Efficient stochastic optimization algorithms for performance-based system design with application to structural control system design for floating offshore wind turbines

Gong, Fanming

2019

Gong, F. (2019). Efficient stochastic optimization algorithms for performance-based system design with application to structural control system design for floating offshore wind turbines. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/103732>

<https://doi.org/10.32657/10356/103732>

**Efficient Stochastic Optimization Algorithms for
Performance-based System Design with Application
to Structural Control System Design for Floating
Offshore Wind Turbines**

GONG FANMING

School of Civil and Environmental Engineering

2019

Efficient Stochastic Optimization Algorithms for Performance-based System Design with Application to Structural Control System Design for Floating Offshore Wind Turbines

GONG FANMING

School of Civil and Environmental Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirement for the degree of
Doctor of Philosophy

2019

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

10/06/2019

Date

A handwritten signature in black ink, appearing to read 'Gong Fanming', on a light-colored rectangular background.

Gong Fanming

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

10/06/2019

Date

Asst. Prof. Cheung Sai Hung

Authorship Attribution Statement

Please select one of the following; *delete as appropriate:

~~*(A) This thesis **does not** contain any materials from papers published in peer-reviewed journals or from papers accepted at conferences in which I am listed as an author.~~

*(B) This thesis contains material from [1] paper(s) published in the following peer-reviewed journal(s) / from papers accepted at conferences in which I am listed as an author.

Chapter 3 is published as F-M Gong, S. H. Cheung, Optimal Tuned Mass Damper for a Spar Buoy Floating Wind Turbine to Mitigate Fatigue Damage. Proceedings of the 22nd ISSAT International Conference on Reliability and Quality in Design August 4-6, 2016 - Los Angeles, California, U.S.A.

The contributions of the co-authors are as follows:

- Prof Cheung provided the initial research topic direction and edited the manuscript drafts.
- I prepared the manuscript drafts. The manuscript was revised together with Prof. Cheung.
- Prof. Cheung provided the initial idea and direction of the proposed algorithm.

- I developed the program code to realize the proposed algorithm. I discussed with Prof. Cheung on improving the efficiency of the proposed algorithm and then revised the code to perform case studies.
- I developed the statistical model of the environmental conditions based on the historical field measurements.
- I modified the source code of FAST to perform dynamic simulation of the floating wind turbine.

10/06/2019

A handwritten signature in black ink on a light-colored background. The signature is written in a cursive style and appears to read 'Gong Fanming'.

Date

Gong Fanming

Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisor Prof. Joseph CHEUNG Sai Hung for the continuous support of my Ph.D. study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D. study.

Special thanks also go to my colleagues Shao Zhe, Queen, Sahil and Liu Wangsheng. Thanks for your help in solving problem of the work station and sharing research interest with me.

Last but not the least, I would like to thank my family: my parents and my brother for supporting me spiritually throughout writing this thesis and my life in general.

Table of Contents

Statement of Originality.....	I
Supervisor Declaration Statement.....	II
Authorship Attribution Statement.....	III
Acknowledgements.....	V
Table of Contents.....	VI
Abstract.....	X
List of tables.....	XII
List of figures.....	XIII
Chapter 1 Introduction	1
1.1 Background	1
1.1.1 Challenges of floating offshore wind turbines.....	4
1.1.2 Structural control techniques and controller design.....	6
1.1.3 Challenges of stochastic design optimization of controllers for floating offshore wind turbines	8
1.2 Objectives.....	9
1.3 Organization of thesis.....	10
Chapter 2 Robust design of structural control system for floating offshore wind turbine and review on stochastic optimization algorithms	11
2.1 Introduction	11
2.2 Review on structural control for floating offshore wind turbines.....	12
2.3 Proposed robust stochastic design optimization of structural controller for floating offshore wind turbine.....	16
2.3.1 Mathematical formulation.....	16
2.3.2 Probabilistic model of Environmental conditions.....	19
2.3.3 Dynamic analysis tool for floating wind turbine	25
2.4 Review on stochastic optimization algorithms.....	32

2.4.1	Methods for probability integral estimation	32
2.4.2	Stochastic optimization algorithms	36
2.4.3	Techniques for reducing the estimation error of the objective function 40	
2.4.4	Challenges of “traditional” stochastic optimization algorithms	42
2.5	Chapter summary	43
Chapter 3 A new sampling algorithm for improving the efficiency of Stochastic Subset Optimization algorithm		45
3.1	Introduction	45
3.2	Stochastic Subset Optimization.....	46
3.2.1	Generic formula	47
3.2.2	Augmented problem.....	48
3.2.3	Investigating the sensitivity of $\pi(\boldsymbol{\varphi})$ by subsets	50
3.2.4	Framework of SSO algorithm	55
3.3	Review of existing sampling algorithms and Subset Simulation	56
3.3.1	Review of the existing sampling algorithms.....	57
3.3.2	Review of Subset Simulation.....	61
3.4	Proposed sampling algorithm.....	62
3.4.1	Acceptance-Rejection Sampling with Subset Simulation.....	62
3.5	Simultaneous Perturbation Stochastic Approximation	69
3.6	Illustrative examples	71
3.6.1	Case description	72
3.6.2	Initial parameters for the proposed algorithm.....	75
3.6.3	Results and discussion	76
3.7	Conclusions	84

Chapter 4	An efficient stochastic optimization algorithm using proposed modified version of Bayesian Sequential Partitioning	86
4.1	Introduction	86
4.2	Proposed stochastic optimization algorithm	87
4.2.1	Augmented problem.....	87
4.2.2	Bayesian Sequential Partitioning	89
4.2.3	Proposed method for the identification of the optimal Binary Partition and the optimal sub-region	92
4.2.4	Iterative approach.....	94
4.2.5	Proposed stochastic optimization algorithm	95
4.3	Implementation details of the proposed stochastic optimization algorithm 99	
4.3.1	Stochastic simulation techniques	99
4.3.2	Choice of parameters for prior densities	100
4.3.3	Copula of the joint marginal PDF $\pi(\boldsymbol{\varphi})$	101
4.4	Illustrative examples	102
4.4.1	Case description	102
4.4.2	Optimization algorithm characteristics	104
4.4.3	Results and discussion	105
4.5	Conclusions	112
Chapter 5	The proposed algorithm for simultaneously solving multiple optimization problems for a single system	114
5.1	Introduction	114
5.2	The proposed algorithm	114
5.2.1	BSP with ARS-SS for a single objective optimization.....	115
5.2.2	Proposed sampling algorithm for multiple PDFs simultaneously ...	118

5.2.3	Proposed algorithm for simultaneously solving multiple optimization problems	124
5.3	Illustrative example	125
5.3.1	Case description	125
5.3.2	Algorithm characteristics	127
5.3.3	Results and discussion	127
5.4	Conclusions	134
Chapter 6	Effects of statistical models for the environmental conditions on the system performance and optimal design	135
6.1	Introduction	135
6.2	Copula method	136
6.3	Illustrative example	138
6.4	Conclusions	144
Chapter 7	Conclusions and future works	145
7.1	Conclusions	145
7.2	Future works.....	147
References		149
Appendix A		153
Appendix B		154

Abstract

Performance-based system design optimization is always required in modern engineering applications. However, the available information about the system is never complete due to the uncertainties in the model parameters of the system, the future environmental conditions subjected to the system and the measurements or prediction errors in the system performance. This leads to a robust stochastic design optimization where all uncertainties of interest can be taken into account and the design objective function is regarded as the expectation of the performance function in the corresponding uncertain space. In practice, the objective function of such optimization problem can rarely be evaluated analytically, and its estimation may require much computational cost and involve unavoidable estimation errors. These features make robust performance-based system design challenging. This thesis aims to propose efficient algorithms to solve these stochastic optimization problems. The proposed algorithms are based on the basic idea by regarding the design variables as artificial random variables such that the sensitivity of the objective function can be explored by investigating the marginal probability density function (PDF) of the auxiliary PDF which is related to the performance function. This thesis proposes a new sampling algorithm ARS-SS (Acceptance-rejection Sampling with Subset Simulation) to reduce the computational cost required by simulating samples from the auxiliary PDF. Moreover, the BSP (Bayesian Sequential Partitioning) method, which is suitable for estimating high-dimensional PDFs, is adopted to estimate the marginal PDF based on the simulated samples and a new stochastic optimization algorithm (BSP with ARS-SS) is proposed by integrating the new sampling algorithm with BSP to solve stochastic optimization problems. This work also proposes an efficient algorithm to simultaneously solve multiple optimization problems for a single system with the computational cost which is comparable to the cost required by a single optimization problem. The effect of the statistical models for the environmental conditions on the performance of the system and the optimal design are studied in this work.

The proposed algorithms are used to identify the optimal design of structural control system for floating wind turbines. The random variables of interest include four main random variables which characterize the sea states and millions of white

Gaussian noise random variables to model the wind turbulences and stochastic waves. The dynamic response of the coupled controller and floating wind turbine system is obtained by a high-fidelity nonlinear simulator FAST-SC (Fatigue, Aerodynamics, Structures and Turbulence, Structural Control). This design optimization is also used as the case study to demonstrate the efficiency of the proposed algorithms. The optimal structural control system identified by the proposed algorithm can reduce the fatigue damage at the tower base by as much as 33% compared with the case without a control system. The case studies show the efficiency enhancement of ARS-SS, the ability and advantages of the proposed stochastic optimization algorithm in solving performance-based system design problems, the ability in saving computational cost of the proposed algorithm in solving multiple optimization problems for a single system and the effect of the statistical models on the optimal design.

List of tables

Table 3.1 Natural Frequencies of OC3-Hywind Floating Wind Turbine (Matha 2010)	73
Table 3.2 Sampling efficiency of ARS-SS for case 1	78
Table 3.3 Sampling efficiency of ARS-SS for case 2	78
Table 3.4 Results for a sample run of SSO with ARS-SS for case 1	80
Table 3.5 Results for a sample run of SSO with ARS-SS for case 2	80
Table 3.6 Comparison of SSO with ARS-SS to SPSA	83
Table 4.1 Results from a sample run of the proposed algorithm for design cases C_1	105
Table 4.2 Results from a sample run of the proposed algorithm for design cases C_2	105
Table 4.3 Comparison between the sub-optimal design (ϕ_s) identified by the proposed algorithm and the final optimum (ϕ^*) by SPSA for the two design cases	106
Table 4.4 Comparison of modified BSP with ARS-SS, SSO with ARS-SS and SPSA	111
Table 5.1 Properties of TLP and mooring lines (Matha 2010)	125
Table 5.2 Natural frequencies of TLP wind turbine (Matha 2010)	126
Table 5.3 The number of samples simulated at each sampling level	128
Table 5.4 The number of samples at each optimization level	130
Table 5.5 Computational cost of the proposed algorithm and separately solving each problem	133
Table 6.1 Results of marginal probability distributions for each random variable	139
Table 6.2 Parameters for Gaussian and T copula model	142
Table 6.3 AIC and BIC value for Gaussian copula. T copula and the traditional model	142
Table 6.4 The value of performance function by different statistical models	143
Table 6.5 The optimal design identified with different statistical models	143

List of figures

Figure 1.1 Floating offshore wind turbines.....	4
Figure 2.1 JONSWAP spectrums for different significant wave heights and peak spectral periods	21
Figure 2.2 Joint distribution showing mean wind speed U versus significant wave height H_s	24
Figure 2.3 Joint distribution showing significant wave height H_s versus spectral peak period T_p	25
Figure 2.4 Schematic of FAST Modules (Lackner and Rotea 2011)	27
Figure 2.5 TurbSim Simulation Method (Jonkman and Kilcher 2012)	28
Figure 2.6 Summary of HydroDyn Calculation Procedure (Wayman, Sclavounos et al. 2006)	29
Figure 3.1 Characteristic trend of the plot of $\ln(P(X \geq a))$ versus a	67
Figure 3.2 Illustrations of the NREL 5-MW wind turbine on the OC3-Hywind spar (Jonkman 2010)	72
Figure 3.3 Typical installation of Platform-based Tuned Mass Damper for a floating spar buoy wind turbine.....	73
Figure 3.4 Plot of $\ln(P(X \geq a))$ versus a based on samples simulated by ARS-SS for case 1	77
Figure 3.5 Simulated samples by ARS-SS (mean wind speed U versus significant wave height H_s).....	79
Figure 3.6 Simulated samples by ARS-SS (significant wave height H_s versus spectral peak period T_p)	79
Figure 3.7 Simulated samples by ARS-SS and identified subset by SSO for case 181	
Figure 3.8 Projections of the optimal subset and the initial design space onto planes of all pairs of the design variables for case 2	83
Figure 4.1 Possible further divisions of a given BP containing two sub-region.....	90
Figure 4.2 Flowchart of modified BSP with ARS-SS	97

Figure 4.3 Illustrative process of modified BSP with ARS-SS for a two-dimensional design case, the red star corresponds to the final optimum and the rectangles with red edges are the optimal sub-region identified in the corresponding iteration.....	98
Figure 4.4 Sample of wind speed and wave elevation in time domain.....	104
Figure 4.5 Simulated samples, the optimal BP identified by modified BSP, and the corresponding estimated density in the initial design space for design case C_1 ...	107
Figure 4.6 The maximum posterior probability of the BP versus the size of the BP for C_1	108
Figure 4.7 Optimal sub-region A_1^* and the initial space Φ_1 for design case C_1 ...	108
Figure 4.8 Projections of the optimal sub-region A_2^* and the initial design space Φ_2 onto planes of all pairs of the design variables for design case C_2	109
Figure 5.1 Plot of $\ln P(X_1 > a)$ versus a	128
Figure 5.2 Plot of $\ln P(X_2 > a)$ versus a	129
Figure 5.3 Optimal BSP and optimal sub-region I_{11}^* for P_1 at the first optimization level.....	130
Figure 5.4 Optimal BSP and optimal sub-region I_{12}^* for P_2 at the first optimization level.....	131
Figure 5.5 Optimal sub-region I_1^* identified at the first optimization level.....	131
Figure 5.6 Optimal BSP and optimal sub-region I_{21}^* for P_1 at the second optimization level.....	132
Figure 5.7 Optimal BSP and optimal sub-region I_{22}^* for P_2 at the second optimization level.....	132
Figure 6.1 Histogram and marginal PDF of the wind speed U	139
Figure 6.2 Histogram and marginal PDF of the significant wave height H_s	140
Figure 6.3 Histogram and marginal PDF of the spectral peak period T_p	140
Figure 6.4 Histogram and marginal PDF of the wind and wave misalignment angle θ_{Angle}	141

Figure 6.5 Optimal sub-region and optimal solution identified with different
statistical models..... 143

Chapter 1 Introduction

1.1 Background

1) Background on performance-based design optimization

Performance-based design optimization is always required in modern engineering applications. The performance of the system has to be evaluated based on the realistic modeling of external environmental conditions, the structural response, the material constitutive behavior and so on, which the system may encounter during its lifetime. However, the information about the system is never complete and there always exist uncertainties in the environmental conditions and the structural response. The future environmental conditions are not accurately known and always predicted based on a chosen model and the history measurements of the loads. The structural response is obtained by a particular structural model which is chosen by the designer to predict the behavior of the system. Uncertainties may arise because of the parameters of the chosen models. Moreover, uncertainties also exist due to the uncertain prediction error between the structural model and the realistic system responses. For performance-based engineering design, uncertainties in external environmental conditions and structural model should be explicitly accounted for the lifetime performance of the system. As is well known, a probabilistic approach can be used to reflect the incomplete information available and quantify the uncertainties involved in the performance of the system in a scientific and rational way. This provides the lifetime performance of the system with application of probability concepts.

The lifetime performance of the system is concerned with the performance of the system under some uncertain environmental conditions. The performance of the system under some specified design can be evaluated mathematically as follows. Let $\boldsymbol{\theta} = [\theta_1, \dots, \theta_n] \in \Theta \subset \mathbb{R}^n$ be a parameter vector including uncertain parameters of interest, such as model parameters for environmental loads. These uncertain parameters should lie in the corresponding space Θ which includes all possible values for each parameter. Let $\boldsymbol{\varphi} = [\varphi_1, \dots, \varphi_d] \in \Phi \subset \mathbb{R}^d$ be a d -dimensional vector

containing some controllable parameters of the system which determine the system design. This vector is also called design vector which includes all design variables. In practice, the design variables should be defined in an admissible space Φ containing all possible design. Let $p(\boldsymbol{\theta}|\boldsymbol{\varphi}):\mathbb{R}^n \mapsto [0,\infty)$ be a probability density function (PDF) quantifying the plausibility of the uncertain parameters contained in $\boldsymbol{\theta}$ under some specified design $\boldsymbol{\varphi}$. This PDF incorporates the available information about the system. To be formulated in a generic form, the performance of the system can be given as

$$H(\boldsymbol{\varphi}) = E_{\boldsymbol{\theta}}[h(\boldsymbol{\theta}, \boldsymbol{\varphi})] = \int_{\Theta} h(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(\boldsymbol{\theta}|\boldsymbol{\varphi}) d\boldsymbol{\theta} \quad (1.1)$$

where $h(\boldsymbol{\theta}, \boldsymbol{\varphi})$ represents the performance function of the system defined by $\boldsymbol{\varphi}$ under the environmental conditions specified by $\boldsymbol{\theta}$. The performance of the system defined by $\boldsymbol{\varphi}$ is given as $E_{\boldsymbol{\theta}}[h(\boldsymbol{\theta}, \boldsymbol{\varphi})]$, denoting the expectation of the performance function with respect to the PDF for $\boldsymbol{\theta}$. In realistic engineering design, the performance of the system may be considered in reliability constraints of a reliability-based design optimization problem. However, this work focuses on the stochastic design problems where the objective function is a stochastic integral as given in Equation (1.1). The stochastic optimization problem in this case can be formulated in a generic form as follows

$$\boldsymbol{\varphi}^* = \arg \min_{\boldsymbol{\varphi} \in \Phi} \int_{\Theta} h(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(\boldsymbol{\theta}|\boldsymbol{\varphi}) d\boldsymbol{\theta} \quad (1.2)$$

2) Background on floating offshore wind turbines

The installed wind capacity has seen a rapid increase in the last two decades since wind energy is an indigenous, inexhaustible and clean resource. Generating electricity from traditional fossil energy will inevitably release harmful chemical substances into the air. This may cause greenhouse effect and air pollution. Moreover, the fossil and oil resources are not renewable. Compared to traditional fossil energy, wind energy is a renewable natural resource and electricity is produced by a mechanical system without releasing any harmful substances. These

advantages make wind energy competitive and attractive. The worldwide installed wind capacity has reached 435 GW in 2015 (Musial, Butterfield et al. 2006) (<https://wwindea.org/>).

Recently more and more wind turbines are installed in offshore areas. At the early stage, wind turbines are located at windiest sites such as remote areas where the wind speed is relatively high. The electricity is transmitted to the coastal cities which may require much electricity. The transmission cost is high since these cities are far away from remote areas. Owing to the abundant offshore wind resource near the coastal cities, an alternative way is generating electricity from offshore wind energy by installing offshore wind turbines. Compared to remote areas for land-based wind turbines, offshore areas have higher wind speed and lower wind turbulence because those areas are close to wind center. Moreover, offshore areas have less space limitation on the array of offshore wind turbines and less consideration of the noise effect. Offshore wind turbines also need a much lower electricity transmission cost. Offshore wind capacity has increased rapidly in the last five years. For some coastal cities, the electricity generated from wind energy in shallow water area (the depth of which is less than 60m) (Jonkman 2007) is not enough for satisfying the demand. In such case, it is necessary to install wind turbines in deep water areas (the depth of which is more than 60m) to generate more electricity.

The feasible and economical installations for deep water areas are floating offshore wind turbines. For remote windiest sites and shallow water areas, the fixed-bottom structure can be used to support the topside nacelle. Since the height of the tower has to increase with the increasing water depth, the fixed-bottom structure is not feasible for deep water wind turbines. The increasing height of the tower will lead the tower to be very slender and vulnerable as well. As a result, more materials are required to make the tower stiff enough to withstand the large structural load. According to (Jonkman 2005), floating support structures should be used in deep water areas. Unlike fixed-bottom structures, floating support structures are connected to the seabed by several mooring lines.

In general, floating wind turbines can be divided into three types based on the source of the stabilized force. The support structure is stabilized by a combination of buoyancy force, ballast force and tension force from tensed mooring lines. The wind turbine supported by a barge platform gains stabilized force from the buoyancy force due to its relatively large water plane area. The spar type turbine is stabilized by ballast, while tension legs provide the major force to stabilize the tension leg floating wind turbine.

Figure 1.1 Floating offshore wind turbines

As shown in Figure 1.1, a typical floating wind turbine will consist of a topside nacelle, a support tower, a platform and several mooring lines. The nacelle contains the generator, gearbox and the rotor blade. The tower is used to support the nacelle and give enough space for the blade rotation. The platform will provide support force for the tower. The mooring lines allow the platform to connect to the seabed and keep the platform stable with restoring forces.

1.1.1 Challenges of floating offshore wind turbines

Compared to fixed-bottom wind turbines, floating offshore wind turbines have to withstand larger structural load due to the flexible structure and harsh

environmental conditions. The floating platform will add six more degrees of freedom to the structure. Since the platform is connected to the seabed by mooring lines, the floating wind turbine is a compliant structure. Furthermore, floating wind turbines located in deep water area suffer the effect of both the turbulent wind and the irregular wave. These factors make some components of the floating wind turbine being very vulnerable during its lifetime.

Improving the structural performance is a very urgent task for the design of floating offshore wind turbines. Because of the large structural load, floating offshore wind turbines may have a higher failure probability than fixed-bottom wind turbines. The failure of the structure may lead the normal production to shut down or even the collapse of the support structure. The maintenance cost is very high due to the difficult marine operation and the long maintenance time. Moreover, improving the structural reliability may allow us to use less material for reducing the cost.

The fatigue damage at the tower base is a critical structural performance for floating wind turbines. Fatigue damage is considered due to the cyclic stress caused by periodic waves and stochastic winds. The tower base is important because the tower base load is the coupled effect of the tower bending motion and the platform motion. Research has shown that the fore-aft fatigue damage load at the tower base is two times of that of onshore wind turbines(Jonkman 2007).

One way to reduce the structural load is to increase the damping of the problematic motion. The pitching motion of the platform is affected by the thrust force from the blades. Thus appropriate blade pitching command is able to increase the damping of the platform pitching motion. An advanced blade pitch control has been developed by Jonkman (Jonkman 2007). The blade pitch can achieve a bit of reduction on structural load with the sacrifice of additional input energy for the actuator.

An alternative method is to install a damper into the turbine to absorb energy. In civil engineering, dampers have been widely used to reduce the wind-induced motion of the bridge and the motion of tall buildings under the earthquake. A passive damper was adopted to improve the motion of the platform under wave load.

Recently there are some early researches on structural control for floating offshore wind turbines.

1.1.2 Structural control techniques and controller design

Numerous structural controllers with several different actuation strategies and controller designs have been proposed in the last three decades. Structural control techniques can be divided into three different types based on their energy requirement. The three types may be defined as passive control, active control and semi-active control.

Passive control is the simplest control system without the consumption of external energy. The motion of the passive controller is activated by the motion of the structure. As a result, the vibration of the controller will dissipate energy and reduce the motion of the structure. Due to its low cost, there has been a growing interest in the application of passive controllers for reducing the dynamic response of the structure. There are various types of passive controllers, including tuned mass dampers, tuned liquid dampers, tuned liquid column dampers, viscoelastic dampers and friction dampers.

An active control system requires a large input of external energy. It consists of a passive controller and an actuator which provides a force to the structure based on the command from the response of the structure. It may improve the performance of the structure a bit more than a passive control system, while it consumes a large amount of power for the operation of the actuator.

A semi-active control system also involves a passive controller and an actuator. Unlike the active control system, it utilizes the actuator to alter the characteristics of the passive controller to adapt to the response of the structure in real time. Compared to the active control, the semi-active control requires much less external power. Several types of semi-active controllers have been proposed for civil structures, such as controllable fluid damper, variable stiffness damper and variable friction damper.

Among these three types of structural control techniques, active control may obtain the greatest improvement of the performance of the structure under dynamic

excitations. However, it demands a significant amount of external power, which may be unrealistic for local supplies. Moreover, the cost of external power supply may exceed the benefit from the reduction of the response of the structure. Due to this limitation, this thesis focuses on the application of the passive control system for improving the structural performance of the floating wind turbine.

The controller design plays an important role in the effectiveness of the controller for reducing the response of the structure under dynamic excitations. A well-designed controller can obtain a remarkable improvement of the structural performance. On the contrary, the response of the structure may even increase by a “bad” controller. For the passive controller, a simple way to design the controller is to tune the natural frequency of the damper at the modal frequency of the structure. However, the controller design may require a selection of the modal frequency when the response to be reduced depends on several modal frequencies. Furthermore, it is not an easy task to design the damping ratio of the controller when there are space limitations of the displacements of the controller.

The optimal controller is the design which has the greatest improvement of the structural performance under realistic dynamic excitations. The external excitations are always defined by the stochastic environmental conditions, which require numerous random variables to specify the stochastic field, such as the wind turbulence and the irregular wave. The performance of the structure, such as the long-term fatigue damage and the structural reliability, is given by the expectation to account for the uncertainties in environmental conditions. The stochastic design optimization for the controller is given as Equation (1.2).

The design parameters of the controller are denoted by $\boldsymbol{\phi}$ with the corresponding design space $\boldsymbol{\Phi}$. The uncertainties in dynamic excitations are represented by $\boldsymbol{\theta}$, which are distributed according to $p(\boldsymbol{\theta})$. $h(\boldsymbol{\theta}, \boldsymbol{\phi})$ is the performance function of the structure with the passive controller designed as $\boldsymbol{\phi}$ under the dynamic excitation given by $\boldsymbol{\theta}$. The objective function $H(\boldsymbol{\phi})$ is the long-term structural performance which is to be improved by the controller.

1.1.3 Challenges of stochastic design optimization of controllers for floating offshore wind turbines

Gradient-based optimization algorithms and genetic algorithm can be used to identify the optimal solution of the stochastic optimization in Equation (1.2). However, the computation effort requirement of these algorithms is very high due to the demand of estimation of the objective function. A single estimation of the objective function may require thousands of evaluations of the performance function in the integrand (the number of evaluations may be larger when the structural failure probability is regarded as the objective function). The performance function does not have an analytic expression because of the complexity and nonlinearity of the structural motions. Each evaluation of the performance function requires one run of the dynamic simulation of the structure under the given external excitation. For the floating offshore wind turbine considered in this thesis, the computational time for a 10-min dynamic simulation of the structure is almost the same as the simulated time, 10 minutes. Both gradient-based algorithms and genetic algorithms require numerous estimations (of the order of thousands) of the objective function to obtain the optimal solutions as they both need to estimate the value of the objective function. The failure probability function approach proposed by Ching (Ching and Hsieh 2007) and Jensen (Jensen, Valdebenito et al. 2008) can be used to solve the optimization problem where the objective function is the failure probability function while the optimization considered in this work is not limited to failure probability function. Therefore, the optimal solution may be identified based on millions of dynamic simulations of the structure, which requires 10,000,000 minutes (about 7,000 days) for the case of floating offshore wind turbines. Due to the extremely high computational time requirement, neither the gradient-based algorithm nor genetic algorithm is applicable for solving the stochastic design optimization of the controller for the floating offshore wind turbine.

An alternative method to solve the stochastic design problem is to use surrogate model to construct an analytical expression for the performance function. However, the surrogate model is not suitable to deal with a high-dimensional function, which is the case for the floating wind turbine that the performance function involves millions of variables accounting for the uncertainties in environmental conditions.

1.2 Objectives

The ultimate goal of this thesis is to propose algorithms to solve the performance-based system design optimization problem efficiently and the efficient algorithms are adopted to identify the optimal design of the structural control system for floating offshore wind turbines.

The first objective is to propose a sampling algorithm to reduce the number of dynamic simulations required by simulating samples from the auxiliary PDF (Probability Density Function) defined in the augmented problem. The new proposed sampling algorithm is integrated with the existing optimization algorithm SSO (Stochastic Subset Optimization) to identify the optimal design of the TMD (Tuned Mass Damper) for a spar buoy floating wind turbine.

The second objective is to propose a new stochastic optimization algorithm to efficiently identify better solutions for a performance-based design optimization problem. The newly proposed efficient sampling algorithm is used and integrated with a density estimation method which is modified to handle high-dimensional PDFs. The proposed algorithm is used to identify the optimal design of multiple TMDs for floating wind turbines and the optimal design compared to the results of SSO.

The third objective is to solve multiple performance-based design optimization problems for a single system simultaneously. The proposed algorithm saves computational cost compared to the total computational cost required by solving each optimization problem separately. The algorithm is adopted to identify the optimal design of the TMD for a tension leg platform floating wind turbine for two different system performance functions.

The fourth objective is to establish the best statistical model for the environment conditions. Copula models are compared with the existing traditional model. The effects of the statistical models on the estimated system performance and the optimal design are investigated.

1.3 Organization of thesis

In this thesis, Chapter 2 summarizes the state-of-the-art related to the topics, including methods to design the control system for wind turbines, algorithms for solving stochastic optimization problems and sampling algorithms. Chapter 2 also constructs the mathematical formula for robust stochastic design optimization of structural control system for floating offshore wind turbines with consideration of the stochastic environmental conditions. Chapter 3 proposes a new sampling approach and integrates with SSO to identify the optimal design of the TMD for a spar buoy floating wind turbine. Chapter 4 presents an efficient stochastic optimization algorithm with the modified density estimation method BSP and the algorithm is used to identify the optimal design of multiple TMDs for a spar buoy floating wind turbine. Chapter 5 proposes an algorithm to simultaneously solving multiple stochastic optimization problems and the approach is adopted to identify the optimal designs for a tension leg platform floating wind turbine for two different objective functions. Chapter 6 considers the effects of the statistical models of the environmental conditions on the performance of the system and the optimal design. Copula models are compared to the traditional conditional joint distribution method. Chapter 7 summarizes the conclusions and recommends the future works.

Chapter 2 Robust design of structural control system for floating offshore wind turbine and review on stochastic optimization algorithms

2.1 Introduction

Structural control techniques have been widely used to suppress the vibrations of structures in civil engineering. The wind-induced dynamic response of the bridge can be reduced by the passive damper. Moreover, structural control system has been also designed to mitigate the floor vibrations of the tall building under the seismic excitation. The dynamic motion of the tower can be reduced by installing a well-designed controller on the top.

Typically, structural control systems are divided into three types, including passive control system, semi-active control system and active control system. Among them, passive control system is the simplest and requires no extra energy input. The semi-active control system consists of a passive controller and an actuator to alter the parameters of the passive control to adapt to the time-varying excitation. The active control system is the most complicated system and requires extra energy to command the actuator to apply a force on the structure. This may cause stability problem of the coupled structure and controller system and require high cost on input energy and maintenance. Due to these factors, this work focuses on the passive control system.

With the experience of structural control techniques in civil engineering, this work investigates the application of structural control system to offshore floating wind turbines. The floating wind turbine located in deep water is subjected to both wind-induced and wave-induced loads. The wind turbulence leads to a broad band thrust force on the tower and topside nacelle. On the contrary, the stochastic wave spectrum is always narrow band and the wave-induced forces are applied on the floating platform. The control system can be installed in the top nacelle, the tower or the platform to improve the performance of the floating offshore wind turbine under stochastic excitations.

Although a number of studies claimed the effectiveness of passive control system for vibration reduction, the main issue seems to be their ability to identify the optimal design for the structure subjected to various stochastic excitations. The effectiveness of the passive control system relies on its optimal design. The optimal design of passive controller is affected by the dynamic characteristics of the primary structure, such as natural frequency and damping ratio. The effectiveness will also vary with the external excitation, which will affect the optimal design of the controller. Moreover, the design optimization of the control has to consider the nonlinear behavior of the structure.

Motivated by the above issues, a robust stochastic design optimization of structural control system for the floating offshore wind turbine is proposed. The objective function is regarded as the long-term structural performance of the wind turbine. In addition, various possible external excitations are considered by constructing probabilistic model to take into account the uncertainties in wind and wave loads. More importantly, the dynamic response of the coupled turbine and controller system is obtained by a high-fidelity nonlinear simulator.

2.2 Review on structural control for floating offshore wind turbines

Suppressing vibrations and reducing structural loads are urgent tasks for floating offshore wind turbines. Compared to land-based wind turbines, offshore wind turbines may encounter larger structural load due to the additional of wave load and the increasing wind speed. Unlike fixed-bottom wind turbines in shallow water, floating wind turbines in deep water use mooring lines to connect platform to the seabed to achieve stabilization and restoring force. This adds additional degrees of freedom to the wind turbine system and increases the flexibility of the structure. Research has shown the tower base bending moment of the floating wind turbine is several times larger than that of onshore wind turbine.

Recent research studies have applied passive controllers on floating offshore wind turbines for structural load reduction. These studies investigated several different types of passive controller, including tuned mass damper, tuned liquid column damper and ball absorber. The passive controllers aimed to either suppress the

motion at the tower top or reduce the pitch motion of the platform as those two modes of motion mainly contribute to the tower base bending moment.

Among all types of passive controllers studied in floating wind turbines, tuned mass damper is the richest in terms of the number of studies. The ideal tuned mass damper consists of a mass, a spring and a damper. Due to its simplicity, tuned mass damper has been well studied for vibration reduction in civil engineering. Based on the application in civil structures, researchers have designed tuned mass damper for floating wind turbines to control the problematic motions.

Lackner (Lackner and Rotea 2011) studied the application of tuned mass damper for both fixed-bottom and floating offshore wind turbines. For fixed-bottom wind turbine, such as monopile wind turbine, the tuned mass damper is tuned around the natural frequency of the first tower fore-aft mode by using the empirical optimal design of damper frequency. Then a parametric study is conducted to identify the optimal damping ratio by fixing the damper frequency. The parametric study was done by optimizing the index of the free-decay tower top motion under a specified disturbance. The turbine system was assumed to be a single degree of freedom system in the parametric study. The optimized tuned mass damper can reduce the damage equivalent load at the tower base by 10% compared to the baseline case without damper.

In (Stewart and Lackner 2013), optimal tuned mass dampers were identified for several different types of floating offshore wind turbines, including barge-type, spar buoy and tension leg platform. The optimal design was obtained by numerical optimization. The objective function is the standard deviation of the tower top displacement under free vibration. The model for the coupled turbine-damper system only considers two degrees of freedom of the turbine, which are the first tower fore-aft mode and the pitching mode of the platform. These two degrees of freedom were chosen due to their critical influence on the tower top displacement. The one-directional motion of tuned mass damper is also involved in the model. The simplified structural model includes three degrees of freedom. A constant horizontal force was applied at the tower top to represent the rotor thrust force for realistic modeling. The constraints on the tuned mass damper displacement were

took into account. The optimal design for minimizing the standard deviation was obtained by genetic algorithm.

Si (Si, Karimi et al. 2014) also developed a limited degrees-of-freedom model to obtain the dynamic response of the spar buoy floating wind turbine. To obtain a more accurate model, additional degrees of freedom were enabled when compare to Lackner's model. The surge motion and heave motion were considered since these motions are highly coupled with the pitch motion for the spar buoy platform. The parameters for the surge-heave-pitch structural model are estimated from the output of the high-fidelity nonlinear simulator. The optimal solutions were obtained for minimizing several performance indexes of the free decay response, such as the standard deviations of the tower top displacement and the platform pitch angle, the ranges of the tower top displacement and the platform pitch angle. The standard deviation and the range were considered for reducing the fatigue damage and the extreme response, respectively. These optimal solutions show quite different effectiveness on reducing the tower base fatigue damage and the effectiveness also depends on the external load.

For the tuned liquid column damper, Lackner conducted a similar process as the design optimization of tuned mass damper to identify the optimal parameters. The tuned liquid column damper was believed to be more effective on suppressing pitch motion of the platform. Barge-type, spar buoy and tension leg platform floating wind turbine were investigated in his research. The simplified model was used to obtain the free decay response of the coupled turbine-damper system. Unlike tuned mass damper, the high-fidelity nonlinear simulator currently is not able to model the floating wind turbine with a tuned liquid column damper. Therefore, the effectiveness of the tuned liquid column damper was checked by applying the estimated forces to the simplified model.

Although the previous studies on design optimization of tuned mass damper used simplified structural model in the optimization process, the effectiveness of the optimal design was checked based on the simulation results of the high-fidelity nonlinear simulator. The simulation results of the optimal design show different abilities on reducing the tower base fatigue damage for different external load

conditions. The identified optimal tuned mass damper is able to achieve 11% and 7% reduction on the tower base fore-aft fatigue damage equivalent load for 10m/s and 18m/s hub-height wind speed, respectively.

There are several limitations for previous studies. For saving computational time, the simplified or limited degrees-of-freedom model is used instead of the high-fidelity nonlinear simulator. Thus the simplified model omitted the nonlinear behavior of the system and the coupled impact from the degrees of freedom which are not considered. Moreover, the estimation error may exist since parameters of the model were estimated based on the output of a more realistic model. This may cause off-tuning of the tuned mass damper and significantly reduce the effectiveness of the damper. Although the nonlinear full-degrees-of-freedom simulation costs much time, it obtains the structural motion more accurately compared to vastly simplified model.

Furthermore, the optimization process only considered the characteristics of the free decay response. The simplified structural model with the damper is subjected to an initial disturbance to obtain the free vibration response. This neglects the impact of the external excitation on the optimal design. However, the effectiveness of the tuned mass damper depends on the frequency characteristics of the excitation. During the lifetime of the floating wind turbine, it may encounter various environmental conditions.

The last but not the least, the selection of the objective function seems to be an issue. The previous studies linked the standard deviation of the tower top displacement or the platform pitch angle with the fatigue damage at the tower base, as well the range to the extreme loads. However, there is no clear relationship between them. The optimal design will be quite different for different performance index and the effectiveness will differ as well. A more direct way is to select the structural performance which is intended to be improved as the objective function.

2.3 Proposed robust stochastic design optimization of structural controller for floating offshore wind turbine

2.3.1 Mathematical formulation

Motivated by the limitations of previous studies, an ideal way to optimize the structural controller parameters is proposed in this work. The objective function would be the long-term structural performance of the floating wind turbine with structural controller under realistic environmental conditions over its lifetime. The objective function is evaluated based on the output from the high-fidelity simulator, FAST (Fatigue, Aerodynamics, Structures and Turbulence) (Jonkman 2005), which will be introduced in the following sections. FAST simulation would pick the specified parameters of the controller to run the dynamic analysis to evaluate the objective function. For traditional optimization algorithms, the process will be repeated for convergence. Although a similar ideal method has been pointed out in Lackner's study, it was not performed due to the high time cost of FAST simulation. A 10-min FAST simulation approximately costs 10-30 minutes and the optimization scheme may require a large number of FAST simulations to obtain the optimal solution. The high computational cost leads to the use of the simplified structural model.

The objective function is regarded as the long-term structural performance of the floating wind turbine over its lifetime. Taking into account the uncertainties in environmental loads, a rigorous mathematical formulation is proposed for computing the long-term structural performance, such as fatigue damage and ultimate strength reliability. The design standard for the offshore wind turbine, IEC 61400-3 (Commission 2009), requires the design to perform fatigue damage analysis and ultimate strength analysis. Several design load cases are defined, including normal power production case, fault, start-up, shut-down and parked case. In this work, the fatigue damage evaluation only considers the normal power production case as it is the major contribution in terms of time.

The lifetime fatigue damage of the wind turbine component is estimated based on the short-term fatigue damage. It is assumed that the lifetime can be a number of equal short-term periods and the environmental condition of each short-term period

has the same probability distribution. In IEC 61400-3, the lifetime fatigue damage is suggested to be evaluated by blocking method, which is to divide the environmental variables into several bins with their corresponding probabilities. However, in this work, the expected lifetime fatigue damage is calculated more realistically as follow:

$$D(\boldsymbol{\varphi}) = E_{\boldsymbol{\theta}}[d(\boldsymbol{\theta}, \boldsymbol{\varphi})] = \frac{L}{T} \int_{\Theta} d(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(\boldsymbol{\theta}) d\boldsymbol{\theta} \quad (2.1)$$

where L is the designed lifetime and T represents the short-term period. The vector $\boldsymbol{\theta}$ includes all random variables to account the uncertain information about the system, such as uncertainties in the structural parameters and external environmental conditions. The corresponding joint probability distribution of random variables is represented by $p(\boldsymbol{\theta})$. Θ denotes the admissible spaces for the random variables. The vector $\boldsymbol{\varphi}$ denotes the design parameters of the structural controllers, such as the spring and damping constant of the tuned mass damper. $d(\boldsymbol{\theta}, \boldsymbol{\varphi})$ is the cumulated short-term fatigue damage of the wind turbine component.

The short-term fatigue damage is calculated based on the output time history stress data from FAST simulation. Miner's rule is applied with S-N curve and rainflow-counting method is used to obtain the number of cycles and the corresponding amplitude. S-N curve describes the relationship between the number of cycles for fatigue failure and the amplitude of the stress cycle.

$$N(S) = C * S^{-m}$$

$$d(\boldsymbol{\theta}, \boldsymbol{\varphi}) = \sum_{k=1}^K \frac{n(S_k)}{N(S_k)} \quad (2.2)$$

where N is the number of cycles which the material can withstand under the stress series with a constant amplitude of S . C and m are material and structural shape dependent parameters. Based on the rainflow-counting method, the number of cycles $n(S_k)$ at K different stress amplitude levels S_k .

The ultimate strength failure is critical to floating offshore wind turbines. IEC 61400-3 requires the ultimate strength analysis for several design load cases. The 1-

year and 50-year recurrence period extreme structural loads have to be estimated based on the extreme external environmental condition with 1-year and 50-year recurrence period, respectively. The method mentioned in the standard is more like an engineering or empirical way to obtain the extreme structural load. The external environmental loads have to be divided into a number of bins and the extreme structural load for a given bin is estimated by distribution fitting. The quality of the distribution fitting depends on the selected basic distribution and the used samples for each bin. The main issue is that the accuracy of the tail of the distribution is not assured, while the tail significantly affects the estimate of the extreme structural load. The ultimate strength failure probability can also be computed with the reliability framework.

$$P(F|\boldsymbol{\varphi}) = \int_{\Theta} I_F(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(\boldsymbol{\theta}) d\boldsymbol{\theta} \quad (2.3)$$

$$I_F(\boldsymbol{\theta}, \boldsymbol{\varphi}) = \begin{cases} 0, & \text{if } \max_{0 \leq t \leq T} \{S(\boldsymbol{\theta}, \boldsymbol{\varphi}, t)\} < S_0 \\ 1, & \text{otherwise} \end{cases} \quad (2.4)$$

where the vectors $\boldsymbol{\theta}$ and $\boldsymbol{\varphi}$ represent the same variables as before. $S(\boldsymbol{\theta}, \boldsymbol{\varphi}, t)$ denotes the time history stress of the turbine component from FAST and t lies between zero and the simulation period. The indicator function $I_F(\boldsymbol{\theta}, \boldsymbol{\varphi})$ would be zero if the maximum of the stress output for the short-term simulation never exceeds the threshold S_0 , which is defined as the resistance of the stress. $P(F|\boldsymbol{\varphi})$ is the ultimate strength failure probability of the floating wind turbine with a structural controller defined by $\boldsymbol{\varphi}$.

This work focuses on the fatigue damage of the critical component of the floating wind turbine, such as the tower base, the blade root and the mooring lines. The mathematical formulation for stochastic design optimization of structural controller is given as follow:

$$\boldsymbol{\varphi}^* = \arg \min_{\boldsymbol{\varphi} \in \Phi} D(\boldsymbol{\varphi}) = \arg \min_{\boldsymbol{\varphi} \in \Phi} \int_{\Theta} d(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(\boldsymbol{\theta}) d\boldsymbol{\theta} \quad (2.5)$$

Φ represents the design space of the design vector $\boldsymbol{\varphi}$. The next sections will introduce the joint probability distribution of the environmental parameters and the high-fidelity simulator FAST.

2.3.2 Probabilistic model of Environmental conditions

For land-based wind turbine, the major external load is the wind-induced load, which applies on the blade and the tower. However, despite wind-induced loads, the floating wind turbine encounters the loads caused by waves, currents and ices. The seismic load would be taken into account for some particular locations. Compared to wind-induced and wave-induced loads, the current-induced loads are relatively less important. The external environmental conditions considered in this work include the wind and wave loads.

The wind coming to the wind turbine will lead to aerodynamic forces on the turbine blades, tower and nacelle. The aerodynamic forces depend on the speed difference between the wind speed and the turbine component. The rotation and vibration of the blades are motivated by the aerodynamic force on the blades. Electric power generation relies on the rotating blades. On the other hand, there exists a horizontal thrust force caused by the blade, the value of which depends on the blade pitch angle. The rotor thrust force will lead the tower to bend and the floating platform to pitch.

The 10-min mean wind speed is often used to represent the wind condition. The random wind field can be described by the wind turbulence intensity. The mean wind speed and turbulence intensity can be obtained from site measurement and they are often measured at the height of 10 meters. To obtain the wind speed at the hub height of the wind turbine, the power-law equation can be used. This equation allows obtaining the mean wind speed at any given height based on the wind speed at reference height.

$$U(z) = U_{10} \left(\frac{z}{10} \right)^\alpha \quad (2.6)$$

where α denotes the exponential number and z represents the height.

The waves will generate hydrostatic and hydrodynamic forces on the merged part of the floating wind turbine. Hydrodynamic forces are caused by the pressure difference on the interaction surface of the floating platform. For floating offshore wind turbines, the wave may approach the platform from different directions with random wave height and length. The wave may approach the structure from various directions, which is represent by the wave direction θ_{Angle} . Typically, the elevation of the irregular wave is viewed as a non-stationary process and the 3-hour sea state is described with a wave spectrum based on the assumption of stationary process. The wave spectrum is characterized by the significant wave height H_s and the peak spectral period T_p . Pierson-Moskowitz spectrum and JONSWAP spectrum are two widely used wave spectrums. For fully developed seas, the Pierson-Moskowitz spectrum is routinely used to describe the statistical properties. The JONSWAP wave spectrum is often used in limited fetch situations. As stated in the IEC 61400-3 design standard, the one-sided JONSWAP wave spectrum is given as follows:

$$S_{\zeta}^{1-sided}(\omega) = \frac{1}{2\pi} \frac{5}{16} H_s^2 T_p \left(\frac{\omega T_p}{2\pi} \right)^{-5} \exp \left[-\frac{5}{4} \left(\frac{\omega T_p}{2\pi} \right)^4 \right] \left[1 - 0.287 \ln(\gamma) \right] \gamma^{\exp \left\{ -0.5 \left[\frac{\frac{\omega T_p}{2\pi} - 1}{\sigma(\omega)} \right]^2 \right\}} \quad (2.7)$$

where γ represents the peak shape parameter of the irregular wave condition and σ denotes the scaling factor. Based on the recommendations from the IEC 61400-3 design standard, the peak shape parameter and the scaling factor can be derived from the significant wave height and the peak spectral period as follows:

$$\gamma = \begin{cases} 5 & , \frac{T_p}{\sqrt{H_s}} \leq 3.6 \\ \exp \left(5.75 - 1.15 \frac{T_p}{\sqrt{H_s}} \right) & , 3.6 < \frac{T_p}{\sqrt{H_s}} \leq 5 \\ 1 & , \frac{T_p}{\sqrt{H_s}} > 5 \end{cases} \quad (2.8)$$

$$\sigma(\omega) = \begin{cases} 0.07 & , \omega \leq \frac{2\pi}{T_p} \\ 0.09 & , \omega > \frac{2\pi}{T_p} \end{cases} \quad (2.9)$$

In the above two equations, the units of the significant wave height and the peak spectral period are assumed to be meters and seconds, respectively. The JONSWAP spectrums of three combinations of parameters are graphically shown in Figure 2.1.

Figure 2.1 JONSWAP spectrums for different significant wave heights and peak spectral periods

The two-sided wave spectrum, as the input for FAST, is given as follow:

$$S_{\zeta}^{2-sided}(\omega) = \begin{cases} \frac{1}{2} S_{\zeta}^{1-sided}(\omega) & , \omega \geq 0 \\ -\frac{1}{2} S_{\zeta}^{1-sided}(\omega) & , \omega < 0 \end{cases} \quad (2.10)$$

The statistical properties of the environmental condition in the short-term period are described by the four parameters, the mean wind speed U , the significant wave height H_s , the peak spectral period T_p and the wave direction θ_{Angle} . As wave is generated by wind far away, there exists some correlation between the mean wind speed and the wave spectrum parameters, as well the wave direction. The joint probability distribution can be used to represent the long-term statistical properties of the mean wind speed, the significant wave height, the peak spectral period and the wave direction. Based on the measured history data from site, the parameters of the joint probability distribution can be obtained.

In this work, the floating wind turbines are assumed to be located at a site near the Shetland Islands, also specified by $61^{\circ}20'N$ latitude and $0^{\circ}0'E$ longitude. This site is selected because it has fairly harsh wind and wave conditions which are believed to lead large structural load on floating wind turbines. The history data of the reference site is downloaded from <http://waveclimate.com>. The environmental conditions are represented by the 10-min mean wind speed U at 10-m height, significant wave height H_s , spectral peak period T_p and wind-wave misalignment angle θ_{Angle} .

The joint probability density function (PDF) of the environmental parameters for the selected site is obtained by fitting the marginal distribution and conditional distribution. The joint PDF is assumed to be as follows.

$$p(U, H_s, T_p, \theta_{wave}) = p_u(U) * p_{h_s|u}(H_s|U) * p_{t_p|h_s}(T_p|H_s) * p_{\theta_{angle}|h_s}(\theta_{Angle}|H_s) \quad (2.11)$$

The 2-parameter Weibull distribution is adopted to fit the marginal PDF for U and the conditional PDF for H_s given U . The log-normal distribution and truncated Weibull distribution are used to determine parameters for $p_{t_p|h_s}(T_p|H_s)$ and $p_{\theta_{angle}|h_s}(\theta_{Angle}|H_s)$, respectively.

As we only consider the mean wind speed for the normal production case, the PDF for U has to be truncated within the range of $[U_{10m-in}, U_{10m-out}]$. The transformation between the hub-height wind speed and 10-m wind speed is achieved by the power-law equation given in Equation (2.6).

The cut-in and cut out wind speed at the hub height are $3m/s$ and $25m/s$, respectively. $p_u(U)$ is given as follow:

$$p_u(U) = \begin{cases} \frac{\beta_U}{\alpha_U} \left(\frac{U}{\alpha_U} \right)^{\beta_U-1} \exp \left(- \left(\frac{U}{\alpha_U} \right)^{\beta_U} \right) \frac{1}{F_U(U_{10m-out}) - F_U(U_{10m-in})}, & U_{10m-in} \leq U \leq U_{10m-out} \\ 0, & \text{otherwise} \end{cases} \quad (2.12)$$

where $F_u(U) = \exp \left(- \left(\frac{U}{\alpha_U} \right)^{\beta_U} \right)$, $\alpha_U = 10.5613$ and $\beta_U = 2.1774$.

Based on the history data, the conditional PDF $p_{h_s|u}(H_s|U)$ is obtained, given by the following equation.

$$p_{h_s|u}(H_s|U) = \begin{cases} \frac{\beta_{H_s}}{\alpha_{H_s}} \left(\frac{H_s}{\alpha_{H_s}} \right)^{\beta_{H_s}-1} \exp \left(- \left(\frac{H_s}{\alpha_{H_s}} \right)^{\beta_{H_s}} \right), & H_s \geq 0 \\ 0, & H_s < 0 \end{cases} \quad (2.13)$$

The parameters are defined as:

$$\begin{aligned} \alpha_{H_s}(U) &= 0.09612U^{1.344} + 1.498 \\ \beta_{H_s}(U) &= 0.6396U^{0.4723} + 0.5233 \end{aligned} \quad (2.14)$$

The log-normal conditional PDF $p_{t_p|h_s}(T_p|H_s)$ is shown as below:

$$p_{t_p|h_s}(T_p|H_s) = \frac{1}{T_p \sigma_{\ln(T_p)} \sqrt{2\pi}} \exp \left[-\frac{1}{2} \left(\frac{\ln(T_p) - \mu_{\ln(T_p)}}{\sigma_{\ln(T_p)}} \right)^2 \right] \quad (2.15)$$

The parameters are given by:

$$\begin{aligned}\mu_{\ln(T_p)}(H_s) &= 0.1074H_s^{0.7595} + 2.002 \\ \sigma_{\ln(T_p)}(H_s) &= -0.2599H_s^{0.2859} + 0.5813\end{aligned}\quad (2.16)$$

The wave direction, which denotes the misalignment angle between wind and wave direction conditional on significant wave height is distributed according to the following PDF:

$$p_{\theta_{angle}|h_s}(\theta_{Angle}|H_s) = \begin{cases} \frac{\beta_{\theta_{Angle}}}{\alpha_{\theta_{Angle}}} \left(\frac{\theta_{Angle}}{\alpha_{\theta_{Angle}}} \right)^{\beta_{\theta_{Angle}}-1} \exp\left(-\left(\frac{\theta_{Angle}}{\alpha_{\theta_{Angle}}}\right)^{\beta_{\theta_{Angle}}}\right) \frac{1}{F_{\theta_{Angle}}(360)}, & 0 < \theta_{Angle} \leq 360 \\ 0, & \text{otherwise} \end{cases} \quad (2.17)$$

The corresponding parameters are given by:

$$\begin{aligned}\alpha_{\theta_{Angle}}(H_s) &= 219.2 \exp(-0.01374H_s) - 40.36 \exp(-1.439H_s) \\ \sigma_{\theta_{Angle}}(H_s) &= 0.00658H_s^{2.814} + 2.845\end{aligned}\quad (2.18)$$

The joint probability distribution based on 10,000 samples is shown in Figure 2.2 and Figure 2.3.

Figure 2.2 Joint distribution showing mean wind speed U versus significant wave height H_s

Figure 2.3 Joint distribution showing significant wave height H_s versus spectral peak period T_p

2.3.3 Dynamic analysis tool for floating wind turbine

Among the existing simulation tools, FAST which is developed at the NREL (National Renewable Energy Laboratory) and evaluated and certificated by Germanischer Lloyd WindEnergie, is widely used by researchers to model and analyse the structural behaviour of the land-based wind turbine system. For floating offshore wind turbines, FAST is used with HydroDyn which computes the hydrodynamic forces acting at the submerged structures. FAST with HydroDyn is a fully coupled aero-hydro-servo-elastic code which considers both the effect of aerodynamics and hydrodynamics (Wayman, Sclavounos et al. 2006).

When running a simulation using FAST, the AeroDyn subroutine that uses the well-known equilibrium blade element-momentum (BEM) approach calculates the aerodynamic loads applied at the blades. The AeroDyn calculations include the effects of axial and tangential induction, tip and hub losses (using the Prandtl model) and the Beddoes-Leishman dynamic stall model. TurbSim is used to generate wind inflow which is the input of AeroDyn. TurbSim is a stochastic, full-field, turbulent-wind simulator. It uses a statistical model (as opposed to a physics-based model) to

numerically simulate time series of three-component wind-speed vectors at points in a two-dimensional vertical rectangular grid that is fixed in space.

HydroDyn subroutine which enables FAST to model the floating offshore wind turbine calculates the hydrodynamic loads acting on the platform and mooring system, including hydrostatic restoring contributions of buoyancy and water plane area, viscous drag calculated using Morison's equation, added mass and damping contributions from wave radiation, including free surface memory effects and the incident wave excitation from scattering in regular or irregular seas.

Mooring systems provide restoring forces to hold the floating platform against wind, waves and current. The restoring forces of the mooring lines are obtained by quasi-static mooring system module. The quasi-static approach assumes that the inertia and damping of the mooring system are insignificant and the mooring system compliance is inherently linear.

After the calculation of aerodynamic and hydrodynamic loads acting on the structures, FAST is used to model the behaviour of the wind turbine system. The FAST code can model the dynamic response of both two- and three-bladed, conventional, horizontal-axis wind turbines. Rotor-furling, tail-furling, and tail aerodynamic which are useful in the analysis of small wind turbines can be optionally selected in the wind turbine configuration. The FAST model is a combination of modal and multi-body dynamics formulation. For a three-bladed turbine system, the model includes seven rigid bodies (support platform, nacelle, armature, gears, hub, tail and structure furling with rotor) and five flexible bodies (tower, three blades, and drive shaft) with distributed mass and stiffness properties and has 24 DOFs (degrees of freedom). The flexibility in the blades and tower is characterized using a linear modal representation that assumes small deflections. The DOFs are further described below.

The support platform has six DOFs, including the three translational (surge, sway, and heave) and rotational (roll, pitch, and yaw) motions relative to the inertia frame. FAST models the first and second bending mode of the tower in the longitudinal and transverse directions. The tower is modelled as a cantilever beam rigidly

attached to the support platform. Another DOF accounts for the nacelle yaw motion, which can be free or fixed with a torsional yaw spring. This DOF enables the rotor to be either upwind or downwind with the rotor providing yaw loads. The next DOF is the variations generator speed and another DOF accounts for drivetrain flexibility associated with torsional motion between the generator and the hub/rotor. The teeter motion of the blades about a pin located on the hub, which is controlled by the combination of springs and dampers, is another DOF modelled by FAST. For each flexible blade, the first and second flapwise bending modes and edgewise motion are modelled. In total, nine DOFs account for the motion of the three blades. Furling of the rotor and tail about the yawing-portion of the structure atop the tower are the last two DOFs.

The control of the wind turbine can be realized by running FAST through an interface with Matlab/Simulink. The non-linear equations of motion of the fully coupled model are solved with the time-marching simulation. Figure 2.4 is a schematic detailing of the various modules in FAST.

Figure 2.4 Schematic of FAST Modules (Lackner and Rotea 2011)

The output of FAST includes the displacements, velocities, accelerations and internal load of the structure. As shown in Figure 2.4, TurbSim generates the wind inflow in the grid and HydroDyn calculates the hydrodynamic loads caused by

waves and currents. The random wind and wave are described by spectrums. TurbSim transforms the wind velocity spectrum to velocity time series at each grid point and wave spectrum is converted to time histories of wave elevation by HydroDyn. A further discussion about the transformation is shown below.

Figure 2.5 TurbSim Simulation Method (Jonkman and Kilcher 2012)

TurbSim simulates the wind velocity time series at points in the two-dimensional grid. As the spectra of velocity components and spatial coherence are defined in the frequency domain, the inverse Fourier transform is used to produce the time series with the assumption that the time series of the wind velocity follow a stationary process. The simulation method of TurbSim is shown in Figure 2.5. When using the inverse Fourier transform, a large number of random variables which follow the standard normal distribution are required. To obtain the wind speed field around the turbine blade rotating space, the space has to be divided into a number of grids. At each grid point, there exists three wind speeds in three directions. The wind speed at each simulation time step interval is required in the simulation. Thus the simulation of wind speed field for a 10-min simulation may require 20,000,000 white Gaussian noise variables. Because each point has three wind velocities in three directions, the number of random variables is three times the number of frequencies and the number of grid points. To allow the TurbSim to simulate the stochastic wind field using the specified white Gaussian noise variables rather than the Box-Muller

generation, the source code of TurbSim was modified in this work. TurbSim was recompiled to account the modification of the source code.

Figure 2.6 Summary of HydroDyn Calculation Procedure (Wayman, Sclavounos et al. 2006)

As shown in Figure 2.6, HydroDyn module contains hydrodynamic subroutines which finally generate the hydrodynamic forces, including hydrostatic restoring force, viscous damping force and added mass force. The nonlinear restoring forces from the mooring system are obtained from a quasi-static mooring line module that accounts for the elastic stretching of an array of homogenous taut or slack catenary lines with seabed interaction. For regular wave whose periodic elevation is represented as a sinusoid propagating at a single amplitude and frequency or wavelength, its kinematics is described by Airy wave theory. The irregular or random waves, which are determined by the wave spectrum and describe various stochastic sea states, are modelled as the summation or superposition of multiple wave components. The expression for the wave elevation is shown below (Jonkman 2009).

$$\xi(t)|_{t=n\Delta t} = \frac{1}{N} \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} W[k] \sqrt{\frac{2\pi}{\Delta t} S_{\xi}^{2-Sided}(\omega)|_{\omega=k\Delta\omega}} e^{j\frac{2\pi kn}{N}}, n = 0, 1, 2, \dots, N-1$$

(2.19)

This equation is the inverse Fourier transform of the wave spectrum, where j is the imaginary number, n is the discrete-time-step counter, k is the discrete-frequency-step counter and N is the number of discrete steps, which is related to the time step and total analysis time and is equal to the total analysis time divided by the time step. $S_{\xi}^{2-Sided}$ represents the two-sided power spectral density of the wave elevation per unit time, which depends on the incident waves. $W[k]$ is the discrete Fourier transform of a realization of a white Gaussian noise discrete-time-series process with a zero mean and a unit variance. HydroDyn uses Box-Muller method to realize the white Gaussian noise process.

The source code of hydrodynamic subroutine is modified in this work such that the random numbers or variables which are used to generate the white Gaussian noise process can be controlled. In total, each simulation of the wave elevation requires N random variables which follow the standard normal distribution.

The structural response of floating offshore wind turbine is obtained by running FAST with TurbSim and HydroDyn. Each simulation calculates the response of one combination of mean wind speed, significant wave height, peak spectral period, wave direction, wind random variables and wave random variables.

In order to use structural control techniques in floating offshore wind turbines, Matthew developed a new simulation tool, named FAST-SC (FAST, Structural Control), by modifying the aero-elastic code FAST (Lackner and Rotea 2011). The passive mass damper is modelled as a single DOF rigid body, whose equation of motion is added to the original equations. By solving the equations of motion of the floating offshore wind turbine system coupled with mass dampers, the structural response is obtained. FAST-SC is able to simulate two mass dampers located at the nacelle or platform in two different directions, axial and side-side direction.

The newest version (version 8) of FAST (FASTv8), which is released in 2016, also allows simulating the dynamic response of the coupled turbine-damper system. FASTv8 enables the nacelle-based and tower-based tuned mass damper. The tuned mass damper is allowed to move either in a single direction or unidirectional. Moreover, the nonlinear spring can be specified by the user-defined input file. In addition to passive tuned mass damper, FASTv8 is able to model the semi-active control, such as velocity-based damping control, inverse velocity-based damping control and displacement-based damping control.

To sum up, in this work the uncertainties for the floating wind turbine include the wind spectrum parameters, wave spectrum parameters and the white Gaussian noise variables. The wind spectrum is described by the mean wind speed U and the turbulence intensity. The significant wave height H_s and spectral peak period T_p are used to construct the wave spectrum. The misalignment between wind and wave direction is denoted by the wave direction θ_{Angle} . The white Gaussian noise random variables θ_w are used to generate the time series of wind speed and wave elevation from the wind and wave spectrum, respectively. The number of the white Gaussian noise variables depends on the time step of the dynamic analysis. Taking all these uncertainties into account, the robust stochastic design optimization of the structural controller for floating offshore wind turbines can be rewritten in a generic form as follows:

$$\boldsymbol{\varphi}^* = \arg \min_{\boldsymbol{\varphi} \in \Phi} \int_{\Theta} h(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(\boldsymbol{\theta}) d\boldsymbol{\theta} \quad (2.20)$$

and

$$\boldsymbol{\theta} = [U, H_s, T_p, \theta_{Angle}, \boldsymbol{\theta}_w]^T \quad (2.21)$$

It can be seen that the objective function is a high-dimensional integral which includes four short-term environmental variables and millions of white noise Gaussian variables. Solving this proposed optimization problem is not an easy task especially when the dynamic analysis required by the evaluation of the short-term performance is time-consuming. The following chapters aim to propose

computationally efficient algorithm to solve the proposed optimization problem (may include stochastic optimization of multiple objectives) and identify the optimal design of the structural controller as well.

2.4 Review on stochastic optimization algorithms

To identify the optimal solution of the proposed problem as given in Equation (2.20), the evaluation of the objective function is inevitable for both gradient-based and gradient-free optimization algorithms. This section gives a review on methods for the probability integral estimation, followed by the review on stochastic optimization algorithms.

2.4.1 Methods for probability integral estimation

This section provides an investigation on the existing mathematical methods for the estimation of the probability integral. As the objective function is evaluated at some given design, the design vector $\boldsymbol{\varphi}$ is fixed in the equations of the following methods.

1) Perturbation approach

The perturbation approach, which is widely used to estimate the mean of an arbitrary function $h(\boldsymbol{\theta}, \boldsymbol{\varphi})$, uses the second-order Taylor series expansion of the function about $\bar{\boldsymbol{\theta}}$, where the bar represents the mean of the random variables $\boldsymbol{\theta}$. The estimate of the perturbation approach is given as follow

$$\hat{E}_{\boldsymbol{\theta}}[h(\boldsymbol{\theta}, \boldsymbol{\varphi})] = h(\bar{\boldsymbol{\theta}}, \boldsymbol{\varphi}) + \frac{1}{2} \sum_{j=1}^J \frac{\partial^2 h(\bar{\boldsymbol{\theta}}, \boldsymbol{\varphi})}{\partial \theta_j^2} \text{var}(\theta_j) + \sum_{j=1}^J \sum_{k>j}^J \frac{\partial^2 h(\bar{\boldsymbol{\theta}}, \boldsymbol{\varphi})}{\partial \theta_j \partial \theta_k} \text{cov}(\theta_j, \theta_k) \quad (2.22)$$

where $\text{var}(\cdot)$ and $\text{cov}(\cdot)$ denote the variance and the covariance, respectively. The dimension of the random vector $\boldsymbol{\theta}$ is J and θ_j is the j th component of $\boldsymbol{\theta}$. The central difference scheme is implemented to numerically estimate the derivatives in the equation. The step sizes of the variables should be chosen appropriately because the smaller step sizes will minimize the discretization errors at the expense of amplifying the round-off errors. Thus a larger step size is preferred in a general case. The estimate can be generalized to the higher order.

2) Asymptotic approximation

The asymptotic approximation method evaluates the probability integrals by expanding the natural logarithm of the integrand $J(\boldsymbol{\theta}, \boldsymbol{\varphi}) = h(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(\boldsymbol{\theta})$ about its local maximum point. If $J(\boldsymbol{\theta}, \boldsymbol{\varphi})$ has one local maximum at $\boldsymbol{\theta} = \boldsymbol{\theta}^*$, a second-order asymptotic approximation is written as

$$\hat{E}_{\boldsymbol{\theta}}[h(\boldsymbol{\theta}, \boldsymbol{\varphi})] = \frac{(2\pi)^{J/2} J(\boldsymbol{\theta}^*, \boldsymbol{\varphi})}{\sqrt{\det[H(\boldsymbol{\theta}^*, \boldsymbol{\varphi})]}} \quad (2.23)$$

where $H(\boldsymbol{\theta}^*, \boldsymbol{\varphi})$ is the positive definite Hessian matrix of $V(\boldsymbol{\theta}, \boldsymbol{\varphi}) = -\ln J(\boldsymbol{\theta}, \boldsymbol{\varphi})$.

For cases that have multiple peaks, the estimate is applied at each peak first and the final estimate is the sum of the contribution from all maximum points. This asymptotic approximation requires solving an optimization problem to identify all the local maximum point of the integrand in the first step.

3) Direct numerical integration

Direct numerical integration evaluates the integral by meshing the integral space into a number of small sub-regions. The upper and lower limits are determined such that the truncated regions have negligible contribution to the integral, while the mesh size should be chosen appropriately to ensure that it captures the peaks and the important local fluctuations. The computational cost of numerical integration will increase exponentially with the dimension of the random variables due to the mesh of the integration space.

4) Blocking method

The blocking method is a long-term fatigue analysis method recommended by the industrial standard. This method is not a general approach for integral evaluation. The joint probability density function is first partitioned into a number of representative blocks. Then the probability of each block is calculated and a representative combination of random variables located in each block is selected to evaluate the probability integral. The dimension of the random variables affects the

number of the blocks, thus the computational cost will grow with the dimension. Another disadvantage is that there is no consensus for the choice of the representative combination for each block.

5) Monte Carlo estimation

Monte Carlo simulation can be used to estimate the probability integral when it is able to obtain samples from the underlying distribution $p(\boldsymbol{\theta})$. The Monte Carlo estimator is given as follow

$$\hat{E}_{\boldsymbol{\theta}}[h(\boldsymbol{\theta}, \boldsymbol{\varphi})] = \frac{1}{N} \sum_{i=1}^N h(\boldsymbol{\theta}^{(i)}, \boldsymbol{\varphi}) \quad (2.24)$$

where N is the number of simulated samples that are distributed according to $p(\boldsymbol{\theta})$ and $\boldsymbol{\theta}^{(i)}$ represents the i th sample. Based on the knowledge of Strong Law of Large Numbers, this estimator is more accurate as the number of samples increases and will approach the true value $E_{\boldsymbol{\theta}}[h(\boldsymbol{\theta}, \boldsymbol{\varphi})]$ as N approaches infinity. The quality of the Monte Carlo estimator is judged by the coefficient of variation (CoV), which is given as follow

$$CoV = \frac{std[h(\boldsymbol{\theta}, \boldsymbol{\varphi})]}{\sqrt{N} E_{\boldsymbol{\theta}}[h(\boldsymbol{\theta}, \boldsymbol{\varphi})]} \quad (2.25)$$

where $std[h(\boldsymbol{\theta}, \boldsymbol{\varphi})]$ is the standard deviation of the performance function for all simulated samples. As shown in Equation (2.25), the CoV will decrease when the number of simulated samples increases. A large number of samples will lead to a small estimation error, although N will be very large for some cases, such as the estimation of the failure probability which is often small. The main advantage of Monte Carlo estimation is that the computational cost will not grow significantly with the dimension of the random variables.

6) Importance sampling

The importance sampling method uses a proposed importance sampling density $g(\boldsymbol{\theta})$ with the aim to reduce the CoV of the estimation. The estimator for importance sampling is given as follow

$$\hat{E}_{\boldsymbol{\theta}}[h(\boldsymbol{\theta}, \boldsymbol{\varphi})] = \frac{1}{N} \sum_{i=1}^N \frac{h(\boldsymbol{\theta}^{(i)}, \boldsymbol{\varphi}) p(\boldsymbol{\theta}^{(i)})}{g(\boldsymbol{\theta}^{(i)})} \quad (2.26)$$

where $\{\boldsymbol{\theta}^{(i)}\}_{i=1:N}$ are samples that are simulated from the proposed density $g(\boldsymbol{\theta})$. To avoid the variance of the importance sampling estimator going to infinite, it is imperative that the support for $g(\boldsymbol{\theta})$ exceeds the support for $p(\boldsymbol{\theta})$. By the Strong Law of Large Numbers, the importance sampling estimator will converge to the true value $E_{\boldsymbol{\theta}}[h(\boldsymbol{\theta}, \boldsymbol{\varphi})]$ as N approaches infinity. The quality of this estimator depends on the choice of the proposed density $g(\boldsymbol{\theta})$, which requires good prior information about the behavior of the integrand in the random space. An appropriate importance sampling density allows the sampler to obtain more samples in regions that give significant contribution to the integral, and thus reduces the CoV of the estimator. The importance sampling estimator will be the same as the Monte Carlo estimator when the importance sampling density $g(\boldsymbol{\theta})$ is chosen as the initial probability density function $p(\boldsymbol{\theta})$.

For the proposed stochastic design optimization problem as given in Equation (2.20), the dimension of the random variables is large since a large number of white noise Gaussian random variables are required for transforming the load spectrum to time series. Thus, among the methods introduced previously, only the Monte Carlo simulation and importance sampling estimation are suitable to evaluate the probability integral, which is the objective function in Equation (2.20). The Monte Carlo estimation will be used when the prior information about the integrand is not available.

2.4.2 Stochastic optimization algorithms

Based on the estimation of the objective function, the stochastic optimization problem may be solved by two types of stochastic optimization algorithms, including gradient-free and gradient-based algorithm.

1) Gradient-free optimization algorithm

Gradient-free optimization algorithms, including blind random search, localized random search and genetic algorithm, are based on comparisons of the design choices that are distributed in large regions of the design space. These methods only require the information about the objective function, instead of obtaining the derivative information.

Blind random search

Blind random search is the simplest random search method, where the chosen of the current sample $\boldsymbol{\varphi}_k$ is not affected by the previous samples. The search process is called blind because the sampling strategy for the current sample does not take into account the information of the previous samples generated in the search process. The current independent sample is simulated from a chosen probability distribution, which covers all possible design choices. The search process is terminated when the specified stopping criterion is satisfied and the obtained design is believed to be the optimal design. This method is able to deal with problems where the design vector $\boldsymbol{\varphi}$ has continuous, discrete, or hybrid elements. For cases that the dimension of the design vectors is low, the blind random search may be an appropriate method. However, this method is computationally expensive when the number of the design variables is large or even moderate. This is because the size of the design space will increase exponentially as the dimension of the design vector increases.

Localized random search

Unlike the blind random search method, the localized random search adopts the information of the previous samples generated in the search process to choose the current sample. The idea of the localized random search is that the current sample is generated in the neighborhood of the estimate which is the best estimate of the

previous samples. This method is called localized because the current sample is generated in the localized region of the previous sample. In this way, the search process will terminate when the appropriate stopping criterion is satisfied.

Genetic algorithm

Genetic Algorithm (GA) is an adaptive heuristic search algorithm on the basis of the evolutionary property of natural selection and genetics. GA has been widely used to obtain the global solution of the optimization problems in the random search way. Although randomized, GA adopts the historical information to direct the search to the region of better performance in the original design space. The search process of GA will mimic the process in natural systems necessary for evolution, which follows the principle of “survival of the fittest”. The previous random search starts with one single design and moves toward the optimum by simulating new samples in an iterative way, while GA starts with a population of design choices. This population is regarded as seeds to produce a new population of better performance based on the techniques of crossover and mutation. The parents are selected according to the principle of survival of the fittest, which is based on the value of the objective function for the design choices in the population. The crossover represents mating between parents and mutation will introduce random modifications for the offspring generated by the crossover. Elitism is used to ensure that at least one best solution is copied without changes to the new population such that the best solution found can survive to the end of the run. GA allows us to obtain the global solution rather than the local optimum by exploring the design space.

2) Gradient-based algorithms

For the gradient-based algorithms, it is necessary to obtain the derivative information of the objective function and the steepest descent is identified. Based on the derivatives of the objective function, the unconstrained optimization problem in Equation (2.20) is solved by the following generic form:

$$\boldsymbol{\varphi}_{k+1} = \boldsymbol{\varphi}_k - \alpha_k \mathbf{g}(\boldsymbol{\varphi}_k) \quad (2.27)$$

where α_k is the gain or the step size which satisfies $\alpha_k > 0$ and $\mathbf{g}(\cdot)$ is the gradient vector of the objective function. Under certain fairly general conditions, this algorithm will converge to the true optimum $\boldsymbol{\varphi}^*$.

As given in Equation (2.27), the gradient-based algorithms require the exact knowledge of the gradient vector. However, in most engineering system design applications, it is difficult or impractical to analytically obtain the derivatives of the objective function with respect to the design variables due to the complexity of the system. For stochastic optimization problem in Equation (2.20), the gradient vector may be estimated by stochastic approximation method. Two stochastic approximation methods have been proposed, finite-difference stochastic approximation and simultaneous perturbation method.

Finite-difference stochastic approximation: the finite-difference method is a traditional way to approximate the gradient vector in Equation (2.27). The gradient vector can be estimated by either one-sided approximation or two-sided approximation. Let $\hat{H}(\boldsymbol{\varphi}) = \hat{E}_\theta[h(\boldsymbol{\theta}, \boldsymbol{\varphi})]$ denote the estimation of the objective function. The two-sided approximation for the gradient vector is given as follow:

$$\hat{\mathbf{g}}(\boldsymbol{\varphi}_k) = \begin{bmatrix} \frac{\hat{H}(\boldsymbol{\varphi}_k + c_k \boldsymbol{\xi}_1) - \hat{H}(\boldsymbol{\varphi}_k - c_k \boldsymbol{\xi}_1)}{2c_k} \\ \vdots \\ \frac{\hat{H}(\boldsymbol{\varphi}_k + c_k \boldsymbol{\xi}_d) - \hat{H}(\boldsymbol{\varphi}_k - c_k \boldsymbol{\xi}_d)}{2c_k} \end{bmatrix} \quad (2.28)$$

where c_k is the difference magnitude which satisfies $c_k > 0$ and $\boldsymbol{\xi}_i$ is the vector with a 1 in the i th component and 0's elsewhere, which only gives a perturbation in the i th direction. The gains $\{\alpha_k, c_k\}$ for finite-difference stochastic approximation have to be chosen appropriately to improve the performance of the method.

The computational cost of the finite-difference stochastic approximation increases linearly with the dimension of the design vector. This can be seen from Equation

(2.28), where a single evaluation of the gradient vector requires $2d$ estimates of the objective function. Thus the number of estimates of the objective function may become prohibitive when the number of design variables becomes large. The recent proposed simultaneous perturbation stochastic approximation method is able to obtain the gradient vector independent of the dimension of the design vector and this algorithm is introduced as follows.

Simultaneous perturbation stochastic approximation: this algorithm is based on the idea that the simultaneous perturbations in all directions in a single estimation provide as much information for the optimization process in the long run as the perturbation in a single direction at each time. Similarly, both one-sided and two-sided approximation can be obtained. The two-sided approximation for the gradient vector using simultaneous perturbation is given as follow:

$$\hat{\mathbf{g}}(\boldsymbol{\varphi}_k) = \begin{bmatrix} \frac{\hat{H}(\boldsymbol{\varphi}_k + c_k \boldsymbol{\Delta}_k) - \hat{H}(\boldsymbol{\varphi}_k - c_k \boldsymbol{\Delta}_k)}{2c_k \Delta_{k1}} \\ \vdots \\ \frac{\hat{H}(\boldsymbol{\varphi}_k + c_k \boldsymbol{\Delta}_k) - \hat{H}(\boldsymbol{\varphi}_k - c_k \boldsymbol{\Delta}_k)}{2c_k \Delta_{kd}} \end{bmatrix} \quad (2.29)$$

where $\boldsymbol{\Delta}_k = [\Delta_{k1}, \dots, \Delta_{kd}]^T$ is the d -dimensional perturbation vector which defines the perturbation in each component of the design vector. As shown in Equation (2.29), the estimation of the gradient vector only requires two evaluation of the objective function, $\hat{H}(\boldsymbol{\varphi}_k + c_k \boldsymbol{\Delta}_k)$ and $\hat{H}(\boldsymbol{\varphi}_k - c_k \boldsymbol{\Delta}_k)$. This approximation is independent of the dimension d and this makes simultaneous perturbation stochastic approximation competitive to finite-difference stochastic approximation in solving stochastic optimization problem where the number of the design variables is large.

3) meta-model, failure probability function approximation approach

Meta-model approach: this approach is to construct a meta-model to estimate the response of dynamic simulation of the structural system under different combinations of random variables (Jensen 2005, Dubourg, Sudret et al. 2011). The

meta-model is inexpensive to evaluate the function value so that it is very attractive for reliability-based design optimization. However, it is quite challenging for cases where the dimension of random variables is high.

Failure probability function approximation approach: for the reliability-based optimization problem where the objective function is the failure probability function, the derivatives of the objective function can be estimated by the approaches proposed by Ching (Ching and Hsieh 2007) and Jensen (Jensen, Valdebenito et al. 2008). These failure probability function methods are based on the assumption that the log of the failure probability varies linearly with the design variables. However, the failure probability function methods cannot directly be used to solve the particular optimization considered in this work as the objective functions are not failure probability functions. Moreover, there is no prior information to help us select the basic functions to describe the relationship between the objective function and the design variables.

Decoupling approach: the basic idea of the decoupling approach is to use the extracted information from the reliability analysis stage to improve the numerical efficiency at the optimization stage. In this way, it may not necessary to perform a full reliability analysis each time at a new design point in the space of the design variables which is explored by the optimization algorithm iteratively. This approximate reliability method may not applicable to the cases where the number of the design variables is large as discussed in Katafygiotis and Zuev and Valdebenito (Katafygiotis and Zuev 2008, Katafygiotis and Wang 2009, Valdebenito and Schu ¨ dler 2010, Valdebenito and Schu ¨ dler 2011).

2.4.3 Techniques for reducing the estimation error of the objective function

For both gradient-free and gradient-based stochastic optimization algorithms, a number of estimations of the objective function are required to obtain the optimal solution during the identification process. However, the selected appropriate estimation method for the probability integral, Monte Carlo estimation, involves an estimation error when a limited number of samples are simulated, as indicated before, which is reflected by the coefficient of variation of the estimator. The

estimation error will affect the efficiency of the optimization algorithm. Thus it is important to reduce the estimation error of the objective function.

One possible method is to reduce the absolute value of the estimation error. From the equation for the coefficient of variation of Monte Carlo estimator, increasing the number of simulated samples is able to reduce the estimation error of the probability integral. However, this will lead to an increase in the computational cost, especially for the case that the integrand is not an analytical function and each evaluation of the integrand is computationally expensive.

Reducing the relative importance of the estimation error is an alternative way to enhance the efficiency of the optimization algorithm without increasing the computational burden. This task may be achieved by creating consistent estimation error for two different design choices $\boldsymbol{\varphi}_1$ and $\boldsymbol{\varphi}_2$, i.e. by minimizing the variance between the two estimates $\hat{E}(\Omega_1, \boldsymbol{\varphi}_1)$ and $\hat{E}(\Omega_2, \boldsymbol{\varphi}_2)$. The simulated samples are represented by Ω . To realize this goal, the technique of common random number streams is adopted by generating the same samples for the two different estimates, i.e. $\Omega_1 = \Omega_2$. The influence of using common random numbers may be better understood by investigating the variability of the two estimates, which is written as

$$\begin{aligned} \text{var}\left(\hat{E}(\Omega_1, \boldsymbol{\varphi}_1) - \hat{E}(\Omega_2, \boldsymbol{\varphi}_2)\right) &= \text{var}\left(\hat{E}(\Omega_1, \boldsymbol{\varphi}_1)\right) + \text{var}\left(\hat{E}(\Omega_2, \boldsymbol{\varphi}_2)\right) \\ &\quad - 2\text{cov}\left(\hat{E}(\Omega_1, \boldsymbol{\varphi}_1), \hat{E}(\Omega_2, \boldsymbol{\varphi}_2)\right) \end{aligned} \quad (2.30)$$

by adopting the common random number technique, the dependence between the two estimates is introduced as long as the design choices are not significantly different. As a result, the dependence increases the covariance between the two estimates and thus reducing their variability and decreasing the relative importance of the estimation error. The common random number technique is often used in the stochastic approximation algorithms since the derivative information only requires the comparison between local design choices.

For stochastic design optimization problems, exterior sampling approach is another method which is closely related to common random number. The difference is that the exterior sampling approach transfers the stochastic optimization problem into a deterministic one which can be solved by an appropriate routine, by adapting the same stream random numbers for all iterations in the identification process. This transformation of optimization problem often requires the number of simulated samples for all estimates to be “large enough”. To assess the quality of the solution using exterior sampling approach, it is recommended to solve the optimization problem multiple times by using different independent random sample streams. Even though the computational cost for the deterministic optimization is typically smaller than that of the original stochastic search problem, the overall efficiency may be worse because of the requirement of performing the deterministic optimization multiple times.

2.4.4 Challenges of “traditional” stochastic optimization algorithms

The “traditional” stochastic optimization algorithms include both the gradient-free and gradient-based algorithms introduced in section 2.4.2. For the proposed design optimization in this study, the “traditional” optimization algorithms are challenging since the objective function is not analytically known and the evaluation of the performance function is computationally expensive.

One limitation is that the identified optimum by random search methods may not be the true optimum due to the estimation error of the objective function. Since the objective function is not analytically known and has a generic form of the probability integral, the estimation error will exist when using Monte Carlo simulation to estimate the objective function based on a limited number of simulated samples. The estimation error can be reduced by increasing the number of simulated samples with the expense of high computational cost. It can be seen that the optimum using a limited number of simulated samples may differ from the true optimum and this is caused by the noise in the estimation of the objective function. The noise in the estimation of the objective function may also lead to challenges for the gradient-based optimization algorithms since noise will also exist in the

estimation in the gradient vector. The noise in the gradient vector may be reduced by adopting the common random number technique.

The other limitation is the requirement of high computational cost. Both gradient-based and gradient-free optimization algorithm require the estimation of the objective function a number of times and a single estimate needs a number of evaluations of the performance function. This will lead to a large number of evaluations of the performance function and a single evaluation requires an amount of considerable time for complex system in realistic engineering applications. As a result, the computational cost is very high and even unaffordable for some cases.

The dimension of the design vector also affects the efficiency of the optimization algorithm. For the random search algorithms, the computational cost of the identification process will increase exponentially as the number of the design variables increases due to the size of the design space. The number of evaluations of the performance function will increase linearly with the dimension as the estimation of the gradient vector depends on the number of the design variables. Simultaneous perturbation stochastic approximation is independent of the dimension as the gradient vector is estimated based on two estimates of the objective function, while it still requires the estimation of the objective function. As the aforementioned limitation, the computational cost will be high.

2.5 Chapter summary

This chapter provides literature reviews on the design optimization of structural control system for floating offshore wind turbines in section 2.2. The previous studies use vastly simplified structural models which only consider two or four degrees of freedom. To save simulation time, the simplified models are used to simulate the free decay response rather than the dynamic response under realistic external excitations. Moreover, the objective function is regarded as the performance indexes of the free decay response, such as the standard deviation and the range. The selection of the objective function is based on the assumption that optimizing these performance indexes equals to optimizing the structural performance of the floating wind turbine under realistic external environmental conditions.

The robust stochastic design optimization of structural control system for floating wind turbines is proposed in section 2.3. The optimization scheme is based on the ideal way to design a controller to optimize the realistic structural performance of the turbine. The objective function is selected as the long-term structural performance such as fatigue damage and ultimate strength failure probability. Uncertainties in external environmental conditions are taken into account in the rigorous mathematic formulation of the objective function. Based on history data of site measurement, probabilistic model of these random variables are given in section 2.3.2. The high-fidelity simulator, FAST, is introduced to simulate the dynamic response of the coupled turbine-controller system under realistic external environmental loads, including turbulent wind and irregular waves.

A review on the stochastic optimization algorithms is presented in section 2.4. The random search and stochastic approximation method are introduced. This section also reviews the methods for estimating the probability integral and the techniques for reducing the estimation error. The existing stochastic optimization algorithms are challenging in solving the proposed optimization problem due to the high computational cost. Moreover, the dimension of the design variables will significantly affect the efficiency of the algorithm.

The main role of this chapter is to propose a mathematical formulation for the design optimization of structural control system and present the motivation of proposing an efficient stochastic optimization algorithm based on the review on the “traditional” stochastic optimization algorithms. This provides basics for the following chapters to propose algorithms to obtain the optimal design.

Chapter 3 A new sampling algorithm for improving the efficiency of Stochastic Subset Optimization algorithm

3.1 Introduction

As given in Chapter 2, the generic formula of the design optimization of the structural control system is $\boldsymbol{\varphi}^* = \arg \min_{\boldsymbol{\varphi} \in \Phi} \int_{\Theta} h(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(\boldsymbol{\theta}) d\boldsymbol{\theta}$, where the objective function is a probability integral. The only constrain of the optimization problem considered in this thesis is the upper and lower bound of the design variables. In some real practices, there may exist some constrains which correspond to the requirements that the system has to satisfy. The constrains are functions of the design variables and the random variables, such as the failure probability functions. It is important to consider the constrains to identify the optimal design, but the current work only considers the unconstrained optimization problem. The estimation of the objective function is computationally expensive since each estimation requires numerous evaluations of the performance function $h(\boldsymbol{\theta}, \boldsymbol{\varphi})$ and each evaluation is obtained from the result of a structural analysis, which is time-consuming. This leads to the traditional gradient-free optimization algorithms being challenging as the estimation of the objective function is inevitable for these algorithms such as direct search and the convergence of the optimization procedure requires a number of estimations of the objective function. Moreover, gradient-based optimization algorithms which directly require the derivatives of the objective function are not applicable since the derivatives cannot be obtained analytically for this probability integral. Although the failure probability function can be used approximate the objective function value to obtain the derivatives, it cannot be directly adopted to solve the problem in this work as the objective function is not a failure probability function. An alternative way is to estimate the derivatives of the objective function based on the estimation of the objective function although the estimation may need much computational cost. Although the estimation of the objective function is required, Simultaneous perturbation stochastic approximation (SPSA) (Spall 1992) is computationally affordable for some cases since it obtains the derivatives of the objective function with respect to all design variables from

two estimates of the objective function and this significantly reduces the total number of estimates of the objective function. In this work, the optimal solution obtained by SPSA is regarded as benchmark for the proposed algorithms.

Stochastic Subset Optimization (SSO) algorithm was initially proposed by Taflanidis and Beck (Taflanidis and Beck 2008) for solving reliability-based design optimization problem where the objective function is a failure probability function and the optimal design was obtained by stochastic simulation. Due to the advantage of avoiding estimating the objective function, SSO is suggested to be adopted to solve the proposed optimization problem. Although SSO is relatively efficient without the requirement of estimation of the objective function, it still requires a number of evaluations of the performance function $h(\boldsymbol{\theta}, \boldsymbol{\varphi})$ as it is necessary to obtain samples distributed according to the auxiliary probability density function (PDF) defined by SSO. The major computational cost of SSO lies in simulating samples from the auxiliary PDF. Therefore, improving the efficiency of the sampling algorithm can directly reduce the computational effort required by SSO. A new sampling algorithm is proposed in this chapter and it is based on the sampling procedure of the Acceptance-Rejection sampling (ARS) algorithm and integrated with Subset Simulation (SS).

This chapter is organized as follows. Section 3.2 introduces SSO and summarizes the framework of SSO. In section 3.3, the review of several existing sampling algorithms is presented and followed by the introduction of SS. The new sampling algorithm, ARS-SS, is proposed subsequently. Section 3.4 presents the SPSA algorithm and the corresponding implementation details of SPSA. Case studies are conducted in section 3.5 and two design optimization problems are considered, including a 2-dimensional design vector and a 4-dimensional design vector. The improvement of the efficiency by ARS-SS and the quality of the optimal solution are presented. The conclusions are given in section 3.6.

3.2 Stochastic Subset Optimization

In order to reduce the computational cost, Taflanidis and Beck (Taflanidis and Beck 2008, Taflanidis and Beck 2008) proposed a new optimization method, called

Stochastic Subset Optimization (SSO), by identifying subsets which may contain the optimal design choice step by step. SSO was initially developed to solve the optimal reliability design problems where the objective function is the reliability or failure probability. The main idea behind SSO is investigating the sensitivity of the objective function within the design space through the simulated samples from an auxiliary probability density function (PDF) which is proposed in the augmented problem. The following sections give the details about the augmented problem and the iterative approach for investigating the sensitivity of the objective function.

3.2.1 Generic formula

For simplicity, the generic formula for the system design optimization problem is proposed as follows in this work. Let $\boldsymbol{\theta} = [\theta_1, \dots, \theta_n] \in \Theta \subset \mathbb{R}^n$ be a parameter vector including all uncertain parameters of interest, such as the parameters of the simulation model and environmental conditions. These uncertain parameters should lie in the corresponding space which includes all possible values for each parameter. Let $\boldsymbol{\varphi} = [\varphi_1, \dots, \varphi_d] \in \Phi \subset \mathbb{R}^d$ be a d -dimensional vector containing some controllable parameters of the system which determine the system design. This vector is also called the design vector which includes all design variables. In practice, the design variables should be defined in an admissible space Φ containing all possible design choices. Let $p(\boldsymbol{\theta}): \mathbb{R}^n \mapsto [0, \infty)$ be a PDF quantifying the plausibility of the uncertain parameters contained in $\boldsymbol{\theta}$. This PDF incorporates the available information about the system. To be formulated in a generic form, the performance of the system can be given as:

$$H(\boldsymbol{\varphi}) = E_{\boldsymbol{\theta}}[h(\boldsymbol{\theta}, \boldsymbol{\varphi})] = \int_{\Theta} h(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(\boldsymbol{\theta}) d\boldsymbol{\theta} \quad (3.1)$$

where $h(\boldsymbol{\theta}, \boldsymbol{\varphi})$ represents the performance function of the system defined by $\boldsymbol{\varphi}$ under the uncertain parameters specified by $\boldsymbol{\theta}$. The performance of the system defined by $\boldsymbol{\varphi}$ is given as $E_{\boldsymbol{\theta}}[h(\boldsymbol{\theta}, \boldsymbol{\varphi})]$, denoting the expectation of the performance function with respect to the PDF for $\boldsymbol{\theta}$. In realistic engineering design, the performance of the system may be used in the constraints, such as the reliability-based design optimization problem which contains reliability constraints. This work

focuses on the stochastic design optimization problem where the objective function is a stochastic integral as given in Equation (3.1). The stochastic optimization problem in this case can be formulated in a generic form as follow:

$$\boldsymbol{\varphi}^* = \arg \min_{\boldsymbol{\varphi} \in \Phi} \int_{\Theta} h(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(\boldsymbol{\theta}) d\boldsymbol{\theta} \quad (3.2)$$

3.2.2 Augmented problem

The augmented problem is the basis of SSO and it was initially proposed by Au and Beck (Au 2005) for investigating the sensitivity of the failure probability function with respect to the system design. In this concept, the design variables are artificially regarded as uncertain with some specified PDF $p(\boldsymbol{\varphi})$ over the admissible design space Φ . Based on this setting, an auxiliary PDF is defined as follow:

$$\pi(\boldsymbol{\theta}, \boldsymbol{\varphi}) = \frac{h(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(\boldsymbol{\theta}) p(\boldsymbol{\varphi})}{\int_{\Theta} \int_{\Phi} h(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(\boldsymbol{\theta}) p(\boldsymbol{\varphi}) d\boldsymbol{\theta} d\boldsymbol{\varphi}} \quad (3.3)$$

the denominator in the above equation is a normalizing constant which represents the expectation of the performance function over the design space Φ and the random variable space Θ . Let C be the normalizing constant and $\pi(\boldsymbol{\theta}, \boldsymbol{\varphi})$ can be rewritten as follow:

$$\pi(\boldsymbol{\theta}, \boldsymbol{\varphi}) = C \cdot h(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(\boldsymbol{\theta}) p(\boldsymbol{\varphi}) \propto h(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(\boldsymbol{\theta}) p(\boldsymbol{\varphi}) \quad (3.4)$$

$$C = \frac{1}{\int_{\Theta} \int_{\Phi} h(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(\boldsymbol{\theta}) p(\boldsymbol{\varphi}) d\boldsymbol{\theta} d\boldsymbol{\varphi}} = \frac{1}{E_{\boldsymbol{\theta}, \boldsymbol{\varphi}}[h(\boldsymbol{\theta}, \boldsymbol{\varphi})]} \quad (3.5)$$

the exact value of C can be evaluated by stochastic simulation techniques, while the value may not be required by simulating samples from $\pi(\boldsymbol{\theta}, \boldsymbol{\varphi})$ in case the appropriate stochastic sampling algorithms are chosen.

Since $\pi(\boldsymbol{\theta}, \boldsymbol{\varphi})$ should satisfy the characteristics of the PDF, $h(\boldsymbol{\theta}, \boldsymbol{\varphi}) \geq 0$ is required such that $\pi(\boldsymbol{\theta}, \boldsymbol{\varphi}) \geq 0$ will be true for all pairs of $[\boldsymbol{\theta}, \boldsymbol{\varphi}]$. This condition can be achieved by the following modification of the performance function:

$$h_m(\boldsymbol{\theta}, \boldsymbol{\varphi}) = h(\boldsymbol{\theta}, \boldsymbol{\varphi}) - \min_{\boldsymbol{\theta} \in \Theta, \boldsymbol{\varphi} \in \Phi} \{h(\boldsymbol{\theta}, \boldsymbol{\varphi})\} \quad (3.6)$$

In terms of the above auxiliary PDF, the marginal PDF of $\pi(\boldsymbol{\theta}, \boldsymbol{\varphi})$ with respect to the design variables $\boldsymbol{\varphi}$ can be given as follow:

$$\begin{aligned} \pi(\boldsymbol{\varphi}) &= \int_{\Theta} \pi(\boldsymbol{\theta}, \boldsymbol{\varphi}) d\boldsymbol{\theta} \\ &= \int_{\Theta} C \cdot h(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(\boldsymbol{\theta}) p(\boldsymbol{\varphi}) d\boldsymbol{\theta} \\ &= C \cdot p(\boldsymbol{\varphi}) H(\boldsymbol{\varphi}) \end{aligned} \quad (3.7)$$

the above equation indicates that the objective function $H(\boldsymbol{\varphi})$ is related to the marginal PDF $\pi(\boldsymbol{\varphi})$ and the chosen artificial PDF $p(\boldsymbol{\varphi})$. The objective function $H(\boldsymbol{\varphi})$ can be rewritten as follow:

$$H(\boldsymbol{\varphi}) = \frac{\pi(\boldsymbol{\varphi})}{C \cdot p(\boldsymbol{\varphi})} \quad (3.8)$$

Since C is a constant, the objective function $H(\boldsymbol{\varphi})$ is proportional to the value of $\frac{\pi(\boldsymbol{\varphi})}{p(\boldsymbol{\varphi})}$. The artificial PDF $p(\boldsymbol{\varphi})$ can be chosen as any arbitrary PDF over the design space Φ , while in most cases $p(\boldsymbol{\varphi})$ is chosen as a uniform PDF since no prior information about the system design is available. In such case, the objective function $H(\boldsymbol{\varphi})$ becomes directly proportional to the marginal PDF $\pi(\boldsymbol{\varphi})$ since $p(\boldsymbol{\varphi})$ is also a constant for all $\boldsymbol{\varphi} \in \Phi$. Let C_h be the constant. Equation (3.8) can be written as follow:

$$H(\boldsymbol{\varphi}) = C_h \cdot \pi(\boldsymbol{\varphi}) \quad (3.9)$$

Thus, the original task for minimizing the objective function $H(\boldsymbol{\varphi})$ is equivalent to minimizing the marginal PDF $\pi(\boldsymbol{\varphi})$. In other words, the sensitivity of the objective function can be explored by investigating the sensitivity of the marginal PDF with respect to the design variables over the design space.

The marginal PDF $\pi(\boldsymbol{\varphi})$ can be explored in case the samples distributed according to $\pi(\boldsymbol{\varphi})$ are available. These samples can be obtained by taking the $\boldsymbol{\varphi}$ component of the sample pairs $[\boldsymbol{\theta}, \boldsymbol{\varphi}]$ simulated from the auxiliary PDF $\pi(\boldsymbol{\theta}, \boldsymbol{\varphi})$. The algorithms for simulating samples from $\pi(\boldsymbol{\theta}, \boldsymbol{\varphi})$ will be presented in section 3.3. The method for exploring $\pi(\boldsymbol{\varphi})$ based on the simulated samples will be given in the following section.

3.2.3 Investigating the sensitivity of $\pi(\boldsymbol{\varphi})$ by subsets

Based on the samples distributed according to $\pi(\boldsymbol{\varphi})$, the marginal PDF $\pi(\boldsymbol{\varphi})$ may be evaluated by some approximation methods such as kernel density estimation methods. However, kernel density estimation methods may be challenging or problematic for some cases, where the dimension of design variables is large or the sensitivity of the objective function over the design space is complicated. For the high-dimensional case, the accuracy of the kernel density estimation also depends on the kernel function, the selection of which is not an easy work. Furthermore, the kernel density estimation methods always underestimate the densities around the boundary of the design space which is bounded in most engineering design cases.

To avoid evaluating the value of $\boldsymbol{\varphi}$ at any specified design choice in the design space, SSO investigates the sensitivity of $\pi(\boldsymbol{\varphi})$ by estimating the average value of $\pi(\boldsymbol{\varphi})$ over any specified subset I within the design space Φ . The average value, which is also equal to the volume density, is denoted by $S(I)$:

$$S(I) = \frac{1}{V_I} \int_I \pi(\boldsymbol{\varphi}) d\boldsymbol{\varphi} \propto \frac{1}{V_I} \int_I H(\boldsymbol{\varphi}) d\boldsymbol{\varphi} \quad (3.10)$$

Instead of looking at the average value of $\pi(\boldsymbol{\varphi})$ within any subset I , SSO investigates the average relative sensitivity of $\pi(\boldsymbol{\varphi})$ with respect to $\boldsymbol{\varphi}$ in any subset I and the initial design space Φ , which is denoted as follow:

$$S(I|\Phi) = \frac{\frac{1}{V_I} \int_I \pi(\boldsymbol{\varphi}) d\boldsymbol{\varphi}}{\frac{1}{V_\Phi} \int_\Phi \pi(\boldsymbol{\varphi}) d\boldsymbol{\varphi}} = \frac{\frac{1}{V_I} \int_I H(\boldsymbol{\varphi}) d\boldsymbol{\varphi}}{\frac{1}{V_\Phi} \int_\Phi H(\boldsymbol{\varphi}) d\boldsymbol{\varphi}} \quad (3.11)$$

$$= \frac{V_\Phi \int_I \pi(\boldsymbol{\varphi}) d\boldsymbol{\varphi}}{V_I \int_\Phi \pi(\boldsymbol{\varphi}) d\boldsymbol{\varphi}}$$

In the above equation, $\int_\Phi \pi(\boldsymbol{\varphi}) d\boldsymbol{\varphi}$ is always equal to 1 and $\int_I \pi(\boldsymbol{\varphi}) d\boldsymbol{\varphi}$ can be estimated based on the number of the simulated samples within the subset I . Based on the simulated samples distributed according to $\pi(\boldsymbol{\varphi})$, the relative sensitivity $S(I|\Phi)$ can be estimated as follow:

$$\hat{S}(I|\Phi) = \frac{V_\Phi N_I}{V_I N_\Phi} \quad (3.12)$$

where N_I and N_Φ represent the number of the simulated samples belonging to the spaces I and Φ , respectively. V_I and V_Φ are the volume of the spaces I and Φ , respectively.

In terms of this estimate for the ratio of the average volume density between I and Φ , the direct estimation of the volume density for any arbitrary subset is avoided and the ratio enables us to reduce the effect of the noise of the estimation. The coefficient of variation (CoV) for this estimation can be given as follow:

$$CoV(\hat{S}(I|\Phi)) \approx \sqrt{\frac{1 - P(\boldsymbol{\varphi} \in I)}{N \cdot P(\boldsymbol{\varphi} \in I)}} = \sqrt{\frac{1 - \frac{N_I}{N_\Phi}}{N \cdot \frac{N_I}{N_\Phi}}} \quad (3.13)$$

where $N = \frac{N_\Phi}{1+\gamma}$ is the effective number of the independent samples. The factor γ is a coefficient depending on the types of sampling methods. γ is zero for the direct Monte Carlo (MC) simulation technique or Acceptance-Rejection sampling (ARS) method, while this number will be greater than zero when the Markov Chain Monte Carlo (MCMC) sampling method is used since the simulated samples are correlated. For the estimation of γ , more details are provided in the reference by Berg (Berg 2004).

Based on the above, the original optimal design task becomes the following stochastic optimization of $\hat{S}(I|\Phi)$ to identify the optimal subset $\hat{I}$:

$$\hat{I} = \arg \min_{I \in A} \hat{S}(I|\Phi) = \arg \min_{I \in A} \frac{V_\Phi N_I}{V_I N_\Phi} = \arg \min_{I \in A} \frac{N_I}{V_I} \quad (3.14)$$

where A is a family of admissible subsets of the design space Φ with the constraint that each subset satisfies a specified geometrical shape and some restrictions on the size of the subset. Since the CoV of the estimate of $\hat{S}(I|\Phi)$ is related to the value of $\frac{N_I}{N_\Phi}$, an appropriate method to select the size constraint for the admissible subsets is to maintain the same number of samples belonging to each subset such that $\frac{N_I}{N_\Phi}$ is a constant for keeping the same CoV of the estimate for all subsets. Based on the simulated samples, SSO will identify a subset within A which contains the final optimal design with the highest likelihood. This likelihood represents the quality of the identified subset and depends on the corresponding value of $\hat{S}(I|\Phi)$. Smaller the value of $\hat{S}(I|\Phi)$ is, higher the quality of the identified optimal subset is. This is due to smaller value of $\hat{S}(I|\Phi)$ indicates the greater difference between the average value of the objective function within the optimal subset and the original design space. Therefore, a value of $\hat{S}(I|\Phi)$ that is close to unity indicates that the average values of the objective function within the optimal subset I and the design space Φ are almost the same. This implies the

search for the optimal subset is challenging because the objective function is not sensitive to the selection of subsets and ultimately means that the objective function has small sensitivity to the design variables. For the case where $\hat{S}(I|\Phi)$ is close to one, one way to identify the final optimal design choice is to use the traditional optimization algorithms by regarding the center point of the optimal subset $\hat{I}$ as the starting point of the optimization process and the search is conducted within the optimal subset $\hat{I}$. The results from SSO will save the computational cost of the traditional algorithms. A simpler way is just regarding the center point of the optimal subset $\hat{I}$ as the suboptimal design.

It is notable that SSO is performed with respect to the vector that is used to constrain the shape and the size of the admissible subsets and this optimization in Equation (3.14) is not an easy task due to the discontinuity of its objective function. However, compared to the objective function of the original optimization problem in Equation (3.2), this objective function in Equation (3.14) can be evaluated with insignificant computational cost because the evaluation of the performance function which is time-consuming is not required and the major computational cost lies in simulating samples from the auxiliary PDF. This advantage allows us to calculate many candidate solutions with a small amount of computational effort and the optimal subset can be identified with an appropriate algorithm such as the direct search optimization algorithms.

In order to identify a relatively small subset which may contain the optimal design, an iterative approach is adopted by SSO since it increases the efficiency. Since the size of the admissible subsets depends on the choice of the value of $\frac{N_I}{N_\Phi}$, obtaining

a small subset in a single step requires a small value of $\frac{N_I}{N_\Phi}$. However, a large number of the simulated samples are required for maintaining the CoV of the objective function since $\frac{N_I}{N_\Phi}$ is small and this will lead to high computational cost.

Let k denote the index of the optimization level. At optimization level 1, I_0

represents the initial design space Φ . The samples for optimization level 1 will be discussed in the next section. At optimization level k , the MCMC technique is used to simulate a number of additional sample pairs $[\theta, \phi]$ which are located within $\hat{I}_{k-1}$ and distributed according to $\pi(\theta, \phi)$. MCMC is used because the samples that are distributed according to the target PDF are available and these samples are the samples in optimization level $k-1$ whose ϕ component is located within $\hat{I}_{k-1}$. The details about simulating additional samples based on the available seeds by MCMC are given in the next section. Based on the simulated samples in optimization level k , the optimal subset $\hat{I}_k$ is identified by solving the modified optimization problem in Equation (3.14)

$$\hat{I}_k = \arg \min_{I_k \in A} \hat{S}(I_k | \hat{I}_{k-1}) = \arg \min_{I \in A} \frac{V_{\hat{I}_{k-1}} N_{I_k}}{V_{I_k} N_{\hat{I}_{k-1}}} = \arg \min_{I \in A} \frac{N_{I_k}}{V_{I_k}} \quad (3.15)$$

and searching within $\hat{I}_{k-1}$. At the end of each optimization level, the stopping criterion is checked in terms of the value of $\hat{S}(\hat{I}_k | \hat{I}_{k-1})$ and the optimization will terminate on condition that the stopping criterion is satisfied.

This section provides some suggestions for the decision of the stopping criterion and the selection of the shape and size constraints for the admissible subsets. The optimization iteration may terminate when the estimate $\hat{S}(\hat{I}_k | \hat{I}_{k-1})$ exceeds the specified threshold c_s . The value of c_s depends on the choice of $\frac{N_I}{N_\Phi}$ and the

effective number of the simulated samples N at each optimization level. For the selection of the admissible subsets, the efficiency of SSO significantly depends on the constraints on the geometrical shape and size of the subsets. In order to fully explore the sensitivity of the objective function with respect to each design variable, an appropriate geometrical shape should be chosen so that the challenging, non-smooth optimization problem in Equation (3.15) can be efficiently solved. An appropriate choice may be the hyper-rectangle or hyper-ellipse since the dimension of them in each direction can be adaptively adjusted so that the sensitivity of the

objective function with respect to all design variables can be investigated simultaneously. The constraint on the size of the admissible subsets is very critical due to its effect on the CoV of $\hat{S}(\hat{I}_k | \hat{I}_{k-1})$ and the quality of the identification. Typically, the size of the subsets may be constrained by either: (i) the volume ratio $\delta = \frac{V_I}{V_\Phi}$, or (ii) the sample ratio $\rho = \frac{N_I}{N_\Phi}$. The criterion or suggestion on the selection of δ is not straightforward since the CoV of $\hat{S}(\hat{I}_k | \hat{I}_{k-1})$ is not directly affected by δ . On the contrary, the relationship between the CoV of $\hat{S}(\hat{I}_k | \hat{I}_{k-1})$ and the value of ρ is obvious and this enables some suggestions on the selection of ρ . In spite of the relationship between $\hat{S}(\hat{I}_k | \hat{I}_{k-1})$ and ρ , the selection of ρ is a challenging task due to the iterative approach. The large value for ρ indicates that the size of the identified subsets decreases slowly with optimization level and this leads to more optimization levels for SSO to converge to a relatively small subset. More optimization levels indicate a larger amount of computational cost since additional samples need to be simulated at each optimization level. However, a small value for ρ will increase the CoV of $\hat{S}(\hat{I}_k | \hat{I}_{k-1})$ in case the number of the simulated samples is maintained and the only way to maintain the same CoV of $\hat{S}(\hat{I}_k | \hat{I}_{k-1})$ is to simulate more samples. The choice of ρ need balance the number of the simulated samples at each optimization level and the number of steps for the convergence of SSO. Based on the investigation by Taflanidis and Beck, the selection of $\rho \in [0.1, 0.2]$ can provide good efficiency.

3.2.4 Framework of SSO algorithm

Based on the augmented problem which artificially regards the design variables as uncertain, the sensitivity of the objective function $H(\boldsymbol{\varphi})$ can be explored by investigating the sensitivity of the marginal PDF $\pi(\boldsymbol{\varphi})$ of the auxiliary PDF $\pi(\boldsymbol{\theta}, \boldsymbol{\varphi})$ with respect to the design variables. To explore the marginal PDF $\pi(\boldsymbol{\varphi})$, the optimization problem in (3.15) is solved based on the samples that are

distributed according to $\pi(\boldsymbol{\varphi})$ and these samples are obtained by taking the $\boldsymbol{\varphi}$ component of the samples pairs $[\boldsymbol{\theta}, \boldsymbol{\varphi}]$ from $\pi(\boldsymbol{\theta}, \boldsymbol{\varphi})$. Instead of directly obtaining the optimal design which minimizes the objective function, SSO adopts the iterative approach to identify a sequence of subsets $\hat{I}_k \subset \hat{I}_{k-1} \subset \dots \subset \hat{I}_0 = \Phi$ which may contain the optimal design with the highest likelihood at the corresponding optimization level. SSO finally converges to a relatively small subset when the stopping criterion is satisfied. The procedure of SSO is outlined as follows:

SSO algorithm

Input: N_Φ , the number of samples at each optimization level

ρ , the ratio of the samples in the subset

Algorithm: at optimization level k

Step 1: regard the retained sample pairs $\pi(\boldsymbol{\theta}, \boldsymbol{\varphi} | \boldsymbol{\varphi} \in \hat{I}_{k-1})$ at optimization level $k-1$ as seeds and simulate additional sample pairs until the total number of samples reaches N_Φ .

Step 2: based on the $\boldsymbol{\varphi}$ component of the simulated sample pairs, solve the optimization problem in Equation (3.15) and identify the optimal subset $\hat{I}_k \subset \hat{I}_{k-1}$ which has the highest likelihood of containing the optimal design.

Step 3: check the stopping criterion $\hat{S}(\hat{I}_k | \hat{I}_{k-1}) > c_s$ and terminate the optimization iteration if it was satisfied. Otherwise, retain the sample pairs whose $\boldsymbol{\varphi}$ component belongs to $\hat{I}_k$ and denote these sample pairs by $\pi(\boldsymbol{\theta}, \boldsymbol{\varphi} | \boldsymbol{\varphi} \in \hat{I}_k)$.

Suboptimal design: the center point of the optimal subset $\hat{I}_k$ at the last optimization level.

3.3 Review of existing sampling algorithms and Subset Simulation

The major computational cost of SSO lies in the simulation of sample pairs $[\boldsymbol{\theta}, \boldsymbol{\varphi}]$ from the auxiliary PDF $\pi(\boldsymbol{\theta}, \boldsymbol{\varphi})$ since many system analyses are required.

Therefore, it is important to adopt an appropriate sampling algorithm to reduce the computational cost and consequently increase the efficiency of SSO. The review of several existing sampling algorithms is first presented and followed by the proposed sampling algorithm. The proposed algorithm is based on Acceptance-Rejection sampling (ARS) algorithm and Subset Simulation (SS) algorithm such that SS is introduced before the proposed algorithm.

3.3.1 Review of the existing sampling algorithms

For some standard and widely-used distributions, samples can be directly simulated by the software such as Matlab. However, the auxiliary PDF $\pi(\boldsymbol{\theta}, \boldsymbol{\phi})$ is not analytically known due to the performance function and this prohibits simulating samples from $\pi(\boldsymbol{\theta}, \boldsymbol{\phi})$ directly. The following sampling algorithms are presented for simulating samples from an arbitrary PDF.

1) Acceptance-Rejection sampling algorithm

The basic idea of ARS is to use a simpler proposal PDF $g(\boldsymbol{\theta})$ from which the samples can be easily simulated. The fundamental theorem of ARS can be found in this book. Let $f(\boldsymbol{\theta})$ be the target PDF which the simulated samples are distributed according to. $\boldsymbol{\theta} = [\theta_1, \dots, \theta_n]$ is a vector including all random variables of interest. $g(\boldsymbol{\theta})$ denotes the proposal PDF which the candidate samples are simulated from. The random variable u is simulated from the uniform distribution in the range from 0 to 1. The procedure of the ARS is shown as follows.

ARS algorithm

Initialization: $g(\boldsymbol{\theta})$, the proposal PDF

$$M, \text{ the constant satisfies } M \geq \max_{\boldsymbol{\theta} \in \Theta} \left\{ \frac{f(\boldsymbol{\theta})}{g(\boldsymbol{\theta})} \right\}$$

Step 1: simulate a candidate sample $\boldsymbol{\theta}$ from $g(\boldsymbol{\theta})$ and simulate u from $U_{[0,1]}$

Step 2: compute the ratio $r = \frac{f(\boldsymbol{\theta})}{M \cdot g(\boldsymbol{\theta})}$

Step 3: if $r > u$, accept the candidate sample. Otherwise, reject the candidate sample.

The procedure can be repeated as many times until the specified number of the accepted candidate samples is obtained. The constant M has to satisfy

$$M \geq \max_{\boldsymbol{\theta} \in \Theta} \left\{ \frac{f(\boldsymbol{\theta})}{g(\boldsymbol{\theta})} \right\} \text{ for all possible } \boldsymbol{\theta} \in \Theta \text{ such that the accepted samples will be}$$

distributed according to the target PDF. On the contrary, smaller M is preferred since it affects the acceptance rate of the candidate samples. In realistic applications,

M is selected as the maximum of the ratio $\frac{f(\boldsymbol{\theta})}{g(\boldsymbol{\theta})}$ such that M is as small as

possible. Since the candidate samples are generated independently, the simulated samples by ARS are independent samples. The efficiency of ARS is represented by the acceptance rate which depends on M and the proposal PDF $g(\boldsymbol{\theta})$. Therefore, a good choice of the proposal PDF is very critical to ARS.

2) Markov Chain Monte Carlo algorithm

MCMC is indeed the Metropolis-Hastings (MH) method, which is a powerful technique for simulating samples from an arbitrary PDF. MCMC is based on the assumption of ergodicity which ensures that the limited stationary distribution of the determined Markov chain is the target distribution. The initial state of the Markov chain is chosen as an arbitrary sample in the sample space. A chosen proposal PDF $g(\cdot|\boldsymbol{\theta})$ is used to generate the candidate sample for the next state and the next state of the Markov chain may accept the candidate sample or just regard the current state as the next state. A brief introduction of the MCMC algorithm is given as follows.

MCMC algorithm

Initialization: $g(\cdot|\boldsymbol{\theta})$, the proposal PDF

$\boldsymbol{\theta}_k$, current state of the Markov chain

Step 1: simulate a candidate sample $\tilde{\boldsymbol{\theta}}_{k+1} \sim g(\tilde{\boldsymbol{\theta}}_{k+1} | \boldsymbol{\theta}_k)$ and simulate u from $U_{[0,1]}$

Step 2: compute the ratio

$$r = \frac{f(\tilde{\boldsymbol{\theta}}_{k+1}) g(\boldsymbol{\theta}_k | \tilde{\boldsymbol{\theta}}_{k+1})}{f(\boldsymbol{\theta}_k) g(\tilde{\boldsymbol{\theta}}_{k+1} | \boldsymbol{\theta}_k)} \quad (3.16)$$

Step 3: determine the next state of the chain

$$\boldsymbol{\theta}_{k+1} = \begin{cases} \tilde{\boldsymbol{\theta}}_{k+1}, & \text{if } r > u \\ \boldsymbol{\theta}_k, & \text{otherwise} \end{cases} \quad (3.17)$$

These steps show the iteration for the Markov chain moving from the current state to the next state. The next state of the Markov chain may remain as the current state or move to the candidate state. The efficiency of MCMC depends on the choice of the proposal PDF. The proposal PDF should be simple for easily simulating the candidate samples and resemble the target PDF closely to improve the acceptance rate of the candidate samples. Typically, there are several widely-used proposal PDFs, including the symmetric PDF $g(\tilde{\boldsymbol{\theta}}_{k+1} | \boldsymbol{\theta}_k) = g(\boldsymbol{\theta}_k | \tilde{\boldsymbol{\theta}}_{k+1})$ and the independent PDF which indicates the candidate sample doesn't depend on the current state, i.e. $g(\tilde{\boldsymbol{\theta}}_{k+1} | \boldsymbol{\theta}_k) = g(\tilde{\boldsymbol{\theta}}_{k+1})$. For the symmetric proposal PDF, the ratio in step 2 can be

rewritten as $r = \frac{f(\tilde{\boldsymbol{\theta}}_{k+1})}{f(\boldsymbol{\theta}_k)}$.

Compared to the ARS algorithm, the samples generated by MCMC simulation are correlated since the next state of the chain depends on the current state. The Markov chain will be stationary after the burn-in period and all the states of the chain will be distributed according to the target PDF $f(\boldsymbol{\theta})$.

The MCMC algorithm is particularly efficient when the samples that are distributed according to the target PDF $f(\boldsymbol{\theta})$ are already available. Each initial sample will be regarded as the initial state to generate a new Markov chain by MCMC and multiple Markov chains can be obtained. Since the initial samples are distributed according

to the target PDF, the Markov chains generated by MCMC in this way are always in the stationary state and all the simulated states will be distributed according to the target PDF. This feature avoids the burn-in problem for the single Markov chain.

3) Modified MCMC

The effective number of the simulated samples by MCMC depends on the correlation of the samples and one source of the correlation is the repeated samples (Papaioannou, Betz et al. 2015). The correlation of the samples is extensively high for the high-dimensional vector $\boldsymbol{\theta}$ due to many repeated samples. It has been found that the acceptance rate of the candidate samples tends to zero (Au and Beck 2001) for the high-dimensional case due to the n -variate proposal PDF. To solve the low acceptance rate problem, Au and Beck modified the original MCMC and proposed the modified MCMC for the reliability problem based on the idea of using n univariate proposal PDFs. In such case, each dimension of the candidate sample is separately generated by the corresponding proposal PDFs and the new candidate samples will be checked for failure to determine its acceptance. Let $f(\boldsymbol{\theta}) = I_F(\boldsymbol{\theta})p(\boldsymbol{\theta})$ be the target PDF and the joint PDF of all the variables can be written as $p(\boldsymbol{\theta}) = \prod_{i=1}^n p_i(\theta_i)$. The procedure of the modified MCMC is given as follow:

Modified MCMC algorithm

Initialization: $g_i(\cdot|\theta_i)$, the proposal PDF for the i -th dimension

$\boldsymbol{\theta}_k = [\theta_k^1, \dots, \theta_k^n]$, current state of the Markov chain

Step 1: generate the candidate sample $\tilde{\boldsymbol{\theta}}_{k+1}$

for $i = 1 : n$

generate the i -th pre-candidate sample $\xi^i \sim g_i(\xi^i|\theta_k^i)$ and $u_i \sim U_{[0,1]}$

compute the ratio $r_i = \frac{p(\xi^i)g_i(\theta_k^i|\xi^i)}{p(\theta_k^i)g_i(\xi^i|\theta_k^i)}$

$$\tilde{\theta}_{k+1}^i = \begin{cases} \xi^i, & \text{if } r_i > u \\ \theta_k^i, & \text{otherwise} \end{cases}$$

end

the candidate sample is obtained as $\tilde{\boldsymbol{\theta}}_{k+1} = [\tilde{\theta}_{k+1}^1, \dots, \tilde{\theta}_{k+1}^n]$

Step 2: determine the next state $\boldsymbol{\theta}_{k+1}$

$$\boldsymbol{\theta}_{k+1} = \begin{cases} \tilde{\boldsymbol{\theta}}_{k+1}, & \text{if } \tilde{\boldsymbol{\theta}}_{k+1} \in F \\ \boldsymbol{\theta}_k, & \text{otherwise} \end{cases}$$

3.3.2 Review of Subset Simulation

Subset Simulation (SS) is proposed by Au and Beck (Au and Beck 2001, Au and Beck 2003) for efficiently estimating the small failure probability. The direct Monte Carlo simulation (MC) is a straightforward method for the estimation of failure probability, while it will be computationally expensive for estimating the small failure probability due to the difficulty of obtaining the failure samples. Instead of directly simulating the failure samples for the rare failure event, SS simulates the failure samples for a sequence of more frequent failure events $F_1 \supset F_2 \supset \dots \supset F_{m+1} = F$ and the failure samples for the current failure event can be efficiently simulated based on the failure samples for the previous failure event. Let $P(F)$ defined as follow be the failure probability to be estimated.

$$\begin{aligned} P(F) &= \int_{\Theta} I_F(\boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta}, \\ I_F(\boldsymbol{\theta}) &= \begin{cases} 1, & \boldsymbol{\theta} \in F \\ 0, & \boldsymbol{\theta} \notin F \end{cases} \\ \text{and } F &= \{g(\boldsymbol{\theta}) \geq b\} \end{aligned} \tag{3.18}$$

The corresponding sequence of failure events are defined as follow

$$\begin{aligned} F_i &= \{g(\boldsymbol{\theta}) \geq b_i\}, \text{ for } i = 1, \dots, m+1 \\ b_1 &\leq b_2 \leq \dots \leq b_{m+1} = b \end{aligned} \tag{3.19}$$

such that

$$\begin{aligned} P(F_i|F_{i-1}) &= p_0, \text{ for } i = 1, \dots, m \\ \text{and } P(F_0) &= 1 \end{aligned} \quad (3.20)$$

The estimator for $P(F)$ can be written as follow

$$P(F) = P(F_m) = P(F_0)P(F_1|F_0) \cdots P(F_{m+1}|F_m) \quad (3.21)$$

Let $i = 1, \dots, m$ denote the simulation level for SS and $p(\theta|F_i)$ denote the PDF of the failure samples for the failure event F_i . Let N_{ss} be the number of the simulated samples for estimating $P(F_i|F_{i-1})$ at the i -th simulation level. At the i -th simulation level, the value of b_i is adaptively determined based on the simulated samples $\{\theta_k^i\}_{k=1, \dots, N}$ that are distributed according to $p(\theta|F_{i-1})$.

3.4 Proposed sampling algorithm

3.4.1 Acceptance-Rejection Sampling with Subset Simulation

To simulate samples from the auxiliary PDF $\pi(\theta, \varphi)$ defined in Equation (3.4), the ARS and MCMC algorithm may be used since they don't require the exact value of the normalizing constant C which is related to the expectation. However, it is a nontrivial task for MCMC algorithm to determine whether the Markov chain has passed the burn-in period and reached the stationary state. ARS is often adopted to simulate samples from $\pi(\theta, \varphi)$ due to its convenience and the independence between the samples. Let $p(\theta, \varphi) = p_\theta(\theta)p_\varphi(\varphi)$ denote the joint probability density function of the random variables θ and the design variables φ . Let $g(\theta, \varphi)$ denote the proposal PDF for simulating the candidate samples. The process of simulating samples from $\pi(\theta, \varphi)$ by ARS is given as follows:

Step 1: simulate a candidate sample pair $[\theta, \varphi]$ from $g(\theta, \varphi)$ and simulate u from $U_{[0,1]}$.

Step 2: compute the acceptance ratio $r = \frac{h(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(\boldsymbol{\theta}, \boldsymbol{\varphi})}{M \cdot g(\boldsymbol{\theta}, \boldsymbol{\varphi})}$.

Step 3: accept the candidate sample pair $[\boldsymbol{\theta}, \boldsymbol{\varphi}]$ if $r > u$. Otherwise, reject it.

The process can be run as many times as the required number of samples are obtained. Since the prior information about $\pi(\boldsymbol{\theta}, \boldsymbol{\varphi})$ is not available for most cases, the proposal PDF $g(\boldsymbol{\theta}, \boldsymbol{\varphi})$ is often chosen as the original PDF $p(\boldsymbol{\theta}, \boldsymbol{\varphi})$. Therefore, the acceptance ratio becomes $r = \frac{h(\boldsymbol{\theta}, \boldsymbol{\varphi})}{M}$ and the constant M can be chosen with the satisfactory of $M \geq \max_{\boldsymbol{\theta}, \boldsymbol{\varphi}} \{h(\boldsymbol{\theta}, \boldsymbol{\varphi})\}$. However, this choice of the proposal PDF may lead to a very low acceptance rate for some practical engineering applications and this will make ARS being computationally expensive. For the case considered in this work, the acceptance rate is as low as 4% and the original ARS will be inefficient for simulating samples from $\pi(\boldsymbol{\theta}, \boldsymbol{\varphi})$. A very efficient sampling algorithm is proposed as follows based on ARS and SS.

1) Basic idea

Let $p(u)$ denote the uniform PDF $U_{[0,1]}$. In terms of the process of ARS algorithm, it can be seen that the accepted sample pairs $[\boldsymbol{\theta}, \boldsymbol{\varphi}, u]$ are distributed according to the PDF defined as follows:

$$q(\boldsymbol{\theta}, \boldsymbol{\varphi}, u | F) = \frac{I_F(\boldsymbol{\theta}, \boldsymbol{\varphi}, u)}{P_F} g(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(u) \quad (3.22)$$

where the indicator function is given by:

$$I_F(\boldsymbol{\theta}, \boldsymbol{\varphi}, u) = \begin{cases} 1, & \text{if } l(\boldsymbol{\theta}, \boldsymbol{\varphi}, u) \geq 0 \\ 0, & \text{otherwise} \end{cases} \quad (3.23)$$

and $l(\boldsymbol{\theta}, \boldsymbol{\varphi}, u) = \frac{h(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(\boldsymbol{\theta}, \boldsymbol{\varphi})}{M \cdot g(\boldsymbol{\theta}, \boldsymbol{\varphi})} - u$

The failure probability P_F , which is a constant, is given as follows:

$$P_F = \int_{\Theta} \int_{\Phi} \int_U I_F(\boldsymbol{\theta}, \boldsymbol{\varphi}, u) g(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(u) d\boldsymbol{\theta} d\boldsymbol{\varphi} du = \frac{E_{\boldsymbol{\theta}, \boldsymbol{\varphi}}[h(\boldsymbol{\theta}, \boldsymbol{\varphi})]}{M} \quad (3.24)$$

P_F is equal to the acceptance rate of the ARS algorithm.

Therefore, the sample pairs $[\boldsymbol{\theta}, \boldsymbol{\varphi}]$ which are distributed according to $\pi(\boldsymbol{\theta}, \boldsymbol{\varphi})$ can be obtained by taking the $\boldsymbol{\theta}$ and $\boldsymbol{\varphi}$ components of the failure sample pairs $[\boldsymbol{\theta}, \boldsymbol{\varphi}, u]$ which are distributed according to $q(\boldsymbol{\theta}, \boldsymbol{\varphi}, u | F)$. For the cases where the acceptance rate of the ARS algorithm is very low, direct MC simulation will be very computationally expensive for simulating failure samples from $q(\boldsymbol{\theta}, \boldsymbol{\varphi}, u | F)$. In such cases where the sampling task is to simulate failure samples from $q(\boldsymbol{\theta}, \boldsymbol{\varphi}, u | F)$, the SS algorithm can be adopted to efficiently obtain failure samples for the rare failure event F (Au and Beck 2001).

2) Modification of the formula of the failure event F

In the above definition of the failure event F in Equation (3.19), the constant M is required before running the SS algorithm. It should satisfy

$$M \geq \max_{\boldsymbol{\theta}, \boldsymbol{\varphi}} \left\{ \frac{h(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(\boldsymbol{\theta}, \boldsymbol{\varphi})}{g(\boldsymbol{\theta}, \boldsymbol{\varphi})} \right\} \text{ and the estimate of } \max_{\boldsymbol{\theta}, \boldsymbol{\varphi}} \left\{ \frac{h(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(\boldsymbol{\theta}, \boldsymbol{\varphi})}{g(\boldsymbol{\theta}, \boldsymbol{\varphi})} \right\} \text{ is not an}$$

easy task especially when no evaluations of $h(\boldsymbol{\theta}, \boldsymbol{\varphi})$ are available. Therefore, the value of M may be updated with the SS simulation levels based on the evaluated values of $h(\boldsymbol{\theta}, \boldsymbol{\varphi})$. However, the modification in M requires us to rerun all SS simulation levels when the failure event F is defined as Equation (3.19). This will lead to additional computational cost for the sampling process.

The formula of the failure event F is rewritten as follows:

$$F = \{X(\boldsymbol{\theta}, \boldsymbol{\varphi}, u) \geq \ln M\}$$

$$X(\boldsymbol{\theta}, \boldsymbol{\varphi}, u) = \ln \left(\frac{h(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(\boldsymbol{\theta}, \boldsymbol{\varphi})}{u \cdot g(\boldsymbol{\theta}, \boldsymbol{\varphi})} \right) \quad (3.25)$$

The logarithm is used such that X is a well-defined random variable based on the definition of the driving variable proposed by DiazDelaO (DiazDelaO, Garbuno-Inigo et al. 2017). This definition of X facilitates the adaptively chosen of M and the details will be given in the following part.

By the modification in Equation (3.25), the left part of the limit state equation doesn't depend on M and M is shifted to the right part which gives the threshold. Therefore, the simulation results of the previous SS simulation levels can still be used even if the value of M has been changed based on the simulation results of the current SS simulation level. This can be seen in the definition of the sequence of frequent conditional failure events in SS.

In terms of the failure event F defined in Equation (3.25), the sequence of the conditional failure events $F_0 \supset F_1 \supset \dots \supset F_m = F$ are given as follows:

$$P(F_0) = 1, F_i = \{X(\boldsymbol{\theta}, \boldsymbol{\varphi}, u) \geq b_i\}, i = 1, \dots, m \quad (3.26)$$

and the corresponding sequence of thresholds $\{b_i\}$ are chosen such that

$$P(F_i | F_{i-1}) = p_{ss}, i = 1, \dots, m-1 \quad (3.27)$$

and $b_m = \ln M$

Therefore, the update in M only changes the value of b_m and a larger M will require more SS simulation levels to obtain the failure samples.

3) Adaptive selection of M

The selection of M is a nontrivial task. M should satisfy

$$M \geq \max_{\boldsymbol{\theta}, \boldsymbol{\varphi}} \left\{ \frac{h(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(\boldsymbol{\theta}, \boldsymbol{\varphi})}{g(\boldsymbol{\theta}, \boldsymbol{\varphi})} \right\} \text{ to ensure that the simulated samples will be distributed}$$

according to the target PDF $\pi(\boldsymbol{\theta}, \boldsymbol{\varphi})$. The maximum may be determined based on a larger number of evaluations of $h(\boldsymbol{\theta}, \boldsymbol{\varphi})$. A larger M is preferred to reduce the risk

$$\text{of } M < \max_{\boldsymbol{\theta}, \boldsymbol{\varphi}} \left\{ \frac{h(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(\boldsymbol{\theta}, \boldsymbol{\varphi})}{g(\boldsymbol{\theta}, \boldsymbol{\varphi})} \right\}. \text{ However, the increase in } M \text{ will require more SS}$$

simulation levels. This will increase the computational cost and the correlation of the simulated samples as well. Therefore, the theoretical optimal selection of M is

$$M = \max_{\theta, \varphi} \left\{ \frac{h(\theta, \varphi) p(\theta, \varphi)}{g(\theta, \varphi)} \right\}.$$

This section proposes a method for adaptively selecting M based on the failure samples obtained in the SS simulation levels. If $a \geq \ln \left\{ \max_{\theta, \varphi} \left\{ \frac{h(\theta, \varphi) p(\theta, \varphi)}{g(\theta, \varphi)} \right\} \right\}$,

$P(X \geq a)$ is given by:

$$\begin{aligned} P(X \geq a) &= P \left(\ln \left(\frac{h(\theta, \varphi) p(\theta, \varphi)}{u \cdot g(\theta, \varphi)} \right) \geq a \right) \\ &= P \left(\frac{h(\theta, \varphi) p(\theta, \varphi)}{u \cdot g(\theta, \varphi)} \geq e^a \right) \\ &= \int_{\theta, \varphi} \int_0^1 I \left(\frac{h(\theta, \varphi) p(\theta, \varphi)}{u \cdot g(\theta, \varphi)} \geq e^a \right) g(\theta, \varphi) du d\theta d\varphi \quad (3.28) \\ &= \int_{\theta, \varphi} \int_0^1 I \left(u \leq \frac{h(\theta, \varphi) p(\theta, \varphi)}{e^a \cdot g(\theta, \varphi)} \right) du g(\theta, \varphi) d\theta d\varphi \\ &= \int_{\theta, \varphi} \frac{h(\theta, \varphi) p(\theta, \varphi)}{e^a \cdot g(\theta, \varphi)} g(\theta, \varphi) d\theta d\varphi \\ &= e^{-a} E_{\theta, \varphi} [h(\theta, \varphi)] = \frac{e^{-a}}{C} \end{aligned}$$

where C is the constant defined in Equation (3.5). By taking the logarithm on both sides, Equation (3.28) can be rewritten as:

$$\ln(P(X \geq a)) = -a - \ln C \quad (3.29)$$

For any specified a , the corresponding failure probability $P(X \geq a)$ can be estimated based on the failure samples obtained in SS. By plotting $\ln(P(X \geq a))$ versus a , it can be seen from Equation (3.29) that the curve will become a straight line with a slope of -1 for the part where $a \geq \ln \left(\max_{\theta, \varphi} \left(\frac{h(\theta, \varphi) p(\theta, \varphi)}{g(\theta, \varphi)} \right) \right)$. This

important characteristic trend of the curve is useful for us to determine the value of

$\max_{\theta, \varphi} \left(\frac{h(\theta, \varphi) p(\theta, \varphi)}{g(\theta, \varphi)} \right)$, which suggests the selection of M . As shown in Figure 3.1,

$a_{\min}$ denotes the location where the characteristic trend begins. The maximum can

be estimated as $\max_{\theta, \varphi} \left(\frac{h(\theta, \varphi) p(\theta, \varphi)}{g(\theta, \varphi)} \right) = e^{a_{\min}}$.

Figure 3.1 Characteristic trend of the plot of $\ln(P(X \geq a))$ versus a

The above arguments provide a theoretical basis for determining M and the SS simulation level to stop. In the plotted curve of a Subset Simulation run, random deviation from theoretical trend will exist due to the statistical estimation error of $P(X \geq a)$, which depends on the number of samples obtained in each SS simulation level. Taking the random deviation into account, the simulation stops at the simulation level believed to have the characteristic trend.

4) Framework of ARS-SS

Based on the process of ARS algorithm, the original task of simulating samples from the target PDF $\pi(\boldsymbol{\theta}, \boldsymbol{\phi})$ is equivalent to simulating failure samples from the proposed PDF $q(\boldsymbol{\theta}, \boldsymbol{\phi}, u | F)$. Therefore, the failure samples can be efficiently obtained by the SS algorithm. Moreover, a method is proposed for adaptively selecting M which is a critical constant in the formula of the definition of $q(\boldsymbol{\theta}, \boldsymbol{\phi}, u | F)$ and this method also gives the stopping criterion for the SS algorithm. The procedure of the ARS-SS algorithm is given as follows:

ARS-SS algorithm

Input: $g(\boldsymbol{\theta}, \boldsymbol{\phi})$, the proposal PDF in ARS

N_{ss} , the number of samples simulated in each SS simulation level

p_{ss} , the conditional failure probability for SS

N_{ϕ} , the required number of samples for the target PDF $\pi(\boldsymbol{\theta}, \boldsymbol{\phi})$

Algorithm: at the SS simulation level i ,

Step 1: based on the retained failure samples that are distributed according to $q(\boldsymbol{\theta}, \boldsymbol{\phi}, u | F_{i-1})$, use modified MCMC algorithm to simulate additional failure samples until the number of failure samples reaches N_{ss} . (for the SS simulation level 1, the samples are directly simulated by MC.)

Step 2: based on the N_{ss} failure samples from $q(\boldsymbol{\theta}, \boldsymbol{\phi}, u | F_{i-1})$, determine b_i with the satisfactory of $P(F_i | F_{i-1}) = p_{ss}$. Retain the failure samples for F_i .

Step 3: plot $P(X \geq a)$ versus a and check whether the characteristic trend begins. If yes, terminate the SS process and determine $a_{\min}$ using the plot. Let $m = i$ and reset $b_m = a_{\min}$. Retain the failure samples for $F_m = F$ and these failure samples will be distributed according to $q(\boldsymbol{\theta}, \boldsymbol{\phi}, u | F)$.

After the SS algorithm terminates, use modified MCMC algorithm to simulate additional failure samples until the number of failure samples reaches N_{ϕ} . Take the $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ of the N_{ϕ} failure samples from $q(\boldsymbol{\theta}, \boldsymbol{\phi}, u | F)$.

3.5 Simultaneous Perturbation Stochastic Approximation

This section reviews the SPSA algorithm which is computationally efficient compared to other stochastic optimization algorithms listed in the literature review. The optimal solution obtained by SPSA is used as benchmark in this work.

1) Basic SPSA

The main idea behind SPSA is that in the long run the carefully chosen simultaneous random perturbation in all components of the design vector $\boldsymbol{\varphi}$ is able to provide much information for the optimization purpose compared to the set of a single perturbation in one dimension at each time. In terms of this idea, the gradient vector is approximated by two evaluations of the objective function for the chosen two perturbation design configurations. The iteration process of SPSA is given as follow:

$$\boldsymbol{\varphi}_{k+1} = \boldsymbol{\varphi}_k - \alpha_k \hat{\mathbf{g}}(\Omega_N, \boldsymbol{\varphi}_k) \quad (3.30)$$

and the gradient vector is estimated as follow:

$$\hat{\mathbf{g}}_{k,j} = \frac{\hat{H}(\boldsymbol{\varphi}_k + c_k \boldsymbol{\Delta}_k, \Omega_{N,k}) - \hat{H}(\boldsymbol{\varphi}_k - c_k \boldsymbol{\Delta}_k, \Omega_{N,k})}{2c_k \Delta_{k,j}} \quad (3.31)$$

where c_k and $\boldsymbol{\Delta}_k$ define the perturbation of the design variables, and α_k determines the size of the moving step of the design vector. k is the index of the iteration which starts from 1 and $\boldsymbol{\varphi}_1$ is the chosen starting point of the iteration. $\Omega_{N,k}$ is a set containing a number of samples representing $\boldsymbol{\theta}$ and the same set is used to estimate both of the objective function in the numerator. $\boldsymbol{\Delta}_k$ is a d -dimensional vector of mutually independent random variables that provides the direction of the perturbation in each component of $\boldsymbol{\varphi}_k$. Typically, the component in $\boldsymbol{\Delta}_k$ is chosen from a symmetric Bernoulli ± 1 distribution. To ensure the identified optimum converges to the actual optimum, the choice of parameter should satisfy the following: $\alpha_k = \frac{\alpha_1}{(k+w)^r}$ and $c_k = \frac{c_1}{k^\varsigma}$, where $r-4\varsigma < 0$, $r-\varsigma > 0.5$, with

$w, \varsigma > 0$ and $0 < r \leq 1$. Suggestions on selection of these parameters are given in (Berg 2004). w is chosen to be 10% of the expected number of iteration for convergence and c_1 is chosen to be the standard deviation of the estimation error of $\hat{H}(\boldsymbol{\varphi}_0)$. α_1 is adaptively chosen such that the desired step size for the first iteration is obtained and this can be determined based on the average of the gradient vector in the first iteration.

2) Implementation suggestions

Based on the basic SPSA algorithm, an adaptive initial step size selection method has been proposed by Keiichi Ito. This adaptive chosen method reduces the risk of divergence of the SPSA iteration and allows us to identify a better solution than basic SPSA by selecting a larger initial step size. The initial step size is chosen as $\alpha_1 = \delta\varphi_{0\min} \cdot (1+w)^r$. Let $\boldsymbol{\varphi}_b$ denote the design vector which has the minimum objective function of all the estimations in previous iterations. The adaptive initial step size method is given as follows:

Adaptive initial step size method

At iteration k , **if** $\min\{\hat{H}(\boldsymbol{\varphi}_k + c_k \boldsymbol{\Delta}_k), \hat{H}(\boldsymbol{\varphi}_k - c_k \boldsymbol{\Delta}_k)\} \geq \hat{H}(\boldsymbol{\varphi}_1)$ **then**

$$\boldsymbol{\varphi}_{k+1} = \boldsymbol{\varphi}_b, \alpha_1 = \alpha_1 / 2$$

endif

The blocking rules are suggested to be used for the bounded optimization. Let $\Phi = (\varphi_{1l}, \varphi_{1u}) \otimes \cdots \otimes (\varphi_{dl}, \varphi_{du})$ denote the bounded space of the design variables. Since the design vector $\boldsymbol{\varphi}_{k+1}$ is regarded as the center to estimate the gradient in the next iteration, $\boldsymbol{\varphi}_{k+1}$ should lie in the region $\Omega_{k+1} = (\boldsymbol{\varphi}_{1l} + c_{k+1}, \boldsymbol{\varphi}_{1u} - c_{k+1}) \otimes \cdots \otimes (\boldsymbol{\varphi}_{dl} + c_{k+1}, \boldsymbol{\varphi}_{du} - c_{k+1})$. $\boldsymbol{\varphi}_{k+1}$ is blocked to move to the points lie outside Ω_{k+1} .

The SPSA iterations will terminate when the specified stopping criterion is satisfied. The stopping criterion can be based on either the change in the objective function value or the design vector.

Using the adaptive initial step size selection method and the blocking rules for the bounded design space, the SPSA algorithm used in this work is given as follows:

SPSA algorithm

Initialization: select appropriate values for the parameters α_1 , c_1 , r and ζ .

Determine the maximum number of the iterations $N_{\max}$ and set $w = 0.1 * N_{\max}$.

Choose the starting point of the iteration $\boldsymbol{\varphi}_1 \in \Omega_1$.

Algorithm: evaluate $\hat{H}(\boldsymbol{\varphi}_1)$ and let $\boldsymbol{\varphi}_b = \boldsymbol{\varphi}_1$

for $k = 1 : N_{\max}$ **do**

 Determine Δ_k and c_k

 Evaluate $\hat{H}(\boldsymbol{\varphi}_k - c_k \Delta_k, \Omega_{N,k})$ and $\hat{H}(\boldsymbol{\varphi}_k + c_k \Delta_k, \Omega_{N,k})$

 Renew $\boldsymbol{\varphi}_b$ if better minimum in $\hat{H}(\cdot)$ is obtained.

 Compute the gradient vector $\hat{g}(\boldsymbol{\varphi}_k)$ and α_k

 Compute $\boldsymbol{\varphi}_{k+1} = \boldsymbol{\varphi}_k - \alpha_k \hat{g}(\boldsymbol{\varphi}_k)$

 Check the blocking rules and set

$$\boldsymbol{\varphi}_{k+1,i} = \min\left(\max\left(\boldsymbol{\varphi}_{il} + c_{k+1}, \boldsymbol{\varphi}_{k+1,i}\right), \boldsymbol{\varphi}_{iu} - c_{k+1}\right)$$

 Perform the adaptive initial step size algorithm

 Check the stopping criterion and break out the for loop if it is satisfied.

Endfor

3.6 Illustrative examples

In this section, SSO is used to identify the optimal design of the structural control system for a spar buoy floating wind turbine and ARS-SS is adopted to simulate samples which are required by SSO. Two cases are considered in this work. Case 1 involves two design variables as the structural control system has only a single

damper, while case 2 has 4 design variables since the structural control system consists of two dampers.

3.6.1 Case description

In this section, the TMD (tuned mass damper) coefficients for minimizing the expected lifetime fore-aft fatigue damage at the tower base of a spar-type floating wind turbine are identified. This work investigates a representative 5-MW wind turbine model (Jonkman, Butterfield et al. 2009), which has been developed by NREL for simulation and widely used by researchers for testing the design of a wind turbine. This wind turbine is a three-bladed machine with a 90-m hub height and a 120-m rotor diameter. As shown in Figure 3.2, the NREL 5-MW baseline wind turbine is supported by a ballast-stabilized spar buoy concept, which is named as OC3-Hywind spar buoy by Jonkman (Jonkman 2010). The spar buoy is connected with the seabed by three catenary mooring lines. More details about the NREL 5-MW wind turbine can be found in Jonkman's thesis (Jonkman 2007) and Table 3.1 lists the natural frequencies of the spar buoy floating wind turbine. The baseline control strategy for OC3-Hywind system, which is developed by Jonkman, is used for improving the damping of the problematic motions of the platform.

Figure 3.2 Illustrations of the NREL 5-MW wind turbine on the OC3-Hywind spar buoy (Jonkman 2010)

Table 3.1 Natural Frequencies of OC3-Hywind Floating Wind Turbine (Matha 2010)

Mode	Natural Frequency (rad/s)	Mode	Natural Frequency (rad/s)
Platform roll	0.215	Platform surge	0.050
Platform pitch	0.215	Platform sway	0.050
Platform yaw	0.760	Platform heave	0.203
1 st Tower F-A	2.972	1 st Tower S-S	2.872

Figure 3.3 Typical installation of Platform-based Tuned Mass Damper for a floating spar buoy wind turbine

This design work is aimed to determine the optimal spring and damping coefficients of the TMDs which are located in the platform and the nacelle. Let PTMD and

NTMD denote the TMD in the platform and the nacelle, respectively. The installation of the PTMD is shown in Figure 3.3 and the NTMD is installed in the topside nacelle other than inside the platform. Due to the limited moving space inside the platform and the nacelle and the difficult marine installation, the mass of PTMD and NTMD are selected as 200,000 kg and 20,000 kg, about 2.5% of the platform weight and the nacelle mass, respectively. This work considers the TMDs which are allowed to move in the fore-aft direction. Based on the dimensions of the platform and the nacelle, the PTMD displacement range is defined as ± 4 m and the NTMD displacement range is defined as ± 7 m. In FAST-SC simulation, the TMD stop position is modeled by the large spring and damping coefficients. The neutral position of PTMD is defined as 10 m below the sea water level and the neutral position of NTMD is defined as the center of the nacelle. According to the natural frequencies shown in Table 3.1, the design space of TMD parameters are selected as PTMD natural frequency $\omega_{PTMD} \in [0, 5]$ rad/s, PTMD damping ratio $\xi_{PTMD} \in [0, 1]$, NTMD natural frequency $\omega_{NTMD} \in [0, 5]$ rad/s, NTMD damping ratio $\xi_{NTMD} \in [0, 1]$.

Two cases are studied in this work. Case 1 considers a single PTMD and the control system of case 2 includes two dampers, PTMD and NTMD. The design vector for case 1 is $[\omega_{PTMD}, \xi_{PTMD}]$, and case 2 has a 4-dimensional design vector $[\omega_{PTMD}, \xi_{PTMD}, \omega_{NTMD}, \xi_{NTMD}]$. Case 1 and case 2 are summarized as follows:

$$\text{Case 1: } \boldsymbol{\varphi}_1^* = \arg \min_{\boldsymbol{\varphi}_1 \in \Phi_1} \int_0 h(\boldsymbol{\theta}, \boldsymbol{\varphi}_1) p(\boldsymbol{\theta}) d\boldsymbol{\theta}$$

$$\text{Case 2: } \boldsymbol{\varphi}_2^* = \arg \min_{\boldsymbol{\varphi}_2 \in \Phi_2} \int_0 h(\boldsymbol{\theta}, \boldsymbol{\varphi}_2) p(\boldsymbol{\theta}) d\boldsymbol{\theta}$$

where the uncertain variables of interest are in the vector $\boldsymbol{\theta} = [U, H_S, T_P, \theta_{angle}, \boldsymbol{\theta}_{wind}, \boldsymbol{\theta}_{wave}]$. The joint distribution of the variables $[U, H_S, T_P, \theta_{angle}]$ can be found in Chapter 2 and $[\boldsymbol{\theta}_{wind}, \boldsymbol{\theta}_{wave}]$ includes a large number of white Gaussian noise variables that are used to generate wind speed and

wave height time series. The design vector for case 1 is $\boldsymbol{\varphi}_1 = [\omega_{PTMD}, \xi_{PTMD}]$ with the corresponding design space $\Phi_1 = [0,5] \otimes [0,1]$. The design vector for case 2 is a 4-dimensional vector $\boldsymbol{\varphi}_2 = [\omega_{PTMD}, \xi_{PTMD}, \omega_{NTMD}, \xi_{NTMD}]$ with the corresponding design space $\Phi_2 = [0,5] \otimes [0,1] \otimes [0,5] \otimes [0,1]$.

In this study, the wind speed at the hub height is determined by using a power law of 0.14 in this study. The wind data are simulated by using Kaimal spectra and a turbulence intensity of 15% and 20,000,000 white noise random variables are required when transferring the wind spectrum to short-term time series. The stochastic wave data are generated by JONSWAP spectrum and 20,000 white noise random variables.

In this work, FASTv8 is used to simulate the short-term nonlinear dynamic response of the wind turbine system with all degrees of freedom enabled. The short term period is selected as 600 s and each simulation run 630 s as the wind turbine generator torque and blade pitch motion need the first 30 s to reach normal operation state. For all simulations, the TMD initial displacement is zero. Based on the simulation results, the post-processor MLife (Hayman 2012) (MATLAB-based estimator of fatigue life) is utilized to calculate the short-term fatigue damage at the tower base, with the S/N (stress/number of cycles to failure) curve slope of 3. The workstation used in the simulation has 28 processors of Intel Xeon E5-2680 2.40GHz and 128 GB memory.

3.6.2 Initial parameters for the proposed algorithm

The SSO algorithm is used to perform the optimization of the natural frequencies and damping ratios of the TMDs in this study. To save more computational cost, samples distributed according to the auxiliary PDF in the first optimization level are obtained by the proposed sampling algorithm ARS-SS. For ARS-SS, N_{ss} is chosen as 400 and p_{ss} is defined as 0.2. When performing SSO, the number of samples obtained in each optimization level is $N_{\phi} = 2,000$ for case 1 and $N_{\phi} = 4,000$ for

case 1. The shape of the admissible subset is hyper-ellipse and the size ratio ρ is selected as 20% . The adaptive optimization will stop when $\hat{S}(I|\Phi)$ is around 0.8.

The effectiveness of the optimal TMD design is checked by comparing the lifetime fatigue damage of the spar buoy wind turbine with TMD with that of the baseline case without TMD. The expected value of the lifetime fatigue damage is estimated by Monte Carlo simulation with 400 samples.

The final optimal design is identified by performing SPSA. The SPSA searches within the initial design space and starts from the sub-optimal design identified by SSO in the design space. The parameter selection is given as follows: $w=100, r=0.8, \zeta=0.25, c_1=0.1, N=400$. The selection of α_1 depends on the average of the gradient vector at the starting point and follows the guidelines given in section 3.4 and blocking rules are adopted. SPSA stops when the norm $\|\boldsymbol{\Phi}_{k+1} - \boldsymbol{\Phi}_k\|_\infty$ for each of the five last iterations is less than 0.5% of the dimension of the search space. The converged solution is assumed to be the final optimal solution.

3.6.3 Results and discussion

The efficiency of ARS-SS for case 1 is demonstrated in Table 3.2. Samples from the first ARS-SS level $p(\boldsymbol{\theta})$, the second ARS-SS level $q(\boldsymbol{\theta}|F_1)$ and the target PDF $\pi(\boldsymbol{\theta}, \boldsymbol{\varphi})$ with respect to $\{U, H_s, T_p\}$ are shown in Figure 3.5 and 3.6. Because of the lack of prior knowledge about the short-term fatigue damage function, the PDF $g(\boldsymbol{\theta}, \boldsymbol{\varphi})$ is chosen as the original PDF $p(\boldsymbol{\theta}, \boldsymbol{\varphi})$ of the random variables. As shown in Table 3.2 and Figure 3.4, two SS simulation levels are performed to seek the $a_{\min}$. The information of these two simulation levels indicates that the acceptance rate of ARS will be about 5.8% as it should be equal to the “failure” probability. The failure samples in the second simulation level are used as seeds to generate more samples by MMH algorithm which can be used in the first optimization level. As γ is calculated to be 2 , the effective number of these 2,000 correlated samples is about 667 uncorrelated samples. To obtain the same number of samples, ARS requires 11,494 evaluations of the short-term fatigue damage approximately. In

total, ARS-SS requires 2,604 evaluations. The ratio of the number of evaluations required by ARS to ARS-SS is about 4 .

For case 2, the acceptance rate of ARS is 1.6% and this decrease may be caused by the addition of the design variables. As presented in Table 3.3, ARS-SS requires 4,880 evaluations and ARS needs 83,380 evaluations to obtain the equal number of effective samples as ARS-SS. The ratio of the number of evaluations required by ARS-SS to ARS is about 17 . The advantage of ARS-SS is more obvious for case 2 as the dimensions of the design vector increase.

Figure 3.4 Plot of $\ln(P(X \geq a))$ versus a based on samples simulated by ARS-SS for case 1

Table 3.2 Sampling efficiency of ARS-SS for case 1

	ARS-SS				ARS	Computational cost ratio ($\frac{ARS}{ARS-SS}$)
	Sampling level 1	Sampling level 2	MCMC simulation	In total		
Number of FAST-SC simulations	400	320	1884	2604	11494	4 : 1
Actual time used (workstation with parallel computing)				26 hours	114 hours	

Table 3.3 Sampling efficiency of ARS-SS for case 2

	ARS-SS					ARS	Computational cost ratio ($\frac{ARS}{ARS-SS}$)
	Sampling level 1	Sampling level 2	Sampling level 3	MCMC simulation	In total		
Number of FAST-SC simulations	400	320	320	3840	4880	83380	17 : 1
Actual time used (workstation with parallel computing)					48.8 hours	833.8 hours

Figure 3.5 Simulated samples by ARS-SS (mean wind speed U versus significant wave height H_s)

Figure 3.6 Simulated samples by ARS-SS (significant wave height H_s versus spectral peak period T_p)

Table 3.4 Results for a sample run of SSO with ARS-SS for case 1

	$\hat{S}(I_1 \Phi_1)$	Suboptimal design		Fatigue damage reduction (%)
		ω_{PTMD} (rad/s)	ξ_{PTMD}	
Optimization results	0.806	2.718	0.3117	15

Table 3.5 Results for a sample run of SSO with ARS-SS for case 2

	$\hat{S}(I_1 \Phi_2)$	$\hat{S}(I_2 I_1)$	$\hat{S}(I_3 I_2)$	Suboptimal design $[\omega_{PTMD}, \xi_{PTMD}, \omega_{NTMD}, \xi_{NTMD}]$	Fatigue damage reduction (%)
Optimization results	0.72	0.78	0.85	[2.75, 0.26, 3.5, 0.3]	28

Results for a sample run of SSO with ARS-SS for case 1 are presented in Table 3.4. The iteration process is stopped at the first optimization level as $\hat{S}(I_1 | \Phi_1)$ is larger than 0.8. The optimal solution identified by SSO, shown as a red star point in Figure 3.7, is (2.7184, 0.3117), corresponding to 1.4779×10^6 N/m for the PTMD spring constant and 3.3893×10^5 N·s/m for the PTMD damping coefficient. The optimal PTMD can reduce the fore-aft fatigue damage at the tower base by 15%.

Table 3.5 presents the results for a sample run of SSO with ARS-SS for case 2. The optimization iteration terminates at the third optimization level where $\hat{S}(I_3 | I_2) = 0.85 > 0.8$. The projections of the optimal subset I_3 and the initial design space Φ_2 onto planes of all pairs of the design variables are presented in Figure 3.8. The optimal solution which is chosen as the center of I_3 and denote as the red star in Figure 3.8, is [2.75, 0.26, 3.5, 0.3], corresponding to 1.5125×10^6 N/m for the PTMD spring constant and 2.86×10^5 N·s/m for the

PTMD damping coefficient, 2.45×10^5 N/m for the NTMD spring constant and 4.5×10^4 N·s/m for the NTMD damping coefficient. With the optimal PTMD and NTMD for the structural control system, the fore-aft fatigue damage of the tower base is reduced by 28% . Compared to a single PTMD in case 1, case 2 with two TMDs is more effective in fatigue reduction.

Figure 3.7 Simulated samples by ARS-SS and identified subset by SSO for case 1

(a)

(b)

(c)

Figure 3.8 Projections of the optimal subset and the initial design space onto planes of all pairs of the design variables for case 2

Table 3.6 Comparison of SSO with ARS-SS to SPSA

		Number of dynamic simulations	Optimal design	Fatigue damage	Fatigue damage reduction(%)
Case 1	SSO with ARS-SS	2604	[2.718,0.3117]	0.458	15
	SPSA	25600	[3.09,0.1416]	0.3735	30
Case 2	SSO with ARS-SS	11280	[2.75,0.26,3.5,0.3]	0.388	28
	SPSA	47200	[2.75,0.175,3.25,0.15]	0.3566	34

The efficiency of the proposed algorithm SSO with ARS-SS is compared to that of SPSA. As shown in Table 3.6, SSO with ARS-SS requires much less number of dynamic simulations for both case 1 and case 2. The ratio of computational cost for SSO with ARS-SS to SPSA is $\frac{1}{10}$ and $\frac{1}{4}$ for case 1 and case 2, respectively. The comparison of the effectiveness of the identified optimal design by SSO with ARS-SS and SPSA is also presented in Table 3.6. It shows a large difference for case 1 and slight difference for case 2. This may due to case 2 performed more optimization levels to achieve the optimal design. It also indicates that we may increase the existing stopping criterion of SSO (which is empirically chosen as $\hat{S}(I|\Phi) > 0.8$) for case 1.

The advantage in efficiency of the proposed algorithm SSO with ARS-SS for case 2 is less than case 1. This may be caused by the increase of the dimension of the design variables. SSO needs to perform more optimization levels to terminate the iteration to obtain the optimal design and the number of samples required in each optimization level has to increase as the addition of the dimension. Moreover, the size of the identified optimal subsets for both cases are relatively large compared to the original design space. These factors motivate me to develop a new algorithm in the next chapter to identify better optimal designs.

3.7 Conclusions

In this chapter, ARS-SS is proposed for improving the efficiency of simulating samples from the auxiliary PDF which is the most critical step of SSO. Based on the sampling procedure of ARS, the simulated samples by ARS can be regarded as failure samples of a proposed reliability problem. SS is adopted to simulate failure samples due to its efficiency for reliability problems. In the case studies, the design optimization of the structural control system for the spar buoy wind turbine is considered. Two design cases are involved: case 1 is to design a single controller which includes 2 design variables and case 2 is to design two controllers simultaneously which includes 4 design variables. Both cases are solved by SSO with ARS-SS and SPSA, respectively.

ARS-SS shows significant enhancement in the efficiency of the sampling task for both cases. Although the solution obtained by SSO with ARS-SS is a suboptimal as the center of the subset is considered as the optimum, the effectiveness of the optimal design obtained by SSO with ARS-SS is much close to that of the optimal design by SPSA. Moreover, SSO with ARS-SS is more efficient than SPSA for both cases.

The advantage in efficiency for SSO with ARS-SS is slighter as the increase of the dimension of the design variables. SSO requires more optimization levels to converge to a relatively small subset which has a certain size reduction in each dimension for a high-dimensional design optimization problem because the size of the subset is pre-specified in each optimization level. Moreover, the size of the identified optimal subsets for both cases are relatively large compared to the original design space. These factors motivate me to develop a new stochastic optimization algorithm in the next chapter to identify better optimal designs.

Chapter 4 An efficient stochastic optimization algorithm using proposed modified version of Bayesian Sequential Partitioning

4.1 Introduction

This chapter also considers the system design optimization problem, $\boldsymbol{\varphi}^* = \arg \min_{\boldsymbol{\varphi} \in \Phi} \int_{\Theta} h(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(\boldsymbol{\theta}) d\boldsymbol{\theta}$. As pointed out in chapter 2, traditional optimization algorithms are computationally expensive for solving this optimization problem. SSO is able to obtain the optimal design with a relatively low computational cost due to its advantage of avoiding estimating the objective function. The major computational effort of SSO lies in simulating samples from some specified PDFs at each optimization level. However, the computational cost of SSO will increase significantly as the increase of the dimension of the design variables. This is due to the size of the subset is specified in each optimization level and it requires more optimization levels to converge to a relatively small subset which has a certain size reduction in each dimension when the dimension of the design variables is large. The increasing of the computational cost can be seen by compared case 2 to case 1 in Chapter 3.

This chapter is to propose a new stochastic optimization algorithm to efficiently identify smaller optimal sub-regions and better optimal designs. Based on the augmented problem where the design variables are artificially considered as uncertain, the sensitivity of the objective function in the initial design space can be explored by investigating the marginal PDF of the auxiliary PDF. A multivariate density estimation method, proposed modified version of BSP, is adopted due to its advantage of adaptively dividing the design space based on the simulated samples and estimating the high-dimensional densities.

The proposed stochastic optimization algorithm is presented in section 4.2. The augmented problem, which is the basis of the algorithm, is given and followed by the proposed modified version of BSP which enables estimating the high-dimensional probability densities. The framework of the proposed algorithm is also outlined. Section 4.3 gives some suggestions for the implementation of the

proposed stochastic optimization algorithm. Case studies and discussion are presented in section 4.4. Section 4.5 shows the conclusions.

4.2 Proposed stochastic optimization algorithm

4.2.1 Augmented problem

The augmented problem was initially proposed by Au and Beck (Au 2005) for investigating the sensitivity of the failure probability with respect to the system design. In this concept, the design variables are artificially regarded as uncertain with some specified PDF $p(\boldsymbol{\varphi})$ over the admissible design space. Based on this setting, an auxiliary PDF is defined as follow

$$\pi(\boldsymbol{\theta}, \boldsymbol{\varphi}) = \frac{h(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(\boldsymbol{\theta}|\boldsymbol{\varphi}) p(\boldsymbol{\varphi})}{\int_{\Theta} \int_{\Phi} h(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(\boldsymbol{\theta}|\boldsymbol{\varphi}) p(\boldsymbol{\varphi}) d\boldsymbol{\theta} d\boldsymbol{\varphi}} \propto h(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(\boldsymbol{\theta}|\boldsymbol{\varphi}) p(\boldsymbol{\varphi}) \quad (4.1)$$

the denominator in the above equation is a normalizing constant, which is related to the expectation of the performance function over the design space and the random variable space. The exact value of the constant can be evaluated with stochastic simulation techniques, while it may not be required in practice. Since $\pi(\boldsymbol{\theta}, \boldsymbol{\varphi})$ should satisfy the characteristics of the PDF, $h(\boldsymbol{\theta}, \boldsymbol{\varphi}) \geq 0$ is required such that $\pi(\boldsymbol{\theta}, \boldsymbol{\varphi}) \geq 0$. The marginal PDF of the auxiliary PDF with respect to the design variables $\boldsymbol{\varphi}$ is given as

$$\begin{aligned} \pi(\boldsymbol{\varphi}) &= \int_{\Theta} \pi(\boldsymbol{\theta}, \boldsymbol{\varphi}) d\boldsymbol{\theta} \\ &= \frac{1}{\int_{\Theta} \int_{\Phi} h(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(\boldsymbol{\theta}|\boldsymbol{\varphi}) p(\boldsymbol{\varphi}) d\boldsymbol{\theta} d\boldsymbol{\varphi}} \int_{\Theta} h(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(\boldsymbol{\theta}|\boldsymbol{\varphi}) p(\boldsymbol{\varphi}) d\boldsymbol{\theta} \\ &= \frac{1}{\int_{\Theta} \int_{\Phi} h(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(\boldsymbol{\theta}|\boldsymbol{\varphi}) p(\boldsymbol{\varphi}) d\boldsymbol{\theta} d\boldsymbol{\varphi}} p(\boldsymbol{\varphi}) H(\boldsymbol{\varphi}) \end{aligned} \quad (4.2)$$

This indicates that the objective function is related the marginal PDF and the chosen artificial PDF. It is given as below

$$H(\boldsymbol{\varphi}) \propto \frac{\pi(\boldsymbol{\varphi})}{p(\boldsymbol{\varphi})} \quad (4.3)$$

since the denominator in Equation (4.2) is a normalizing constant, the objective function is proportional to the value of $\frac{\pi(\boldsymbol{\varphi})}{p(\boldsymbol{\varphi})}$. In most cases, $p(\boldsymbol{\varphi})$ is chosen as a uniform PDF over the design space since no prior information about the system design is available. In such case, the objective function is directly proportional to the marginal PDF $\pi(\boldsymbol{\varphi})$. Thus minimizing the objective function is equivalent to minimizing the marginal PDF. In other words, the sensitivity of the objective function can be explored by investigating the sensitivity of the marginal PDF with respect to its variables over the design space.

The marginal PDF $\pi(\boldsymbol{\varphi})$ can be explored based on the samples distributed according to its PDF. These samples can be obtained by taking the $\boldsymbol{\varphi}$ component of the sample pairs $[\boldsymbol{\theta}, \boldsymbol{\varphi}]$ simulated from the auxiliary PDF $\pi(\boldsymbol{\theta}, \boldsymbol{\varphi})$. The samples distributed according to the joint PDF $\pi(\boldsymbol{\theta}, \boldsymbol{\varphi})$ may be simulated by stochastic sampling techniques, such as Acceptance-Rejection Sampling (ARS) algorithm and Markov Chain Monte Carlo (MCMC) simulation. Based on these samples, the marginal PDF $\pi(\boldsymbol{\varphi})$ may be evaluated by some approximation methods such as kernel density methods. However, kernel density methods may be challenging or problematic for some cases, where the dimension of the design variables is large or the sensitivity of the objective function over the design space is complicated. For high-dimensional case, the accuracy of the kernel density estimation also depends on the kernel function, the selection of which is not an easy work. Furthermore, the kernel density methods always underestimate the densities around the boundary of the sample space which corresponds to the bounded design space. In this work, a proposed approach is provided to investigate the sensitivity of the marginal PDF with respect to the design variables, especially for high-dimensional and complicated problems.

4.2.2 Bayesian Sequential Partitioning

To explore the sensitivity of the marginal PDF $\pi(\boldsymbol{\varphi})$ over the design space, $\pi(\boldsymbol{\varphi})$ is approximated by a multivariate density estimation method based on Bayesian Sequential Partitioning (BSP) (Lu, Jiang et al. 2013, Liu and Wong 2014, Liu and Wong 2015). This method was proposed by Lu for density estimation and classification and it has been proved to be able to obtain good approximation for high-dimensional PDFs such as the 50-dimensional PDF. The basic idea of BSP is using the posterior PDF of each Binary Partition (BP) to evaluate its quality. This is based on the proof that maximizing the posterior PDF of a BP is asymptotically equivalent to minimizing the Kullback-Leibler (KL) divergence between the true density and the estimated density. The posterior PDF is evaluated based on the simulated samples and some chosen prior densities.

Let $D = (\boldsymbol{\varphi}_1, \dots, \boldsymbol{\varphi}_{N_0})$ be the $\boldsymbol{\varphi}$ component of the i.i.d. simulated samples, the number of which is N_0 . The design vector $\boldsymbol{\varphi}_i$ is a d -dimensional vector given as $\boldsymbol{\varphi}_i = [\boldsymbol{\varphi}_{i1}, \dots, \boldsymbol{\varphi}_{id}] \in \mathbb{R}^d$. The corresponding sample space for the given data set D is defined as $\Omega = I_1 \otimes \dots \otimes I_d$, where $I_j = (\min_i(\boldsymbol{\varphi}_{ij}), \max_i(\boldsymbol{\varphi}_{ij}))$ represents the estimated support for the j th dimension. In some case, this sample space can be directly selected as the admissible design space Φ .

To explore the density over the sample space, Ω is divided into a number of regions by BP. Let t be the number of partitions for a BP. The BP is defined as follows: the BP of size $t+1$ is obtained by a further division of one of the regions generated by a BP of size t , where the further division is a bisection along one of the coordinates. Figure 4.1 shows the possible further divisions of a BP of size 2. The BPs are obtained in an iterative process and a sequential way. Let $x_t = A_1 \cup \dots \cup A_t$ be the BP of size t , where the sample space Ω is divided into t regions and A_i represents the i th region.

Figure 4.1 Possible further divisions of a given BP containing two sub-region

In terms of the division of a given BP, the corresponding density function is defined as follows:

$$f(\boldsymbol{\varphi}) = \sum_{k=1}^t \mu_k I_{A_k}(\boldsymbol{\varphi}) \quad (4.4)$$

$$I_{A_k}(\boldsymbol{\varphi}) = \begin{cases} 1 & , \boldsymbol{\varphi} \in A_k \\ 0 & , \text{otherwise} \end{cases} \quad (4.5)$$

where $\mu_k = \frac{p_k}{|A_k|}$, $p_k = P(\boldsymbol{\varphi} \in A_k)$ and $|A_k|$ is the volume of the region A_k . p_k represents the probability of the samples belonging to A_k . Thus p_k has to be chosen such that $0 \leq p_k \leq 1$ and $\sum_{k=1}^t p_k = 1$. The density function can be rewritten as follows:

$$f(\boldsymbol{\varphi}) = \sum_{k=1}^t \frac{p_k}{|A_k|} I_{A_k}(\boldsymbol{\varphi}) \quad (4.6)$$

To build the posterior probability of a BP, it is necessary to propose prior PDFs for the BP and the parameters containing in the density function f . It is assumed that all partitions of the same size t have an equivalent prior density, which is proportional to $\exp(-\beta t)$. Therefore, the prior density for a BP is given as

$$P(x_t) \propto \exp(-\beta t) \quad (4.7)$$

where β is a specified constant representing the preference on the size of BP. Considering the characteristics of the parameters p_k , the prior PDF for parameters $(p_1, \dots, p_t)$ conditional on a given BP can be assumed to be a Dirichlet distribution with parameters $(\alpha, \dots, \alpha)$. The prior PDF is given as

$$f(p_1, \dots, p_t | A_1, \dots, A_t) = \frac{1}{D(\alpha, \dots, \alpha)} \prod_{k=1}^t p_k^{\alpha-1} \quad (4.8)$$

$$\text{where } D(\alpha, \dots, \alpha) = \frac{\prod_{k=1}^t \Gamma(\alpha)}{\Gamma\left(\sum_{k=1}^t \alpha\right)}.$$

In terms of the above prior densities, the posterior probability of a BP $x_t = (A_1, \dots, A_t)$ on a given data set D is given as follows:

$$\begin{aligned} \pi^0(x_t) &= P(A_1, \dots, A_t | D) \propto P(A_1, \dots, A_t) P(D | A_1, \dots, A_t) \\ &= P(A_1, \dots, A_t) \int_{p_1, \dots, p_t} P(D | A_1, \dots, A_t, p_1, \dots, p_t) \times f(p_1, \dots, p_t | A_1, \dots, A_t) dp_1 \dots dp_t \\ &\propto \exp(-\beta t) \int \left(\prod_{k=1}^t \mu_k^{n_k} \right) \left(\frac{1}{D(\alpha, \dots, \alpha)} \prod_{k=1}^t p_k^{\alpha-1} \right) dp_1 \dots dp_t \\ &\propto \exp(-\beta t) \frac{D(n_1 + \alpha, \dots, n_t + \alpha)}{D(\alpha, \dots, \alpha)} \prod_{k=1}^t \frac{1}{|A_k|^{n_k}} \end{aligned} \quad (4.9)$$

where n_k is the number of points in the data set D belonging to the region A_k . Let C be the normalizing constant. The posterior probability is written as

$$\pi^0(x_t) = C \exp(-\beta t) \frac{D(n_1 + \alpha, \dots, n_k + \alpha)}{D(\alpha, \dots, \alpha)} \prod_{k=1}^t \frac{1}{|A_k|^{n_k}} \quad (4.10)$$

Let f and $\hat{f}_{x_t}$ be the true density function and the estimated density function based on the BP x_t , respectively. The KL divergence between f and $\hat{f}_{x_t}$ is given as follow:

$$D_{KL} \left(f \parallel \hat{f}_{x_t} \right) = \int f(\phi) \log \frac{f(\phi)}{\hat{f}_{x_t}(\phi)} d\phi \quad (4.11)$$

The following equation has been proved to be true for any BP x_t on condition that $\alpha \in (0, 0.5]$ [6].

$$\lim_{n \rightarrow \infty} \frac{\log \pi^0(x_t)}{n} = -D_{KL} \left(f \parallel \hat{f}_{x_t} \right) + \int \log(f) f dx \quad (4.12)$$

It can be seen that larger posterior probability will lead to smaller KL divergence between the true density function and the estimated density function. Thus minimization of the KL divergence is asymptotically equivalent to maximization of the posterior probability. The BP can better approximate the true density when its posterior is larger.

4.2.3 Proposed method for the identification of the optimal Binary Partition and the optimal sub-region

The original method of BSP for identifying the optimal partition is not applicable to this stochastic optimization problem. This work proposes a feasible and applicable method to identify the optimal BP.

In this work, the optimal division of the sample space is defined as the optimal BP which has the maximum posterior probability. The optimal BP x^* is defined as follow

$$x^* = \arg \max_{1 \leq t \leq t_{\max}} \pi^0(x_t) \quad (4.13)$$

where $t_{\max}$ is a pre-specified maximum number of partitions. Based on the optimal BP, a piecewise constant function is constructed as follow to approximately estimate the true density function.

$$\hat{f}(\boldsymbol{\varphi}) = \sum_{k=1}^t \frac{n_k}{N_0 |A_k|} I_{A_k}(\boldsymbol{\varphi}) \quad (4.14)$$

this estimated function is used to investigate the sensitivity of the marginal PDF over the design space. The region that has the low estimated function value is able to minimize the margin PDF with high probability. In this work, the best sub-region A^* is chosen such that $A^* = \arg \min_{1 \leq k \leq t} \hat{f}(\boldsymbol{\varphi} \in A_k)$.

To obtain the optimal BP for a given data set D , an optimization scheme is proposed to identify an approximate optimal BP based on the property of BP. Since the BP is built up in a sequential way, the BPs of size $t+1$ that have high posterior probabilities will always be a further division of the BPs of size t that have high posterior probabilities. In terms of this property, a number of BPs that have higher posterior probabilities are obtained for a given size t and these BPs will be regarded as seeds to obtain a number of BPs of size $t+1$ that have higher posterior probabilities. Let a specified M_0 be the number of BPs with higher posterior probabilities for each size. Each of these M_0 BPs of size t will generate K_0 BPs of size $t+1$, which contribute to the maximum K_0 posterior probabilities of all possible BPs by a further division. By sorting these $M_0 K_0$ posterior probabilities in the decreasing order, the maximum M_0 posterior probabilities are retained and the corresponding BPs will be remained. The selections of M_0 and K_0 are related to the dimension of the design space and the maximum size of the BPs. The optimization scheme is given as follows:

Initialization: provide the maximum size of the BPs $t_{\max}$, the number of BPs for each size M_0 , and the number of BPs generated by each seed K_0 . Let B_t be the set containing the M_0 BPs for each size. B_1 will only contain one BP which is the

design space. Decide t_s accounted for the stopping criterion.

for $t = 2 : t_{\max}$

Step 1: Generate all possible BPs for a selected BP in B_{t-1} . Sort the posterior probabilities of these BPs in the decreasing order and bring the corresponding BPs of the first K_0 posterior probabilities to the set B'_t . Repeat this for all BPs in B_{t-1} .

Step 2: Sort the posterior probabilities of the BPs in the set B'_t in the decreasing order and remain the corresponding BPs of the first M_0 posterior probabilities for the set B_t .

Step 3: Check the stopping criterion. Break out of the loop if the stopping criterion is satisfied.

end

Stopping criterion: The maximum posterior probability will not change with increasing the size of the BPs by a pre-specified number t_s .

4.2.4 Iterative approach

The quality of the estimated density function is related to the size of the simulated samples, which also defines the convergence rate of the BSP method. A certain number of samples are required for investigating the sensitivity of the PDF over the design space. To explore the sensitivity of the true density function over the regions with lower densities, it is necessary to obtain more samples in such regions for a better estimation. An alternative way is to simulate more samples that are distributed according to $\pi(\phi|\phi \in \Phi)$. However, the probability of the simulated samples lie in each region is related to the density in the corresponding region and the simulated samples tend to lie in regions with high densities. Thus obtaining more samples in regions with low densities will significantly increase the computational cost.

To better investigate the sensitivity of the density over regions with lower densities and to increase the efficiency of the optimization process, an iterative approach can be adopted, rather than using a single iteration. At iteration m , it requires additional samples in A_{m-1}^* ($A_0^* = \Phi$) that are distributed according to the marginal PDF

$\pi(\boldsymbol{\phi} | \boldsymbol{\phi} \in A_{m-1}^*)$. Based on the simulated samples in A_{m-1}^* , BSP is performed by regarding A_{m-1}^* as the sample space. The optimal BP is identified as Equation (4.13) and the corresponding region A_m^* that has the lowest estimated density is obtained. It can be seen that A_m^* is a sub-region of A_{m-1}^* and the size of the identified region at each iteration will be reduced with iteration process. The iteration process will terminate when the optimal BP identified based the simulated samples in A_m^* is the design space A_m^* . This indicates that the estimated density function is a uniform PDF over the whole space A_m^* .

4.2.5 Proposed stochastic optimization algorithm

In terms of the augmented problem, minimizing the objective function is equivalent to minimizing the marginal PDF $\pi(\boldsymbol{\phi})$ of the auxiliary PDF $\pi(\boldsymbol{\theta}, \boldsymbol{\phi})$ with respect to the design vector $\boldsymbol{\phi}$. Based on the simulated samples that are distributed according to $\pi(\boldsymbol{\phi})$, modified BSP is used to approximate the PDF and the region that has the lowest density is identified. This region has high plausibility of containing the optimal design variables. The framework of modified BSP with ARS-SS is outlined as follows (Figure 4.2 shows the flowchart of the proposed algorithm (modified BSP with ARS-SS) and Figure 4.3 visually illustrates some important steps):

Initialization: define the desired number of samples N_0 and the bounded design space Φ .

Decide on the parameters of the prior densities α and β , and the maximum size of the BPs $t_{\max}$.

At optimization iteration m ($A_0^ = \Phi$ for the first iteration):*

Step 1: Obtain samples that are distributed according to $\pi(\boldsymbol{\phi} | \boldsymbol{\phi} \in A_{m-1}^*)$. For the first iteration, samples are obtained by taking the $\boldsymbol{\phi}$ component of the simulated sample pairs $[\boldsymbol{\theta}, \boldsymbol{\phi}]$ simulated from $\pi(\boldsymbol{\theta}, \boldsymbol{\phi})$. For the iteration $m > 1$, samples that are distributed according to $\pi(\boldsymbol{\theta}, \boldsymbol{\phi} | \boldsymbol{\phi} \in A_{m-1}^*)$ may be

obtained by MCMC simulation based on the retained samples of $\pi(\boldsymbol{\theta}, \boldsymbol{\varphi} | \boldsymbol{\varphi} \in A_{m-2}^*)$, whose $\boldsymbol{\varphi}$ component are in A_{m-1}^* .

Step 2: Identify the optimal BP x_m^* based on the samples obtained in step 1.

$$x_m^* = \arg \max_{1 \leq t \leq t_{\max}} \pi^0(x_t)$$

where $\pi^0(x_t)$ is the posterior probability of the BP x_t . Let t_m be the size of the optimal BP x_m^* . The estimated density is obtained as:

$$\hat{f}_m(\boldsymbol{\varphi}) = \sum_{k=1}^{t_m} \frac{n_{km}}{N_0 |A_{km}|} I_{A_{km}}(\boldsymbol{\varphi})$$

where n_{km} and $|A_{km}|$ are the number of samples in A_{km} and the volume of A_{km} , respectively. A_{km} is the sub-region of A_{m-1}^* such that $A_{m-1}^* = \bigcup_{k=1}^{t_m} A_{km}$. The region that has the lowest density is selected as $A_m^* = A_{km} = \arg \min_{1 \leq k \leq t_m} \hat{f}_m(\boldsymbol{\varphi} \in A_{km})$.

Step 3: Retain the sample pairs $[\boldsymbol{\theta}, \boldsymbol{\varphi}]$ whose $\boldsymbol{\varphi}$ component are in A_m^* . These samples will be distributed according to $\pi(\boldsymbol{\theta}, \boldsymbol{\varphi} | \boldsymbol{\varphi} \in A_m^*)$ and can be used as seeds at the next iteration.

Stopping criterion: The iteration stops when the following is satisfied.

$$A_m^* = A_{m-1}^*$$

Figure 4.2 Flowchart of modified BSP with ARS-SS

Figure 4.3 Illustrative process of modified BSP with ARS-SS for a two-dimensional design case, the red star corresponds to the final optimum and the rectangles with red edges are the optimal sub-region identified in the corresponding iteration

The proposed stochastic optimization algorithm, which estimates the multivariate density by proposed modified version of BSP, will adaptively converge to a relatively small region for the optimal solution within the origin design space. The identification of the region in each iteration is related to the sensitivity of the objective function within the corresponding design space. It also depends on the number of the simulated samples as the identification of the optimal BP over the design space is based on the maximum of the posterior probability. The sensitivity of the objective function within the design space can be well explored if enough samples were simulated. As the iteration evolves and converges to a smaller region

where the sensitivity is smaller, the size of the optimal BP will become one and the estimated density is the same within the design space in the last iteration. This indicates that the value of the objective function will be nearly the same for the design configurations within the design space of the last iteration. Therefore, the center point of this identified region can be considered as the sub-optimal design.

4.3 Implementation details of the proposed stochastic optimization algorithm

This section provides some suggestions on simulating samples, the choice of parameters for prior densities, the copula of the high dimensional density.

4.3.1 Stochastic simulation techniques

To obtain samples that are distributed according the marginal PDF $\pi(\boldsymbol{\phi})$ in the first iteration, it is necessary to simulate sample pairs $[\boldsymbol{\theta}, \boldsymbol{\phi}]$ from the auxiliary PDF $\pi(\boldsymbol{\theta}, \boldsymbol{\phi})$. As defined in Equation (4.1), there exists an explicitly unknown normalization constant which is the expectation of the performance function with respect to both the random variables and design variables. In terms of this, Acceptance-Rejection Sampling (ARS) method or Markov Chain Monte Carlo (MCMC) simulation may be adopted as evaluation of the normalization constant is not required for these sampling methods. ARS can be used for simulating independent samples which are required for density estimation using BSP. However, the acceptance rate may be very low in practice and this will lead to a large number of evaluations of the performance function $h(\boldsymbol{\theta}, \boldsymbol{\phi})$. To save computational cost, samples that are distributed according to $\pi(\boldsymbol{\theta}, \boldsymbol{\phi})$ can be obtained by the ARS with Subset Simulation (ARS-SS) which is proposed by the authors in chapter 3. Although the samples that are generated by ARS-SS are correlated, asymptotically independent samples can be obtained by taking one sample out of every n samples in the Markov Chain of Subset Simulation. The number n is selected based on the correlation of the samples γ , which can be evaluated as specified in (Berg 2004). To use the modified MCMC method proposed for SS (Au and Beck 2001), the correlated random variables are transformed to independent variables such that the

PDF of the random variables is $p(\boldsymbol{\theta}) = \prod_{i=1}^{n_\theta} p(\theta_i)$. The artificial uncertain design variables within the bounded design space are transformed to the standard normal variables.

In the iteration m , the sample pairs $[\boldsymbol{\theta}, \boldsymbol{\varphi}]$ retained in the previous can be regarded as seeds to generate additional samples in A_{m-1}^* by MCMC. The information about the retained samples can be used to better choose the proposal PDF for MCMC. The asymptotically independent samples are obtained as the first iteration.

4.3.2 Choice of parameters for prior densities

The choice of the parameter β decides the preferences on the size of the BP. The prior density for BP x_t is proportional to $\exp(-\beta t)$. Empirically, it is found that the results are satisfactory on condition that $\beta \in [0.5, 1]$ (Lu, Jiang et al. 2013). Such choice indicates that the BPs of smaller size have more priorities. This reduces the number of division over the design space and thus mitigates the risk of overfitting problem. For cases that prior information is available, the prior can be modified to involve the available information such as the size of the sub-regions to be balanced.

The parameter α in the prior PDF for the probability mass has to be chosen to prevent overfitting problem and to maintain the diversity of the BP as well. Due to the property of p_k , which sum up to one, the Dirichlet distribution is chosen as the prior for the probability mass of each sub-region generated by a given BP. Overfitting problem can be prevented by this distribution when $\alpha \in (0, 0.5]$. For simplicity, the parameter α may be selected as 0.5. However, this choice may inject much preference on the prior when the dimension of the design variables is large and the number of simulated samples is relatively small. To avoid potential serious biases, the parameter α is chosen based on the experience from [6]. The main idea is to select smaller α when the number of simulated samples in the sub-region is small. Thus the parameter α is chosen as follow:

$\alpha = \max \left(\min \left(\min_{1 \leq k \leq t} \left(\frac{n_k}{200} \right), 0.5 \right), 0.1 \right)$, where n_k is the number of simulated samples in the corresponding region A_k , which is any sub-region divided by a BP of size t . This truncated linear function of n_k is adopted such that the sum $t\alpha$ is small compared to n_k .

4.3.3 Copula of the joint marginal PDF $\pi(\boldsymbol{\varphi})$

When using BSP to estimate the multivariate density, the size of the optimal BP, which is related to the number of cut of the design space, depends on the difference between the true density and the uniform density. More cuts are required when the true density is further away from uniform. This will lead to a bigger value of $t_{\max}$ and thus increase the computational complexity in the identification of the optimal BP. The number of cuts will be extremely large when the dimension of the design variables is large and the true density is further away from uniform. In such cases, copula of the joint PDF $\pi(\boldsymbol{\varphi})$ may be adopted to reduce the size of the optimal BP and alleviate the computational complexity.

The joint marginal PDF $\pi(\boldsymbol{\varphi})$, considered in this work, can be written as follow:

$$\pi(\boldsymbol{\varphi}) = \prod_{j=1}^d f_j(\varphi_j) \times f_F(F_1(\varphi_1), \dots, F_d(\varphi_d)) \quad (4.18)$$

where $\boldsymbol{\varphi} = [\varphi_1, \dots, \varphi_d]$. $f_j(\cdot)$ is the marginal PDF for the j th dimension and $F_j(\cdot)$ is the cumulative density function (CDF) for the j th dimension. The copula $f_F(\cdot)$ is the joint distribution of the CDF transformed data. This copula explores the dependence between the design variables and is very helpful in improving the computational efficiency when the dimension of the design variables is large.

The marginal PDF $f_j(\cdot)$ can be estimated based on the j th component of the simulated samples. The optimal BP for this one-dimensional data can be identified and the corresponding piecewise constant density function can be considered as an

estimate of the true marginal PDF. Also the one-dimensional CDF $F_j(\cdot)$ can be built based on the estimate of $f_j(\cdot)$. Thus the original data are mapping to the cumulative probability space $\Omega = [0,1]^d$ by the CDF $F_j(\cdot)$. The dependence between the design variables is investigated based on the transformed data in $[0,1]^d$. BSP is adopted to estimate the copula PDF $f_F(\cdot)$. Fewer cuts are required for the transformed data when compared to the original data because the true density of the transformed data is closer to the uniform over the transformed space. The reduction on computational complexity can be easily seen for the case when all design variables are independent. In such case, let t_j be the size of the optimal BP with respect to the j th dimension and the size of the optimal BP for the transformed data will be 1 as all design variables are independent. Thus the total number of cuts is $\sum_{j=1}^d t_j$. However, the number of cuts will be $\prod_{j=1}^d t_j$ without the copula. For cases that the dimension of the design variables is larger than 3, the copula is suggested to be adopted.

4.4 Illustrative examples

4.4.1 Case description

For illustrative purpose, the proposed stochastic optimization algorithm is used to identify the optimal design of the structural control system for a spar buoy floating wind turbine with respect to minimizing the fatigue damage at the tower base. Two cases are presented in this section. These two cases are the same as the cases in Chapter 3.

The two cases in this work are summarized as follows:

$$C_1 : \boldsymbol{\varphi}_1^* = \arg \min_{\boldsymbol{\varphi}_1 \in \Phi_1} \int_{\Theta} h(\boldsymbol{\theta}, \boldsymbol{\varphi}_1) p(\boldsymbol{\theta}) d\boldsymbol{\theta} \quad (4.19)$$

and

$$C_2 : \boldsymbol{\varphi}_2^* = \arg \min_{\boldsymbol{\varphi}_2 \in \Phi_2} \int_{\Theta} h(\boldsymbol{\theta}, \boldsymbol{\varphi}_2) p(\boldsymbol{\theta}) d\boldsymbol{\theta} \quad (4.20)$$

where $\boldsymbol{\theta}$ is the random variables of interest and $\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2$ denote the design vector for case 1 and case 2, respectively. The random variables of this design problem are the environmental conditional parameters $[U, H_s, T_p, \theta_{Angle}]$, and the white Gaussian noise variables $\boldsymbol{\theta}_w$, so $\boldsymbol{\theta} = [U, H_s, T_p, \theta_{Angle}, \boldsymbol{\theta}_{wind}, \boldsymbol{\theta}_{wave}]$. The dimension the white Gaussian noise variables depends on the simulation time, the size of the time step, and the dimension of the wind grid. In practice, the dimension of $\boldsymbol{\theta}_{wind}$ will has the magnitude of a million. The wave elevation time series of the stochastic waves are generated based on the white Gaussian noise vector $\boldsymbol{\theta}_{wave}$. A sample of wind speed and wave elevation time series is shown in Figure 4.4. The evaluation of the fatigue damage only considers the normal production case such that the space for the mean wind speed U is $[3, 25]$ m/s. The spaces for other random variables are defined as $H_s \in (0, +\infty)$ m, $T_p \in (0, +\infty)$ s, $\theta_{Angle} \in [0, 2\pi)$, and $\boldsymbol{\theta}_w \in \mathbb{R}$. For the case C_1 , the design vector is $\boldsymbol{\varphi}_1 = [\omega_1, \xi_1]$ and the design spaces are defined as $\omega_1 \in (0, 5)$ rad/s and $\xi_1 \in (0, 1)$. For the case C_2 , the design vector is $\boldsymbol{\varphi}_2 = [\omega_1, \xi_1, \omega_2, \xi_2]$ and the corresponding design spaces are defined as $\omega_1 \in (0, 5)$ rad/s, $\xi_1 \in (0, 1)$, $\omega_2 \in (0, 5)$ rad/s, and $\xi_2 \in (0, 1)$. Compared to the two cases in chapter 3, $[\omega_1, \xi_1, \omega_2, \xi_2]$ corresponds to $[\omega_{PTMD}, \xi_{PTMD}, \omega_{NTMD}, \xi_{NTMD}]$.

Figure 4.4 Sample of wind speed and wave elevation in time domain

4.4.2 Optimization algorithm characteristics

The proposed stochastic optimization algorithm, modified BSP with ARS-SS, is used to identify a sub-region and a sub-optimal design for both C_1 and C_2 cases. The number of simulated samples at each iteration is selected as $N_0 = 2000$ for both cases. The marginal PDF $\pi(\boldsymbol{\phi})$ is estimated by modified BSP and the chosen of the parameters for the prior densities follows the guideline given in section 3.2. The selections are $\alpha = \max\left(\min\left(\min_{1 \leq k \leq t}\left(\frac{n_k}{200}\right), 0.5\right), 0.1\right)$ and $\beta = 0.5$. For the maximum size of the BPs, $t_{\max} = 15$ and $t_{\max} = 20$ are chosen for C_1 and C_2 , respectively. To identify the optimal BP, the optimization parameters are chosen as $M_0 = 400, K_0 = 4$.

4.4.3 Results and discussion

Results for a sample run are shown in Table 4.1-3 and Figure 4.5-8. For design case C_1 , the iteration terminates after the first iteration and converges to a small sub-region where the sensitivity of the objective function is small. The samples from a sample run are presented in Figure 4.5a and the optimal BP is given as well. Figure 4.5b shows the estimated density function based on the optimal BP. For design case C_2 , two iterations are performed based on the stopping criterion and the optimal sub-region identified by the proposed algorithm is presented in Figure 4.8. The center point of the optimal sub-region is denoted by ϕ_s and the optimal solution that was identified by SPSA is represented by ϕ^* .

Table 4.1 Results from a sample run of the proposed algorithm for design cases C_1

	Design space A_{m-1}^*	Size of the optimal BP t_m	Optimal sub-region A_m^*	$\frac{ A_m^* }{ A_{m-1}^* }$	$\frac{ A_m^* }{ \Phi_1 }$
Iteration $m = 1$	$\omega_1 \in (0, 5)$ $\xi_1 \in (0, 1)$	8	$\omega_1 \in (2.5, 3.75)$ $\xi_1 \in (0.125, 0.25)$	$\frac{1}{64}$	$\frac{1}{64}$

Table 4.2 Results from a sample run of the proposed algorithm for design cases C_2

	Design space A_{m-1}^*	Size of the optimal BP t_m	Optimal sub-region A_m^*	$\frac{ A_m^* }{ A_{m-1}^* }$	$\frac{ A_m^* }{ \Phi_2 }$
Iteration $m = 1$	$\omega_1 \in (0, 5)$ $\xi_1 \in (0, 1)$ $\omega_2 \in (0, 5)$ $\xi_2 \in (0, 1)$	7	$\omega_1 \in (2.5, 3.75)$ $\xi_1 \in (0, 1)$ $\omega_2 \in (2.5, 5)$ $\xi_2 \in (0, 0.5)$	$\frac{1}{16}$	$\frac{1}{16}$

Iteration $m = 2$	$\omega_1 \in (2.5, 3.75)$ $\xi_1 \in (0, 1)$ $\omega_2 \in (2.5, 5)$ $\xi_2 \in (0, 0.5)$	6	$\omega_1 \in (2.5, 3.125)$ $\xi_1 \in (0.125, 0.25)$ $\omega_2 \in (2.5, 5)$ $\xi_2 \in (0, 0.5)$	$\frac{1}{16}$	$\frac{1}{256}$
----------------------	---	---	---	----------------	-----------------

Table 4.3 Comparison between the sub-optimal design ($\boldsymbol{\varphi}_s$) identified by the proposed algorithm and the final optimum ($\boldsymbol{\varphi}^*$) by SPSA for the two design cases

Design case	$\boldsymbol{\varphi}_s$	$\hat{H}(\boldsymbol{\varphi}_s)$	$\boldsymbol{\varphi}^*$	$\hat{H}(\boldsymbol{\varphi}^*)$
C_1	$\omega_1 = 3.125$ $\xi_1 = 0.1875$	0.3902	$\omega_1 = 3.09$ $\xi_1 = 0.1416$	0.3735
C_2	$\omega_1 = 2.8125$ $\xi_1 = 0.1875$ $\omega_2 = 3.75$ $\xi_2 = 0.25$	0.3632	$\omega_1 = 2.75$ $\xi_1 = 0.175$ $\omega_2 = 3.25$ $\xi_2 = 0.15$	0.3566

Figure 4.5 Simulated samples, the optimal BP identified by modified BSP, and the corresponding estimated density in the initial design space for design case C_1

Figure 4.6 The maximum posterior probability of the BP versus the size of the BP for C_1

Figure 4.7 Optimal sub-region A_1^* and the initial space Φ_1 for design case C_1

Figure 4.8 Projections of the optimal sub-region A_2^* and the initial design space Φ_2 onto planes of all pairs of the design variables for design case C_2

The ability of BSP in adaptively dividing the design space can be seen in Figure 4.5. For the region where the sensitivity of the marginal PDF is large, it is divided into more sub-regions. Fewer cuts exist in the region where the sensitivity is small. The optimal division of the design space is identified based on the posterior probabilities and this allows the adaptive partitioning of the design space. As presented in Figure 4.6, the optimal BP is chosen as the one which has the maximum posterior probability and the size of the optimal BP depends on the sensitivity of the marginal PDF over the design space. Due to the choice of the prior densities, the posterior probability of the BP will not increase with the size of the BP and the overfitting problem is prevented. Therefore, the optimal BP has a limited size.

As the proposed algorithm evolves, the size of the optimal sub-region at each iteration is reduced for both design case and the reduction of the size depends on the

sensitivity of the objective function in the corresponding design space. This is shown in the last two columns of Table 4.1 and 4.2. For design case C_1 , the size of the optimal sub-region is $1/64$ of the initial design space. For design case C_2 , the total reduction of the size of the design space is $1/256$, with $1/16$ in each of the two iterations. The proposed algorithm is able to adaptively identify a relatively small sub-region based on the sensitivity of the objective function in the initial design space. The size of the optimal sub-region identified in the last iteration depends on the sensitivity of the objective function in the initial design space, which also determines the number of iterations required by the proposed algorithm. Based on the simulated samples that are distributed according to the marginal PDF $\pi(\boldsymbol{\varphi})$, BSP is used to estimate this multivariate density by dividing the corresponding design space into a number of sub-regions with BPs. The optimal BP is identified by maximizing the posterior probability. If enough samples are obtained, the size of the optimal BP for a given iteration is only related to the sensitivity of the marginal PDF within the design space, rather than a pre-specified value.

The quality of the sub-optimal design identified by the proposed stochastic optimization algorithm can be evaluated by comparison between the value of the objective function for $\boldsymbol{\varphi}_s$ and $\boldsymbol{\varphi}^*$. This is presented in Table 4.3: $H(\boldsymbol{\varphi}_s)$ is 4% and 2% larger than $H(\boldsymbol{\varphi}^*)$ for design case C_1 and C_2 , respectively. These slight differences allow us to consider the center point of the optimal sub-region as the sub-optimal solution. If it is required to pinpoint to an accurate optimal design, SPSA can be performed within the optimal sub-region and regard $\boldsymbol{\varphi}_s$ as the starting point for the iterative process. The results from the proposed algorithm may accelerate the convergence of SPSA for $\boldsymbol{\varphi}^*$ due to the reduction of the search space and the appropriate selection of the starting point.

The efficiency of the proposed algorithm can be demonstrated by the total computational cost. For both design cases, the samples are simulated by ARS-SS in the first iteration and the modified MCMC is used to simulate additional samples in

the latter iterations. SPSA is performed within the initial design space. To accelerate the convergence of SPSA, the starting point is chosen as the center point of the identified optimal sub-region. For design case C_1 , the total number of dynamic analyses used for the proposed algorithm is 6,640 and SPSA requires 32 iterations to converge to the optimal solution, which costs 25,600 dynamic analyses. For design case C_2 , the proposed algorithm requires 12,530 dynamic analyses and SPSA needs 59 iterations which equals 47,200 dynamic analyses. The number of dynamic analyses required by the proposed algorithm is 4 times smaller than SPSA for cases C_1 and C_2 , respectively, when SPSA starts from Φ_S .

Table 4.4 Comparison of modified BSP with ARS-SS, SSO with ARS-SS and SPSA

		Number of dynamic simulations (actual time used)	Optimal design	Fatigue damage	Fatigue damage reduction (%)	Volume ratio $\frac{V_{subset}}{V_{\Phi}}$
Case 1	Modified BSP with ARS-SS	6640 (66.4hours)	[3.125,0.1875]	0.3902	27	0.0625
	SSO with ARS-SS	6640 (66.4 hours)	[2.875,0.275]	0.442	18	0.239
	SPSA	25600 (256 hours)	[3.09,0.1416]	0.3735	30	
Case 2	Modified BSP with ARS-SS	12530 (125.3 hours)	[2.8125,0.1875, 3.75,0.25]	0.3632	33	0.0039
	SSO with ARS-SS	11280 (112.8 hours)	[2.75,0.26, 3.5,0.3]	0.388	28	0.017
	SPSA	47200 (472 hours)	[2.75,0.175, 3.25,0.15]	0.3566	34	

The advantage of the proposed stochastic optimization algorithm is shown in Table 4.4. For both cases, with almost the same computational cost, modified BSP with ARS-SS identifies much smaller optimal sub-regions and better optimal designs than SSO with ARS-SS. The optimal design identified by modified BSP with ARS-SS can reduce the fatigue damage by 33%. The proposed algorithm identified a much smaller optimal subset, the volume of which is $\frac{1}{4}$ of the optimal subset identified by SSO with ARS-SS for case 1. Moreover, the optimal design identified by modified BSP with ARS-SS is more effective than that of SSO with ARS-SS and is very close to the optimal design identified by SPSA.

4.5 Conclusions

This study aims to solve the stochastic design optimization problem where the objective function is the expected value of any performance measure. A stochastic optimization algorithm is proposed in this study for identifying a sub-region which has high likelihood of containing the optimal solution. The proposed algorithm is based on the augmented problem where the design variables are artificially considered as uncertain variables. This allows that minimizing the objective function is equivalent to minimizing the marginal PDF (Probability Density Function). The sensitivity of the objective function is explored by the estimated density based on the simulated samples that are distributed according to the marginal PDF. This estimated density is built by a multivariate density estimation method, proposed modified version of Bayesian Sequential Partitioning (BSP), which is able to estimate the high-dimensional density. To obtain the optimal estimated density, the design space is divided into a number of sub-regions by Binary Partitions (BPs) and the optimal BP is identified with the maximum posterior probability. Based on the optimal estimate of the marginal PDF, the sub-region that has the lowest density is believed to have the highest likelihood of containing the optimal solution. To improve the efficiency of the identification, an iterative approach is adopted and the samples from the previous iteration are used to help simulate additional samples for the current iteration. The center point of the optimal sub-region identified in the last iteration is regarded as the sub-optimal design.

An illustrative example, which is the design optimization of the structural control system for a floating wind turbine, is shown to demonstrate the ability of the proposed algorithm in identifying a sub-optimal design and its efficiency. The center point of the optimal sub-region identified by the proposed algorithm is regarded as the sub-optimal design and SPSA (Simultaneous Perturbation Stochastic Approximation) started this center point to search the final optimal design within the initial design space. In the illustrative example, it is found that the value of the objective function between the final optimum and the sub-optimal are close to each other, and modified BSP with ARS-SS requires much less computational cost.

Compared to SSO with ARS-SS in Chapter 3, the proposed algorithm, modified BSP with ARS-SS, identifies much smaller optimal sub-regions and better optimal designs. The future studies will investigate the method of improving the efficiency by adaptively choosing the number of simulated samples.

Chapter 5 The proposed algorithm for simultaneously solving multiple optimization problems for a single system

5.1 Introduction

The former two chapters focus on the design optimization problem where the objective is to optimize a single objective for a single system. However, in most engineering practice, there will be more than a single performance function that is required to be optimized for a single system. The multiple performance functions of the system can be put in a single optimization problem which is a multi-objective optimization problem. The optimization considered in this chapter is totally different from multi-objective optimization problem. This optimization is called as multiple optimization problems and the proposed optimization algorithm is developed to solve multiple optimization problems simultaneously and identifies the optimal solution for each optimization problem. However, the multi-objective optimization problem will identify solutions which compromise multiple objective functions. The optimal solutions of multi-objective optimization problem may be not the optimal solution for each of the performance function, but achieves a balance on all objective functions of interest. This work aims to solve multiple single-objective optimization problems for a single system simultaneously other than solving each optimization problem separately.

The proposed algorithm for simultaneously solving multiple optimization problems for a single system is presented in section 5.2. The illustrative examples are given in section 5.3 and section 5.4 outlines the conclusions.

5.2 The proposed algorithm

This work focuses on the stochastic optimization problem which has multiple objectives and each objective function corresponds to a single optimization problem. For simplicity, the multiple stochastic optimization problems are given as follow:

$$\boldsymbol{\varphi}_j^* = \arg \min_{\boldsymbol{\varphi} \in \Phi} \int_{\Theta} h_j(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(\boldsymbol{\theta}) d\boldsymbol{\theta} \quad j = 1, \dots, N_j \quad (5.1)$$

where N_j denotes the number of objective functions and the performance function of the j th objective is given by $h_j(\boldsymbol{\theta}, \boldsymbol{\varphi})$. $\boldsymbol{\theta}$ denotes the random variables which are distributed according to $p(\boldsymbol{\theta})$ within the space Θ . The design vector $\boldsymbol{\varphi}$ has a design space Φ . $\boldsymbol{\varphi}_j^*$ is the optimal design which minimizes the j th objective function.

The above multiple stochastic optimization problems can be solved by identifying the optimal design for each objective independently. However, the total computational cost will increase with the number of objectives and become extremely high when N_j is large. Taking into account the fact that a single dynamic analysis is time consuming, the required computational cost for the multiple stochastic optimization problems is unaffordable.

This section proposes a computationally efficient algorithm to solve the multiple stochastic optimization problems. It is assumed that the N_j performance functions can be evaluated through a single dynamic analysis and it is the case in the stochastic design optimization considered in this work. The proposed algorithm allows us to identify the optimal solutions for all optimization problems of interest simultaneously. In such way, the total number of dynamic analyses is reduced.

5.2.1 BSP with ARS-SS for a single objective optimization

Based on the idea of the augment problem proposed by Au and Beck (Au 2005), the design vector is artificially regarded as uncertain variables within the design space. The corresponding probability density function (PDF) of the artificial uncertain variables is given by $p(\boldsymbol{\varphi})$. For simplicity, the uniform distribution is chosen. An auxiliary PDF is defined as follow:

$$\pi_j(\boldsymbol{\theta}, \boldsymbol{\varphi}) \propto h_j(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(\boldsymbol{\theta}) p(\boldsymbol{\varphi}) \quad (5.2)$$

Then the marginal PDF of $\pi_j(\boldsymbol{\theta}, \boldsymbol{\varphi})$ with respect to $\boldsymbol{\varphi}$ can be evaluated as follow:

$$\pi_j(\boldsymbol{\varphi}) \propto p(\boldsymbol{\varphi}) \int_{\boldsymbol{\theta}} h_j(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(\boldsymbol{\theta}) d\boldsymbol{\theta} \quad (5.3)$$

In case that $p(\boldsymbol{\varphi})$ is chosen as uniform PDF over the design space, it can be seen that the j th objective function is proportional to the marginal PDF $\pi_j(\boldsymbol{\varphi})$. This indicates that minimizing the objective function equals to minimizing the marginal PDF of the auxiliary PDF $\pi_j(\boldsymbol{\theta}, \boldsymbol{\varphi})$ with respect to $\boldsymbol{\varphi}$.

The sensitivity of the marginal PDF $\pi_j(\boldsymbol{\varphi})$ with respect to $\boldsymbol{\varphi}$ can be investigated based on the samples $\{\boldsymbol{\varphi}\}$ distributed according to $\pi_j(\boldsymbol{\varphi})$. These samples can be obtained from sample pairs $\{\boldsymbol{\theta}, \boldsymbol{\varphi}\}$ distributed according to $\pi_j(\boldsymbol{\theta}, \boldsymbol{\varphi})$. Thus, the major task is to simulate sample pairs $\{\boldsymbol{\theta}, \boldsymbol{\varphi}\}$ from $\pi_j(\boldsymbol{\theta}, \boldsymbol{\varphi})$.

To simulate samples from $\pi_j(\boldsymbol{\theta}, \boldsymbol{\varphi})$ efficiently, the authors proposed the ARS-SS (Acceptance-Rejection Sampling with Subset Simulation) sampling algorithm. The ideas behind ARS-SS are the reliability view of the ARS procedure and the adoption of SS for obtaining samples for the proposed reliability problem efficiently. To simulate samples from $\pi_j(\boldsymbol{\theta}, \boldsymbol{\varphi})$, ARS follows the procedure as below:

Step 1: simulate a candidate sample $\{\boldsymbol{\theta}, \boldsymbol{\varphi}\}$ from the proposal PDF $g(\boldsymbol{\theta}, \boldsymbol{\varphi})$ and u from the uniform PDF $g(u)$ within the range $[0,1]$.

Step 2: run the dynamic analysis and evaluate the performance function $h_j(\boldsymbol{\theta}, \boldsymbol{\varphi})$, compute the ratio $\frac{h_j(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(\boldsymbol{\theta}) p(\boldsymbol{\varphi})}{M_j g(\boldsymbol{\theta}, \boldsymbol{\varphi})}$.

Step 3: accept the candidate sample $\{\boldsymbol{\theta}, \boldsymbol{\varphi}\}$ if $\frac{h_j(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(\boldsymbol{\theta}) p(\boldsymbol{\varphi})}{M_j g(\boldsymbol{\theta}, \boldsymbol{\varphi})} > u$, otherwise reject it.

The above steps can be run as many times as the number of the accepted samples achieves the required number. The constant M_j is chosen such that

$$M_j \geq \max_{\boldsymbol{\theta}, \boldsymbol{\varphi}} \left\{ \frac{h_j(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(\boldsymbol{\theta}) p(\boldsymbol{\varphi})}{g(\boldsymbol{\theta}, \boldsymbol{\varphi})} \right\}.$$

Taking a reliability view on the procedure of ARS, the accepted samples in step 3 can be regarded as failed samples. Thus, sampling from $\pi_j(\boldsymbol{\theta}, \boldsymbol{\varphi})$ equals to generating samples from $q_j(\boldsymbol{\theta}, \boldsymbol{\varphi}, u | F_j)$ which is defined as follows:

$$q_j(\boldsymbol{\theta}, \boldsymbol{\varphi}, u | F_j) = I_{F_j}(\boldsymbol{\theta}, \boldsymbol{\varphi}, u) g(\boldsymbol{\theta}, \boldsymbol{\varphi}) g(u) \quad (5.4)$$

$$I_{F_j}(\boldsymbol{\theta}, \boldsymbol{\varphi}, u) = \begin{cases} 1, & \text{if } \frac{h_j(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(\boldsymbol{\theta}) p(\boldsymbol{\varphi})}{u g(\boldsymbol{\theta}, \boldsymbol{\varphi})} > M_j \\ 0, & \text{otherwise} \end{cases} \quad (5.5)$$

In Equation (5.5), M_j is shifted to the right. This allows us to perform Subset Simulation without obtaining the value of M_j at the beginning. In [], the authors proposed a method for selecting M_j based on the simulated samples.

As soon as the samples from $q_j(\boldsymbol{\theta}, \boldsymbol{\varphi}, u | F_j)$ are obtained, the $\boldsymbol{\varphi}$ component of those samples can be used to investigate the sensitivity of $\pi_j(\boldsymbol{\varphi})$. Based on the simulated samples $\{\boldsymbol{\varphi}\}$, Bayesian Sequential Partitioning (BSP) can be used to estimate the marginal PDF $\pi_j(\boldsymbol{\varphi})$ as proposed in chapter 4. BSP is able to identify a sub-region which contains the optimal solution and obtain a relatively small sub-region in an iterative way based on additional simulated sample at the new optimization level.

Let k denote the optimization level. Suppose the optimal subset is I_{kj} by BSP based on the components $\{\boldsymbol{\varphi}\}$ of sample pairs $\{\boldsymbol{\theta}, \boldsymbol{\varphi}\}$ simulated from the PDF

$q_j(\boldsymbol{\theta}, \boldsymbol{\varphi}, u | F_j, \boldsymbol{\varphi} \in I_{(k-1)j})$. The retained sample pairs whose $\{\boldsymbol{\varphi}\}$ components are inside the optimal subset I_{kj} is used to generate additional sample pairs $\{\boldsymbol{\theta}, \boldsymbol{\varphi}\}$ that are distributed according to $q_j(\boldsymbol{\theta}, \boldsymbol{\varphi}, u | F_j, \boldsymbol{\varphi} \in I_{kj})$ by MCMC. For the first optimization level, $I_{0j} = \Phi$. The iterative process terminates when $I_{kj} = I_{(k-1)j}$ is satisfied. When the iteration terminates, a small subset is identified. The traditional gradient-based optimization algorithm can be used to pinpoint an optimal solution within the identified subset if the computational cost is affordable. Otherwise, the center point of identified subset is regarded as the suboptimal solution for simplicity.

5.2.2 Proposed sampling algorithm for multiple PDFs simultaneously

Based on the Subset Simulation proposed by Au and Beck for estimating failure probability, the authors proposed ARS-SS to generate samples from a single $q_j(\boldsymbol{\theta}, \boldsymbol{\varphi}, u | F_j)$ efficiently. The simulated samples from $q_j(\boldsymbol{\theta}, \boldsymbol{\varphi}, u | F_j)$ can be used to investigate the sensitivity of $\pi_j(\boldsymbol{\varphi})$.

To save the number of dynamic analyses, the author proposed a sampling algorithm which allows simulating failed samples for multiple failure events simultaneously. The sampling algorithm is proposed based on the idea of evaluating multiple failure probability curves through a single run of SS.

The original SS can estimate the small failure probability for high-dimensional reliability problem in a sequential way. To perform SS, a sequence of failure events are defined as $F_1 \supset F_2 \cdots \supset F_m = F$ (Bansal and Cheung 2017). Based on these failure events, the failure probability can be computed by the product of the conditional failure probabilities of a sequence of more frequent failure events. $P(F)$ can be evaluated as follows:

$$P(F) = P(F_1) \prod_{i=1}^{m-1} P(F_{i+1} | F_i) \quad (5.6)$$

and the failure event F_i is defined such that:

$$\begin{aligned} P(F_1) &= p_0 \\ P(F_{i+1}|F_i) &= p_0 \quad i = 1, \dots, m-1 \end{aligned} \quad (5.7)$$

p_0 is a pre-specified value for the conditional failure probability. The estimation of $P(F_1)$ is based on the samples simulated by Monte Carlo (MC) simulation. To estimate $P(F_{i+1}|F_i)$, additional samples are required to simulated based on the available failed samples $q(\boldsymbol{\theta}, \boldsymbol{\varphi}, u|F_i)$. These additional samples can be simulated by Markov Chain Monte Carlo (MCMC) simulation techniques. For high dimensional problem, the Modified Metropolis-Hastings (MH) algorithm can be used to avoid low acceptance rate problem.

The samples simulated at the i th SS level are distributed according to $q(\boldsymbol{\theta}, \boldsymbol{\varphi}, u|F_i)$ and these samples are used to determine the failure event F_{i+1} . The retained samples at the i th SS level will follow the distribution $q(\boldsymbol{\theta}, \boldsymbol{\varphi}, u|F_{i+1})$. At the last SS simulation level, the failure event F_m has to be the same as the failure event F_j defined in Equation (5.5). Thus, the retained samples at the last SS level are distributed according to $q_j(\boldsymbol{\theta}, \boldsymbol{\varphi}, u|F_j)$. These samples can be used as seeds to generate additional samples by MCMC simulation techniques. In this way, the samples from a single $q_j(\boldsymbol{\theta}, \boldsymbol{\varphi}, u|F_j)$ can be obtained.

For the proposed sampling algorithm for multiple distributions, the method for estimating multiple failure probabilities is considered since the failure probability estimation is based on the failed samples. It is assumed that the N_j failure probabilities are required to estimate and the j th failure probability is defined as follows:

$$P(F_j) = \int_{\Theta} \int_{\Phi} \int_U I_{F_j}(\boldsymbol{\theta}, \boldsymbol{\varphi}, u) g(\boldsymbol{\theta}, \boldsymbol{\varphi}) g(u) d\boldsymbol{\theta} d\boldsymbol{\varphi} du \quad (5.8)$$

the failure event F_j is defined as Equation (5.5).

The idea for the proposed algorithm is based on the special definition of a sequence of failure events. At the $(i-1)$ th SS level, the failure event F_i is defined as follows:

$$F_i = \bigcup_{j=1}^{N_j} F_{ij} \quad (5.9)$$

The failure event F_i is the union of the N_j failure events. The failure event F_{ij} is defined such that the conditional failure probability $P(F_{(i-1)j} | F_{ij})$ equals to the pre-specified value p_0 . Thus, the failure event F_{ij} is given as below:

$$I_{F_{ij}}(\boldsymbol{\theta}, \boldsymbol{\varphi}, u) = \begin{cases} 1, & \text{if } H_{ij}(\boldsymbol{\theta}, \boldsymbol{\varphi}, u) = \frac{h_j(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(\boldsymbol{\theta}) p(\boldsymbol{\varphi})}{ug(\boldsymbol{\theta}, \boldsymbol{\varphi})} > c_{ij} \\ 0, & \text{otherwise} \end{cases} \quad (5.10)$$

At the $(m-1)$ -th simulation level, the failure probability of the failure event F_{mj} can be computed as follow:

$$P(F_{mj}) = P(F_{mj} | F_{m-1}) \prod_{i=0}^{m-2} P(F_{i+1} | F_i) \quad (5.11)$$

the failure event F_0 is defined such that $P(F_0) = 1$

The value of c_{ij} is determined based on the simulated samples. For the zero simulation level, let $\{H_{0j}(\boldsymbol{\theta}, \boldsymbol{\varphi}, u)\}$ denote a series of values corresponding to the simulated samples. The number of the simulated samples at the zero level is assumed to be N_0 and these samples are simulated by MC simulation. Let $Hsort_{0j}$ denote the vector returned by sorting H_{0j} in the decreasing order. The value of c_{1j} is selected as the $p_0 N_0$ -th element of $Hsort_{0j}$ such that $P(F_{1j} | F_0) = p_0$. Based on the determination of c_{1j} , the samples belonging to the failure event F_1 are

retained at level zero and used as seeds to simulated additional samples for the next level.

For simulation level i , the simulated samples provide a vector of $\{H_{ij}(\boldsymbol{\theta}, \boldsymbol{\varphi}, u)\}$. The elements of H_{ij} may include values smaller than c_{ij} since the failure event F_i is the union of the N_j failure events F_{ij} . It is assumed the number of the simulated samples at level i to be N_i . Among the N_i simulated samples, N_{ij} samples which belong to failure event F_{ij} are selected based on the criterion that H_{ij} has to be larger than c_{ij} . The vector H_{ij} is sorted in the decreasing order and the vector $H_{sort_{ij}}$ is obtained. The value of $c_{(i+1)j}$ is selected as the $p_0 N_{ij}$ -th element of $H_{sort_{ij}}$. In this way, the failure probability $P(F_{(i+1)j} | F_{ij})$ equals the pre-specified value p_0 . The failure event F_{i+1} can be determined when the N_j failure events $F_{(i+1)j}$ are determined. The samples which belong to F_{i+1} are retained at level i and used as seeds to generate additional samples for level $i+1$.

Based on the simulated levels, a method is proposed for determining the stopping criterion and the value of M_j in Equation (5.5). After i simulation levels, $c_{1j} \leq c_{2j} \leq \dots \leq c_{(i+1)j}$ are obtained. A well-defined random variable X_j is given as follow:

$$X_j(\boldsymbol{\theta}, \boldsymbol{\varphi}, u) = \ln \frac{h_j(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(\boldsymbol{\theta}) p(\boldsymbol{\varphi})}{ug(\boldsymbol{\theta}, \boldsymbol{\varphi})} \quad (5.12)$$

The corresponding failure probability is defined as $P(X_j > a)$. After i simulation levels, the part of failure probability curves corresponding to $a < \ln c_{(i+1)j}$ are obtained. Based on the theoretical basis provided in [1], a characteristic trend can be found in the plot of $\ln P(X_j > a)$ versus a . The characteristic trend is that the slope

of the curve for the part $a > a_0$ will be a constant of -1 . a_0 is the starting point of this trend. The simulation level stops when the characteristic trend is found in the plotted curve and the value of M_j is selected as $M_j = e^{a_0}$.

Let $m-1$ denote the stopping simulation level and the failure event F_{mj} is defined as $F_{mj} = F_j$. The corresponding c_{mj} equals to M_j . Thus, the failure event F_m is given by the union of F_{mj} , $F_m = \bigcup_{j=1}^{N_j} F_{mj}$. Then the samples which belong to the failure event F_m are retained and used as seeds to simulate additional samples for the requirement of optimization.

The number of samples required for level i , N_i , is based on the overlapping of the failure events F_{ij} . To maintain a certain level of accuracy, a certain number of samples are required for estimating the conditional failure probability $P(F_{(i+1)j} | F_{ij})$ for each performance function at each simulation i . The number of samples which belong to F_{ij} has to be about N_0 , which is the number of samples used at level zero. Since the number of samples for level i are simulated from the retained samples at level $i-1$, N_i can be determined based on the overlapping of the retained samples. Let Nr_{i-1} denote the number of samples retained at level $i-1$ and $Nr_{(i-1)j}$ denote the number of samples which belong to the failure event F_{ij} . N_i can be selected as follow:

$$N_i = N_0 \max_j \left\{ \frac{Nr_{(i-1)}}{Nr_{(i-1)j}} \right\} \quad (5.13)$$

Let NS denote the number of samples required for the optimization. This means at least NS samples have to be obtained for each objective function. It is assume that the SS simulation stops at level $m-1$. Therefore, the number of samples required at the first optimization level is given as follow:

$$NS_1 = NS \max_j \left\{ \frac{Nr_m}{Nr_{mj}} \right\} \quad (5.14)$$

Initialization: decide the number of samples at the first level N_0 and the value for the pre-specified probability p_0 .

Step 1: simulate N_0 sample pairs $\{\boldsymbol{\theta}, \boldsymbol{\varphi}\}$ from the distribution $p(\boldsymbol{\theta})p(\boldsymbol{\varphi})$, set $i = 1$

Step 2: determine the value for $c_{ij}, i = 1, j = 1, \dots, N_j$ based on p_0 . This defines the failure events $\{F_{ij}\}_{i=1}$ and the failure event $F_i = \bigcup_{j=1}^{N_j} F_{ij}$.

Step 3: retain the sample pairs $\{\boldsymbol{\theta}, \boldsymbol{\varphi}\}$ that are distributed according to $q(\boldsymbol{\theta}, \boldsymbol{\varphi} | F_i)$.

The total number of the retained samples is Nr_i . The number of the retained samples that belong to the failure event F_{ij} is Nr_{ij} . Determine the number of

samples for the next level $N_{i+1} = N_0 \max_j \left\{ \frac{Nr_i}{Nr_{ij}} \right\}$

Step 4: plot $\ln P(X_j > a)$ versus a to check whether c_{ij} achieves M_j . Keep $F_{nj} = F_{ij}, n > i$ for the upcoming levels if $c_{ij} > M_j$.

Step 5: check the stopping criterion. If $c_{ij} > M_j$ is true for any j , stop the iteration and set $m = i$. Determine the corresponding failure event F_{mj} based on M_j and

the failure event $F_m = \bigcup_{j=1}^{N_j} F_{mj}$. Otherwise, simulate $(N_{i+1} - Nr_i)$ additional sample pairs $\{\boldsymbol{\theta}, \boldsymbol{\varphi}\}$ by MCMC regarding the retained samples that are distributed according to $q(\boldsymbol{\theta}, \boldsymbol{\varphi} | F_i)$ and return to step 2 by setting $i = i + 1$

Step 6: retain the sample pairs $\{\boldsymbol{\theta}, \boldsymbol{\varphi}\}$ that are distributed according to $q(\boldsymbol{\theta}, \boldsymbol{\varphi} | F_m)$.

The total number of the retained samples is Nr_m . The number of the retained samples that belong to the failure event $F_{mj} = F_j$ is Nr_{mj} . Determine the number of

samples for the optimization level $NS_1 = NS \max_j \left\{ \frac{Nr_m}{Nr_{mj}} \right\}$. Simulate $(NS_1 - Nr_m)$ additional sample pairs $\{\boldsymbol{\theta}, \boldsymbol{\varphi}\}$ by MCMC regarding the retained samples that are distributed according to $q(\boldsymbol{\theta}, \boldsymbol{\varphi} | F_m)$.

5.2.3 Proposed algorithm for simultaneously solving multiple optimization problems

At the first optimization level ($k=1$), the proposed sampling algorithm in the previous section allows obtaining NS_1 samples distributing according to $q(\boldsymbol{\theta}, \boldsymbol{\varphi}, u | F_m)$. Taking those samples which belong to the failure event F_{mj} gives us the samples following $q_j(\boldsymbol{\theta}, \boldsymbol{\varphi}, u | F_j)$ since F_{mj} is the same as F_j . Therefore the samples from each $\pi_j(\boldsymbol{\varphi})$ are obtained. These samples can be used to raw data by BSP to identify the optimal subset I_{kj}^* for each objective function. Given that overlapping exists in the identified N_j subsets $\left\{ I_{kj}^* \right\}_{j=1 \dots N_j}$, the optimal subset at the optimization level k is defined as follow:

$$I_k^* = \bigcup_{j=1}^{N_j} I_{kj}^* \quad (5.15)$$

Therefore the samples whose $\boldsymbol{\varphi}$ component belong to the subset I_k^* are retained and distributed according to $q\left(\boldsymbol{\theta}, \boldsymbol{\varphi}, u \middle| F_m, \boldsymbol{\varphi} \in I_k^* \right)$. For some cases, I_k^* is chosen as the hyper-rectangle which contains all N_j subsets $\left\{ I_{kj}^* \right\}_{j=1 \dots N_j}$. Let NSr_k denote the number of the retained samples at the optimization level k and NSr_{kj} denote the number of the retained samples belonging to the failure event F_{mj} . These retained

samples are regarded as seeds to simulate additional samples by MCMC techniques. The number of samples required for the optimization level $k+1$ is given as follow:

$$NS_{k+1} = NS \max_j \left\{ \frac{NSr_k}{NSr_{kj}} \right\} \quad (5.16)$$

The simulated samples can be used for identifying the optimal subset at the optimization level $k+1$. The optimization iteration proceeds until the stopping criterion is satisfied. When there is no overlapping between the identified subsets, it equals to run the optimization independently for each objective function.

5.3 Illustrative example

For illustrative purpose, this work aims to identify the optimal TMD in the platform of a TLP (Tension Leg Platform) floating offshore wind turbine to optimize two performance functions, respectively. These two optimization problem were solved simultaneously by the proposed algorithm to obtain two optimal design choices.

5.3.1 Case description

The TLP floating offshore wind turbine consists of a representative 5-MW wind turbine model as presented in chapter 3 and the tension leg platform. The wind turbine is supported by the tension leg platform and the platform connects to the seabed by 8 mooring lines. Details of the wind turbine model can be found in chapter 3. The properties of TLP and mooring lines are listed in Table 5.1. Table 5.2 lists the natural frequencies of the tension leg platform.

Table 5.1 Properties of TLP and mooring lines (Matha 2010)

Property	Value
Number of mooring lines	8
Fairlead distance from center	18m
Unstretched mooring-line length	151.73m
Line diameter	0.127m
Line mass per unit length	116.03kg/m
Line extensional stiffness	1500000000N

Average steel density	7850kg/m ³
Average concrete density	2562.5kg/m ³
Steel-wall thickness	0.015m

Table 5.2 Natural frequencies of TLP wind turbine (Matha 2010)

Mode	Value (rad/s)	Mode	Value (rad/s)
Platform roll	1.4	Platform surge	0.1036
Platform pitch	1.389	Platform sway	0.1036
Platform yaw	0.611	Platform heave	2.75
1 st tower F-A	3.965	1 st tower S-S	3.61
2 nd tower F-A	19.2	2 nd tower S-S	19.76

This work considers two critical performance functions of the TLP floating wind turbine, which are the fore-aft fatigue damage at the tower base and the fatigue damage of the mooring lines at the location close to the seabed. A single TMD is located inside the platform and its mass is selected as 20,000kg which will not significantly reduced the tension force of the mooring lines.

The two optimization problems are given as follows:

$$P_1 : \boldsymbol{\varphi}_1^* = \arg \min_{\boldsymbol{\varphi} \in \Phi} \int_{\Theta} h_1(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(\boldsymbol{\theta}) d\boldsymbol{\theta}$$

$$P_2 : \boldsymbol{\varphi}_2^* = \arg \min_{\boldsymbol{\varphi} \in \Phi} \int_{\Theta} h_2(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(\boldsymbol{\theta}) d\boldsymbol{\theta}$$

where $\boldsymbol{\theta} = [U, H_s, T_p, \theta_{Angle}, \boldsymbol{\theta}_{wind}, \boldsymbol{\theta}_{wave}]$ is the random variables of interest as defined in chapter 4 with the corresponding space Θ . The design vector is $\boldsymbol{\varphi} = [\omega, \xi]$, including the natural frequency ω and damping ratio ξ of the TMD. The design spaces for ω and ξ are selected as $[0, 5]$ rad/s and $[0, 1]$, respectively. The corresponding design space Φ is $[0, 5] \otimes [0, 1]$.

5.3.2 Algorithm characteristics

The proposed sampling algorithm is used to simulate samples for two PDFs simultaneously. The parameters for the proposed sampling algorithm are chosen as follows: $N_0 = 800$, $p_0 = 0.2$. The number of samples required by the proposed optimization algorithm is chosen as $NS = 2000$.

5.3.3 Results and discussion

Table 5.3 shows the number of samples simulated at each sampling level. By using the proposed sampling algorithm, 800 samples are simulated in the first sampling level. The number of retained samples is 190 with 160 samples for both the failure event F_{11} and the failure event F_{12} . Thus the second sampling level needs 950 samples and MCMC is used to simulate additional 760 samples based on the retained 190 samples. Based on the samples simulated at the sampling level 1 and level 2, the plots of $\ln P(X_1 > a)$ versus a and $\ln P(X_2 > a)$ versus a are shown in Figure 5.1 and Figure 5.2, respectively. In both plots, the slope of -1 is identified and the values of M_1 and M_2 are determined correspondingly. As shown in Figure 5.1 and Figure 5.2, M_1 and M_2 are determined as $M_1 = 1.545 \times 10^{-8}$ and $M_2 = 8.7 \times 10^{-9}$, respectively. Thus, the sampling iteration terminates at the second level as $c_{21} > M_1$ and $c_{22} > M_2$ are satisfied. Finally, the number of the retained samples is 250. Among the retained samples, there are 187 samples and 230 samples that are distributed according to the target distribution $q(\theta, \phi | F_{21})$ and $q(\theta, \phi | F_{22})$ for P_1 and P_2 , respectively. These retained samples are regarded as seeds to simulate additional samples for optimization by MCMC. The length of the Markov chain is 10 as $NS_1 = 2674$. Thus, MCMC simulated 2640 additional samples and the total number of samples is 2750 with 2137 for P_1 and 2302 for P_2 . This ensures that the number of samples for both optimization problems is more than $NS = 2000$.

Table 5.3 The number of samples simulated at each sampling level

	Simulated samples in total	Simulated sample for P_1	Simulated sample for P_2	Retained sample for P_1	Retained sample for P_2	Retained sample in total
Level 1	800	800	800	160	160	190
Level 2	950	802	805	187	230	250
Level 3	2750	2137	2302

Figure 5.1 Plot of $\ln P(X_1 > a)$ versus a

Figure 5.2 Plot of $\ln P(X_2 > a)$ versus a

Based on the simultaneously simulated 2750 samples, BSP is used to separately estimate the marginal PDF $\pi_1(\boldsymbol{\varphi})$ and $\pi_2(\boldsymbol{\varphi})$ by the corresponding 2137 samples and 2302 samples. For P_1 , the optimal partition over the initial design space Φ is presented in Figure 5.3 and the rectangle with red boundary line is the optimal sub-region I_{11}^* which has the lowest density. Similarly, Figure 5.4 shows the optimal partition and the optimal sub-region I_{12}^* for P_2 . By taking the union of both optimal sub-regions, the optimal sub-region I_1^* at the first optimization level is shown in Figure 5.5, which is $I_1^* = I_{11}^* \cup I_{12}^* = I_{11}^*$. As the optimal sub-region I_1^* is determined, the 2750 samples that are located inside I_1^* will be retained. The number of retained samples is 170. The number of retained samples for P_1 and P_2 is 151 and 165, respectively. Thus, the number of samples required at the second optimization level is 2252 and each retained sample has to generate 13 additional samples by MCMC. The total number of simulated samples at the second optimization level is 2210,

with 2013 for P_1 and 2085 for P_2 . Table 5.4 summarizes the number of samples at each optimization level.

Table 5.4 The number of samples at each optimization level

	Total simulated	Simulated for P_1	Simulated for P_2	Total retained	Retained for P_1	Retained for P_2
Level 1	2750	2137	2302	170	151	165
Level 2	2210	2013	2085	160	147	150
Level 3	2240	2041	2115

Figure 5.3 Optimal BSP and optimal sub-region I_{11}^* for P_1 at the first optimization level

Figure 5.4 Optimal BSP and optimal sub-region I_{12}^* for P_2 at the first optimization level

Figure 5.5 Optimal sub-region I_1^* identified at the first optimization level

Figure 5.6 Optimal BSP and optimal sub-region I_{21}^* for P_1 at the second optimization level

Figure 5.7 Optimal BSP and optimal sub-region I_{22}^* for P_2 at the second optimization level

After the second optimization level, the optimal sub-regions I_{21}^* and I_{22}^* were identified. The optimal sub-region at the second optimization is $I_2^* = I_{21}^* \cup I_{22}^* = I_{21}^*$. The samples retained inside I_2^* are used as seeds by MCMC to simulate additional samples to achieve the number of samples required at the third optimization level. The optimization iteration terminates at the third level as I_{31}^* equals to I_{21}^* and I_{32}^* equals to I_{22}^* for both cases. The optimal solution for each case is chosen as the center point of the corresponding optimal sub-regions I_{21}^* and I_{22}^* , respectively. The optimal design for P_1 is $[1.094, 0.094]$ and the optimal design for P_2 is $[1.016, 0.094]$.

Table 5.5 Computational cost of the proposed algorithm and separately solving each problem

	Number of dynamic simulations (actual time used)		Optimal solutions	Fatigue damage reduction
Proposed algorithm	8195 (81.95 hours)		$P_1: [1.094, 0.094]$ $P_2: [1.016, 0.094]$	$P_1: 25\%$ $P_2: 30\%$
Separately solving	$P_1: 7445$	In total: 15148 (151 hours)	$[1.094, 0.094]$	
	$P_2: 7693$		$[1.016, 0.094]$	

For comparison, P_1 and P_2 are solved separately by the proposed algorithm by setting the number of optimization problems $N_j = 1$. Table 5.5 shows the advantage in efficiency of the proposed algorithm for solving multiple stochastic optimization problems simultaneously. The proposed algorithm obtains the same optimal solution as solving each optimization problem separately. However, the proposed only requires about half computational cost and the advantage in saving

computational cost may be more significant as the number of optimization problems increases. The percentage of saving computational cost depends on the overlapping of the performance functions.

5.4 Conclusions

This chapter focuses on simultaneously solving multiple stochastic optimization problems for a single system. The ARS-SS algorithm for a single PDF which is proposed in chapter 3 is reviewed for its idea of simulating failure samples from the failure event for the corresponding auxiliary PDF of the augmented problem. This work proposes a new sampling algorithm to simulate samples simultaneously for multiple failure events based on the idea of conditional failure event in a sequential way. Based on the simulated samples for each optimization problem, BSP is used to identify the optimal sub-region separately and the optimal sub-region is the union of the optimal sub-regions for all optimization problems. The optimization iteration terminates when there is no overlap of the optimal sub-regions and then each optimization is solved separately by regarding the corresponding optimal sub-region as design space.

An illustrative example is performed to demonstrate the advantage of the proposed algorithm in simultaneously solving multiple stochastic optimization problems. The case aims to design a TMD located inside the platform of the TLP floating offshore wind turbine. Two performance functions are considered in this case study, including the fore-aft fatigue damage at the tower base and the fatigue damage of the mooring line close to the seabed. The proposed algorithm costs almost the same computational effort as solving a single optimization problem and identifies the same optimal solutions.

The proposed algorithm is able to simultaneously solving multiple stochastic optimization problems for a single system with the computational cost comparable to solving a single stochastic optimization problem. Future work will consider the multi-objective stochastic optimization problem.

Chapter 6 Effects of statistical models for the environmental conditions on the system performance and optimal design

6.1 Introduction

In real engineering practice, the lifetime performance of the engineering system needs to consider all possible environmental conditions that the system will be subjected to during its service life. The accuracy of the evaluation of the system performance function depends on the statistical model of the environmental conditions. Moreover, the statistical model of the sea states may also affect the optimal design of the structural control system for the floating offshore wind turbines considered in this work. In most cases, the environmental conditions are characterized by more than one variable and it is often the case that each variable is not independent. Thus, it is a challenging task to build an accurate model to characterize the environmental conditions.

The traditional method for building the statistical model which has several dependent variables is the conditional joint distribution model as used in chapter 2 by constructing the distributions to characterize the sea environment (Johannessen, Meling et al. 2002). However, the conditional joint distribution model depends on the assumption on the relationship and dependence of the random variables. It may neglect dependence of some random variables. The choice of a certain distribution type for the conditional PDF will imply a certain dependence structure between the random variables.

In addition to the probabilistic method which uses a single value of the probability, there are several non-probabilistic approaches for modelling uncertainties (Aven, Baraldi et al. 2013). These methods are based on probability-bound analysis, imprecise probability, possibility theory and evidence theory. The non-probabilistic methods are categorized into three types, the non-probabilistic method using lower and upper probabilities, the non-probabilistic method using interpretations other than lower and upper probabilities, and the hybrid approaches for combining probabilistic and non-probabilistic representations. So far, no comprehensive authoritative guidance exists on which method is better for modelling uncertainties.

However, the probabilistic approach may be more appropriate when a sufficient amount of data is available.

This work uses copula model to construct the statistical model for the environmental as copula model is able to model both the linear and nonlinear dependence of the random variables and consider the dependence of multiple random variables at the same time. The proposed ARS-SS sampling algorithm is able to simulate samples from any arbitrary PDF including the copula model. Therefore, the efficiency of the proposed optimization algorithm will not be affected much by the used statistical model. Section 6.2 presents the copula model method and gives the criterion of model selection. An illustrative example is presented in section 6.3 to investigate the effect of the statistical model of the environmental conditions on the performance function and the optimal design. The illustrative example studied a tension leg platform floating offshore wind turbine. Conclusions are given in section 6.4.

6.2 Copula method

As shown in the original meaning of the word “copula”, copula model is able to model the dependence of the random variables. However, the copula function is not directly used to model the dependence of the random variables. Instead, the copula function models the dependence of the random variables by considering the dependence of the marginal probability density function of each random variable. The definition of the copula function is given as follows:

$$[U_1, \dots, U_n] = [F_1(\theta_1), \dots, F_n(\theta_n)] \quad (6.1)$$

$$C(u_1, \dots, u_n) = P(U_1 \leq u_1, \dots, U_n \leq u_n) \quad (6.2)$$

where the random variables are denoted by $\boldsymbol{\theta} = [\theta_1, \dots, \theta_n]$ with the corresponding cumulative distribution functions $[F_1(\theta_1), \dots, F_n(\theta_n)]$ and $\mathbf{U} = [U_1, \dots, U_n]$ is the vector with each component satisfies $0 \leq U_i \leq 1$. U_i represents the cumulative distribution function for each random variable. Then, all the dependence information of the random variables is included in the copula function C and the

marginal distribution function involves the information of each random variable separately. There are several types of copula functions, such as Gaussian copula and T copula.

The procedure of copula method to construct the multivariate distribution based on the raw data is summarized as follows:

Step 1: based on the marginal samples for each random variable, construct the optimal marginal probability distribution function which has the maximum loglikelihood value.

Step 2: choose several types of copula functions to model the information obtained from the marginal distribution of each random variable.

Step 3: select the best copula model based on the given criterion

The parameters of the marginal probability distribution function are identified based on the criterion of the maximum loglikelihood. For each random variable, several types of distributions may be chosen, including Lognormal distribution, Gamma distribution and Weibull distribution. The best distribution is the one which obtains the largest maximum loglikelihood.

As the marginal probability distribution function of each random variable is obtained in step 1, the corresponding cumulative distribution function can be derived accordingly. These marginal cumulative distribution functions are used to identify the parameters for several types of copula functions.

The best copula model can be selected based on Akaike Information Criterion (AIC) (Akaike 1974) or Bayesian Information Criterion (BIC) (Schwarz 1978), which are widely used in model selection. The minimum value of the AIC and BIC is given as follows:

$$\begin{aligned} AIC &= -2l(k) + 2k \\ BIC &= -2l(k) + k \ln(m) \end{aligned} \tag{6.3}$$

where k denotes the number of parameters of the copula model. The maximum loglikelihood value is represented by $l(k)$, the value of which is defined as follows:

$$l(k) = \sum_{j=1}^m \ln C(F_1(\theta_{1j}), \dots, F_n(\theta_{nj})) + \sum_{j=1}^m \sum_{i=1}^n \ln f_i(\theta_{ij}) \quad (6.4)$$

where n is the number of copula dimension which equals to the number of random variables. m denotes the number of samples or raw data. $f_i(\theta_{ij})$ represents the marginal probability distribution function value and $F_i(\theta_{ij})$ is the corresponding cumulative distribution function.

The best copula model is selected as the one which gives the lowest AIC or BIC value.

6.3 Illustrative example

For illustrative purpose, a tension leg platform floating offshore wind turbine is studied. The details about the platform and wind turbine can be found in chapter 5. The floating wind turbine is assumed to be located at a site near the Shetland Islands because this site has fairly harsh wind and wave conditions which are believed to lead large structural load and bad performance functions of the system. The raw data about the environmental conditions is downloaded from <http://waveclimate.com>. There are four random variables to characterize the wind and wave conditions, including the wind speed U , the significant wave height H_s , the spectral peak period T_p and the wind-wave misalignment angle θ_{Angle} . The raw data involves 67208 samples. The statistical model of the environmental conditions that constructed by the traditional conditional joint distribution method can be found in Chapter 2.

Table 6.1 Results of marginal probability distributions for each random variable

	U	H_s	T_p	θ_{Angle}
Lognormal distribution $f(x \mu, \sigma) = \frac{1}{x\sigma\sqrt{2\pi}} \exp\left(-\frac{(\ln x - \mu)^2}{2\sigma^2}\right)$	$\mu = 2.096$ $\sigma = 0.5673$ $Log = -198170$	$\mu = 1.00$ $\sigma = 0.5595$ $Log = -123560^*$	$\mu = 2.224$ $\sigma = 0.2735$ $Log = -157730$	$\mu = 5.14$ $\sigma = 0.5486$ $Log = -400470$
Gamma distribution $f(x k, \theta) = \frac{1}{\Gamma(k)\theta} x^{k-1} e^{-\frac{x}{\theta}}$	$k = 3.774$ $\theta = 2.475$ $Log = -194580$	$k = 3.4132$ $\theta = 0.9288$ $Log = -124600$	$k = 14.137$ $\theta = 0.6779$ $Log = -156640^*$	$k = 5.3139$ $\theta = 35.4075$ $Log = -386800$
Weibull distribution $f(x k, \lambda) = \frac{k}{\lambda} \left(\frac{x}{\lambda}\right)^{k-1} \exp\left[-\left(\frac{x}{\lambda}\right)^k\right]$	$k = 10.5613$ $\lambda = 2.177$ $Log = -193970^*$	$k = 3.592$ $\lambda = 1.8735$ $Log = -127290$	$k = 10.5387$ $\lambda = 4.1013$ $Log = -157870$	$k = 208.56$ $\lambda = 3.0962$ $Log = -378340^*$

*the best distribution model which has the maximum Loglikelihood

Figure 6.1 Histogram and marginal PDF of the wind speed U

Figure 6.2 Histogram and marginal PDF of the significant wave height H_s

Figure 6.3 Histogram and marginal PDF of the spectral peak period T_p

Figure 6.4 Histogram and marginal PDF of the wind and wave misalignment angle θ_{Angle}

The copula method is used to construct the statistical model of the environmental conditions based on the raw data. The results of marginal probability distribution function for each random variable are shown in Table 6.1. The best marginal probability distributions for the wind speed U , the significant wave height H_s , the spectral peak period T_p , and wind-wave misalignment angle θ_{Angle} are Weibull distribution, Lognormal distribution, Gamma distribution and Weibull distribution, respectively. And Figure 6.1-4 show the histogram of the raw data and the best marginal probability distribution function.

Table 6.2 Parameters for Gaussian and T copula model

	Parameters
Gaussian copula	$\begin{bmatrix} 1 & 0.6315 & 0.0343 & 0.0519 \\ 0.6315 & 1 & 0.4507 & 0.0168 \\ 0.0343 & 0.4507 & 1 & -0.0014 \\ 0.0519 & 0.0168 & -0.0014 & 1 \end{bmatrix}$
T copula	$\begin{bmatrix} 1 & 0.6399 & 0.04 & 0.0378 \\ 0.6399 & 1 & 0.4651 & 0.0072 \\ 0.04 & 0.4651 & 1 & 0.0021 \\ 0.0378 & 0.0072 & 0.0021 & 1 \end{bmatrix}$ <p>Degrees of freedom: 24.209</p>

Table 6.3 AIC and BIC value for Gaussian copula. T copula and the traditional model

	AIC	BIC
Gaussian copula	1646000	1646100
T copula	1644400*	1644600*
Traditional model	1654000 ⁺	1653400 ⁺

*the lowest value means the best model +the highest value

Table 6.2 shows the results of Gaussian copula and T copula. Table 6.3 presents the AIC and BIC values for Gaussian copula model, T copula model and the traditional conditional joint distribution method. As shown in Table 6.3, the T copula model has the lowest AIC and BIC values and the traditional method has the highest AIC and BIC values. Thus, the T copula model is the best model to construct the statistical model for the environmental conditions.

To check the effect of the statistical model on the value of the performance function of the system, the lifetime fore-aft fatigue damage at the tower base and the mooring line fatigue damage at the seabed are evaluated based on 2000 samples simulated by the best statistical model and the model constructed by the traditional method, respectively. As shown in Table 6.4, both the mooring line fatigue damage

and the tower base fatigue is underestimated by the statistical model constructed by the traditional when compared to the best statistical model.

Table 6.4 The value of performance function by different statistical models

	Tower base fatigue damage	Mooring line fatigue damage
T copula	0.6453	0.6239
Traditional model	0.5392	0.5504

Table 6.5 The optimal design identified with different statistical models

	Optimal design	Fatigue damage reduction using T copula
T copula	[1.016, 0.0625]	28%
Traditional model	[1.094, 0.094]	20%

Figure 6.5 Optimal sub-region and optimal solution identified with different statistical models

This work also studied the effect of the statistical model for the environmental conditions on the optimal design of the TMD for minimizing the fore-aft fatigue damage at the tower base. The optimal design is identified by the algorithm proposed in chapter 4 with the density estimated by BSP. Figure 6.5 shows the optimal sub-region for the best statistical model and the model constructed by the traditional method and Table 6.5 lists the optimal solutions. The optimal design by T copula is more effective in reducing fatigue damage.

6.4 Conclusions

To construct the statistical model for the environmental conditions, this work uses the more realistic model, copula model, to describe the complicated dependence on the random variables characterized the environmental conditions. The procedure to construct the copula model and the criterion to select the best copula model are given. An illustrative example is presented to investigate both the effect of statistical model for the environmental conditions on the absolute value of the performance function of the system and the optimal design of the control system. The example considers a tension leg platform floating wind turbine located in fairly harsh environmental conditions and the raw data is used to construct the statistical model.

Compared to the statistical model built by the best copula model, the statistical model which is constructed by the traditional method and used in the previous chapters underestimates the performance function such as the fatigue damage at the tower base and the mooring line fatigue damage. This underestimation may due to the lack of dependence of the random variables in the statistical model constructed by the condition joint distribution method. The optimal design by T copula is more effective in reducing fatigue damage than the traditional joint distribution.

Chapter 7 Conclusions and future works

7.1 Conclusions

This thesis focuses on efficiently identifying the optimal solution for a performance-based design optimization problem and the proposed algorithms are used to identify the optimal design of the structural control system for floating offshore wind turbines. The efficiency is critical due to the time-consuming dynamic simulation of the system in time domain, such as the 10-min dynamic analysis of the floating wind turbine considered in this work may cost almost 10 minutes to obtain the dynamic response. The performance-based system design optimization $\boldsymbol{\varphi}^* = \arg \min_{\boldsymbol{\varphi} \in \Phi} \int_{\Theta} h(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(\boldsymbol{\theta}) d\boldsymbol{\theta}$ is a stochastic optimization problem as the uncertainties in the system model, environmental conditions and material properties need to be taken into account. By regarding the design variables as artificial uncertain variables uniformly distributed in the design space, the sensitivity of the objective function can be explored by investigating the marginal PDF $\pi(\boldsymbol{\varphi})$ of the auxiliary PDF $\pi(\boldsymbol{\theta}, \boldsymbol{\varphi}) \propto h(\boldsymbol{\theta}, \boldsymbol{\varphi}) p(\boldsymbol{\theta}) p(\boldsymbol{\varphi})$. The optimization task is divided into two parts, including simulating samples from the auxiliary PDF $\pi(\boldsymbol{\theta}, \boldsymbol{\varphi})$ and investigating the sensitivity of the marginal PDF $\pi(\boldsymbol{\varphi})$ with respect to the design variables. This thesis proposes algorithms to reduce the computational cost for both parts and investigates the effects of the statistical models of the environmental conditions on the optimal design using the proposed algorithms. The main work of the thesis is divided into four parts as follows:

- i. An efficient sample algorithm for simulating samples from the auxiliary PDF $\pi(\boldsymbol{\theta}, \boldsymbol{\varphi})$

The computational cost of the optimization task directly relies on the sampling efficiency of $\pi(\boldsymbol{\theta}, \boldsymbol{\varphi})$ as it requires dynamic simulations. The ARS-SS algorithm is proposed to simulate failure samples from $q(\boldsymbol{\theta}, \boldsymbol{\varphi} | F)$ by defining a failure event F such that these failure samples are distributed according to $\pi(\boldsymbol{\theta}, \boldsymbol{\varphi})$. The ARS-SS

algorithm is integrated with SSO to identify the optimal design of the TMDs for a spar buoy floating wind turbine. The optimal design obtained by SPSA is regarded as the benchmark solution. The main findings are summarized as follows:

- ARS-SS shows significant enhancement in the efficiency of the sampling task for case studies. ARS-SS only requires 6% of the computational cost of ARS.
 - The SSO with ARS-SS is able to identify a suboptimal design with less computational cost compared to SPSA, although the optimal design is slightly less effective than the optimal design by SPSA.
 - The optimal design identified by SSO with ARS-SS can reduce the fatigue damage by 28% for the two-TMDs control system.
 - The computational cost by SSO increases rapidly with the number of design variables and the size of the identified subset is relatively large compared to the design space.
- ii. An efficient stochastic optimization algorithm using proposed modified version of Bayesian Sequential Partitioning

In this part, the marginal PDF $\pi(\boldsymbol{\phi})$ is estimated by proposed modified version of BSP and the sub-region that has the lowest estimated PDF is regarded as the optimal sub-region. This part proposes a new algorithm by integrating modified BSP with ARS-SS. The main findings are shown as follows:

- With the same amount of computational cost, modified BSP with ARS-SS identifies better optimal solutions and much smaller optimal sub-regions than SSO with ARS-SS.
 - The optimal design for the two-TMDs control system is able to reduce the fatigue damage by as much as 33%.
- iii. The proposed algorithm for simultaneously solving multiple optimization problems for a single system

This part proposes an algorithm to solve multiple optimization problems simultaneously as a single system always has several performance functions to be

optimized. The purpose of solving them simultaneously is to save computational cost. The main findings are presented as follows:

- The proposed algorithm requires the computational cost which is comparable to the computational cost of solving a single stochastic optimization problem and identifies the optimal solutions for all design problems simultaneously.
- The optimal designs identified by the proposed algorithm are close to the optimal designs identified by solving each optimization problem separately.
- iv. Effects of the statistical models for the environmental conditions on the performance function and optimal design

The work in this part investigates the effects of the statistical models for the environmental conditions on the value of the performance function and the optimal design. The statistical model constructed by the traditional method is compared to the ones obtained by the copula models. The main findings are listed as follows:

- The T copula model is the best model to construct the statistical model for the sea environmental conditions based on the value of AIC and BIC.
- The statistical model constructed by the traditional method underestimates the performance functions compared with the T copula model.
- The statistical model also affects the effectiveness of the identified optimal design. The optimal design by the T copula model is more effective.

7.2 Future works

The future works may consider the problems listed as follows:

- Install more TMDs for better performance. Optimize the number of dampers and the location of each damper besides the parameters of the damper. This may highly increase the number of design variables.
- The multi-objective optimization will be considered. Other than identifying the optimal design for each objective function, the multi-objective optimization can be identified with the optimal designs which compromise all objective functions of interest.

- Consider the semi-active control system for further improving the system performance.
- Consider the optimization of the arrangement of the wind turbines in the wind farm.

References

- Akaike, H. (1974). "A new look at the statistical model identification." IEEE transactions on automatic control **19**(6): 716-723.
- Au, S. (2005). "Reliability-based design sensitivity by efficient simulation." Computers & structures **83**(14): 1048-1061.
- Au, S. and J. Beck (2003). "Subset simulation and its application to seismic risk based on dynamic analysis." Journal of Engineering Mechanics **129**(8): 901-917.
- Au, S. K. and J. L. Beck (2001). "Estimation of small failure probabilities in high dimensions by subset simulation." Probabilistic Engineering Mechanics **16**(4): 263-277.
- Aven, T., P. Baraldi, R. Flage and E. Zio (2013). Uncertainty in risk assessment: the representation and treatment of uncertainties by probabilistic and non-probabilistic methods, John Wiley & Sons.
- Bansal, S. and S. H. Cheung (2017). "On the evaluation of multiple failure probability curves in reliability analysis with multiple performance functions." Reliability Engineering & System Safety **167**: 583-594.
- Berg, B. A. (2004). Markov chain Monte Carlo simulations and their statistical analysis: with web-based Fortran code, World Scientific Publishing Company.
- Ching, J. and Y.-H. Hsieh (2007). "Local estimation of failure probability function and its confidence interval with maximum entropy principle." Probabilistic Engineering Mechanics **22**(1): 39-49.
- Commission, I. E. (2009). IEC 61400-3. Wind Turbines-Part 1. **3**.
- DiazDelaO, F., A. Garbuno-Inigo, S. Au and I. Yoshida (2017). "Bayesian updating and model class selection with Subset Simulation." Computer methods in applied mechanics and engineering **317**: 1102-1121.
- Dubourg, V., B. Sudret and J.-M. Bourinet (2011). "Reliability-based design optimization using kriging surrogates and subset simulation." Structural and Multidisciplinary Optimization **44**(5): 673-690.
- Hayman, G. (2012). MLife theory manual for version 1.00. National Renewable Energy Laboratory, Golden, CO.

- Jensen, H., M. Valdebenito and G. Schu ¨ dler (2008). "An efficient reliability-based optimization scheme for uncertain linear systems subject to general Gaussian excitation." Computer Methods in Applied Mechanics and Engineering **198**(1): 72-87.
- Jensen, H. A. (2005). "Design and sensitivity analysis of dynamical systems subjected to stochastic loading." Computers & structures **83**(14): 1062-1075.
- Johannessen, K., T. S. Meling and S. Haver (2002). "Joint distribution for wind and waves in the Northern North Sea." International Journal of Offshore and Polar Engineering **12**(1): 1-8.
- Jonkman, B. J. and L. Kilcher (2012). TurbSim User's Guide: Version 1.06.00.
- Jonkman, J. (2005). "Engineering challenges for floating offshore wind turbines."
- Jonkman, J. (2007). "Dynamics Modeling and Loads Analysis of an Offshore Floating Wind Turbine."
- Jonkman, J. M. (2005). FAST user's guide.
- Jonkman, J. M. (2007). Dynamics Modeling and Loads Analysis of an Offshore Floating Wind Turbine.
- Jonkman, J. M. (2007). Dynamics modeling and loads analysis of an offshore floating wind turbine, ProQuest.
- Jonkman, J. M. (2009). "Dynamics of Offshore Floating Wind Turbines-Model Development and Verification." Wind Energy **12**(5): 459-492.
- Jonkman, J. M. (2010). Definition of the Floating System for Phase IV of OC3, National Renewable Energy Laboratory Golden, CO, USA.
- Jonkman, J. M., S. Butterfield, W. Musial and G. Scott (2009). Definition of a 5-MW reference wind turbine for offshore system development, National Renewable Energy Laboratory Golden, CO.
- Katafygiotis, L. and J. Wang (2009). "Reliability analysis of wind-excited structures using domain decomposition method and line sampling." Structural Engineering and Mechanics **32**(1): 37-53.
- Katafygiotis, L. S. and K. M. Zuev (2008). "Geometric insight into the challenges of solving high-dimensional reliability problems." Probabilistic Engineering Mechanics **23**(2-3): 208-218.

- Lackner, M. A. and M. A. Rotea (2011). "Passive structural control of offshore wind turbines." Wind Energy **14**: 373-388.
- Liu, L. and W. H. Wong (2014). "Multivariate density estimation via adaptive partitioning (i): sieve mle." arXiv preprint arXiv:1401.2597.
- Liu, L. and W. H. Wong (2015). "Multivariate density estimation via adaptive partitioning (ii): posterior concentration." arXiv preprint arXiv:1508.04812.
- Lu, L., H. Jiang and W. H. Wong (2013). "Multivariate density estimation by bayesian sequential partitioning." Journal of the American Statistical Association **108**(504): 1402-1410.
- Matha, D. (2010). Model Development and Loads Analysis of an Offshore Wind Turbine on a Tension Leg Platform with a Comparison to Other Floating Turbine Concepts: April 2009, National Renewable Energy Laboratory (NREL), Golden, CO.
- Musial, W., S. Butterfield and B. Ram (2006). Energy from offshore wind. Offshore technology conference, Houston, Texas, Offshore Technology Conference.
- Papaioannou, I., W. Betz, K. Zwirgmaier and D. Straub (2015). "MCMC algorithms for subset simulation." Probabilistic Engineering Mechanics **41**: 89-103.
- Schwarz, G. (1978). "Estimating the dimension of a model." The annals of statistics **6**(2): 461-464.
- Si, Y., H. R. Karimi and H. Gao (2014). "Modelling and optimization of a passive structural control design for a spar-type floating wind turbine." Engineering Structures **69**: 168-182.
- Spall, J. C. (1992). "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation." IEEE transactions on automatic control **37**(3): 332-341.
- Stewart, G. and M. Lackner (2013). "Offshore Wind Turbine Load Reduction Employing Optimal Passive Tuned Mass Damping Systems." Control Systems Technology, IEEE Transactions on **21**(4): 1090-1104.
- Taflanidis, A. A. and J. L. Beck (2008). "An efficient framework for optimal robust stochastic system design using stochastic simulation." Computer Methods in Applied Mechanics and Engineering **198**(1): 88-101.

Taflanidis, A. A. and J. L. Beck (2008). "Stochastic Subset Optimization for optimal reliability problems." Probabilistic Engineering Mechanics **23**(2-3): 324-338.

Valdebenito, M. and G. Schuëler (2011). "Efficient strategies for reliability-based optimization involving non-linear, dynamical structures." Computers & Structures **89**(19-20): 1797-1811.

Valdebenito, M. A. and G. I. Schuëler (2010). "A survey on approaches for reliability-based optimization." Structural and Multidisciplinary Optimization **42**(5): 645-663.

Wayman, E. N., P. D. Slavounos, S. Butterfield, J. Jonkman and W. Musial (2006). Coupled Dynamic Modeling of Floating Wind Turbine Systems: Preprint.

Appendix A

The modified Metropolis-Hastings algorithm can be used to generate samples according to $q(\boldsymbol{\theta} | F_i)$ when some samples that follow this distribution are already available. Each sample generates a Markov Chain as follows:

1. Given the current sample $\boldsymbol{\theta}_k$ following $q(\boldsymbol{\theta} | F_i)$. Generate a pre-candidate state $\tilde{\boldsymbol{\theta}}_{k+1}^i$ using the proposal PDF $f_i(\tilde{\boldsymbol{\theta}}_{k+1}^i | \boldsymbol{\theta}_k^i)$, where i is the component counter of $\boldsymbol{\theta}_k$.
2. Compute the acceptance ratio

$$r_{k+1}^i = \frac{q_i(\tilde{\boldsymbol{\theta}}_{k+1}^i) f_i(\boldsymbol{\theta}_k^i | \tilde{\boldsymbol{\theta}}_{k+1}^i)}{q_i(\boldsymbol{\theta}_{k+1}^i) f_i(\tilde{\boldsymbol{\theta}}_{k+1}^i | \boldsymbol{\theta}_k^i)} \quad (\text{A.1})$$

and set

$$\boldsymbol{\theta}_{k+1}^i = \begin{cases} \tilde{\boldsymbol{\theta}}_{k+1}^i & \text{with probability } \min(1, r_{k+1}^i) \\ \boldsymbol{\theta}_k^i & \text{with probability } 1 - \min(1, r_{k+1}^i) \end{cases} \quad (\text{A.2})$$

3. Do 1 and 2 for all components of the vector $\boldsymbol{\theta}_k$. Accept it as the next state if $\boldsymbol{\theta}_{k+1} \in F_i$; otherwise reject it and set $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k$.

In the illustrative study, $\{U, H_s, T_p\}$ are considered as a group to compute the acceptance ratio because of their correlation. The proposal PDFs $f_i(\tilde{\boldsymbol{\theta}}_{k+1}^i | \boldsymbol{\theta}_k^i)$ for random variables $\{\boldsymbol{\theta}_e, \boldsymbol{\phi}\}$ are chosen to be uniform PDFs with width two times of the standard deviations of the available seeds. For the white Gaussian noise random variables, uniform PDFs with width 2 are used.

Appendix B

To solve the simplified optimization problem in Equation (3.15), the set of the admissible subsets has to be defined such that the boundary of the subset is inside the design space of the previous level. In this study, the shape of the admissible subsets is chosen to be hyper-ellipse and the design space is a rectangular box. The procedure which we used to check whether a hyper-ellipse is inside a rectangular box is shown as following.

The boundary of the ellipse is defined as:

$$(\mathbf{x} - \mathbf{x}_0)^T \mathbf{A} (\mathbf{x} - \mathbf{x}_0) = 1 \quad (\text{B.1})$$

where $\mathbf{A}$ is a positive definite symmetric matrix and $\mathbf{x}_0$ is the center of the ellipse.

The boundary of the rectangular box is defined as:

$$\begin{aligned} \mathbf{B}_i \mathbf{x} &= L_i \\ \mathbf{B}_i \mathbf{x} &= U_i \end{aligned} \quad (\text{B.2})$$

where L_i and U_i are the lower and upper bound for the i th component of $\mathbf{x}$, respectively. $\mathbf{B}_i$ is a row vector and only its i th component has a non-zero value of 1. In total, $2n$ (n is the dimension of the box) equations are required to define the boundary of the box.

Let $\mathbf{C}$ be the Cholesky decomposition for $\mathbf{A}$. By the following transformation:

$$\mathbf{y} = \mathbf{C}(\mathbf{x} - \mathbf{x}_0) \quad (\text{B.3})$$

the ellipse in the $\mathbf{x}$ space is transformed into a unit radius hyper-sphere in the $\mathbf{y}$ space.

The boundary of the rectangular box in the $\mathbf{y}$ space is transformed to be:

$$\begin{aligned} \mathbf{B}_i \mathbf{C}^T \mathbf{y} + \mathbf{B}_i \mathbf{x}_0 &= L_i \\ \mathbf{B}_i \mathbf{C}^T \mathbf{y} + \mathbf{B}_i \mathbf{x}_0 &= U_i \end{aligned} \quad (\text{B.4})$$

To ensure the ellipse inside the box, the unit radius hyper-sphere has to be inside the box in the $\mathbf{y}$ space. Equivalently, the minimum distance from the original point to

any boundary of the box should be larger than or equal to 1. The following equation should be satisfied.

$$\min_i \left\{ \left| \frac{\mathbf{B}_i \mathbf{x}_0 - L_i}{\sqrt{(\mathbf{B}_i \mathbf{C}^T)(\mathbf{B}_i \mathbf{C}^T)^T}} \right|, \left| \frac{\mathbf{B}_i \mathbf{x}_0 - U_i}{\sqrt{(\mathbf{B}_i \mathbf{C}^T)(\mathbf{B}_i \mathbf{C}^T)^T}} \right| \right\} \geq 1 \quad (\text{B.5})$$