter, Stacking method is explored for structural reliability modeling. The Stacking method builds the model in a hierarchical way, combining different base learners to produce a meta-learner that generally outperforms any of the base learners. Numerical simulations are performed from various perspectives such that the proposed model is examined in a systematic way. The simulation results show that, with a price of a limited CPU time increment, the Stacking model obtains a significant advantage over its base learners in terms of structural reliability predictions.

Stacking method is compared with RF on the benchmark problem in chapter 4. Non-Gaussian structural random variables are also studied in structural reliability estimation.

Chapter 6

Conclusions

Contents

The structural reliability assessment always demands appropriate characterization of input information such as loading, material parameters, etc. However, these parameters are often not determined uniquely as they are affected by various uncertainties due to interior and exterior factors. Hence, the output of virtual simulation models cannot be specified by a single value. Uncertainties in inputs are propagated to output quantities generally by simulation techniques. Uncertainty quantification is most useful for handling uncertainties and making evaluations on levels of structural safety, maintenance and repair activities. Developments in the field of data mining have provided various powerful tools for predicting performance of structural systems in the framework of virtual simulation models. Therefore, machine learning technics have been extensively explored in this dissertation for structural reliability assessment.

6.1 Dissertation contributions

In summary, this research contributes to methodologies of structural reliability analysis and machine learning theories in structural reliability estimation. The reliability analysis is performed in a new framework of ensemble learning models that models structural parameter uncertainties and map them into failure probabilities. The following list includes the contributions made in the dissertation.

- Conditional failure probability is well defined and studied. By considering the uncertainties both in the ground excitations and the structure properties, the overall failure probability is actually a high-dimension (generally several hundred) integral that is analytically impossible. By exploring the relationship between deterministic analysis and uncertainty analysis, we found that the uncertainty analysis can be approximated by a limited number of deterministic analysis which belong to a certain probability density distribution. The conditional failure probability is defined by assuming that the structural is deterministic, i.e. a sample of the uncertain structural properties is obtained from their joint probability density distribution, and then a failure probability is calculated for this 'deterministic' structure. The conditional failure probability is denoted as a high-dimension integral according to the distribution of all possible deterministic analysis based on the sample space of the excitation process. This task can be accomplished in the theoretic framework Monte-carlo simulations (MCS).

- The KL-IS method is employed to handle very small conditional probabilities. To overcome the shortcomings of standard MCS in approximating very small failure probabilities (for example, less than 1%), the KL-IS method is introduced that is much more efficient than the general MCS. In this method, the Karhunen-Loeve expansion is used to decompose the Gaussian excitation process into deterministic part (i.e. the K-L vectors) and uncertain part (i.e. multi-dimension standard normal vector). Under the assumption that the structure is 'deterministic', the structural response can be directly calculated as the convolution integral of its impulse responses and the excitation process that it is subjected to. Besides, to improve the sampling efficiency, an importance sampling technic has been detailed and used. The KL-IS scheme has been compared with the standard MCS and showed its overwhelming advantage in small failure probability estimations.

- Random Forest model has been used to predict conditional failure probabilities. In machine learning perspective, the estimation of conditional failure

probability is related to the regression problem. Random Forest (RF) is a powerful machine learning tool to accomplish this task. To learn a RF model, a training dataset is firstly generated. In the training set, the input is the samples of the structural properties obtained by random sampling from the assumed joint probability density function (PDF, assuming the variables are relatively independent). The output for each sample is induced by MCS or KL-IS method, depending on the scale of the failure probability to be estimated. Once the RF model is built, it can be used to make predictions on any other samples outside the training set. Finally the overall failure probability can be directly estimated. The RF model has shown its accuracy in conditional failure probability modeling and predictions.

- The Stacking model has been explored and introduced to model the conditional failure probability. In principle, a Stacking model tries to improve the prediction performance by combining various models (called base models) in a special way, then constructing a higher-level model (called meta-model). The base models can be a single learner, such as k-nearest neighbor (k-NN), support vector machine (SVM); it also can be an ensemble learner, such as RF, Gradient boosting (GB) or Extra-Tress (ETs). The meta-model is induced from the predictions of the base models together with the original outputs of the training set. Here, the predictions, by the base models, are made on the input samples of the training set. Cross-validations are carried out to make these predictions in order that the overfittings can be avoided. The simulations have been performed from various perspectives such that the proposed model is examined in a systematic way. The simulation results show that, with a price of a limited CPU time increment, the Stacking model obtains a significant advantage over its base learners in terms of structural reliability predictions.

- In this research, we have successfully introduced machine learning theories into structural reliability assessment domain. Various other machine learning tools that are designed for regression problems can be readily used.

6.2 Future work

The following issues need further research.

- In this research, a basic assumption is that the object structure is a linear system so that the KL-IS method can be applied. However, for non-linear structures, this method may not be well suited. Although linearization techniques can be used to convert nonlinear problems into linear problems, sometimes strong nonlinearity cannot be ignored. Therefore, extension of this method into non-linear domain is an important direction in the future research.

- Correlated random variables have not been studied. The stochastic excitations are modeled as filtered Gaussian white noise process, which means that the excitation values at different time points are relatively independent. However, in reality there may be some correlations between these excitations values. Similar situations happen to the uncertain structural properties that are assumed relatively independent by now. Extension of this assumption into correlated random variables will be meaningful in future research.

- The parameter tuning of Random Forest still needs more study. From the simulation results, it is known that, according to different parameter configurations the Random Forest model has clearly different performances. The parameters to be tuned constitute a 'random vector' that may include the number of trees, the number of features randomly sampled on each node, the depth of the trees and the scale of the sum errors within the leaf nodes. Optimization methods may be considered to accomplish such a task.

- For the Stacking model, except for the parameter tuning problem introduced in Random Forest, the choice of base models and meta-models is still not of high efficiency. This makes the Stacking model consume more time as various base models are involved in the learning process. The organization of these different level models needs further study.

List of abbreviations

AFOSM	advanced first-order second moment
ANN	Artificial Neural Networks
BNs	Bayesian networks
CART	classification and regression tree
CDF	cumulative distribution function
CFP	Conditional failure probability
CI	confidence interval
COV	coefficient of variation
Co-coef	correlation coefficient
CPT	conditional probability tables
CV	cross-validation
DOF	degree of freedom
ERM	empirical risk minimization
ETs	Extremely Randomized trees
FI	feature importance
FORM	First-Order Reliability Method
FOSM	first-order second moment
FOTM	first-order third moment method
GB	Gradient boosting
HPCE	Hermite polynomial chaos expansion
IS	Importance sampling
ISD	importance sampling density
kNN	k-nearest neighbors
LSF	limit state function
MABS	mobile agent based systems
MC	Monte Carlo

MCS	Monte Carlo simulation
ML	Machine learning
MSE	mean square error
OOB	out-of-bag
OOBE	out-of-bag error
PCE	Polynomial chaos expansion
PDF	probability density function
P_f	Probability of failure
RAE	relative absolute error
RBO	Reliability-Based Optimization
REPTree	Reduced Error Pruning Tree
RF	Random forest
RSM	Response Surface Method
RMSE	root mean squared error
SD	standard deviation
SFEM	stochastic finite element method
SNS	standard normal space
SORM	Second-Order Reliability Method
SSE	sum of squared errors
Stacking	Stacked Generalization
SRM	structural risk minimization
SVM	Support Vector Machines
SVR	support vector regression
TMD	Tuned mass damper
UA	uncertainty analysis

List of Figures

List of Tables

List of publications

International Journals

- **W. You**, A. Saidi, A. Zine, X. Zhong and M. Ichchou, Structural reliability modeling and prediction through Random Forest. Int. J. of Relia., Qua. and Saf. Eng. (Under review).
- **W. You**, A. Saidi, A. Zine and M. Ichchou, Mechanical Reliability Assessment by Ensemble Learning. Journal MDPI.
- **W. You**, A. Saidi, A. Zine and M. Ichchou, Machine learning based reliability analysis of linear uncertain structures under Gaussian stochastic excitation. (Under review)
- W. Chai, A. Saidi, A. Zine, C. Droz, **W. You** and M. Ichchou, Comparison of uncertainty quantification process using statistical and datamining algorithms. Structural and Multidisciplinary Optimization. 2019: 1-12.

International Conferences

- **W. You**, A. Saidi, A. Zine and M. Ichchou, Reliability modeling and prediction of passive controlled structures through Random Forest, CSNDD, Tangier, June 2018.
- **W. You**, A. Saidi, A. Zine and M. Ichchou, A new strategy for structural reliability prediction, ECMR, Djerba, July 2018.
- **W. You**, A. Saidi, A. Zine and M. Ichchou, An Ensemble Learning Model for Structural Reliability Prediction, A3m, Hammamet, December 2018.
- **W. You**, A. Saidi, A. Zine and M. Ichchou, Improving Reliability Predictions by Stacking, ASPAI, Parcelona, March 2019.
- **W. You**, A. Saidi, A. Zine and M. Ichchou, Combining Importance Sampling with Machine Learning to Accelerate Structural Reliability Analysis, MEDYNA, Napoli, February 2020.

Bibliography

[1] Bernd Bertsche. *Reliability in automotive and mechanical engineering: determination of component and system reliability*. Springer Science & Business Media, 2008. (Cited on page 1.)

[2] Alessandro Martelli, Massimo Forni, and Paolo Clemente. Recent worldwide application of seismic isolation and energy dissipation and conditions for their correct use. In *Proceedings on CD of the Structural Engineering World Congress (SEW5)*, 2012. (Cited on page 2.)

[3] BF Spencer Jr and S Nagarajaiah. State of the art of structural control. *Journal of structural engineering*, 129(7):845–856, 2003. (Cited on pages 2 and 3.)

[4] TT Soong and BF Spencer Jr. Supplemental energy dissipation: state-of-the-art and state-of-the-practice. *Engineering structures*, 24(3):243–259, 2002. (Cited on page 2.)

[5] CC Chang. Mass dampers and their optimal designs for building vibration control. *Engineering Structures*, 21(5):454–463, 1999. (Cited on page 2.)

[6] H Yoshioka, JC Ramallo, and BF Spencer Jr. âsmartâ base isolation strategies employing magnetorheological dampers. *Journal of engineering mechanics*, 128(5):540–551, 2002. (Cited on pages 3, 12, 13 and 125.)

[7] Osman E Ozbulut, Maryam Bitaraf, and Stefan Hurlebaus. Adaptive control of base-isolated structures against near-field earthquakes using variable friction dampers. *Engineering Structures*, 33(12):3143–3154, 2011. (Cited on page 3.)

[8] Hermann Frahm. Device for damping vibrations of bodies., April 18 1911. US Patent 989,958. (Cited on pages 3 and 133.)

[9] Kefu Liu and Gianmarc Coppola. Optimal design of damped dynamic vibration absorber for damped primary systems. *Transactions of the Canadian Society for Mechanical Engineering*, 34(1):119–135, 2010. (Cited on page 3.)

[10] Quan Li, Jiansheng Fan, Jianguo Nie, Quanwang Li, and Yu Chen. Crowd-induced random vibration of footbridge and vibration control using multiple tuned mass dampers. *Journal of Sound and Vibration*, 329(19):4068–4092, 2010. (Cited on page 3.)

[11] Giuseppe Carlo Marano, Rita Greco, and Sara Sgobba. A comparison between different robust optimum design approaches: application to tuned mass dampers. *Probabilistic Engineering Mechanics*, 25(1):108–118, 2010. (Cited on page 3.)

[12] Giuseppe Carlo Marano, Rita Greco, Francesco Trentadue, and Bernardino Chiaia. Constrained reliability-based optimization of linear tuned mass dampers for seismic control. *International Journal of Solids and Structures*, 44(22-23):7370–7388, 2007. (Cited on page 3.)

[13] Giuseppe Carlo Marano, Sara Sgobba, Rita Greco, and Mauro Mezzina. Robust optimum design of tuned mass dampers devices in random vibrations mitigation. *Journal of Sound and Vibration*, 313(3-5):472–492, 2008. (Cited on page 3.)

[14] Mariantonietta Morga and Giuseppe C Marano. Optimization criteria of tmd to reduce vibrations generated by the wind in a slender structure. *Journal of Vibration and Control*, 20(16):2404–2416, 2014. (Cited on page 3.)

[15] Omer F Tigli. Optimum vibration absorber (tuned mass damper) design for linear damped systems subjected to random loads. *Journal of Sound and Vibration*, 331(13):3035–3049, 2012. (Cited on page 3.)

[16] H Yu, F Gillot, and M Ichchou. Reliability based robust design optimization for tuned mass damper in passive vibration control of deterministic/uncertain

structures. *Journal of Sound and Vibration*, 332(9):2222–2238, 2013. (Cited on pages 3 and 88.)

[17] William L Oberkampf, Jon C Helton, Cliff A Joslyn, Steven F Wojtkiewicz, and Scott Ferson. Challenge problems: uncertainty in system response given uncertain parameters. *Reliability Engineering & System Safety*, 85(1-3):11–19, 2004. (Cited on pages 4 and 155.)

[18] Li-Ping He, Hong-Zhong Huang, Li Du, Xu-Dong Zhang, and Qiang Miao. A review of possibilistic approaches to reliability analysis and optimization in engineering design. In *International Conference on Human-Computer Interaction*, pages 1075–1084. Springer, 2007. (Cited on pages 4 and 155.)

[19] Abhijit Dasgupta and Michael Pecht. Material failure mechanisms and damage models. *IEEE Transactions on Reliability*, 40(5):531–536, 1991. (Cited on page 5.)

[20] Wallace R Blischke and DN Prabhakar Murthy. *Reliability: modeling, prediction, and optimization*, volume 767. John Wiley & Sons, 2011. (Cited on page 5.)

[21] Stephen H Crandall and William D Mark. *Random vibration in mechanical systems*. Academic Press, 2014. (Cited on page 5.)

[22] P Thoft-Cristensen and Michael J Baker. *Structural reliability theory and its applications*. Springer Science & Business Media, 2012. (Cited on page 6.)

[23] Jay Martin and Timothy Simpson. A monte carlo method for reliability-based design opitmization. In *47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference 14th AIAA/ASME/AHS Adaptive Structures Conference 7th*, page 2146, 2006. (Cited on pages 8 and 44.)

[24] Robert E Melchers and André T Beck. *Structural reliability analysis and prediction*. John Wiley & Sons, 2018. (Cited on pages 9, 21, 51 and 155.)

[25] Jorge Eduardo Hurtado. *Structural reliability: statistical learning perspectives*, volume 17. Springer Science & Business Media, 2013. (Cited on pages 10, 35 and 155.)

[26] Dhanesh Padmanabhan, Harish Agarwal, John E Renaud, and Stephen M Batill. A study using monte carlo simulation for failure probability calculation in reliability-based optimization. *Optimization and Engineering*, 7(3):297–316, 2006. (Cited on pages 10 and 80.)

[27] Manolis Papadrakakis and Nikos D Lagaros. Reliability-based structural optimization using neural networks and monte carlo simulation. *Computer methods in applied mechanics and engineering*, 191(32):3491–3507, 2002. (Cited on page 10.)

[28] P Marek, J Brozzetti, M Gustar, and I Elishakoff. Probabilistic assessment of structures using monte carlo simulations, 2002. (Cited on page 10.)

[29] Raymond H Myers, Douglas C Montgomery, and Christine M Anderson-Cook. *Response surface methodology: process and product optimization using designed experiments*. John Wiley & Sons, 2016. (Cited on page 11.)

[30] SM Wong, RE Hobbs, and C Onof. An adaptive response surface method for reliability analysis of structures with multiple loading sequences. *Structural safety*, 27(4):287–308, 2005. (Cited on page 11.)

[31] Byeng D Youn and Kyung K Choi. A new response surface methodology for reliability-based design optimization. *Computers & structures*, 82(2-3):241–256, 2004. (Cited on pages 11 and 32.)

[32] Norbert Wiener. The homogeneous chaos. *American Journal of Mathematics*, 60(4):897–936, 1938. (Cited on pages 12 and 33.)

[33] Shigehiro Sakamoto and Roger Ghanem. Polynomial chaos decomposition for the simulation of non-gaussian nonstationary stochastic processes. *Journal of engineering mechanics*, 128(2):190–201, 2002. (Cited on page 12.)

[34] B Gaspar, AP Teixeira, and C Guedes Soares. Assessment of the efficiency of kriging surrogate models for structural reliability analysis. *Probabilistic Engineering Mechanics*, 37:24–34, 2014. (Cited on pages 12 and 68.)

[35] Małgorzata Kutyłowska. Neural network approach for failure rate prediction. *Engineering Failure Analysis*, 47:41–48, 2015. (Cited on pages 12 and 80.)

[36] Ahmed Z Al-Garni and Ahmad Jamal. Artificial neural network application of modeling failure rate for boeing 737 tires. *Quality and Reliability Engineering International*, 27(2):209–219, 2011. (Cited on page 12.)

[37] Wei-Chiang Hong and Ping-Feng Pai. Predicting engine reliability by support vector machines. *The International Journal of Advanced Manufacturing Technology*, 28(1-2):154–161, 2006. (Cited on pages 12 and 80.)

[38] Hong-shuang Li, Zhen-zhou Lü, and Zhu-feng Yue. Support vector machine for structural reliability analysis. *Applied Mathematics and Mechanics*, 27(10):1295–1303, 2006. (Cited on page 12.)

[39] Zhiwei Guo and Guangchen Bai. Application of least squares support vector machine for regression to reliability analysis. *Chinese Journal of Aeronautics*, 22(2):160–166, 2009. (Cited on page 12.)

[40] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine learning*, 63(1):3–42, 2006. (Cited on pages 13 and 125.)

[41] David H Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992. (Cited on pages 13, 120 and 139.)

[42] Leo Breiman. Stacked regressions. *Machine learning*, 24(1):49–64, 1996. (Cited on pages 13 and 120.)

[43] Bruno Sudret. Uncertainty propagation and sensitivity analysis in mechanical models–contributions to structural reliability and stochastic spectral methods. *Habilitationa diriger des recherches, Université Blaise Pascal, Clermont-Ferrand, France*, page 18, 2007. (Cited on pages 18, 23, 48, 51, 155 and 156.)

[44] Ove Ditlevsen and Henrik O Madsen. *Structural reliability methods*, volume 178. Wiley New York, 1996. (Cited on pages 21, 25 and 64.)

[45] de Maurice Lemaire. Fiabilité des structures: Couplage mécano-fiabiliste statique. *European Journal of Computational Mechanics/Revue Européenne de Mécanique Numérique*, 15(7-8):989–992, 2006. (Cited on page 21.)

[46] Efstratios Nikolaidis, Dan M Ghiocel, and Suren Singhal. *Engineering design reliability handbook*. CRC Press, 2004. (Cited on pages 22 and 155.)

[47] Bilal M Ayyub and Richard H McCuen. *Probability, statistics, and reliability for engineers and scientists*. CRC press, 2016. (Cited on pages 22 and 155.)

[48] S Engelund and R Rackwitz. A benchmark study on importance sampling techniques in structural reliability. *Structural safety*, 12(4):255–276, 1993. (Cited on pages 22 and 155.)

[49] Norbert Kuschel and Rüdiger Rackwitz. Time-variant reliability-based structural optimization using sorm. *Optimization*, 47(3-4):349–368, 2000. (Cited on pages 24 and 34.)

[50] K Breitung. Asymptotic approximations for the outcrossing rates of stationary gaussian vector processes, university of lund, dept. of math. *Statistics*, 1984. (Cited on pages 24, 25 and 34.)

[51] Michael Hohenbichler, Stephan Gollwitzer, W Kruse, and Rüdiger Rackwitz. New light on first-and second-order reliability methods. *Structural safety*, 4(4):267–284, 1987. (Cited on pages 25 and 102.)

[52] Michael D Dettinger and John L Wilson. First order analysis of uncertainty in numerical models of groundwater flow part: 1. mathematical development. *Water Resources Research*, 17(1):149–161, 1981. (Cited on page 26.)

[53] Sankaran Mahadevan and A Haldar. Probability, reliability and statistical method in engineering design. *John Wiley & Sons*, 2000. (Cited on page 26.)

[54] Milík Tichý. First-order third-moment reliability method. *Structural Safety*, 16(3):189–200, 1994. (Cited on page 26.)

[55] Yan-Gang Zhao and Tetsuro Ono. Moment methods for structural reliability. *Structural safety*, 23(1):47–75, 2001. (Cited on page 26.)

[56] Reuven Y Rubinstein and Dirk P Kroese. *Simulation and the Monte Carlo method*, volume 10. John Wiley & Sons, 2016. (Cited on page 27.)

[57] RE Melchers. Importance sampling in structural systems. *Structural safety*, 6(1):3–10, 1989. (Cited on pages 29 and 30.)

[58] Siu-Kui Au and James L Beck. Estimation of small failure probabilities in high dimensions by subset simulation. *Probabilistic engineering mechanics*, 16(4):263–277, 2001. (Cited on pages 29, 31 and 33.)

[59] Sang Hoon Lee and Byung Man Kwak. Response surface augmented moment method for efficient reliability analysis. *Structural safety*, 28(3):261–272, 2006. (Cited on page 32.)

[60] Roger G Ghanem and Pol D Spanos. *Stochastic finite elements: a spectral approach*. Courier Corporation, 2003. (Cited on pages 33, 51 and 58.)

[61] Chun-Ching Li and Armen Der Kiureghian. Mean out-crossing rate of non-linear response to stochastic input. *Proceedings of ICASP-7, Balkema, Rotterdam*, pages 295–302, 1995. (Cited on page 34.)

[62] C Andrieu-Renaud, Bruno Sudret, and Maurice Lemaire. The phi2 method: a way to compute time-variant reliability. *Reliability Engineering & System Safety*, 84(1):75–86, 2004. (Cited on page 34.)

[63] Geoffrey Grimmett, Geoffrey R Grimmett, David Stirzaker, et al. *Probability and random processes*. Oxford university press, 2001. (Cited on page 34.)

[64] Junho Song and Armen Der Kiureghian. Joint first-passage probability and reliability of systems under stochastic excitation. *Journal of Engineering Mechanics*, 132(1):65–77, 2006. (Cited on page 35.)

[65] Herbert Martins Gomes and Armando Miguel Awruch. Comparison of response surface and neural network with other methods for structural reliability analysis. *Structural safety*, 26(1):49–67, 2004. (Cited on page 35.)

[66] Jorge E Hurtado and Diego A Alvarez. Neural-network-based reliability analysis: a comparative study. *Computer methods in applied mechanics and engineering*, 191(1-2):113–132, 2001. (Cited on page 35.)

[67] Jian Deng, Desheng Gu, Xibing Li, and Zhong Qi Yue. Structural reliability analysis for implicit performance functions using artificial neural network. *Structural Safety*, 27(1):25–48, 2005. (Cited on page 35.)

[68] Jorge E Hurtado. Filtered importance sampling with support vector margin: a powerful method for structural reliability analysis. *Structural Safety*, 29(1):2–15, 2007. (Cited on pages 35 and 80.)

[69] Richard O Duda, Peter E Hart, and David G Stork. Pattern classification 2nd edition. *New York, USA: John Wiley&Sons*, 2001. (Cited on page 35.)

[70] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995. (Cited on page 36.)

[71] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001. (Cited on pages 36, 44 and 140.)

[72] Raúl Rojas. *Neural networks: a systematic introduction*. Springer Science & Business Media, 2013. (Cited on page 39.)

[73] J. Ross Quinlan. Simplifying decision trees. *International journal of man-machine studies*, 27(3):221–234, 1987. (Cited on page 44.)

[74] C Lataniotis, S Marelli, and B Sudret. Uqlab user manual–the model module. 2015. (Cited on page 49.)

[75] Bruno Sudret and Armen Der Kiureghian. *Stochastic finite element methods and reliability: a state-of-the-art report*. Department of Civil and Environmental Engineering, University of California â, 2000. (Cited on page 57.)

[76] A Der Kiureghian. The geometry of random vibrations and solutions by form and sorm. *Probabilistic Engineering Mechanics*, 15(1):81–90, 2000. (Cited on pages 62, 64, 65 and 156.)

[77] SK Au and James L Beck. First excursion probabilities for linear systems by very efficient importance sampling. *Probabilistic engineering mechanics*, 16(3):193–207, 2001. (Cited on pages 66, 67, 68, 71, 87 and 159.)

[78] George L Ang, Alfredo H-S Ang, and Wilson H Tang. Optimal importance-sampling density estimator. *Journal of engineering mechanics*, 118(6):1146–1163, 1992. (Cited on page 68.)

[79] Muhammad NS Hadi and Yoyong Arfiadi. Optimum design of absorber for mdof structures. *Journal of Structural Engineering*, 124(11):1272–1280, 1998. (Cited on page 69.)

[80] Chien-Liang Lee, Yung-Tsang Chen, Lap-Loi Chung, and Yen-Po Wang. Optimal design theories and applications of tuned mass dampers. *Engineering structures*, 28(1):43–53, 2006. (Cited on page 69.)

[81] Elyes Mrabet, Mohamed Guedri, Mohamed Ichchou, and Samir Ghanmi. New approaches in reliability based optimization of tuned mass damper in presence of uncertain bounded parameters. *Journal of Sound and Vibration*, 355:93–116, 2015. (Cited on pages 69, 70, 108, 109, 110, 156 and 157.)

[82] Souvik Chakraborty and Rajib Chowdhury. A semi-analytical framework for structural reliability analysis. *Computer Methods in Applied Mechanics and Engineering*, 289:475–497, 2015. (Cited on page 79.)

[83] Ali Hadidi, Bahman Farahmand Azar, and Amin Rafiee. Efficient response surface method for high-dimensional structural reliability analysis. *Structural Safety*, 68:15–27, 2017. (Cited on page 80.)

[84] Qiujing Pan and Daniel Dias. An efficient reliability method combining adaptive support vector machine and monte carlo simulation. *Structural Safety*, 67:85–95, 2017. (Cited on page 80.)

[85] Hongzhe Dai, Boyi Zhang, and Wei Wang. A multiwavelet support vector regression method for efficient reliability assessment. *Reliability Engineering & System Safety*, 136:132–139, 2015. (Cited on page 80.)

[86] Vincent Dubourg, Bruno Sudret, and Franois Deheeger. Metamodel-based importance sampling for structural reliability analysis. *Probabilistic Engineering Mechanics*, 33:47–57, 2013. (Cited on page 80.)

[87] Guofeng Xue, Hongzhe Dai, Hao Zhang, and Wei Wang. A new unbiased metamodel method for efficient reliability analysis. *Structural Safety*, 67:1–10, 2017. (Cited on page 80.)

[88] Enrico Zio. System reliability and risk analysis. In *The Monte Carlo Simulation Method for System Reliability and Risk Analysis*, pages 7–17. Springer, 2013. (Cited on page 80.)

[89] Jose E Ramirez-Marquez and David W Coit. A monte-carlo simulation approach for approximating multi-state two-terminal reliability. *Reliability Engineering & System Safety*, 87(2):253–264, 2005. (Cited on page 80.)

[90] Mosaab Daoud and Qusay H Mahmoud. Monte carlo simulation-based algorithms for estimating the reliability of mobile agent-based systems. *Journal of Network and Computer Applications*, 31(1):19–31, 2008. (Cited on page 80.)

[91] João B Cardoso, João R de Almeida, José M Dias, and Pedro G Coelho. Structural reliability analysis using monte carlo simulation and neural networks. *Advances in Engineering Software*, 39(6):505–513, 2008. (Cited on page 80.)

[92] Guoshao Su, Lifeng Peng, and Lihua Hu. A gaussian process-based dynamic surrogate model for complex engineering structural reliability analysis. *Structural Safety*, 68:97–109, 2017. (Cited on page 80.)

[93] I Nitze, U Schulthess, and H Asche. Comparison of machine learning algorithms random forest, artificial neural network and support vector machine to maximum likelihood for supervised crop type classification. *Proceedings of the 4th GEOBIA, Rio de Janeiro, Brazil*, 79:3540, 2012. (Cited on page 81.)

[94] George H John and Pat Langley. Estimating continuous distributions in bayesian classifiers. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pages 338–345. Morgan Kaufmann Publishers Inc., 1995. (Cited on page 81.)

[95] Pedro Garcia-Teodoro, Jesus Diaz-Verdejo, Gabriel Maciá-Fernández, and Enrique Vázquez. Anomaly-based network intrusion detection: Techniques, systems and challenges. *computers & security*, 28(1-2):18–28, 2009. (Cited on page 81.)

[96] Davy Janssens, Geert Wets, Tom Brijs, Koen Vanhoof, Theo Arentze, and Harry Timmermans. Integrating bayesian networks and decision trees in a sequential rule-based transportation model. *European Journal of operational research*, 175(1):16–34, 2006. (Cited on page 81.)

[97] Edwin Raczko and Bogdan Zagajewski. Comparison of support vector machine, random forest and neural network classifiers for tree species classification on airborne hyperspectral apex images. *European Journal of Remote Sensing*, 50(1):144–154, 2017. (Cited on page 81.)

[98] Hiroshi Tajimi. A statistical method of determing the maximum response of a building structure during an earthquake. In *Proc. 2nd World Conf. Earthq. Eng.*, pages 781–797, 1960. (Cited on page 85.)

[99] Ilaria Venanzi. Robust optimal design of tuned mass dampers for tall buildings with uncertain parameters. *Structural and Multidisciplinary Optimization*, 51(1):239–250, 2015. (Cited on page 86.)

[100] JB Roberts. First-passage probabilities for randomly excited systems: diffusion methods. *Probabilistic Engineering Mechanics*, 1(2):66–81, 1986. (Cited on page 86.)

[101] Theodore Kypraios, Peter Neal, and Dennis Prangle. A tutorial introduction to bayesian inference for stochastic epidemic models using approximate bayesian computation. *Mathematical biosciences*, 287:42–53, 2017. (Cited on page 88.)

[102] Min-Ling Zhang and Zhi-Hua Zhou. A k-nearest neighbor based algorithm for multi-label classification. *GrC*, 5:718–721, 2005. (Cited on page 88.)

[103] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001. (Cited on page 88.)

[104] E Mrabet, M Guedri, MN Ichchou, and S Ghanmi. Stochastic structural and reliability based optimization of tuned mass damper. *Mechanical Systems and Signal Processing*, 60:437–451, 2015. (Cited on page 95.)

[105] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001. (Cited on pages 96 and 102.)

[106] Kai Xu, Min Xie, Loon Ching Tang, and SL Ho. Application of neural networks in forecasting engine systems reliability. *Applied Soft Computing*, 2(4):255–268, 2003. (Cited on page 107.)

[107] Gerhart I Schuëller and Helmuth J Pradlwarter. Benchmark study on reliability estimation in higher dimensions of structural systems–an overview. *Structural Safety*, 29(3):167–182, 2007. (Cited on pages 108, 112, 114, 115 and 159.)

[108] SK Au, J Ching, and JL Beck. Application of subset simulation methods to reliability benchmark problems. *Structural safety*, 29(3):183–193, 2007. (Cited on page 108.)

[109] Hector A Jensen and Marcos A Valdebenito. Reliability analysis of linear dynamical systems using approximate representations of performance functions. *Structural Safety*, 29(3):222–237, 2007. (Cited on page 108.)

[110] LS Katafygiotis and SH Cheung. Application of spherical subset simulation method and auxiliary domain method on a benchmark reliability study. *Structural Safety*, 29(3):194–207, 2007. (Cited on page 108.)

[111] Lambros Katafygiotis, Torgeir Moan, and Sai Hung Cheungt. Auxiliary domain method for solving multi-objective dynamic reliability problems for nonlinear structures. *Structural Engineering and Mechanics*, 25(3):347–363, 2007. (Cited on page 108.)

[112] HJ Pradlwarter, GI Schueller, Phaedon-Stelios Koutsourelakis, and Dimos C Charmpis. Application of line sampling simulation method to reliability benchmark problems. *Structural safety*, 29(3):208–221, 2007. (Cited on page 108.)

[113] Yu-Kweng Lin. Probabilistic theory of structural dynamics(book on probabilistic theory of structural dynamics, analyzing responses of structures to random vibrations). *NEW YORK, MCGRAW-HILL BOOK CO., 1967. 366 P*, 1967. (Cited on page 114.)

[114] Bertrand Clarke. Comparing bayes model averaging and stacking when model approximation error cannot be ignored. *Journal of Machine Learning Research*, 4(Oct):683–712, 2003. (Cited on page 120.)

[115] TG Ditterrich. Machine learning research: four current direction. *Artificial Intelligence Magzine*, 4:97–136, 1997. (Cited on page 120.)

[116] Thomas G Dietterich et al. Ensemble learning. *The handbook of brain theory and neural networks*, 2:110–125, 2002. (Cited on page 121.)

[117] Charu C Aggarwal. *Data classification: algorithms and applications*. CRC press, 2014. (Cited on pages 123 and 159.)

[118] Zhi-Hua Zhou. *Ensemble methods: foundations and algorithms*. Chapman and Hall/CRC, 2012. (Cited on page 135.)