Univerisity of Exeter

Doctoral Thesis

Robustness analysis of VEGA launcher model based on effective sampling strategy

Author:
Siyi Dong

Supervisor:
Dr. Prathyush Menon

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy in Mathematics

November 2016

Declaration of Authorship

I, Siyi Dong, declare that this thesis titled, 'Robustness analysis of VEGA launcher model based on effective sampling strategy' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

UNIVERSITY OF EXETER

Abstract

Colleges of Engineering, Mathematics and Physical Sciences

Doctor of Philosophy in Mathematics

Robustness analysis of VEGA launcher model based on effective sampling strategy

by Siyi Dong

An efficient robustness analysis for the VEGA launch vehicle is essential to minimize the potential system failure during the ascending phase. Monte Carlo sampling method is usually considered as a reliable strategy in industry if the sampling size is large enough. However, due to a large number of uncertainties and a long response time for a single simulation, exploring the entire uncertainties sufficiently through Monte Carlo sampling method is impractical for VEGA launch vehicle. In order to make the robustness analysis more efficient when the number of simulation is limited, the quasi-Monte Carlo(Sobol, Faure, Halton sequence) and heuristic algorithm(Differential Evolution) are proposed. Nevertheless, the reasonable number of samples for simulation is still much smaller than the minimal number of samples for sufficient exploration. To further improve the efficiency of robustness analysis, the redundant uncertainties are sorted out by sensitivity analysis. Only the dominant uncertainties are remained in the robustness analysis. As all samples for simulation are discrete, many uncertainty spaces are not explored with respect to its objective function by sampling or optimization methods. To study these latent information, the meta-model trained by Gaussian Process is introduced. Based on the meta-model, the expected maximum objective value and expected sensitivity of each uncertainties can be analyzed for robustness analysis with much higher efficiency but without loss much accuracy .

Acknowledgements

Firstly, I would like to express my sincere gratitude to my supervisor Dr Prathyush Menon for the continuous support of my Ph.D study and related research, for his patience, motivation and immense knowledge. Besides, I thank my fellow labmates: Anusha Mujumdhar, Atul Kamath, Wenfei Wang and Chiara Mellucci, for the stimulating discussions and kindness support in any matters.In the meanwhile,I would like to thank the European Space Agency for the support of Simulink model and dynamic system of the VEGA launch vehicle. Last but not the least, I would like to thank my family: my parents and my wife for supporting me spiritually throughout my Ph.D study.

Contents

List of Figures

List of Tables

Abbreviations

AVUM	**A**ttitude and **V**ernier **U**pper **M**odule
ANOVA	**AN**alysis **O**f **VA**riance
BB	**B**rownian **M**otion
COG	**C**enter **O**f **G**ravity
DE	**D**ifferenctial **E**volution
ESA	**E**uropean **S**pace **A**gency
ELV	**E**uropean **L**aunch **V**ehicle
EOM	**E**quations **O**f **M**otion
GA	**G**enetic **A**lgorithm
GP	**G**aussian **P**rocess
HDE	**H**ybrid **D**ifferenctial **E**volution
LPV	**L**inear **P**arameter **V**arying
LV	**L**aunch **V**ehicle
PCA	**P**rinciple **C**omponent **A**nalysis
SRM	**S**olid **R**ocket **M**otor
TVC	**T**hrust **V**ector **C**ontrol

Chapter 1

Introduction

For the success of an aerospace mission, flight safety is one of the key aspect. The performance of the flight control system, even in the presence of uncertainties and disturbances, is vital to ensure safety. Typically every aerospace system will have a set of functional and operational limits and the control laws must ensure the operation of the system within these limits irrespective of the presence of uncertainty perturbations and external disturbances. Such control laws are called 'robust' to uncertainties and disturbances. It is vital to determine the level of robustness of the control laws at the design stage itself to ensure the success of the mission. Hence, robustness analysis is performed on the closed loop control system or closed loop model to determine any extreme condition of uncertainties that could make the system unstable, or in the loss of desired functional performance limits. Occurrence of such extreme conditions need to be avoided by all means either by limiting the operating levels (a sub level of performance typical for first flights) or by redesigning the control laws (contrary to aircraft problems, typical for a launch vehicle type of problems as there is limited possibility for a first flight with lower operational capability). If no instability or violation of performance limits, the level up to which the performance may be deteriorated are assessed. Furthermore, while determining the worst case uncertain parameter combination, it is also equally important to understand the sensitivity of each uncertainties to the designed objective functions. When the number of uncertainties present in the model are large, it is of typical interest to find out which uncertain parameters are key sensitive ones, often identified as 'driving uncertain parameters'.

1.1 An overview of simulation based robustness analysis

In aerospace industry, the methods for robustness analysis are normally divided into two categories: analytical based methods and simulation based methods. The analytical based methods, such as gain and phase margin, μ analysis, Sum of Squares (SoS) etc..., require the exact information of the dynamics of the model. For example, μ analysis requires the details of the state space model and the controller in a very specific form, so called Linear Fractional Representations or LFT models. However, in our research case, the model has to be treated as a 'black box' as the controller of the model are present but the precise details of them are unknown (confidential). The 'confidential' in our research means the VEGA company do not wish to release to detail of design parameters of their launch vehicle to a third party for business confidential purpose. This is often the case when dealing with industrial problems; many of them will be available under third party license and details won't be revealed. Under this circumstance, the only choice left to us is the simulation based methods. Normally, the Monte Carlo methods and optimization based methods are favoured by the aerospace industry. The main reasons for this affinity for such methods are simplicity and efficacy of the analysis approach. Furthermore, the popularity of the method stem from the wide adaptability of such methods to different models, applications and disciplines, and limited requirements for specialist newly trained skill set for the industry and agencies (especially when validated simulation/optimisation based tools are readily available from a third party to the industry, for example University of Exeter offering codes to the engineers in industry).

1.1.1 Sampling based method

Several sampling strategies are present in literature to distribute sample points, also in higher dimensional problems. Aerospace industry conventionally focus on Monte Carlo sampling methods for the purpose of robustness analysis. Monte Carlo method distributes sample points in a random manner. Three main advantages of the method are: (1) it is easy to implement; (2) it can be also used for risk assessment to explore the safety regions in uncertainty space that no constraint violation happens , (3) the convergence of covering the entire uncertainty space is independent of the dimension of uncertain parameters. In other word, as the minimal number of samples to best approximate an integration problem is calculated and expressed as a coefficient multiplying

a higher order term, the number of minimal samples based on Monte Carlo method is not related to the higher order term. In this research, a phrase 'order of convergence' in later chapters means 'the speed of convergence' or the minimal number of samples'. This phrase refers to how fast or how efficient the robustness analysis would be.
More details about the Monte Carlo method will be introduced in Chapter 3. However, a few applications of Monte Carlo method in aerospace industry for carrying out robustness analysis are listed here.

Hanson and *Beard* [3] implemented the Monte Carlo method to assist the design the model of launch vehicle and ensure the design launch vehicle is robustness to the uncertainties. The Monte Carlo method is applied to verify the design requirements. In their paper, they also mentioned the types of distribution (such as normal or Gaussian distribution) that is depended on the probability density function and discussed the minimum number of sampling size. *Hanson* and *Hall* [4] applied the Monte Carlo method to assist the design of Ares I type launch vehicle as well. The design requirements, such as structure load analysis, flight control design, failure analysis, under the varying of uncertainties are testes by Monte Carlo method. Then the assessments are carried out to analyze whether the performance of outputs obtained from the Monte Carlo simulation satisfy the design requirements. *Thipphavong* [5] also proposed the Monte Carlo simulation for safety test for scenario of separation of aircraft in air traffic control system. In this paper, the author combined the Monte Carlo method with features of fault tree to accelerate the risk acceessment of air traffic control. *Crespo*, *Giesy* and *Kenny* [6] used the Monte Carlo method for robustness analysis to determine the hard constraints based on the design requirements.

On the other hand, the quasi-Monte Carlo method, which is a low-discrepancy sequence, is not much used yet for robustness analysis in the aerospace industry [3]. The Monte Carlo and quasi-Monte Carlo methods are stated in a very similar way and indented to solve the generic integration problem. The difference between two methods is that the quasi-Monte Carlo is not randomly generate sampling points but in a more deterministic fashion. It has been noticed that for a number of problems that have been studied,especially in Finance industry [7–10], the quasi-Monte Carlo sampling approaches lead to significantly faster convergence with much smaller number of samples

for the mean and standard deviation of the output as compared to Monte Carlo sampling. More details about quasi-Monte Carlo Carlo can be found in Chapter 3. However, quasi-Monte Carlo method is studied by many other authors [11–15], and it is demonstrated as a more efficient way with a better convergence for solving the integration problem compared to Monte Carlo method. It is a popular method in finance, health and natural sciences for model validation and verification.

Furthermore, an approach that can take significant advantage from the sampling schemes is Gaussian Process emulation. This approach considers the simulator as 'black-box' with its inputs, which is modeled as a realization of a stochastic process. Prior information about simulator that is gathered and updated typically by sampling scheme, either Monte Carlo or one of the quasi Monte Carlo methods, is expressed as a Gaussian Process with mean and covariance functions. From a frequentest perspective, the origin of the methodology used a regression model in conjunction with a zero-mean GP to best predict simulator output at an untried input configuration. Later on a Bayesian perspective was taken to incorporate additional sources of uncertainty, for e.g., model discrepancy, parametric uncertainty, found in typical robustness analysis problems [16]. The underlying concept is applicable to uncertainty analysis [17], sensitivity analysis [18], calibration and also to optimization [16]. More details on the Gaussian Process emulation will be provided in Chapter 5. An excellent graduate textbook on this topic is [19].

1.1.2 Optimization based method

Optimization based robustness analysis is also widely used in aerospace engineering[20]. Normally, in order to avoid a local optimum, the global optimization strategies are more welcome than local optimization algorithms, or mixed strategies which bear the advantages from both local and global optimization methods. Especially, the evolutionary based algorithms are most popular when the study subject is complex and nonlinear system, such as Genetic Algorithm (GA) [21] and Differential Evolution (DE) [22]. The study on optimization based robustness analysis normally focus on adapting the global optimization algorithm to the subjective model and comparing the performance with Monte Carlo based worst case validation; or proposing a novel optimization algorithm or a strategy to modify an existing optimization/iteration algorithm tailoring to the

specific problem under consideration in order to improve the efficiency of worst case validation.

Oudin and *Puyou* [23] designed a adaptive control law for aircraft and applied one global optimization algorithm for robustness analysis to verify whether the behavior of controller meets the design requirements under the vary of uncertainties. In this thesis, the optimization algorithm only used as a tool for worst case analysis. *Marcos* and *Marina* [20] proposed different optimization algorithms for robustness analysis based on the ELV launch vehicle. In the meantime, the authors also compared the results of worst case validation with the results obtained through Monte Carlo method. It shows that the optimization based worst case validation has a better efficiency in worst-case search over the Monte Carlo method. The focus of the authors is on the study of results of worst-case results based on different algorithms. *Menon* also implement the different optimization strategies to study the robustness of an Linear Parameter Varying(LPV) controller for a re-entry vehicle [24]. The author mainly focus on proposing two hybrid optimization algorithms for robustness analysis. The hybrid optimization algorithm denotes a strategy to combine the both local and global optimization algorithms in order to achieve a good worst-case result. The hybrid optimization normally could obtain a better worst-case than ordinary a single local or global optimization algorithm [25]. But the drawback is the computational cost could be very expensive.

1.1.3 Brief summary

Simulation based robustness analysis is suitable for the complex models or the models with insufficient details. It is also able to transfer a problem from a continuous domain to discrete domain by scattering a sequence of points. The Table 1.1 followed summaries the application of simulation based robustness analysis and features of each method. In the table, 'L', 'NL', 'Aerospace', 'Other industry', 'Efficiency' and 'CC' denotes 'Linear system', 'Nonlinear system', 'application in Aerospace industry', 'application in other industry except aerospace industry', 'the efficiency when applied to robustness analysis' and 'computational cost/complexity' respectively. The characters '✓' means 'widely used' and '✗' means not or seldom used.

Method	L	NL	Aerospace	Other industry	Efficiency	CC
Monte Carlo	✓	✓	✓	✓	Medium	High
quasi-Monte Carlo	✓	✓	×	✓	Medium to High	High
Optimization	✓	✓	✓	✓	Medium to High	High

TABLE 1.1: Applications of simulation based robustness analysis and features of each method

1.2 Research motivation

Monte Carlo method is widely used for robustness analysis in the aerospace industry. This method is easy to implement and ideally convenient for complex system. However, the computational time required by Monte Carlo method for robustness analysis is not cheap and not quite efficient, especially when the dimension of the uncertainties is high and computational budget is constrained. Moreover, a single simulation of a complex model in aerospace engineering may often need longer simulation time. In this thesis, the main objective is looking for efficient alternate sampling and optimization methods for robustness analysis of a complex system, mainly rooted on the concepts on quasi Monte Carlo methods (for the first time in the case of a European launch vehicle model to best of authors knowledge) and optimization methods (of which few are available ones and others are integrated/tuned with the efficient sampling methods). The ideas developed in the thesis will also lever from the conventional Gaussian Process emulation which will be employed for the first time to a detailed launch vehicle model. To study the efficiency of robustness, we propose the quasi-Monte Carlo method and a modified Differential Evolution(DE) optimization algorithm to compare the efficiency with Monte Carlo. Moreover, the efficiency of robustness analysis can be affected by high dimension of uncertainties. Thus, the variance based sensitivity analysis is proposed to identify the driven variable to reduce the dimension of uncertainties. However, the variance-based sensitivity analysis is an expensive method for sensitivity analysis if the dimension of uncertainties is high. To lower the time cost for sensitivity analysis, the Gaussian Process emulation is introduced to approximate the statistic features of the distribution of the outputs. The statistical model generated by Gaussian process replaces the initial ELV launch model for the sensitivity analysis in order to reduce the computational burden and achieve an acceptable quality of the sensitivity index. The usability of Gaussian Process emulation models in revealing worst case behavior is also demonstrated.

1.3 Research contribution

The main research contributions of this thesis are:

- In this research, the quasi-Monte Carlo sampling methods are considered, applied to robustness analysis of a closed loop model of a launch vehicle of industrial standard, and verified to be more efficient way for analysis (mean, standard deviation and coverage of uncertain space) in our case compared to the conventional Monte Carlo method. Specific performance costs suggested by industrial experts (ELV, Italy) have been considered as metrics for the analysis. The results of the worst case analysis obtained by the quasi-Monte Carlo method is no worse than the Monte Carlo method at least. In most cases of the analysis, the quasi-Monte Carlo methods achieved a better worst case result than Monte Carlo method. Computational budget constraints, which typically followed in industrial practice [1] and affordable in most analysis cases [25], were imposed in all the analysis cases.

- The conventional DE optimization algorithm is modified with a new way for generation of initial population. Specifically the low discrepancy sampling method was used to generate the required good uniformity distribution for the initial population and then a short step-size parameter was used to enhance the efficacy of the conventional optimization scheme. Like many other published studies on population based evolutionary optimization schemes, several numerical experiments using standard test functions were carried out to validate the proposed concept. The modified optimization algorithm DE is approved as a much efficient method for robustness analysis of the launch vehicle control system model than the conventional DE algorithm, Monte Carlo method and quasi-Monte Carlo method.

- For sensitivity analysis, the variance based sensitivity analysis which exploits the sampling schemes is considered, and demonstrated as a good method to sort out the driving uncertain parameters, or the most sensitive uncertain parameters, for the case of relatively small dimensional uncertain parameter space problem. Here also, the computational budget constraints are present. Especially, it is a good

[1] ELV, Italy, ESA private communication

candidate method to study the sensitivity index of uncertain parameters when the model is given as 'black-box' model.

- Gaussian Process is successfully applied to obtain a statistic meta-model based on the ELV launch vehicle model. Based on the meta-model, the sensitivity index for the complete set of uncertain parameters is determined with an affordable computational burden and within an acceptable quality. The sensitivity index weight of each variable is identified and used to reduce the dimension of uncertainties by neglecting the non-important variables. The usefulness of the Gaussian Process methodology to reveal the worst case behaviour of the models effectively and swiftly is also demonstrated

1.4 Thesis organisation

In Chapter 2, a literature review is given to discuss the current research status on Monte Carlo, Quasi-Monte Carlo, sensitivity analysis and Gaussian Process.
In Chapter 3, the mathematical model of the VEGA launch vehicle model, designed and developed by the ELV, Italy, is introduced. The VEGA launch vehicle is a three stage rocket, in which the present study focuses on the ascending phase in the first stage only. The structure, dynamics, uncertainties, outputs and objective criteria for robustness analysis are given in this chapter.
The Monte Carlo, quasi-Monte Carlo, DE and its proposed variant are discussed and applied to determine robustness of the VEGA launch vehicle model in Chapter 4. The results and the computational efficiency of these methods are compared. In the meanwhile, the effects of wind is studied independently which is very important during the ascending phase.
In Chapter 6, the variance based sensitivity analysis is discussed and applied to study the driving uncertain parameters, or the most sensitive uncertain parameters. However, not all of the parameters are investigated because of the excessive computational cost associated with the methodology. Only 15 uncertainties are selected from the entire uncertainty set and studied by the variance based sensitivity analysis methodology.
In Chapter 7, the Gaussian Process is introduced to generate a computationally fast statistical meta-model mainly for the purpose of sensitivity analysis in higher dimensional

uncertainty space. The entire uncertain parameters are considered with this methodology, and sensitivity index of each uncertain parameter is efficiently determined at an affordable computational cost with good accuracy. The usefulness of the Gaussian Process method in determining the worst case perturbation is also discussed and demonstrated in Chapter 5.

In Chapter 8, the main conclusive conclusions and summary are given, along with a few narrated future works.

Chapter 2

Literature Review

A survey of the basic principles of the research approaches in this thesis is elaborated in this chapter, and includes discussion of the Monte Carlo and quasi-Monte Carlo methods, different evolution, sensitivity analysis, and the Gaussian process.

2.1 Monte Carlo method

Complex systems often have elements that behave in a stochastic manner, or have parameters vary within a bounded range. It is important to assess the variability of output responses of such complex systems. The Monte Carlo simulation method is often used to study the properties of such complex systems. Typically a representative mathematical model of suitable level of complexity is simulated on the computer by randomly sampling the parameters within their bounds, or randomly generating variables describing elements with the stochastic behavior. Subsequently the responses of interest are obtained from the simulation data and used for assessing maximum, variability, etc. and used for statistical inference. The Monte Carlo simulation is widely used in several disciplines, including engineering, finance, biology, physics, operations research, and management science. It is widely used in many fields of engineering, especially applied to robustness and reliability analysis for complex system.

For instance, for the launch vehicle considered in this thesis, there are many uncertain parameters that may affect the performance of the launcher including the external wind force on the vehicle, the rotation rate, the actual force provided by the thrusters, the

dynamic force and torques, and possible signal delay from the internal electronic system. Clearly, as the number of uncertain parameters increase, the underlying problem becomes abstruse due to the large dimensionality. An approach is to find out whether the launch vehicle would remain robustly stable (satisfy certain mission objectives or design requirements) even in the presence of the worst scenario (multiple uncertainties are varied simultaneously yielding an extreme output values) or not, by investigating all the outputs of interest. The method of evaluating only at the extremes of the uncertain parameters (i.e., at minimum and maximum value of the parameters, and their combinations for the vector of uncertain parameters) or at regular grid points (i.e, at minimum value to maximum value of the parameter at regular intervals) might not always necessarily reveal the inherent extreme, or worst-case behavior due to possible intricate nonlinear influence of the parameters on the dynamics of the system. Several researches [3–5, 26, 27] have considered basic Monte Carlo method as a tool for assessing the robustness and reliability of systems in aerospace engineering.

The Monte Carlo method is a potential approach to vary the uncertainties under one or many distribution rules in order to find a combination of values of the uncertainties close enough to the worst scenario. Normally it requires a large number of simulations for a good convergence. If the launch vehicle is stable in the presence of the worst case uncertain parameter that is obtained by the Monte Carlo method based simulations with a specific confidence level, then typically industry classify that the control laws of the launch vehicle as robust to these uncertainties and allows to be cleared for the mission (similar many other issues have to overcome prior to certification).

In robustness and reliability analysis, Monte Carlo method normally is employed as a tool for generation of the uncertain parameters and tries to reveal the possible condition that makes the system failure; or it indicates the probability of a system stay in the safety conditions over the entire mission. Industrial favoritism for the method seems to stem from the simplicity of the approach and possibility to adopt the concept easily across different platforms and projects. The principles of basic Monte Carlo method, the sampling requirement subject to specific degree of confidence, and its convergence are discussed.

2.1.1 Monte Carlo integration

The expected value of the Lebesgue integrable function $f(.)$ is expressed as the equation 2.1

$$\int_B f(u)\mathrm{d}u \tag{2.1}$$

Let u be a random variable. The integration domain is B $\subseteq \mathbb{R}^S$ and satisfy $0 < \lambda_S(B) < \infty$ where λ_S is the S-dimensional Lebesgue measure. It is possible to represent the integration domain B in probability space with probability measure as follows

$$\mathrm{d}\mu = \frac{\mathrm{d}u}{\lambda_S(B)} \ . \tag{2.2}$$

Then, it is easy to show that

$$\int_B f(u)\mathrm{d}u = \lambda_S(B)\int_B f\mathrm{d}\mu = \lambda_S(B)E(f) \tag{2.3}$$

where $E(f)$ is the expected value of the random variable f. Viewing from the statistical perspective, the numerical integration turns out to be the calculation of expected value.

Let 'f' be a random variable on an arbitrary probability space (A, N, λ). Monte Carlo estimate for the expected value $E(f)$ is obtained by taking N-independent λ-distributed random samples $a_1, \ldots, a_N \in A$ and letting

$$E(f) \approx \frac{1}{N}\sum_{n=1}^{N} f(a_n) \tag{2.4}$$

The Strong Law of Large Numbers [28] guarantees that this procedure converges almost surely in the asymptotic sense that

$$\lim_{N\to\infty} \frac{1}{N}\sum_{n=1}^{N} f(a_n) = E(f)\lambda^{\infty} - a.e. \tag{2.5}$$

where λ^{∞} is the product measure of numerable many copies of λ. The variance is given as

$$\sigma^2(f) = \int_A (f - E(f))^2\mathrm{d}\lambda \tag{2.6}$$

and is finite, whenever $f \in L^2(\lambda)$ (Lebesgue space).

To elaborate a bit further, if $f \in L^2(\lambda)$, then, for any $N \geq 1$, from the variance relation, we have

$$\int_A \dots \int_A (\frac{1}{N} \sum_{n=1}^{N} f(a_n) - E(f))^2 \mathrm{d}\lambda(a_1) \dots \mathrm{d}\lambda(a_N) = \frac{\sigma^2(f)}{N} \ . \tag{2.7}$$

Let us define $g := f - E(f)$, then ideally

$$\int_A g\lambda d\lambda = 0 \tag{2.8}$$

and

$$\frac{1}{N} \sum_{n=1}^{N} (f(a_n) - E(f)) = \frac{1}{N} \sum_{n=1}^{N} g(a_n) \ . \tag{2.9}$$

Thus, it may be shown as

$$\begin{aligned}
&\int_A \dots \int_A (\frac{1}{N} \sum_{n=1}^{N} f(a_n) - E(f))^2 \mathrm{d}\lambda(a_1) \dots \mathrm{d}\lambda(a_N) \\
&\quad = \int_A \dots \int_A (\frac{1}{N} \sum_{n=1}^{N} g(a_n))^2 \mathrm{d}\lambda(a_1) \dots \mathrm{d}\lambda(a_N) \\
&\quad = \frac{1}{N^2} \sum_{n=1}^{N} \int_A \dots \int_A g(a_n)^2 \mathrm{d}\lambda(a_1) \dots \mathrm{d}\lambda(a_N) \\
&\quad + \frac{2}{N^2} \sum_{1 \leq m < n \leq N} \int_A \dots \int_A g(a_m) g(a_n) \mathrm{d}\lambda(a_1) \dots \mathrm{d}(a_N) \\
&\quad = \frac{1}{N} \int_A g^2 \mathrm{d}\lambda = \frac{\sigma^2(f)}{N}
\end{aligned}$$

The absolute value of error in $E(f)$ is, on average, $\epsilon_E = \sigma(f) N^{-1/2}$ where $\sigma(f) = (\sigma^2(f))^{1/2}$ is the standard deviation of f. The Monte Carlo estimate is given as

$$\int_B f(u) \mathrm{d}u \approx \frac{\lambda_S(B)}{N} \sum_{n=1}^{N} f(x_n) \ , \tag{2.10}$$

where $x_1, \dots, x_N$ are N independent μ -distributed random samples from B. The absolute value of error, on average, is given by $\epsilon = \lambda_S(B) \sigma(f) N^{-1/2}$.

The key features of the Monte Carlo method are: (i) it has a probabilistic error bound $O(N^{-1/2})$, (ii) the order of the magnitude does not depend on dimension S, and preferable for dimension $S \geq 5$, (iii) comparing the classical integration scheme $O(N^{-2/s})$, (iv) the convergence rate is also defined as $O(N^{-1/2})$. Here, the applicability of the methodology to a wide class of problems has to be highlighted. In all the derivations,

the value of f does not necessarily require any analytical form. This can be function evaluated at a set of discrete points, this could be an analytical function, a complex function, a complex model, a simulator, no matter what way they are modeled. The only requirement is that one should be able to supply the necessary input conditions for f, which often will be a black-box complex simulator, and evaluate it at the respective supplied conditions, or instance of parameter occurrence. Ideally, Monte Carlo method is suitable for high dimensional problem according to the features described above. To improve the rate of convergence , some researches has already been done. However, this is not belonged to our research concerns. More details can be found in other researcher's work, such as [11].

The pattern of the random number distribution is based on the probability density function λ. The two categories of distribution are: (i) uniformly, and (ii) non-uniformly. The most commonly used non-uniformly distribution is normal/Gaussian distribution. More types of non-uniformly distribution can be found in the book [29]. It should be aware of that the sampling points generated by a computer is not exactly random. Instead, it is typically pseudo-random, which implies that the generation of the sequence of random number depends on many other properties as well. In this thesis, all random numbers are generated by the Matlab version 2009b due to compatibility with the VEGA model discussed in Chapter 2.

2.1.2 Requirement of Sampling size

Because the error bound ϵ is probabilistic, the precision/accuracy of the Monte Carlo integration can only be ensured with a degree of confidence[11]. According to the Central Limit Theorem [28], a minimal sampling size N is required for the Monte Carlo integration error with the confidence level c. The minimal number of N is expressed as

$$N = \epsilon^{-2}\sigma^2 h \tag{2.11}$$

where the h is the confidence function. The confidence level may be represented as

$$c = \int_{-h}^{h} \frac{1}{\sqrt{2\pi}} e^{-u^2/2} \mathrm{d}u$$

where the term $\lambda(u)$ is the probability density function.

Unfortunately, the σ in equation 2.11 cannot be calculated by the Monte Carlo integration itself. An easy way is to use M independent points over the random sequence N to calculate the empirical error and deviation. The empirical error, which makes use of the indicator function value $I^{(j)}$ and $1 \leq j \leq M$, is:

$$\widetilde{\varepsilon} = (\frac{1}{M}\sum_{j=1}^{M}(I^{(j)} - \bar{I})^2)^{1/2} \tag{2.12}$$

where the term $\bar{I}$ represents the average expressed as

$$\bar{I} = \frac{1}{M}\sum_{j=1}^{M} I^{(j)} \ .$$

Thus, the empirical deviation is $\widetilde{\sigma} = N^{1/2}\widetilde{\varepsilon}$, which can be used to replace the σ in equation 2.11 in order to calculate the value of N for a given confidence level c. This will be used to obtain an idea of the computational budget for a required confidence level, and allows to do the necessary trade off/compromise one has to make between computational limits and quality of solution.

From the subsection above, the most significant advantage of the Monte Carlo method is that the integration error(or convergence rate) only depends on the sampling number N, but not anything to do with the dimension. So Monte Carlo is ideal for high dimensional problem. By increasing the number of sampling points, the integration error can be made smaller. Moreover, the confidence level of the integration error can be approximated in terms of the sampling size N. One has to notice that the method does not guarantee revealing worst case behavior. The respective problem may be posed to be solved using the traditional methodology, but at the expense of significant computational budget.

2.1.3 Random number generation

In practice, the random number generated by the computer is not exactly random. The generation is based on some deterministic algorithms. So this process is called pseudo-random. The pseudo-random generates uniform random numbers. Some details of generating the pseudo-random numbers in theoretical level can be found in [30]. For non-uniformly distributed variables, such as Gaussian distribution, the density transformation method is applied.This method is also the general procedure for random number

production in different probability density function p.

Denote y is a uniform random number, and there is function of y that it makes $x = X(y)$. The x is the non-uniform random variable with probability density $p(x)$. In steady of drawing the non-uniform random variable directly based on its probability density $p(x)$, we product the uniform random numbers and transfer these numbers to non-uniform distributed numbers according to the mapping function $X(y)$. The $X(y)$ is found based on the computation of expectation. For a function f,

$$E_p[f(x)] = E_{unif}[f(X(y))] \tag{2.13}$$

by using the change of variable in integration,

$$\int f(x)p(x)\mathrm{d}x = \int f(X(y))p(X(y)X'(y)\mathrm{d}y \tag{2.14}$$

Because the expectation of a function $f(x)$ based on uniform distribution is:

$$E[f] = I[f] = \int f(x)\mathrm{d}x \tag{2.15}$$

Then

$$p(x)X'(x) = 1 \tag{2.16}$$

So that,

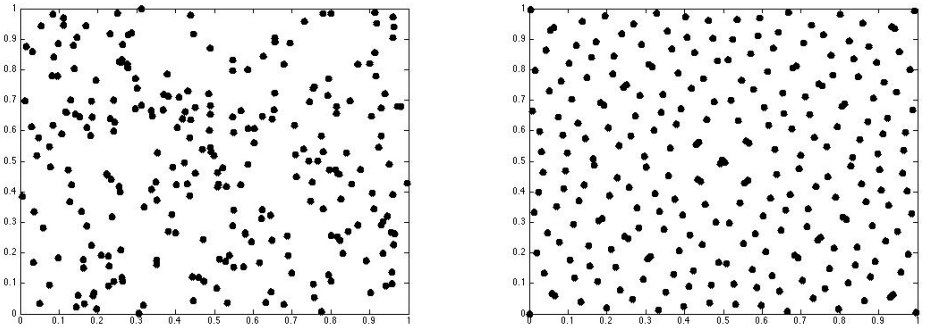
$$\begin{aligned} \int f(x)p(x)dx &= \int f(X(y))p(x)\mathrm{d}y \\ &= \int f(x)(\mathrm{d}y/\mathrm{d}x)\mathrm{d}x \end{aligned} \tag{2.17}$$

The probability density $p(x) = \frac{1}{X'(y)} = \mathrm{d}y/\mathrm{d}x$. Thus,

$$\int^{X(y)} p(x)\mathrm{d}x = y \tag{2.18}$$

The cumulative distribution function of $P(X(y))\mathrm{and}X(y)$ then is:

$$\begin{aligned} P(X(y)) &= y \\ X(y) &= P^{-1}(y) \end{aligned} \tag{2.19}$$



(A) The random sequence (B) The quasi-random sequence

FIGURE 2.1: The distribution of random and quasi-random sequence

Although the density transformation method is easy to implement, the inverse of the cumulative density function $P^{-1}(y)$ may not be easy to obtain.

2.2 Quasi-Monte Carlo Method

Quasi-Monte Carlo is a type of quasi-random sequence or low-discrepancy sequence. Compared to the Monte Carlo method, the quasi-Monte Carlo is a pseudo-random sequence which is deterministic. The term 'discrepancy' mentioned above is a measure of uniformity of a sequence. The quasi-Monte Carlo aims to provide a better uniformity when compared to the Monte Carlo. In the Figure 2.1, it shows the distribution of the 256 points for a two dimensional case. The random sequence is generated following a quasi-random sequence, here specifically following Sobol principle, which is discussed more in sequel. In the left sub figure, marked (a) of Figure 2.1, it is easy to see that certain space has accumulation of points, and certain space has no filling points at all. The granularity of filling of the space, while maintaining the randomness among the sampled points, is crucial to optimal sampling problems. A reason for this phenomena is because the sampled points are independent, and the fact that the points are generated without sharing any information. On the contrary, the quasi-random sequence uses the correlation information between the points for the generation of sampling points. Hence, as evidently seen in Figure (b), a better uniformity for the sampling points can be observed.

When the Quasi-Monte Carlo is applied to solve the integration problem, such as the equation 2.1, the convergence rate is $O((logN)^k N^{-1})$[30]. For large N and relatively

small dimension, the quasi Monte Carlo has a faster convergence rate over the Monte Carlo. However, for a high dimensional problem, the effectiveness of quasi-random sequence is limited. To overcome this drawback, some additional methodologies are introduced to regain the faster convergence of the quasi-random sequence for high dimensional problem in this section. To demonstrate the effectiveness of quasi-Monte Carlo in a realistic industrial research problem, three low discrepancy generation using Sobol sequence, Halton sequence and Faure sequence will be discussed in sequel.

2.2.1 Discrepancy

The discrepancy is the measure of uniformity of a sequence. Suppose a N points sequence $\{x_n\}$ is s-dimensional unit cube $I^s = [0,1)^s$. The local discrepancy for a sub interval J of I^s is defined as[31]:

$$R_N(J) = \frac{1}{N} A(J;N) - m(J) \tag{2.20}$$

where $A(J;N)$ is the number of points N falling in the sub interval, $1 \leq n \leq N$, m is the volume of J. J must be rectangular sub interval, and let E be defined as the set of all such sub intervals. In this additional framework, the local discrepancy R_N is also called the quadrature error in measuring the volume of J [11]. The overall discrepancy of the sequence x_n is defined as:

$$D_N = \sup_{J \in E} |R_N(J)| \tag{2.21}$$

Now, suppose every sub interval J in E has the value 0 for the lower bound, $I_J^s = [0,x)^s$, and denote the modified set of all such sub intervals as E_N^*, the discrepancy is then defined as:

$$D_N^* = \sup_{J \in E_N^*} |R_N(J)| \tag{2.22}$$

For a low-discrepancy sequence, a small value of discrepancy D_N^* is essential. In detailed surveys on quasi-Monte Carlo methods, Hammersley [32] and Halton [12] show that, for any dimension greater than 2, there exist N points in I^s with

$$D_N^* = O((logN)^{s-1} N^{-1}) \tag{2.23}$$

and there exists a sequence of points in I^s with

$$D_N^* = O((logN)^s N^{-1}) \tag{2.24}$$

Both equations indicate that the value of discrepancy is dependent on the order of the sequence and increases exponentially with dimension. This is also a reason why the quasi-Monte Carlo losses the effectiveness for high dimensional problem.

2.2.2 Quasi-Monte Carlo integration error

Because the conventional analysis based on probability theory is no longer able to apply (not i.i.d) to the quasi-Monte Carlo to find the error of the integration, the quasi-Monte Carlo integration error is defined based on an alternative approach called Koksma-Hlawka inequality [11, 13]. For a sequence $\{x_n\}$, the integration error of a bounded variation function $f(.)$ is subject to

$$\epsilon[f] = \left| \int_{I^s} f(x)\mathrm{d}x - \frac{1}{N}\sum_{i=1}^{N} f(x_i) \right| \leq V[f]D_N^* \tag{2.25}$$

The $V[f]$ is the variance function and defined in the Hardy-Krause sense [13] as

$$V[f] = \int_{I^d} \left| \frac{\partial^s f}{\partial t_1 \dots \partial t_s} \right| \mathrm{d}t_1 \dots \mathrm{d}t_s + \sum_{i=1}^{s} V[f_1^{(i)}] \tag{2.26}$$

where $f_1^{(i)}$ is the restriction of the function f to the boundary when $x_i = 1$.
There are various way to construct the quasi-Monte Carlo with small discrepancy, such as Sobol [33], Halton [12] and Faure sequence [34]. The discrepencies of all of these three sequences are bounded by

$$D_N \leq c_s(\log N)^s N^{-1} \tag{2.27}$$

Niederreiter in [14] provides the properties for a low-discrepancy sequence as that the discrepancy must satisfy:

$$D_N \leq c_s(\log N)^s N^{-1} + O((\log N)^{s-1} N^{-1}) \tag{2.28}$$

The bound of the discrepancy indicates that the quasi-random sequence with a small discrepancy has a smaller integration error compared to the random sequence.

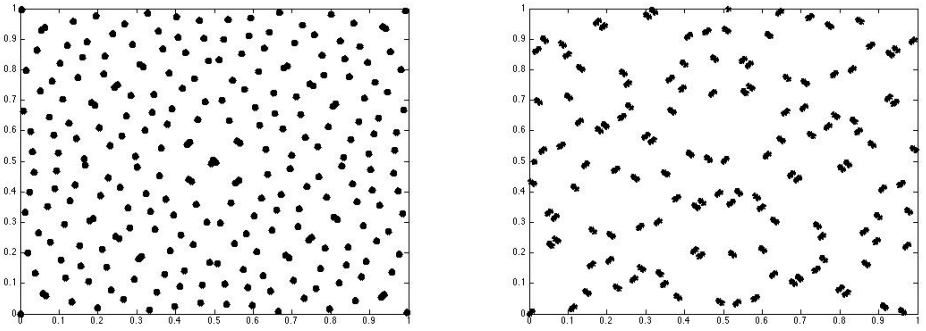
Based on the bound of the discrepancy D_N and the Koksma-Hlawka inequality, the integration error of the quasi-Monte Carlo is given by[11]:

$$\epsilon[f] \leq c_s V[f](\log N)^s N^{-1} \tag{2.29}$$

Thus, the convergence rate is considered as $O(\log N)^s N^{-1}$. A remark here is that, because the integration error using the quasi-Monte Carlo is an inequality relationship, it is not tight as the integration error of random sequence, which has an equality relation.

2.2.3 Discussion on quasi-Monte Carlo

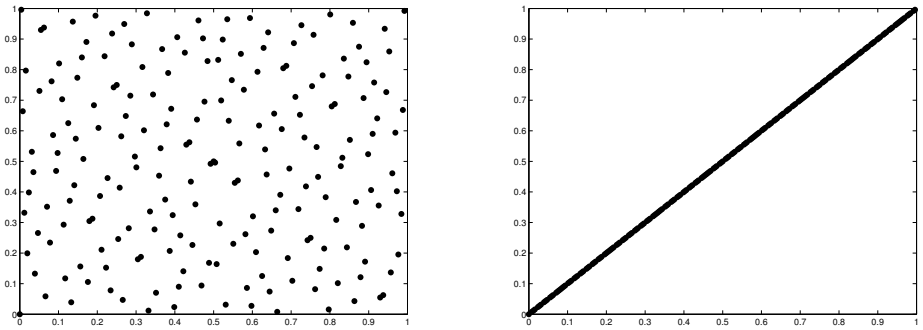
The quasi-Monte Carlo is designed to solve an integration problem and originally Caflisch expressed potentials and concerns about application of the quasi-Monte Carlo to simulation problems[11]. Unlike the crude Monte Carlo method, the quasi-Monte Carlo cannot provide a confident level of accuracy by choosing the sampling size N. Specifically for a large dimension case, the discrepancy bound is dominated by the term $(\log N)^s$ unless $N > 2^s$. In reality, constrained by computational capability of computer and the complexity of the simulation model, the sampling size N cannot tend to infinity or a relatively large value. The effectiveness of quasi-Monte Carlo that depends on the dimension is obvious. By observing the 2D projection of the distribution points, it may indicate whether the quasi-Monte Carlo is effective for the certain size of dimension. The Figure 2.2, 2.3 and 2.4 are examples of such projection based on Sobol, Faure and Halton sequence. The sequence in this figure is constructed by Sobol sequence and the dimension is 30. In the Figure (a), the projection for the lower dimensions (dimension 1 versus 2 as an example) is very uniform; on the other hand, the projection in some higher dimensions (such as the example in Figure (b) for dimension 29 versus 30) is extremely 'bad'. For the 'bad' projection, a larger number of sampling points is required to fill the empty space. In conclusion, the Sobol sequence may loss its effectiveness when the dimension size is 30. Normally, the 'bad' projection is rare observed when the dimension is below 10 for Sobol sequence. In Moskwitz experience [35], the quasi-Monte Carlo can remain in the superior position over crude Monte Carlo when the effective dimension is below 30.



(A) The projection of dimension 1 vs. 2 (B) The projection of dimension 29 vs. 30

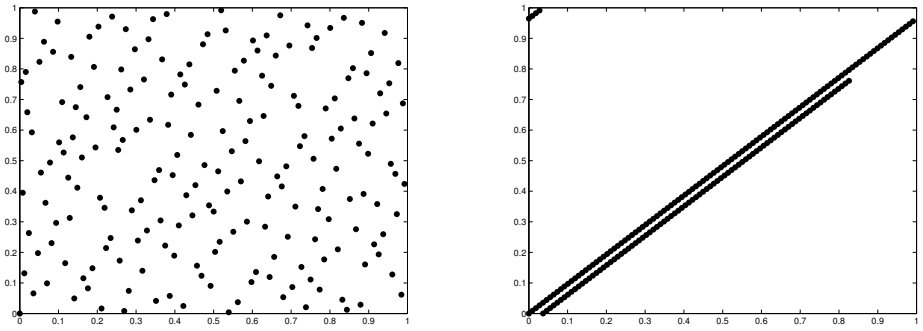
FIGURE 2.2: The 2D project of Sobol sequence

Wang[15], Kucherenko[36] and Yucesan[37], proposed the concept of effective dimension to reduce the dimension of a sequence. Generally speaking, the core concept of the effective dimension is to screen the redundant dimension by satisfying either superposition sense or truncation sense. The definition of the superposition sense and truncation sense will be given below. The Principle Component Analysis (PCA) and Brownian Bridge (BB) method are widely used [11, 15] for effective dimension. The main purpose of these two methods is to put the weight of variance into the first few dimensions because the good 2D projection for lower dimensions in quasi-Monte Carlo. In our research work, instead of using PCA or BB, the global sensitivity analysis is employed for effective dimension. In summary, for the effectiveness of quasi-Monte Carlo for high dimensional problem, the key step is to identify the weight of variables based on variance. Although there are various methods to find the sensitivity indices, a good candidate choice so far for the type of research problem dealt in this thesis is the Variance Based Sensitivity Analysis. The reasons are: (1) Often the detail of the complex simulation model is confidential, and an explicit analytic relationship is not precisely known, or is closed. The implication is that the model has to be considered as black box. We can only access the output data of the simulation. It is impossible to study the sensitivity based on analytic expression level. (2) The core concept of effective dimension needs the information of the variance and correlation for the variables based on the ANOVA(Analysis of Variance) decompensation method. The detail of the sensitivity analysis will be given in next chapter. (3) The computation is affordable for a problem in moderate size of dimension.



(A) The projection of dimension 1 vs. 2 (B) The projection of dimension 28 vs. 30

FIGURE 2.3: The 2D project of Faure sequence



(A) The projection of dimension 1 vs. 2 (B) The projection of dimension 29 vs. 30

FIGURE 2.4: The 2D project of Halton sequence

2.2.4 Effective Dimension

The application of ANOVA decomposition for effective dimension was introduced by Caflisch [38] in first place. The ANOVA is an approach to decompose a function $f(.)$ into a sum of less complex functions. Hence, the function f is expressed as:

$$f(x) = f_0 + \sum_{i=1}^{s} f_i(x_i) + \sum_{i<j}^{s} f_{ij}(x_i, x_j) + \cdots + f_{12\ldots s}(x_1, x_2, ..., x_s) \tag{2.30}$$

In the function above, the components $f_i(x_i)$ are called the first order terms, and the components $f_{ij}(x_i, x_j)$ are called the second order terms, and so on. Suppose the s-dimensional unit cube is defined as $\mathbb{I}^s$, the total variance of the function $f(.)$ is defined as:

$$\sigma^2 = \sum_{n=1}^{s} \sum_{i_1<\ldots i_n}^{s} \sigma^2_{i_1..i_n} \tag{2.31}$$

in which $\sigma^2_{i_1..i_n} = \int_{\mathbb{I}^s} f^2_{i_1..i_n}(x_{i_1}, ..., x_{i_n})\mathrm{d}x_{i_1}, ..., \mathrm{d}x_{i_n}$.

For a subset $y = (x_{i1}, ..., x_{i_m})$, $1 \leq i_1 < \cdots < i_m \leq s$, the variance for the subset y is defined as

$$\sigma^2_y = \sum_{n=1}^{m} \sum_{i_1 < ... i_n}^{m} \sigma^2_{i_1..i_n} \tag{2.32}$$

Thus, the definitions of the effective dimension are:

Definition 1. The effective dimension of $f(.)$ in the superposition sense is the smallest integer d_S, such that

$$\sum_{0<|y|<d_S} \sigma^2_y \geq p\sigma^2 \tag{2.33}$$

where p is the proportion as $0 < p < 1$.

Definition 2. The effective dimension of $f(.)$ in truncation sense is the smallest integer d_T such that

$$\sum_{0<|y|\subseteq 1,2,...,d_T} \sigma^2_y \geq p\sigma^2 \tag{2.34}$$

From the definition of ANOVA decomposition and the definition of superposition sense and truncation sense (variance aspects), it is easy to understand that the idea of the effective dimension is directly related to the sensitivity analysis. To determine the effective dimension, we can thus apply the global sensitivity analysis, which is discussed further in detail in the next chapter. In definition 1, the d_S is always expected to be a small value so that the function $f(.)$ is dominated by the lower order term in ANOVA decomposition. Normally the superposition sense is quite useful when all the variables are equally important. In definition 2, the d_T indicates the weight of variables. In other words, these variables should be most important and dominate the function f. The effective dimension is dependent on p. To obtain a good quality of d_S and d_T, normally the p is expected to be greater than 0.95. For the same p, we always have $d_S \leq d_T$.

2.3 Differential Evolution and Quasi-Monte Carlo

The optimization based worst case validation for robustness analysis is widely used in aerospace engineering, such as [20, 23, 24, 39]. Especially the author in [20] demonstrated that the heuristic based optimization, such as that proposed in [24] has a better

performance in worst-case validation compared to the conventional Monte Carlo method for analyzing the performance of VEGA launch vehicle in the P80 ascending phase. Although the uncertainties and Simulink block considered in [20] are different from our simulation experiment, some mutual objectives are employed for both researches, such as $\max(Q_\alpha(mach))$. The term Q_α is the aerodynamic load performance of the launch vehicle, which is of great importance and of interest to the industry. Thus, it has become standard among the community and a consensus is being arrived that the optimization based robustness analysis is a good strategy in revealing the extreme behaviours of the system typically when the computational budget is limited.

Here, the interest is to demonstrate that the quasi-Monte Carlo based initialization for the population based method considered in [20, 24] significantly improve the convergence of the worst-case validation. Research in [40, 41] shows that an initial population with a better distribution in the case of Genetic Algorithm(GA) help convergence to global solution. Thus, one of our goals in this section is to demonstrate that the quasi-Monte Carlo has good potential to provide a good distribution of the initial population in Differential Evolution(DE). Like GA, DE is also a heuristic based optimization. The basic DE and GA have same or similar strategy of population initialization (normally based on Monte Carlo method with uniform distribution for initialization) and selection, but different in crossover and mutation operators. These concepts 'initialization, selection, crossover and mutation' for DE will be discussed in next subsection. The simple idea here is to make use of the quasi-Monte carlo based initialization of population for the DE algorithm. Only Sobol sequence will be studied in the experiment.

2.3.1 Differential Evolution

DE is a population based global optimisation algorithm having roots on evolutionary optimisation principle and introduced by Storn and Price [22]. Different scheme are available for DE. In this study,the *DE/rand/1/bin* strategy is used, where the '*DE*' denotes Differential Evolution, '*rand*' denotes the way of choosing the individuals for mutation is random, '*1*' denotes the number of pairs of individuals that are selected for mutation and '*bin*' denotes the way of recombination of the pairs of individual is binomial. DE consists of 4 steps based on this scheme:(1)Initialization of population,(2)Mutation, (3) Crossover, (4)Evaluation and selection.

(1)Initialization of population. Suppose the size of population is n, the population set is denoted as $x = [x_1, x_2, ..., x_n]$. For each individual $x_i, i = 1, 2, ..., n$ in the set x, it consist of m-dimensional uncertain parameters. A individual is expressed as $x_i = [x_{i1}, x_{i2}, ..., x_{ij}, ..., x_{im}], j = 1, 2, ..., m$. A conventional DE initialize a certain number of population by following the principle below:

$$x_{ij} = x_{ij}^{min} + (x_{ij}^{max} - x_{ij}^{min}) \times r, \quad i = 1, ..., n \; j = 1, ..., m$$

where x_{ij} denotes the uncertain parameter value in j^{th} dimension of the i^{th} individual. Furthermore, x_{ij}^{max} and x_{ij}^{min} represents the maximum and minimum bound of the uncertain parameters, which typically scaled appropriately to a unit interval. The term r is a 'random' perturbed variable typically uniform distributed within the interval $[0, 1]$. The random perturbed value r in crude DE is generated by Monte Carlo method with uniform distribution.

(2)Mutation To generate the mutated individual for next potential generation, three different individuals are selected from current generation.The potential mutated individual is generated by followed principle:

$$\bar{x}^{G+1} = x_{r1}^{G} + F(x_{r2}^{G} - x_{r3}^{G}) \tag{2.35}$$

where $\bar{x}^{G+1}$ is the mutated individual for crossover operation, $x_{r1}^{G}, x_{r2}^{G}, x_{r3}^{G}$ is the three different individuals randomly picked-up from current generation, F is a scaler factor within the range $[0, 1]$.The term $x_{r2}^{G} - x_{r3}^{G}$ indicates the search direction from point x_{r1}^{G} and the F determines the step size in that direction.

(3)Crossover A crossover operator determines whether an element in the new mutated individual will be remained as a prospective element for the next generation, or not. The definition is followed:

$$p_{ij}^{G+1} = \begin{cases} x_{ij}^{G} & \text{if } \rho_{rand} > \rho_c \\ \bar{x}_{ij}^{G+1} & \text{otherwise} \end{cases}$$

where p_{ij}^{G+1} is the prospective value of the j^{th} element for the i^{th} individual, ρ_{rand} is a uniform random variable and the ρ_c is crossover probability, $i = 1, 2, ..., n$ and $j = 1, 2, ..., m$. If a random variable ρ is greater than the crossover probability, the

value of that element will be kept as current value; otherwise, it will be replaced by the mutated element.

(4) Evaluation and selection After crossover operation, the prospective individual x_i^{G+1} will be evaluation by a objective function. The evaluation result is named as fitness. If the prospective individual x_i^{G+1} has a better fitness value than its parent's (x_i^G) fitness value, this individual will be selected for the next generation as a new parent. Otherwise, the current parent individual x_I^G will be chosen and marked as x_i^{G+1}.

$$x_i^{G+1} = \begin{cases} p_i^{G+1} & \text{if } f(p_i^{G+1}) \leq f(x_i^G) \\ x_i^G & \text{otherwise} \end{cases}$$

The pseudo code of the DE is given in table below.

	Pseudocode of DE
1	Begin
2	Initialize the population by Monte Carlo method with uniform distribution $x_i^G, G = 1, i = 1, 2, ..., n$
3	Evaluate the individual of the population of initialization, $f(x_i)$
4	Obtain the best individual pop_{best}
5	while(termination criteria satisfy)
6	for $i = 1$ to N
7	randomly select x_{r1}, x_{r2}, x_{r3},
8	$\bar{x}_i^{G+1} = x_{r1}^G + F(x_{r2}^G - x_{r3}^G)$
9	for $j = 1$ to m
10	if $\rho_{rand} > \rho_c$ Then
11	$p_{i,j}^{G+1} = x_{i,j}^G$
12	else
13	$p_{i,j}^{G+1} = \bar{x}_{i,j}^{G+1}$
14	End if
15	End for
16	If $(f(p_i^{G+1} \leq f(x_i^G)))$ Then
17	$x_i^{G+1} = p_i^{G+1}$
17	else
18	$x_i^{G+1} = x_i^G$
19	End if
20	End for
21	G = G+1
22	End while
23	End

2.3.2 Differential Evolution with Sobol initialization

Though it has been recommended to have an initial population of 1.5 to 2 times the dimension of the uncertain space, it has been shown in the previous studies that even a lower number of initial population can also perform well with *DE/rand/1/bin* schemes in [20, 24, 25]. For example, the number of initial population, for the 118-dimensional problem we have, is 50 (selected based on . It has been concluded that a better initial population distribution will potentially improve the convergence with a constrained computational cost [42]. Following the convergence plot of f_1 in Figure 4.1, the Sobol sequence is chosen as a candidate sampling method to generate the initial population for DE.

DE with Sobol initialization is essentially the same as crude DE, except the way of generating the initial individuals (i.e., the step 2 of pseudo code of the DE; This would become: 'Initialize the population by Sobol sequence: $x_i^G, G = 1, i = 1, 2, ..., n$'). The random properties occurred in the 'Mutation' and 'Crossover' parts still follow the uniform random distribution strategy. Instead in the case of Sobol initialization DE, the random perturbed value r in equation 2.3.1 is generated by Sobol sequence rather Monte Carlo method. The initialized individuals are distributed in a quasi-random manner in the uncertainty space. Because the Sobol sequence potentially has a better uniformity of the distribution, and a good coverage across the higher dimensional space as discussed earlier, DE should have higher opportunity to converge to global optimum value at a faster pace. In this case, it was observed that a lower value of F in 2.35 yields better results since the sobol sequence covers the space in a better manner.

2.3.3 Hybrid Differential Evolution

The Hybrid Differential Evolution (HDE), proposed in [20, 24] is considered as an extension version of DE. It combines the global optimization algorithm DE and a local optimization algorithm. A trigger is employed to switch the optimization algorithm between the DE and local optimization algorithm. The trigger is activated from DE to local optimization when the current generation in DE has no or very small(the improved fitness is smaller than the user assigned threshold of improvement) improvement in fitness. Then

the local optimization starts to explore the neighborhood of a randomly selected individual from the current population set. If a better optimum value is found, the selected individual will be replaced by the local optimum solution, otherwise, this individual will be remained. Then, the trigger is deactivated and the DE continue the optimization for next generation. In our experiment, the Matlab optimization function *'fmincon'* is applied to the local optimization.The *fmincon* is the gradient based optimization. It uses the Hessian Matrix for the objective function based on the Karush-Kuhn-Tucker(KKT) conditions. The KKT conditions uses the auxiliary Language multiplier. The method of Language multiplier for optimization is expressed as:

$$L(x, \lambda) = f(x) + \sum \lambda_{g,i} g_i(x) + \sum \lambda_{h,i} h_i(x) \tag{2.36}$$

where the $f(x)$ is the objective function of x, $g_i(x)$ is the i^{th} in linear inequality function, h_i is the i^{th} linear equality function, $\lambda_{g,i}$ and $\lambda_{h,i}$ are the Lagrange multiplier for the i^{th} function $g(x)$ and $h(x)$ respectively.

The KKT conditions means there exists a set of λ that satisfy the followed conditions:

$$\begin{aligned} \Delta_x L(x, \lambda) &= 0 \\ \lambda_{g,i} g_i(x) &= 0 \end{aligned} \tag{2.37}$$

So the local optimization is given as:

$$\Delta^2_{xx} L(x, \lambda) = \Delta^2 f(x) + \sum \lambda_{i,g} \Delta^2 g_i(x) + \sum \lambda_{i,h} \Delta^2 h_i(x) \tag{2.38}$$

For a non-differentiable objective function, such as the simulator of the VEGA launch vehicle, the *fmincon* at least require the number $2D + 1$ of function evaluations to construct the gradient.Here the D is the dimension of the uncertainty. In our case, the $D = 118$ and the minimum number of function evaluations for *fmincon* is 237. The *fmincon* will stop the local optimization when the number of evaluations reaches the assigned value or the improvement over each gradient is smaller than threshold 10^{-6}. The threshold value can be re-defined by user as well. And typically due to this fact, in the case of HDE, at the early stages of optimisation DE, which has potential to escape from the local peaks at the initial stages like GA, is preferred. Moroever, since the dimension of the problem is high, and the 'fmincon' optimisation has to evaluate gradient information numerically yielding to additional computational load, it is preferred to be

selected only at the final one stage to provide the final improvement towards the global solution. Instead of 'fmincon', it is possible to use other non-gradient local optimisation methods. However, in the present study we are not aiming neither in this direction, nor improving the conventional HDE with sobol initialization strategy for the same reason. We consider applying HDE for comparison purpose alone for a specified fixed computational budget (typically set to 1000 simulations) as previous studies have considered the methodology. The pseudo-code of the HDE is given below.

	Pseudocode of HDE
1	Begin
2	Initialize the population by Monte Carlo method with uniform distribution $x_i^G, G = 1, i = 1, 2, ..., n$
3	Evaluate the individual of the population of initialization, $f(x_i)$
4	Obtain the best individual pop_{best}
5	while(termination criteria satisfy)
6	If (improvement of fitness satisfies the requirement)
7	continue DE
7	Else
8	(1) Randomly select a individual in the current set of population, say x_0
9	(2) Apply the local optimization with the start point x_0
10	(3) If (any improvement in fitness)
11	replace x_0 by the local optimum solution
12	Else
13	keep x_0
14	End while
15	End

2.4 Gaussian Process

The Gaussian process is a method of estimating a Gaussian probability distribution to predict the objective values under a corresponding set of mean functions, covariance functions, and the inferences on the functions. The Gaussian process is introduced from a function-space point of view. However, to better understand the principle of the Gaussian process, we first introduce an equivalent point of view, the weight-space view [19]. The concept of the Gaussian process is easier to understand from the weight-space point of view than from the function-space point of view. However, the algorithms mentioned in the function-space view are employed in practice.

2.4.1 Weight-space review

In the beginning, we consider a linear model in the view of Bayesian analysis to present a real problem. The model is expressed as:

$$f(x) = x^T w, \quad y = f(x) + \varepsilon \tag{2.39}$$

where x is the input variables, w is the weight of each inputs, f is the value of the linear model, y is the observation of the model with the noise involved and ε is the Gaussian noise. Normally the Gaussian noise is a Gaussian distribution with zero mean and variance σ_n^2:

$$\varepsilon = N(0, \sigma_n^2) \tag{2.40}$$

where N denotes the normal distribution. Thus, the observation value is different from the function value f. To describe the features of the distribution of the linear model associated with the noise, the probability density function for the observations is given in the equation 2.41 below.

$$\begin{aligned} p(y|X, w) &= \prod_{i=1}^{n} p(y_i|x_i, w) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma_n} \exp(-\frac{(y_i - x_i^T w)^2}{2\sigma_n^2}) \\ &= \frac{1}{(2\pi\sigma_n^2)^{n/2}} \exp(-\frac{1}{2\sigma_n^2}|y - X^T w|^2) \\ &= N(X^T w, \sigma_n^2 I) \end{aligned} \tag{2.41}$$

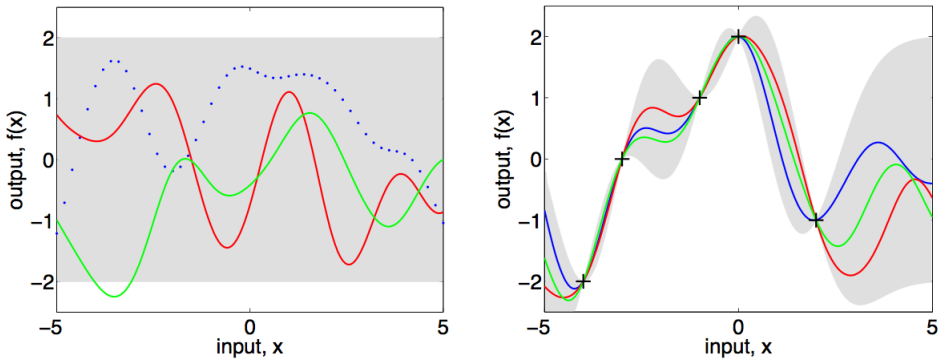
where the X is input matrix as $X = \{x_1, x_2, ..., x_i, ...x_n\}$, the sign '| |' denotes the Euclidean length of the vector.

This probability density is also called the *likelihood* of the Bayesian linear model. The weight w in the equation is also called *prior* in the Bayesian linear model. The prior means the inference about the parameters based on our knowledge of the model before we investigate the inputs and observations. The detail of the properties of the prior will be discussed in the function-space view later. Normally, the weight w is chosen as a Gaussian prior with zero mean and covariance matrix Σ_p.

$$w \sim N(0, \Sigma_p) \tag{2.42}$$

The posterior is the distribution of the weight w based on prior and the investigation of observation. It is provided for the inference of the Bayesian linear model. The Figure

2.5 presents the difference of the probability distribution based on the concepts of 'prior' and 'posterior'. The posterior is obtained by following the Bayes' rule:



(a), prior (b), posterior

FIGURE 2.5: The prior and posterior

$$p(w|y, X) = \frac{p(y|X, w)p(w)}{p(y|X)} \tag{2.43}$$

where the $p(w)$ is the probability density of the prior in the equation 2.42, the $p(y|X)$ is called marginal likelihood and is independent to the prior. Note that the Bayes' rule is only on one condition and has the form as: $p(a|b) = p(b|a)p(a)/p(b)$. Here, we just introduce an additional condition X to all of the terms into the Bayes' rule. However, as the prior has no information with the inputs, the condition X is not given in the prior term.

The marginal likelihood in the equation 2.43 is defined as:

$$p(y|X) = \int p(y|X, w)p(w)\mathrm{d}w \tag{2.44}$$

From this equation, we can see the $p(y|X)$ actually is a constant. So the posterior of the equation 2.43 can be described as:

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}} \tag{2.45}$$

If we only considered the terms are depended on the weight w, the term *marginal likelihood* is neglected. The posterior is then proportional to the multiplication of the

terms *prior* and *likelihood*:

$$\begin{aligned} p(w|X,y) &\propto \exp(-\frac{1}{2\sigma_n^2}(y-X^Tw)^T(y-X^Tw))\exp(-\frac{1}{2}w^T\Sigma_p^{-1}w) \\ &\propto \exp(-\frac{1}{2}(w-\bar{w})^T(\frac{1}{\sigma_n^2XX^T+\Sigma_p^{-1}})(w-\bar{w})) \end{aligned} \tag{2.46}$$

where the $\bar{w} = \sigma_n^{-2}(\sigma_n^{-2}XX^T+\Sigma_p^{-1})^{-1}Xy$. From the equation 2.46, the typical Gaussian distribution format can be identified. So the posterior distribution of the weight with mean $\bar{w}$ and variance $(\sigma_n^{-2}XX^T+\Sigma_p^{-1})^{-1}$ can be written as:

$$p(w|X,y) \sim N(\bar{w}, (\sigma_n^{-2}XX^T+\Sigma_p^{-1})^{-1}) \tag{2.47}$$

As the linear model is based on the Bayesian setting, the prediction of the outputs at the a given point x_* is calculated by averaging the outputs of all linear functions with respect to the posterior:

$$\begin{aligned} p(f_*|x_*,X,y) &= \int p(f_*|x_*,w)p(w|X,y)\mathrm{d}w \\ &= N(\sigma_n^{-2}x_*^TA^{-1}Xy, x_*^TA^{-1}x_*) \end{aligned} \tag{2.48}$$

where the $A = \sigma_n^{-2}XX^T+\Sigma_p^{-1}$. The prediction is also a Gaussian distribution.If the linear model is not a Bayesian setting, normally only one parameter of the weight in posterior is chosen by some criteria rather than all of the parameters are involved in the prediction.

In practice, the input matrix X is not directly assigned into the linear model. Instead, a set of polynomial functions are employed to project the inputs into a higher or equal dimension space. The reason is that the projection would be able to increase the number of expressions for the Bayesian linear model. As long as the polynomial functions are independent with the weight w, the Bayesian model is still linear for the w. Based on this replacement, the linear model is then expressed as:

$$f(x) = \phi(x)^Tw \tag{2.49}$$

where the $\phi(x)$ is the polynomial functions. Thus, the predictive distribution in equation 2.48 is rewritten as:

$$p(f_*|x_*,X,y) = N(\frac{1}{\sigma_n^2}\phi(x_*)^TA^{-1}\phi(X)y, \phi(x_*)^TA^{-1}\phi(x_*)) \tag{2.50}$$

2.4.2 Function-space view

Gaussian process is composed by the mean function and covariance function. We defined the mean and covariance functions for a process $f(x)$ by following expression:

$$\begin{aligned} m(x) &= E[f(x)] \\ k(x, x^{'}) &= E[(f(x) - m(x))(f(x^{'}) - m(x^{'}))] \end{aligned} \tag{2.51}$$

where $m(x)$ is the mean function, $k(x, x^{'})$ is the covariance function, x and $x^{'}$ is either the training data or test data.
Then the Gaussian process is normally written as the format below:

$$f(x) \sim GP(m(x), k(x, x^{'})) \tag{2.52}$$

To better understand the Gaussian process, a simple example is introduced. For the sample Bayesian linear model mentioned in the weight-space view, the $f(x) = \phi(x)^T w$. The prior w is also same and defined as $w \sim (0, \Sigma_p)$.

$$\begin{aligned} &E(f(x)) = \phi(x)^T E[w] = 0; \\ &E[f(x)f(x^{'})] = \phi(x)^T E[ww^T]\phi(x^{'}) = \phi(x)^T \Sigma_p \phi(x^{'}) \end{aligned} \tag{2.53}$$

As the training and test date are both considered in the covariance function, this probability distribution is the joint Gaussian distribution with zero mean and covariance $\phi(x)^T \Sigma_p \phi(x^{'})$. From this example, we made the inference on the mean function as the zero mean. In practice, we need to make the inference on the covariance function as well. The reason is that it is not easy to compute the covariance matrix based on the definition. Thus, a substituted polynomial function is applied to calculate the covariance for each pair of the training and test date. For example, a very common used covariance function is called 'squared exponential' (SE) and defined as:

$$cov(f(x), f(x^{'})) = exp(-\frac{1}{2}|x - x^{'}|^2) \tag{2.54}$$

There are many other covariance functions available as well. The inference for the covariance is also very important to the Gaussian process.
To predict the distribution, we firstly consider a case without any noise in the observation.The noise-free observation means the function value equals to the observation from

the equation 2.39. In practice, we are not interested to put too much effort to draw the random functions $\phi(x)$ from the prior. Thus, the joint prior is introduced according to the training and test data and is expressed as:

$$\begin{bmatrix} f \\ f_* \end{bmatrix} \sim N\left(0, \begin{bmatrix} K(X,X) & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right) \tag{2.55}$$

where the X includes n training points and X_* has n_* test points. $K(,)$ denotes the covariance function. For the joint posterior distribution, those random functions who are not agree with the observations are abandoned. Then the joint Gaussian distribution for prediction then is given as:

$$\begin{aligned} f_*|X_*, X, f \sim N(&K(X_*,X)K(X,X)^{-1}f, \\ &K(X_*,X_*) - K(X_*,X)K(X,X)^{-1}K(X,X_*)) \end{aligned} \tag{2.56}$$

The details to drive the equation 2.55 and 2.56 can be found in [19]. When the noise is involved in the observations, we are unable to directly access the function values. Suppose the noise is also a Gaussian distribution with zero mean and variance σ_n^2. Then, the variance of the prior becomes:

$$cov(y_p, y_q) = k(x_p, x_q) + \sigma_n^2 \delta_{pq} \tag{2.57}$$

where the δ_{pq} is a Kronecker delta that if $p = q$, $\delta_{pq} = 1$; otherwise, $\delta_{pq} = 0$. The reason is the noise is independent for each input point. So the noise matrix is diagonal. By following this principle, the joint prior is rewritten as:

$$\begin{bmatrix} f \\ f_* \end{bmatrix} \sim N\left(0, \begin{bmatrix} K(X,X) + \sigma_n^2 I & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right) \tag{2.58}$$

Then the predicted joint Gaussian prediction with noise is driven to:

$$f_*|X, y, X_* \sim N(\bar{f}_*, cov(f_*)) \tag{2.59}$$

where

$$\begin{aligned} \bar{f}_* &= E[f_*|X, y, X_*] = K(X_*,X)[K(X,X) + \sigma_n^2 I]^{-1} y \\ cov(f_*) &= K(X_*,X_*) - K(X_*,X)[K(X,X) + \sigma_n^2 I]^{-1} K(X,X_*) \end{aligned} \tag{2.60}$$

2.4.3 Hyperparameters

In Gaussian process, there are some parameters that need to be decided by user. These parameters are called free parameters or hyperparameters. The variance Σ_p^2 in the prior and the variance σ_n^2 in the noise are two typical hyperparameters. Normally, the covariance functions have some other hyperparameters. For example, the squared-exponential covariance function is defined as:

$$k(x_p, x_q) = \sigma_f^2 exp(-\frac{1}{2\ell^2}(x_p - x_q)^2) \tag{2.61}$$

where the σ_f^2 is the signal variance and the ℓ is considered as a scaler. Both of them are hyperparameters. The hyperparameters highly affect the predicted joint Gaussian distribution. There are two widely used method to select the proper value of the hyperparameters: Bayesian model section and cross-validation. The details about the two methods can be found in [19]. In this thesis, we are not going to introduce it in detail, but will briefly mention the method that is implemented in the Gaussian process for our experiment.

2.5 Variance based Sensitivity Analysis

2.5.1 Overview

Sensitivity analysis (SA) is a useful approach with which to study the mapping relations between the input uncertainties and the output response for a complex system. SA is also related to robustness and reliability, and is able to estimate whether or not the presence of one uncertainty or a set of uncertainties could affect the stability of a system. SA also can screen out the uncertainties with very little effect on robustness and reliability. Normally, the sensitivity analysis is coupled with uncertainty analysis. However, due to the confidential nature of protocols regarding the VEGA launch vehicle, limited information is available regarding the uncertainties of the vehicle, such as the scaled value, the probabilistic density function of each uncertainty, etc. Thus, the uncertainty analysis in our research is somewhat constrained. Uncertainty analysis studies the distribution of the input of the uncertainties, while sensitivity analysis studies the contribution of each uncertainty in a system according to the input uncertainties. In this research, all

the uncertainties are epistemic. Although we have little information about the uncertainties, we understand the nominal upper and lower bounds of the uncertainties. The nominal values of uncertainties assigned to the system are automatically transferred into the scaled value by the simulator. Moreover, there are many sensitivity and uncertainty analysis methods [43], such as differential analysis, response surface analysis, and Monte Carlo methods, among others.

Normally, the results of sensitivity analysis could be any of the following:
(1) All of the uncertain parameters make a great contribution to the output. In this situation, more investigation is required to enhance the knowledge of the model.
(2) None of the uncertain parameters are sensitive to output response. In this situation, the range of the uncertain parameters may not have been chosen properly, or the output depended on some other factors that were not considered as uncertainties, such as the existence of wind for dynamic loads.
(3) A group of uncertainties do not influence the output response. These uncertainties make few contributions to the output and can be removed from the simulation. This is the desired situation.

The variance decompensation method is one approach by which to study the contribution of each uncertainty of the sampling-based methods (Monte Carlo or quasi-Monte Carlo). It is the fundamental concept of variance-based sensitivity analysis. Variance-based sensitivity is a global sensitivity analysis method. There are many other approaches for sensitivity analysis in sampling-based methods, such as nonparametric regression, squared rank differences/rank correlation coefficient, and the two-dimensional Kolomogorov-Smirnov test among them. More details about these sampling-based methods can be found in [44]. In our research, we only considered variance-based analysis because it is effective, reliable, and closely related to the concept of effective dimensions mentioned in previous chapter, which make it an ideal method by which to improve the effectiveness of the quasi-Monte Carlo method. However, its drawback is that its high computational cost is inflexible when the dimension of the uncertainties is high. To overcome this drawback, we propose using the Gaussian process to train a meta-model to ease the computational burden.
To carry out a variance-based sensitivity analysis, the five basic steps listed below must be followed, based on the steps for performing a sampling-based sensitivity analysis:

(1) Determination of uncertainty distribution.

(2) Generation of sampling points.

(3) Simulation in terms of the sampling points.

(4) Presentation of uncertainty analysis results.

(5) Presentation of sensitivity analysis results

Determination of uncertainty distribution.

The manner of distribution of the uncertainties are considered the most important part in sampling-based sensitivity analysis as it significantly influences the results of sensitivity analysis. Normally the distribution is ideally determined according to an expert review. A preliminary analysis of the uncertainties is recommended to help to define some parameters of the distribution, such as mean value, range, and the quantiles of the corresponding cumulative distribution functions [44]. Thus, the limited computation source could have a better use. In our experiment, the experts on the VEGA launch vehicle model defined the range of the uncertainties, but there is no other information about the uncertainties.

Generation of sampling points.

There are three commonly used strategies to generate the sampling points: random sampling, importance sampling, and Latin hypercube sampling[43–45].Random sampling, introduced in the previous chapter, is easy to implement . However, its drawback is that there is no guarantee that all given sub-regions will be covered unless the sampling size is large enough. Importance sampling exhaustively divides the uncertainty space into many sub-regions without any overlap. In each sub-region, only one value is sampled (has only one sampled point). Importance sampling is used to analyze whether a sampled value has high consequence but low probability. Latin hypercube sampling is based on the idea of importance sampling in that it exhaustively divides the space into many sub-regions, but unlike importance sampling, it does not ensure that there is only one point in each sub-region. More details about generating the sampling points via importance sampling and Latin hypercube sampling can be found in [43]. In our research, we proposed an alternative sampling method: quasi-random sampling, which was introduced in the previous chapter. Because quasi-random sampling has a high uniformity distribution, it is an ideal method with which to obtain good coverage of the uncertainty space. More specifically, we user a Sobol sequence to generate the sampling points. If the uncertainties are not independent, the control of correlation in sampling

is very important. However, in Chapter 2 we have already mentioned that all the uncertainties are independent. Thus, there is no consideration of the correlation issues in our research.

Simulation in terms of the sampling points.

All of the individual points generated by the Sobol sequence are assigned into the VEGA CONTROL simulator, and all of the relevant outputs are recorded for sensitivity analysis.

Presentation of uncertainty analysis results.

In this step, the purpose is to simply display the results of mapping between the input uncertainties and the output of the analysis $[x_i, y(x_i)]$, $i = 1, 2, .., n$. The displayed information normally contains the cumulative distribution function (CDF), mean value, standard deviation, etc.

Presentation of sensitivity analysis results

The presentation of sensitivity analysis results requires more display sophistication since it requires mapping the relation between the input uncertainties and output results. The various approaches to sensitivity analysis have been mentioned in previous chapters, and more details can be found in [44]. For the purposes of our research, we are only interested in the scatter plots and variance decompensation.

2.5.2 Methodology

Variance-based sensitivity analysis is not a new technique for indicating the importance of an input factor with respect to the output response of interest of a model. In our research, we only focus on the Sobol method (or Sobol indices) to study the sensitivity indices. The Sobol method is one of the most commonly used methods in variance-based sensitivity analysis[46], and has been widely studied by many researchers [47–49]. In the aerospace industry, analytical sensitivity analysis is widely used in studying the importance of design parameters, as in[50]. However, the use of variance-based sensitivity to study the importance of the uncertain parameters in the aerospace industry, especially for launch vehicles, is rare. Analytical sensitivity analysis often implements the differential method to study the parameters. The priority of applying the differential method for sensitivity analysis is that the model must be able to expressed by a polynomial function. The differential method is a easy, fast, and efficient way to study the importance of the factors. However, in our research, the VEGA launch vehicle simulator

is treated as a black box, since all of the design parameters of the launch vehicle are confidential. Moreover, our study is focused on the uncertain parameters rather than the design parameters. Thus, the analytical method is not applicable in our research. Therefore, variance-based sensitivity analysis is an ideal approach, since we only need to study the mapping relation between the input variables and the corresponding output.

The key idea of the sensitivity indices is the decompensation of the objective function $f(x)$ into a summation of many components associated with each input factor x. Then, according to the decompensation concept, the relation between the components in the decompensated function $f(x)$ and partial variances in the variance decompensation $V(Y)$ can be found. This relationship is used to identify the sensitive indices. The decompensation of the function $f(x)$ is expressed as

$$f(x) = f_0 + \sum_i f_i + \sum_i \sum_{j>i} f_{ij} + \cdots + f_{12\ldots n} \tag{2.62}$$

where the $x = [X_1, X_2, ..., X_i, ..., X_n]$, $f_i = f_i(X_i)$, $f_{ij} = f_{ij}(X_i, X_j)$. There are total 2^n components in the decompensation function.
The function of f_0,f_i,f_{ij} are defined as:

$$\begin{aligned} f_0 &= E(Y) \\ f_i &= E_{X_{\sim i}}(Y|X_i) - E(Y) \\ f_{ij} &= E_{X_{\sim ij}}(Y|X_i, X_j) - f_i - f_j - E(Y) \end{aligned} \tag{2.63}$$

where the X_i denotes the i^{th} input variable,the $X_{\sim i}$ denotes the matrix of all variables expect X_i, the f_0 is the expected value of output response Y(or objective value), $E_{X_{\sim i}}(Y|X_i)$ is the expected value of Y except the value of X_i is fixed, $E_{X_{\sim ij}}(Y|X_i, X_j)$ is the expected value of Y except the X_i and X_j are fixed. The higher order of the decomposition functions are defined based on the principle of equation 2.63.
The sensitivity indices relies on the variance information of the objective values under the different conditions of uncertainties. It is associated with the method called variance decompensation mentioned above. To find out the sensitivity, the variance decompensation is defined as:

$$V(Y) = \sum_i V_i + \sum_i \sum_{j>i} V_{ij} + \cdots + V_{1,2,\ldots,n} \tag{2.64}$$

where the terms, V_i and V_{ij} , in the variance decompensation $V(Y)$ are defined as:

$$\begin{aligned} V_i &= V(f_i(X_i)) = V_{X_i}[E_{X_{\sim i}}(Y|X_i)] \\ V_{ij} &= V(f_{ij}(X_i, X_j)) \\ &= V_{X_iX_j}(E_{X_{\sim ij}}(Y|X_i, X_j)) - V_{X_i}(E_{X_{\sim i}}(Y|X_i)) \\ &- V_{X_j}(E_{X_{\sim j}}(Y|X_j)) \end{aligned} \tag{2.65}$$

where $V_{X_i}(E_{X_{\sim i}}(Y|X_i))$ denotes the variance of the expected objective value when the variable X_i is fixed with all possible values for other variables, $V_{X_iX_j}(E_{X_{\sim ij}}(Y|X_i, X_j))$ is the variance of the expected objective value when then variable X_i and X_j are fixed with all possible values for other variables. Divid the variance $V(Y)$ for both sides of above equation 2.64, the sensitivity indices is expressed as:

$$\sum_i S_i + \sum_i \sum_{j>i} S_{ij} + \cdots + S_{1,2,\ldots,n} = 1 \tag{2.66}$$

where the S_i is first order sensitivity of a input variable x_i.

For sensitivity analysis based on the Sobol indices, the focus is on the first order sensitivity of a input variable and the total effect sensitivity index for that variable in practice. The reason is the application is simply and effectiveness.

So the first order sensitivity coefficient of a variable x_i is defined as:

$$S_i = \frac{V_{X_i}(E_{X_{\sim i}}(Y|X_i))}{V(Y)} \tag{2.67}$$

This equation presents the relation between the Sobol indices and variance decompensation. It also gives the reason why the function decompensation is the key idea of the sensitivity indices.

The total effect sensitivity index for a variable is defined as:

$$S_{Ti} = \frac{E_{X_{\sim i}}(V_{X_i}(Y|X_{\sim i}))}{V(Y)} = 1 - \frac{V_{X_{\sim i}}(E_{X_i}(Y|X_{\sim i}))}{V(Y)} \tag{2.68}$$

where the $E_{X_{\sim i}}(V_{X_i}(Y|X_{\sim i}))$ is the expected variance with all input factors are fixed except the X_i and the $V_{X_{\sim i}}(E_{X_i}(Y|X_{\sim i}))$ is the variance of expected objective values with all input factors are fixed except X_i. The total effect sensitivity index of a variable x_i means the entire contribution associated with the input factor X_i in all terms of the variance decompensation $V(Y)$. The total effective sensitivity index includes the first order

and higher order sensitivity effects. The higher order sensitivity effects are considered as the interaction effect between the x_i and other input factors. If the $S_{Ti} = S_i$, it means there is no interaction between x_i and other input factors. The term $V_{X_{\sim i}}(E_{X_i}(Y|X_{\sim i}))$ in above equation can be considered as the first order sensitivity effect of the $X_{\sim i}$ and means the effect without X_i involved.

According to definition of variance, the $V(Y)$ can be expressed as:

$$V(Y) = E(Y^2) - (E(Y))^2 \tag{2.69}$$

We extent this definition mentioned above to first order of the variance decompensation $V_{X_i}(E_{X_{\sim i}}(Y|X_i))$:

$$V_{X_i}(E_{X_{\sim i}}(Y|X_i)) = \int E^2_{X_{\sim i}}(Y|X_i)\mathrm{d}X_i - (\int E_{X_{\sim i}}(Y|X_i)\mathrm{d}X_i)^2 \tag{2.70}$$

In above equations 2.69 and 2.70, the second terms $(E(Y))^2$ and $(\int E_{X_{\sim i}}(Y|X_i))^2$ are both always equal to f_0^2. The first term can be expressed as:

$$\begin{aligned}\int E^2_{X_{\sim i}}(Y|X_i)\mathrm{d}X_i = &\int\int\int f(X_1, X_2, ..., X_i, ..., X_n) \\ &\times f(X'_1, X'_2, ..., X'_i, ..., X'_n)\mathrm{d}X_{\sim i}\mathrm{d}X'_{\sim i}\mathrm{d}X_i \\ = &\int\int f(X_1, X_2, ..., X_i, ..., X_n) \\ &\times f(X'_1, X'_2, ..., X'_i, ..., X'_n)\mathrm{d}X\mathrm{d}X'_{\sim i}\end{aligned} \tag{2.71}$$

where the X and X' are two independent matrices of all input variables but generated by same method.

From the equation 2.71, there are actually 2n-1 variables. In our case, the n is equal to 118. Thus, the total dimension of this function is 235. To solve the integral with 235 dimension by numerical method in practice is almost unrealistic. Based on Chapter 3 we know that the Monte Carlo and quasi-Monte Carlo are capable to approximate the integration problem of high dimension. But for such high dimension case, the heavy computational burden is inevitable if a high accuracy is needed. In practice, an alternative approximation is proposed to calculate the first order sensitivity effect and the total sensitivity index. Suppose we have different independent matrices A and B. For each element in the matrices, they are expressed as a_{ji} and b_{ji}. The i denotes i^{th} input

factor and is from 1 to n (n=118 in our case). The j denotes the j^{th} point (or individual or a condition of uncertainties) and is from 1 to N. N is the total number of simulations (or objective function evaluations). $A_B^{(i)}$ denotes a matrix with the elements that are all from the matrix A except the elements in i^{th} column are from matrix B. Similarly, the $B_A^{(i)}$ denotes a matrix with the elements that are all from the matrix B except the elements in i^{th} column are from matrix A. Thus, the approximated first order sensitivity effect is defined as:

$$V_{X_i}(E_{X_{\sim i}}(Y|X_i)) = \frac{1}{N}\sum_{j=1}^{N} f(A)_j f(B_A^{(i)})_j - f_0^2 \tag{2.72}$$

and the total sensitivity effect of X_i is defined as:

$$V_{X_{\sim i}}(E_{X_i}(Y|X_{\sim i})) = \frac{1}{N}\sum_{j=1}^{N} f(A)_j f(A_B^{(i)})_j - f_0^2 \tag{2.73}$$

The equation 2.72 and 2.73 above are considered as the estimator for sensitivity analysis. As both matrices A and B are independent, they can be generated separately with 118 dimensions.

The equation 2.72 is improved in [51, 52] and extent to the new estimator:

$$V_{X_i}(E_{X_{\sim i}}(Y|X_i)) = \frac{1}{N}\sum_{j=1}^{N} f(A)_j (f(B_A^{(i)})_j - f(B)_j) \tag{2.74}$$

The improvement for equation 2.73 is mentioned in [53] and the new estimator is expressed as:

$$V_{X_{\sim i}}(E_{X_i}(Y|X_{\sim i})) = V(Y) - \frac{1}{N}\sum_{j=1}^{N} f(A)_j (f(A)_j - f(A_B^{(i)})_j) \tag{2.75}$$

Chapter 3

VEGA Launcher model

3.1 Background

The VEGA launch vehicle is small to medium weigh payload launcher developed under the European Space Agency(ESA) program. The main contractor for the VEGA launcher is a Italian company called ELV S.p.A Company. The first launching of VEGA was 2012. The history of VEGA program is back to 1990's. The concept of VEGA was proposed by an Italian named DPD Difesa y Spazio in 1988 in order to replace the US Scout launcher that was already retired. The purpose of VEGA is to survey the possibility to complement the Arina launcher family with a small payload launch vehicle by using the solid booster. The VEGA launcher is mainly consist of 4 stages, including 3 solid propellant stages and one liquid propellant stage(also called Attitude and Vernier Upper Module or AVUM). In the first 3 solid stages, the P80FW, ZEFIRO 23 and ZEFIRO 9 Solid Rocket Motor (SRM)are applied respectively. The liquid stage employs the LPS engine instead. Actually in our research, we only interested in the first stage (P80FW stage or P80 stage). The Figure 3.1 is prototype of the VEGA launch vehicle [1].

3.1.1 Launch mission profiles

The typical VEGA launching mission is consist of three phases. **Phase 1** is an ascending phase that deliver the Launch Vehicle(LV) into the low elliptic orbit(sub-orbital

[1] http://www.esa.int/esapub/br/br257/br257.pdf

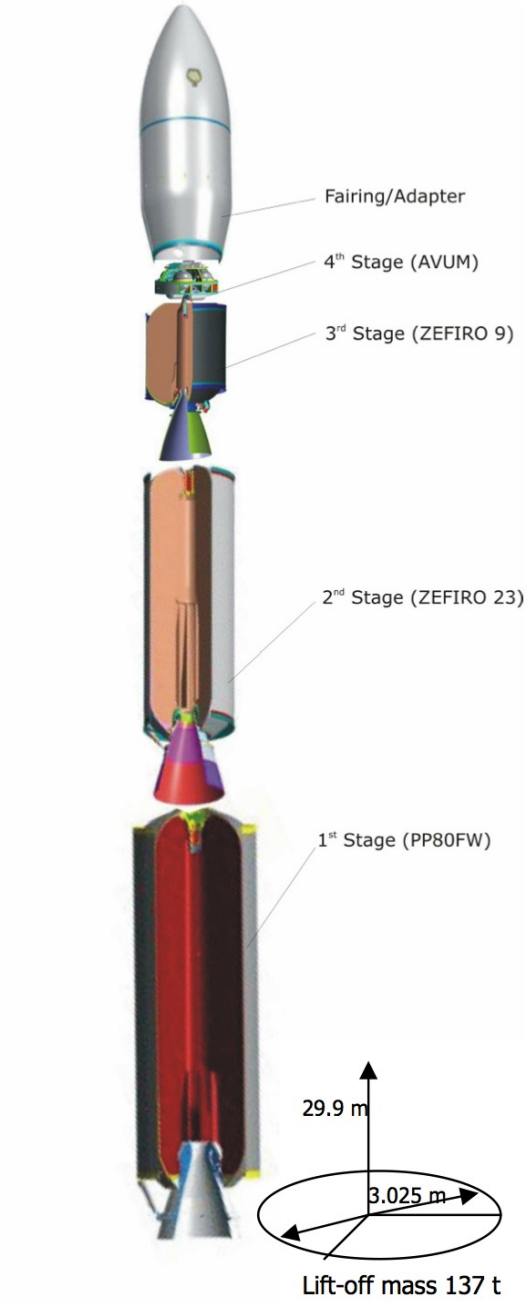


FIGURE 3.1: The VEGA luancher (Courtesy Url Link)

trajectory) by first three stages. The phase 1 includes three events: (i) vertical ascent with programmed pitch maneuver and a zero-incidence flight (for stage i); (ii) incident flight(for stage ii); (iii) fairing separation and payload is delivered into the sub-orbital trajectory(for stage iii). The ascending phase 1 with its associated distinct events is presented in Figure 3.2. At **Phase 2**, the payload and stage 4 are delivered to the intermediate orbit by AVUM after separation with stage 3. Then maneuvering by AVUM stage, the payload eventually is delivered to the final orbit. During **Phase 3**, AVUM is

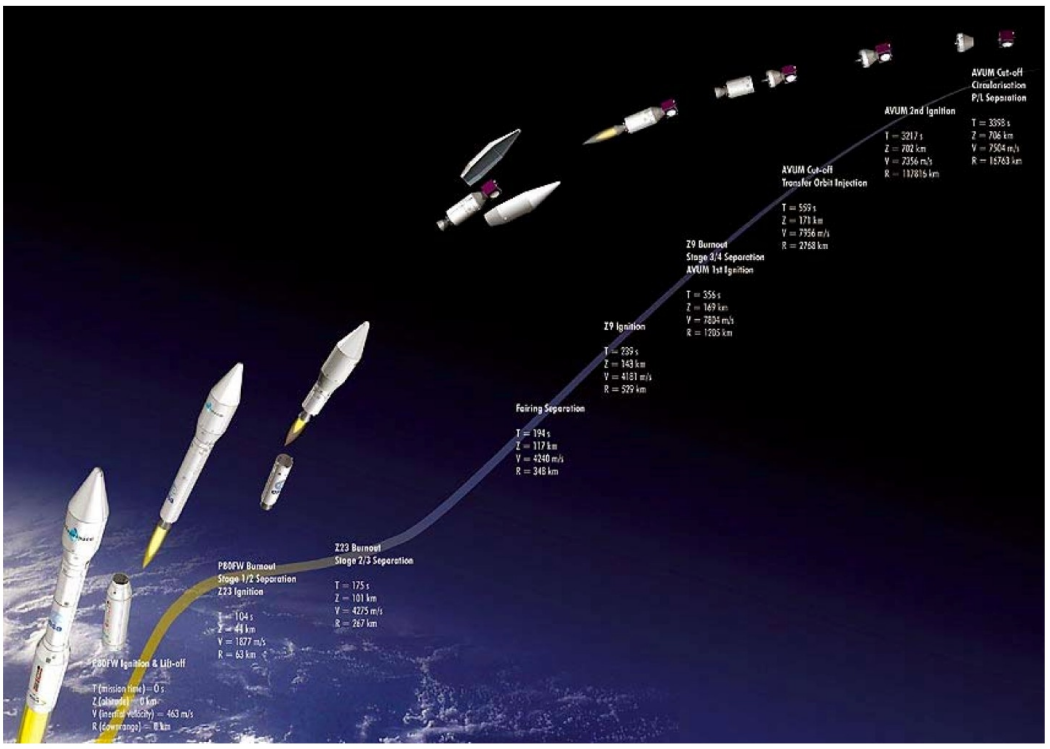


FIGURE 3.2: Sub-orbital trajectory with associated events in phase 1 (Courtesy Url Link)

separated with payload and the AVUM is maneuvered to a safe distance with spacecraft. Then the AVUM carries out the burn of the engine for orbit disposal maneuver. The AVUM re-entry into the earth's atmosphere.

3.2 VEGA launch vehicle trajectory and Dynamics

In this thesis,as the real dynamic structure, control parameters and design parameters of the VEGA launch vehicle are not given due to confidential purpose. We can not offer a general type of launch vehicle to serve a better understanding of the nature of uncertain parameters.The VEGA launch vehicle is a TVC based rocket. The very basic physical concept of ascending trajectory is introduced by the equations of the trajectory. The equations of motion(EOM) are applied to study the dynamics of the rigid body. The concept of EOM is also used for the bending modes of the VEGA launch vehicle. In the end, the TVC actuator is also introduced in this chapter.

3.2.1 Rigid Body

The P80 ascending phase has two sub-phases: the vertical ascent and the turn-over ascent trajectory [1]. The VEGA launcher firstly ascends vertically from the ground. When the launcher vehicle reaches the specified altitude (normally after go through the high air density regions of the atmosphere), the turn-over maneuver makes the launch vehicle carry out the turn-over motion. For vertical ascent phase, the EOM can expressed as:

$$m\frac{\mathrm{d}v}{\mathrm{d}t} = T - D - mg \tag{3.1}$$

where the T is the thrust of VEGA launcher, D is the aerodynamics drag, m is the mass of the VEGA launch vehicle, v is the vertical velocity and g is the local gravitational acceleration. All units in SI.

The thrust T in the equation 3.1 is given by:

$$T = -V_e\frac{\mathrm{d}m}{\mathrm{d}t} + (p_e - p_o)A_e \tag{3.2}$$

where the V_e is the exist velocity of the launch vehicle, p_e is the exit pressure of nozzle, p_o is the local pressure of atmosphere and A_e is the exit area of the nozzle. The negative sign '$-$' is given here because the rate of mass changing is negative. The propellant is losing over the ascending time. The term $-V_e\frac{\mathrm{d}m}{\mathrm{d}t}$ is called momentum thrust and the $(p_e - p_o)A_e$ term is called pressure thrust.

Normally, the thrust due to pressure is very small and neglected from the equation 3.2. In the meantime, the exit velocity V_e is given by:

$$V_e = g_o I_{sp} \tag{3.3}$$

where the g_o is the sea level gravitational acceleration and equals to $9.8m/s^2$, I_{sp} is called specific impulse.

Thus, by neglecting the pressure thrust term and replacing the V_e by the equation into the equation 3.2, the thrust of the launch vehicle is expressed as:

$$T = -V_e\frac{\mathrm{d}m}{\mathrm{d}t} = -g_o I_{sp}\frac{\mathrm{d}m}{\mathrm{d}t} \tag{3.4}$$

The aerodynamic drag D in the equation 3.1 is given by:

$$D = \frac{1}{2}\rho v^2 C_D S \tag{3.5}$$

where the ρ is the aerodynamic density, v is the velocity of the VEGA launch vehicle, C_D is the drag coefficient and S is the reference surface. The drag coefficient C_D changes according to the variation of the altitude and speed(or the mach number: the ratio of the launch vehicle velocity to the speed of sound). The term $\frac{1}{2}\rho v^2$ is called dynamic pressure and often is expressed as $q(t) = \frac{1}{2}\rho v^2$.

To approximate the equation 3.1, the drag term normally is neglected and the thrust is assumed to be the constant. Then the equation 3.1 can be re-written as:

$$m\frac{\mathrm{d}v}{\mathrm{d}t} = -g_o I_{sp}\frac{\mathrm{d}m}{\mathrm{d}t} - mg \tag{3.6}$$

In the turn-over phase, an angle exists between between the trajectory of the LV and the local horizontal axis. Thus, the force and torque that act on the launch vehicle (LV) need to be considered for the EOM during the flight in two dimensional motion. The Figure 3.3 given below provides the information about the force, torque and frames. The meaning of the notation presented in the Figure 3.3 is given below:

- α : angle of attack
- θ_b : pitch angle
- δ : thrust gimbal angle
- U : velocity vector
- W : wind velocity vector
- F_A : aerodynamic force acted on the point A
- T : thrust
- F_E : inertia force of LV nozzle
- M_E : inertia torque of LV nozzle
- m : mass of LV
- g : gravitational acceleration
- x, x_b : the x axis in body frame
- z, z_b : the z axis in body frame
- x_t : the x axis in trajectory frame
- z_t : the z axis in trajectory frame

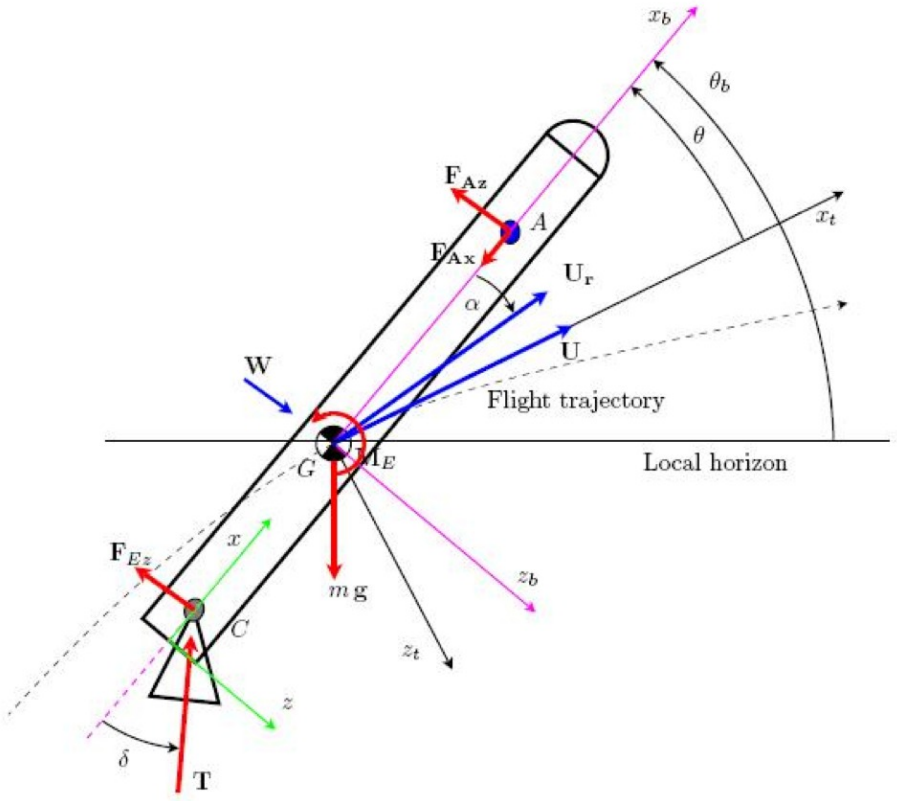


FIGURE 3.3: Two dimensional motion of the VEGA launcher [1]

Gravity The gravity force acted on the LV based on the body frame is expressed as:

$$\begin{aligned} F_{xg} &= -mg\sin\theta_b \\ F_{zg} &= mg\cos\theta_b \end{aligned} \tag{3.7}$$

Thrust The majority lifting force is provided by the thrust in VEGA launch vehicle. The pressure thrust is also neglected for turn-over flight phase. So the expressions of the force and torque generated by thrust are given by:

$$\begin{aligned} F_{Tx} &= T(t)\cos(\delta) \\ F_{Tz} &= -T(t)\sin(\delta) \\ M_T &= -T(t)l_{CG}\sin(\delta) \end{aligned} \tag{3.8}$$

where the $T(t)$ is thrust over the time, δ is the thrust gimbal angle and the l_{CG} is the length between the swivel point C on the nozzle and the center of gravity G(COG).

Aerodynamic force and torque When the VEGA launch vehicle in the ascending phase, the aerodynamic force and torque are generated by the interacting between the

LV and atmosphere. The effect of aerodynamic force and torque are not able to omit in early ascending stage, especially at the turn-over phase. The dynamic load has the most possibility to violate its constraint in the early ascending stage.The aerodynamic force and torque in the pitch motion is expressed as:

$$\begin{aligned} F_{Ax} &= -q(t)SC_A \\ F_{Az} &= -q(t)SC_N \\ M_{Ay} &= q(t)SC_N l_{GA} \end{aligned} \tag{3.9}$$

where $q(t)$ is the dynamic pressure, S is the reference surface, C_A is the coefficient of aerodynamic force at x axis, C_N is the coefficient of aerodynamic force at z axis and l_{GA} is the length between the center of pressure(or aerodynamic center) of LV and COG. The equation 3.9 has the negative sign because the forces on the x and z axis have negative direction to the body frame. F_{Ax} is also called aerodynamic drag and F_{Az} is the force acting through the point A(aerodynamic center). Note that the position of aerodynamic center is also depended on the time(lifting force). The coefficient C_N is a function of the angle of attack α.

When the effect of bending is involved in the EOM, the angle of attack α is no long a global variable over the time. It becomes the local angle of attack that also depends on the position x at the LV. The local angle of attack is then defined as:

$$\alpha_{local}(x) = \alpha + \frac{(l_{OG} - x)}{U}\dot{\theta} - \frac{\partial \xi(x,t)}{\partial x} - \frac{\dot{\xi}(x,t)}{U} \tag{3.10}$$

where l_{OG} is the length between the origin and the COG and $\xi(x,t)$ is the elastic deflection equation of the bending modes that will be introduced in later section. As the coefficient C_N is a function of angle of attack, the local C_N is also depended on the x and defined as:

$$C_{N\alpha} = \frac{\partial C_N}{\partial \alpha} \tag{3.11}$$

The aerodynamic force F_{Az} associated with bending modes is re-defined as:

$$\begin{aligned} F_{Az} &= -q(t)S\int_0^L C_{N\alpha}(x)\alpha_{local}(x)\mathrm{d}x \\ &= -q(t)S(\alpha\int_0^L C_{N\alpha}(x)\mathrm{d}x + \frac{\dot{\theta}}{U}\int_0^L C_{N\alpha}(x)(l_{OG}-x)\mathrm{d}x \\ &+ \sum_i q_i(t)\int_0^L C_{N\alpha}\sigma_i(x)\mathrm{d}x - \sum_i \frac{\dot{q}_i(t)}{U}\int_0^L C_{N\alpha}\phi_i(x)\mathrm{d}) \end{aligned} \tag{3.12}$$

where $q(t)$ is the dynamic pressure and equals to $1/2\rho U^2$, L is the length of VEGA launch vehicle,$q_i(t), \sigma_i(x)$ and ϕ are defined in the bending modes and will be explained in later section.

Rigid body dynamics Denote the $G = [x_b, y_b, z_b]$ as the body frame. The motion of the rigid body can be given by the dynamic expression:

$$\begin{aligned} M_G &= I\dot{\omega} \\ F_G &= m(\dot{V} + \omega \times V) \end{aligned} \tag{3.13}$$

where the M_G is the torque in the body frame, I is the inertia, m is the mass of LV, F_G is force on the LV and V is the linear velocity.
As the EOM of the rigid body only concerns the x-z plane in LV, by submitting $V = [\dot{x}_b, 0, \dot{z}_b]^T$ and $\omega = [0, \dot{\theta}_b, 0]^T$,the dynamic of the rigid in equation 3.13 can be re-defined as:

$$\begin{aligned} M_G &= I_{yy}\ddot{\theta}_b \\ F_x &= m(\ddot{x}_b + \dot{\theta}_b\dot{z}_b) \\ F_y &= m(\ddot{z}_b + \dot{\theta}_b\dot{x}_b) \end{aligned} \tag{3.14}$$

The mass m and the initial I or I_{yy} are assumed as constants or varying over the time slightly when compared to other dynamics. By considering all forces and torques on the LV in the body frame (gravity, thrust force/torque and aerodynamic force/torque), the force and torque in the rigid body frame is expressed as:

$$\begin{aligned} M_G &= l_{GA}q(t)SC_{N\alpha}\alpha - Tl_{CG}\sin(\delta) \\ F_x &= T\cos(\delta) - q(t)SC_A - mg\sin(\phi_b) \\ F_z &= -q(t)SC_{N\alpha}\alpha - T\sin(\delta) + mg\cos(\theta_b) \end{aligned} \tag{3.15}$$

Combined the equation 3.14 and 3.15, then

$$\begin{aligned} I_{yy}\ddot{\theta}_b &= l_{GA}q(t)SC_{N\alpha}\alpha - Tl_{CG}\sin(\delta) \\ m(\ddot{x}_b + \dot{\theta}_b\dot{z}_b) &= T\cos(\delta) - q(t)SC_A - mg\sin(\phi_b) \\ m(\ddot{z}_b + \dot{\theta}_b\dot{x}_b) &= -q(t)SC_{N\alpha}\alpha - T\sin(\delta) + mg\cos(\theta_b) \end{aligned} \tag{3.16}$$

The equation above can be given as a standard dynamic expression:

$$\begin{aligned}
\ddot{\theta}_b &= \frac{q(t)SC_{N\alpha}l_{GA}\alpha}{I_{yy}} - \frac{Tl_{CG}\sin(\delta)}{I_{yy}} \\
\ddot{x}_b &= \frac{T\cos(\delta) - q(t)SC_A}{m} - g\sin(\theta_b) - \dot{\theta}_b\dot{z}_b \\
\ddot{z}_b &= -\frac{q(t)SC_{N\alpha}\alpha}{m} - \frac{T\sin(\delta)}{m} + g\cos(\theta_b) + \dot{\theta}_b\dot{x}_b
\end{aligned} \tag{3.17}$$

3.2.2 Bending modes

Constrained by the capability of the thrust, the weight of the launch vehicle should be as light as possible by employing lighter material for the Launch vehicle rigid body. This sacrificed structure rigidity potentially makes the LV suffering extra flexibility and hence the additional aerodynamic load caused by the flexible modes. To study the elastic property of the LV, the bending modes are introduced. The analysis of EOM of the bending modes needs coupled with the EMO of rigid body. We use finite number of modes to describe the elastic deflection of the LV. Each mode is considered as a mass-spring-damping system. The Figure 3.4 shows the shape of elastic deflection mode coupling with motions of the LV. More detail about the bending modes can be found in [54].The equation of the elastic deflection at a point x along the LV is given as:

$$\xi(x,t) = \sum_{i=1}^{N} q_i(t)\phi_i(x) \tag{3.18}$$

where the x is the abscissa at the LV longitudinal axis, t is the time, ϕ denotes the normalized i^{th} mode shape and $q_i(t)$ is the i^{th} mode generalized coordinate. The q_i also satisfies the equation of the mass-spring-damping system and is expressed as:

$$\ddot{q}_i + 2\zeta_i\omega_i\dot{q}_i + \omega_i^2 q_i = Q_i \tag{3.19}$$

where the ζ_i is the sampling coefficient in i^{th} mode, the ω_i is the frequency of the i^{th} mode and Q_i is the generalized of the i^{th} mode.

Coupling with the LV motions and only considering the first moment of the forces, the

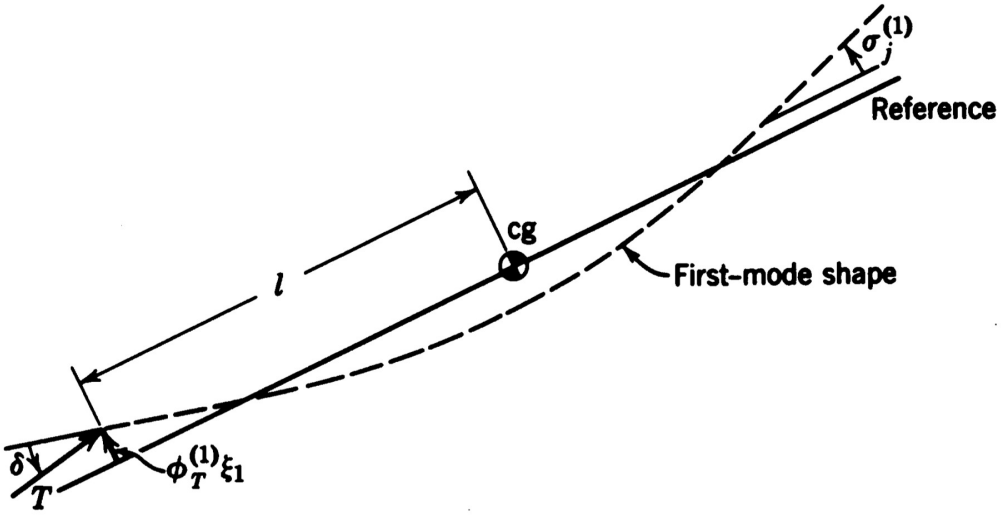


FIGURE 3.4: The bending mode shape for LV [1]

Q_i is expressed as:

$$\begin{aligned} Q_i &= \int_0^L (\sum F_z \phi_i(x) + \sum M_y \sigma_i(x)) \mathrm{d}x \\ &= \int_0^L [-(T\delta + m_E l_{EC} \ddot{\delta}) \phi_i(x) + I_E \ddot{\delta} \frac{\partial \phi(x)}{\partial x}] \mathrm{d}x \\ &= (-T\delta - m_E l_{EC} \ddot{\delta}) \phi_i(l_{OC}) + I_E \ddot{\delta} \frac{\partial \phi(l_{OC})}{\partial x} \end{aligned} \tag{3.20}$$

where F_z represents the force on the mode along the z axis in the rigid body frame, M_y represents the torque on the mode along the y direction in the rigid body frame, l_{OC} is the coordinate of the pivot point, I_E is the local inertia of the mode and

$$\sigma_i = \frac{\partial \phi_i}{\partial x} \tag{3.21}$$

Since the rigid body and elastic motions can be measured by the Inertial Measurement Unit (IMU),combined with the equation 3.18, 3.19,3.20, the equation of the elastic deflection can be solved.

3.2.3 TVC Actuator

The dynamic of the TVC actuator is presented by a transfer function: a second order model and a time delay for engine response:

$$W_{TVC}(s) = \frac{e^{-Ts}}{B_2 s^2 + B_1 s + B_0} \tag{3.22}$$

where the T is the time delay, B_2, B_1 and B_0 are the coefficient of the model. However, the TVC time delay will be neglected in the simulation. The reason will be mentioned later.

3.3 VEGA Simulink model

First flight phase with altitude between 30m and 60km is considered in the present study. The VEGA Simulink model is a nonlinear 6 degrees-of-freedom model simulator called VEGACONTROL or VEGAMATH to perform the simulation for VEGA flight in time domain during the stage 1 (P80 stage). The simulator is consist of 7 sub-blocks, such as the LV model(P80), the TVC (Thrust Vector Control) system, including the delay of computation, the GNC system (Guidance, navigation and control), the propulsion and MCI (Mass-center-inertia) system, the INS (Inertial navigation system), the bending modes and the aerodynamic modes. The figure of Simulink model is given in 3.5 [2] below.

[2]The reference of the simulator figure comes from the project document that is provided by ESA [2]

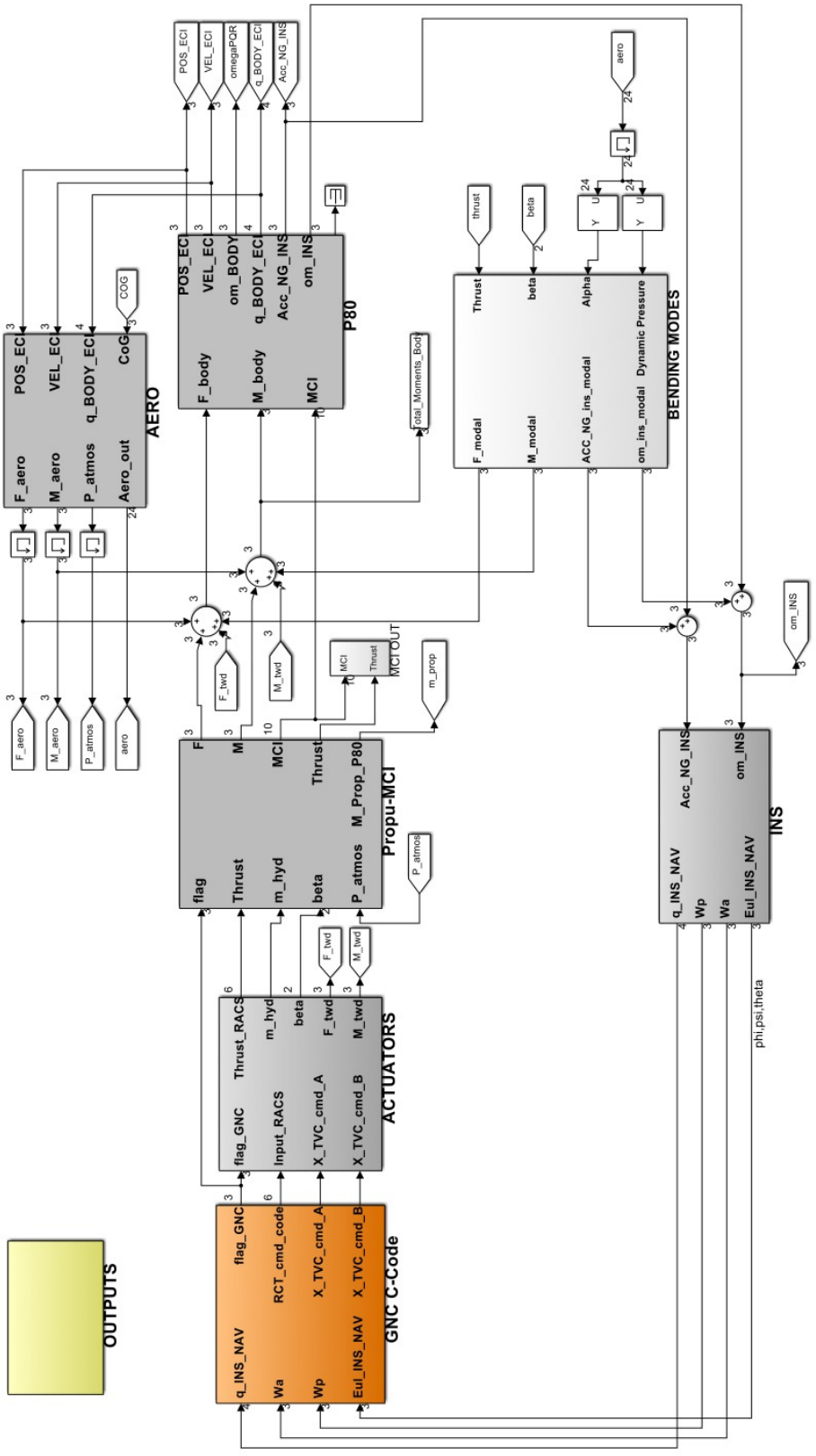


FIGURE 3.5: VEGACONTROL simulator [2]

The simulator has full set of non-linear equations of motion with the normal force, drag and tree axis moments curves depending on Mach and angle of attack, kinematic coupling in all axes, and a non-linear model of the eletro-mechanical actuator (EMA) dynamics with associated backlash and delays. The mathematical model for QUASAR Inertial Sensor Unit with its noise and bias characteristics and the propulsion model of the P80 solid propulsion system with validated thrust curves that include thrust oscillation effects to assess proper execution of the separation dynamics further constitute to the complexity of the simulator.

A high fidelity structural flexible mode model describing the launcher deformation is included to assess proper filtering and stability properties. The atmosphere model includes also a set of measured sizing wind-gust input models representative for the launch site Kourou. The launcher dynamics are driven by the FPSA ADA/C-flight code reflecting the flight management system for the time line sequence command and execution of associated guidance navigation and control system for thrust vector control (TVC) and roll and attitude control (RACS) and other support functions such as acceleration threshold detection (ATD) and pyro-valve command for stage separation. The different blocks made available from Industry were interconnected and configured to realize the simulation of the launch vehicle model in the present thesis.

The computation resource in our experiment for this Simulink model is given in the followed Table 3.1. For a single simulation process, the approximate simulation time is

	Computation Power
CPU	**Intel i5-2410M @2.3Hz dual core**
Memory	**4 GB**
Operation System	**Windows 7, 64-bit**
Matlab Version	**2009b** (Compulsory)
GPU	None

TABLE 3.1: The computation Power

12-32 seconds. The computation is varied depended on the total number of simulations. It is already observed that a longer computing time is required after the simulation count is over 2000. When the number of simulation set to 10000, a single computing process will roughly takes 30-35 seconds.

3.4 Uncertainties

The scaled upper bound values and lower bound values for the uncertainties (Scattering values) are fixed in the p-code file that is applied to the Matlab simulation. Thus, there is no way to access the file and no prior knowledge of these uncertain variables. The scaled scattering variables are multiplied by a nominal value in the interval [-1, 1]. So the uncertainties that we can investigate and vary are the normalized values only.All the uncertain variables are independently. For robustness analysis purpose, we try to explore the entire uncertain with equal opportunity. Thus, the uniform distribution is applied to the Monte Carlo method in this research. The description, variable name in the Matlab code and the category of a uncertainty is given in the Table 3.2.

Category	**Variable identifier**	**Description**
Aeroelasticity	Flag.aeroelastic	To enable the aeroelasticity effect(+10 % on CN coefficient)
Aerodynamics	Flag.disp_CA	Dispersion on Stage 1 Axial coefficient
	Flag.unc_CA	Uncertainty on Stage 1 Axial coefficient
	Flag.disp_CN	Dispersion on Stage 1 Normal Coefficient
	Flag.unc_CN	Uncertainty on Stage 1 Normal Coefficient
	Flag.disp_Xcp	Dispersion on Stage 1 Xcp
	Flag.unc_Xcp	Uncertainty on Stage 1 Xcp
	Flag.aero_roll	To enable Roll motion
Wind	Flag.azimuth_wind_angle	Wind azimuth direction [rad]
	Flag.h_wind	Synthetic wind gust altitude [Km]
	Flag.RealW	Enable Real wind
IRS (Inertial Reference System)	Flag.IRSmountingX	IRS Mounting Error w.r.t X Body Axis
	Flag.IRSmountingY	IRS Mounting Error w.r.t Y Body Axis

Continued on next page

Table 3.2 – *Continued from previous page*

Category	**Variable identifier**	**Description**
	Flag.IRSmountingZ	IRS Mounting Error w.r.t Z Body Axis
	Flag.INS_Noise	Enable INS Noise
Thrust Parameters Scattering	Flag.dISP	Stage 1 impulse scattering
	Flag.dTc	Scattering on time burn
	Flag.SRM_roll	Scattering on P80 Roll Torque
MCI	Flag.stagedM	Structural mass scattering
	Flag.stagedM_Prop	Scattering on propellant mass
	Flag.stagedJx	Scattering on XX MOI
	Flag.stagedJy	Scattering on YY MOI
	Flag.stagedJz	Scattering on ZZ MOI
	Flag.stagedxCOG	Scattering on X CoG(Center of Gravity)
	Flag.stagedyCOG	Scattering on Y CoG
	Flag.stagedzCOG	Scattering on Z Cog
	Flag.stagedJx_S	Scattering on structural Stage XX MOI
	Flag.stagedJy_S	Scattering on structural Stage YY MOI
	Flag.stagedJz_S	Scattering on structural Stage ZZ MOI
	Flag.stagedxCOG_S	Scattering on structural X CoG
	Flag.stagedyCOG_S	Scattering on structural Y CoG
	Flag.stagedzCOG_S	Scattering on structural Z CoG
	Flag.PLdM	Scattering on PL(Payload) Mass
	Flag.PLdJx	Scattering on PL XX MOI
	Flag.PLdJy	Scattering on PL YY MOI
	Flag.PLdJz	Scattering on PL ZZ MOI

Continued on next page

Table 3.2 – *Continued from previous page*

Category	**Variable identifier**	**Description**
	Flag.PLdxCOG	Scattering on PL X CoG
	Flag.PLdyCOG	Scattering on PL Y CoG
	Flag.PLdzCOG	Scattering on PL Z CoG
	Flag.PAdM	Scattering on PA Mass
	Flag.PAdJx	Scattering on PA XX MOI
	Flag.PAdJy	Scattering on PA YY MOI
	Flag.PAdJz	Scattering on PA ZZ MOI
	Flag.PAdxCOG	Scattering on PA X CoG
	Flag.PAdyCOG	Scattering on PA Y CoG
	Flag.PAdzCOG	Scattering on PA Z CoG
MCI	Flag.PL2dM	Scattering on PL2 Mass
	Flag.PL2dJx	Scatter on PL2 XX MOI
	Flag.PL2dJy	Scatter on PL2 YY MOI
	Flag.PL2dJz	Scatter on PL2 ZZ MOI
	Flag.PL2dxCOG	Scattering on PL2 X CoG
	Flag.PL2dyCOG	Scattering on PL2 Y CoG
	Flag.PL2dzCOG	Scattering on PL2 Z CoG
	Flag.RACSdM	Scalltering on RACS(Roll and Attitude Control system) Mass
	Flag.RACSdJx	Scattering on RACS XX MOI
	Flag.RACSdJy	Scattering on RACS YY MOI
	Flag.RACSdJz	Scattering on RACS ZZ MOI
	Flag.RACSdxCOG	Scattering on RACS X CoG
	Flag.RACSdyCOG	Scattering on RACS Y CoG
	Flag.RACSdzCOG	Scattering on RACS Z CoG
Thrust Offset and Misalignment	Flag.TVC_SF_A	Scattering on TVC gain Lane A
	Flag.TVC_bias_A_disp	Scattering on TVC Lane A(Gaussian)

Continued on next page

Table 3.2 – *Continued from previous page*

Category	**Variable identifier**	**Description**
	Flag.TVC_bias_A)unc	Scattering on TVC Lane A(uniform)
	Flag.TVC_SF_B	Scattering on TVC gain Lane B
	Flag.TVC_bias_B_disp	Scattering on TVC Lane B(Gaussian)
	Flag.TVC_bias_B)unc	Scattering on TVC Lane B(uniform)
	Flag.Thrust_misA_dis	Scattering on thrust misalignment first Lane(Gaussian)
	Flag.Thrust_misB_dis	Scattering on thrust misalignment Second Lane(Gaussian)
	Flag.Thrust_misA_unc	Scattering on thrust misalignment first Lane(uniform)
	Flag.Thrust_misB_unc	Scattering on thrust misalignment Second Lane(uniform)
	Flag.PvP_offsetX	Scattering on thrust offset in X
	Flag.PvP_offsetY_disp	Scattering on thrust offset in Y (Gaussian)
	Flag.PvP_offsetZ_disp	Scattering on thrust offset in Z (Gaussian)
	Flag.PvP_offsetY_unc	Scattering on thrust offset in Y (uniform)
	Flag.PvP_offsetZ_unc	Scattering on thrust offset in Z (uniform)
Atmosphere	Flag.air_density_scat	Atmosphere Density
Separation Disturb	Flag.sep_dist_yz	
	Flag.sep_dist_az	
Bending Modes	Flag.flex_freq	Scattering on Bending frequencies
	Flag.TMC_PVP	Scattering on Translation at Pivot point

Continued on next page

Table 3.2 – *Continued from previous page*

Category	**Variable identifier**	**Description**
	Flag.RMC_PVP	Scattering on Rotation at Pivot point
	Flag.TMC_INS	Scattering on Translation at INS
	Flag.RMC_INS	Scattering on Rotation at INS

TABLE 3.2: The description, category and variable name in VEGACONTROL of each uncertainty

3.5 Outputs

The VEGA simulator provides the output response in time domain. All the output responses with their variable name in the simulator, description and unite are provided in the Table 3.3.

Variable identifier	**Description**	**Unit**
out_MCI(:,1)	LV total mass	[Kg]
out_MCI(:,2)	X CoG	[m]
out_MCI(:,3)	Y CoG	[m]
out_MCI(:,4)	Z CoG	[m]
out_MCI(:5)	Jxx	[Kg m^2]
out_MCI(:6)	Jyy	[Kg m^2]
out_MCI(:7)	Jzz	[Kg m^2]
out_MCI(:8)	Jxy	[Kg m^2]
out_MCI(:9)	Jxz	[Kg m^2]
out_MCI(:10)	Jyz	[Kg m^2]
out_altitude	Altitude	[m]
out_mach	Mach number	[-]
out_pdyn	Dynamic pressure	[Pa]

Continued on next page

Table 3.3 – *Continued from previous page*

Variable identifier	**Description**	**Unit**
out_q_alpha	Q-alpha(Dynamic load)	[Pa deg]
out_F_aero	Aerodynamic forces(x,y,z axis)	[N]
out_M_aero	Aerodynamic torques(x,y,z axis)	[Nm]
out_AoA(:,1)	Angle of attack	[deg]
out_AoA(:,2)	Incidence	[deg]
out_AoA(:,3)	Sideslip	[deg]
Attitude_error	Attitude error(pitch, yaw,roll)	[rad]
out_beta(:,1)	TVC dynamics lane A	[rad]
out_beta(:,2)	TVC dynamics lane B	[rad]
out_beta_cmdA	TVC Commanded Value lane A	[rad]
out_beta_cmdB	TVC Commanded Value land B	[rad]
out_betaDotA	TVC rate deflection Value lane A	[rad/s]
out_betaDotB	TVC rate deflection Value land B	[rad/s]
out_omegaPQR	Angular velocity in Body axes(roll, yaw, pitch)	[rad/s]
out_POS_ECI	LV position in ECI RF	[m]
out_VEL_ECI	LV velocity in ECI RF	[m/s]
out_thrust	Thrust	[N]
out_Acc_NG_INS	Non gravitational acceleration sensed at INS	[m/s^2]
out_time	Simulation time	[s]
out_Wa	Velocity along accelerometers	[m/s]
out_Wp	Platform velocity	[m/s]
Wx	Non gravitational velocity	[m/s]
NS_Ctrl(:,1)	Drift velocity - Z axis	[m/s]
NS_Ctrl(:,2)	Drift position - Z axis	[m]
NS_Ctrl(:,3)	Drift velocity - Y axis	[m/s]
NS_Ctrl(:,4)	Drift position - Y axis	[m]
out_Wind_east	Wind East Component vs. Time	[m/s]

Continued on next page

Table 3.3 – *Continued from previous page*

Variable identifier	**Description**	**Unit**
out_Wind_north	Wind North Component vs. Time	[m/s]

TABLE 3.3: The variable name in the VEGACONTROL, description and the unit of each output

For example, the output response of Q_α in terms of the nominal value of uncertainties is given in Figure 3.6.

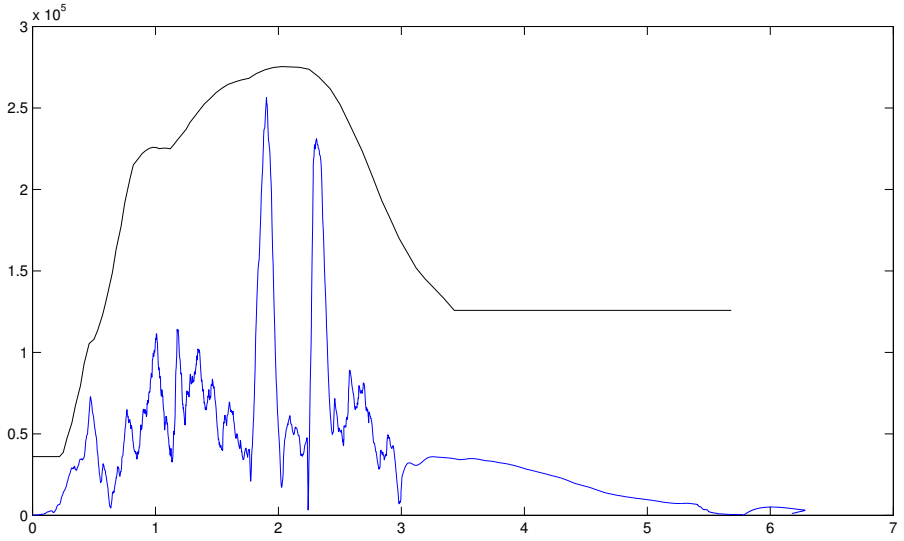


FIGURE 3.6: The nominal output response of Q_α (x-axis - Q_α [Pa deg], y-axis - Mach)

3.6 Objectives

The main research objective is the assessment and understanding of aerodynamic loads constraints during the Stage 1 (P80 phase) subject to the uncertain parameters listed in Table 3.2. The interested problems are: (i) Determine the maximum excursion value for the dynamic aerodynamic load over the given Mach profile; (ii) Determine the uncertain parameter directions corresponding to the maximum excursion of aerodynamic load; (iii) Compare the response due to the uncertain parameter in (ii) against the Q_α vs. Mach

constraint profile depicted in Figure 3.6; (iv) Analysis to be completed within a fixed computational budget that typically favoured by Industry in a reliable and convincing manner; (v) Determine driving uncertain parameters, or active uncertain parameter space, i.e., most sensitive or contributing uncertain parameters among the many; (vi) Develop exploratory analysis using efficient sampling based and krigging based methods to obtain solutions for (i)-(v). In this thesis, a function evaluation requires simulation of the 6 Degrees-of-freedom(DOF) VEGA simulator performing the P80 ascent phase. The main inputs varied for the analysis are mostly those listed uncertainties in Table 3.2. The criteria that are helpful for assessing performance violations, if any, for P80 phase are: (i) Q_α less than a desired profile as shown in black in Figure 3.6; (ii) total angle of attack < 3 [deg]; (iii) lateral drift position < 500 [m] for z and y components; (iv) lateral drift velocity <15 [m/s] for $\dot{z}$ and $\dot{y}$ components. In sequel, efficient quasi Monte Carlo sampling schemes and Gaussian process based methods will be explored to understand this problem in detail.

Chapter 4

Robustness analysis based on Monte Carlo, quasi-Monte Carlo and optimization methods

The basic concept of Monte Carlo method is given in the Chapter 2.1. The objective is to provide elementary concepts of necessary probability theory, to define basic notations and terminology that will be used in the thesis. In sequel, the concept of variance reduction and quasi Monte Carlo techniques are discussed in Chapter 2.2. Finally, the idea of optimization based methods tailored to the robustness analysis problem is briefed in Chapter 2.3. The results of robustness analysis based on these approaches mentioned above are discussed in this chapter in order to highlight the advantage of Quasi-Monte Carlo approach.

4.1 Samples construction based on Quasi-Monte Carlo sequence

4.1.1 Sobol sequence

The Sobol sequence is constructed based on the theory of (t, m, s)-nets and (t, s)-sequences developed by Niederreiter to construct a low discrepancy sequence [31]. The (t, m, s)-nets denotes a class of point sets that have zero discrepancy for several sub

intervals J in I^s. The (t, s)-sequences denotes a class of sequences of points in I^s that form the particular nets (point sets). To generate the Sobol sequence, a conventional algorithm, the so called Algorithm 659 [55], has been made use of and discussed in brief here. To construct the Sobol sequence with low discrepancy in the unit interval, firstly, the *direction number* is needed. The *direction number* is a binary fraction number ($v_i = 0.v_{i1}v_{i2}...$) and is defined as:

$$v_i = m_i/2^i \tag{4.1}$$

where the m_i is an odd integer with $0 < m_i < 2^i$ The m_i can be chosen freely. The way to obtain the direction number is choosing a primitive polynomial in $GF(2)$ field (also called Z_2 field) [31, 55]. In this algorithm, the polynomial is chosen as:

$$P = x^d + a_1x^{d-1} + ... + a_{d-1}x + 1 \tag{4.2}$$

where a_i is 0 or 1 and P is a primitive polynomial of degree d in $GF(2)$. Once the polynomial is settled, its coefficients are used to define a recurrence to compute the v_i:

$$v_i = a_1v_{i-1} \oplus a_2v_{i-2} \oplus ... \oplus a_{d-1}v_{i-d+1} \oplus v_{i-d} \oplus [v_{i-d}/2^d],\ i > d \tag{4.3}$$

where $\oplus$ is the bit-by-bit operation.
The recurrence in terms of the m_i can be expressed as

$$m_i = 2a_1m_{i-1} \oplus 2^2a_2m_{i-2} \oplus ... \oplus 2^{d-1}a_{d-1}m_{i-d+1} \oplus 2^dm_{i-d} \oplus m_{i-d} \tag{4.4}$$

Eventually, for the i^{th} point, it is constructed by followed definition:

$$x_i = i_1v_1 \oplus i_2v_2 \oplus ... \oplus i_kv_k \oplus ...$$

where i_k is the k^{th} binary digit number of the i, $i = (...i_3i_2i_1)_2$. A unique good property of the sequence based on this principle is that the when the points x_i in the sequence are projected to a lower dimensional face, the points would not be distributed over the same location in a higher dimensional space.

4.1.2 Halton Sequence

To construct the Halton Sequence, two definitions - Digit Expansion and Radical Inverse Function - are to be introduced [31]. For any integer $b \geq 2$, there is a set $Z_b = \{0, 1, 2, ..., b-1\}$. So a integer $n \geq 0$ can be expressed by the *digit expansion* in base b:

$$n = \sum_{j=0}^{\infty} a_j(n) b^j \tag{4.5}$$

where $a_j(n) \in Z_b$. $a_j(n) = 0$ when j is large enough. For any integer $b \geq 2$, the *radical inverse function* ϕ_b in base b is expressed as:

$$\phi_b(n) = \sum_{j=0}^{\infty} a_j(n) b^{(-j-1)} \tag{4.6}$$

where n is the digit expansion mentioned above.

The Halton sequence is defined in terms of these two definitions. For any dimension $s \geq 1$ and any integer $b_1, ..., b_s \geq 2$, the Halton sequence x_n in base $b_1, ..., b_s$ is defined as:

$$x_n = \{\phi_{b_1}(n), ..., \phi_{b_s}(n)\} \tag{4.7}$$

where the element $\phi_{b_i}, 1 \leq i \leq s$ in x_n is the radical inverse function mentioned above.

4.1.3 Faure Sequence

The way to construct Faure Sequence has mutual concept as the way to construct Halton sequence. For Faure sequence, the base b is required as $b \geq s$, s is the size of dimension. So there is a set $Z_b = \{0, 1, ..., b-1\}$. The maximum number of sampling size is limited to $N = d^m$, where m is an positive integer and $0 \leq j \leq i \leq m$. Denote $c_{ij} = \binom{i}{j} \bmod d$, where 'mod' is the modulo operation. The modulo operation obtains the reminder after one number is divided by another. Then an integer n can be expressed as:

$$n = \sum_{i=0}^{m-1} a_i(n) b^i \tag{4.8}$$

where $a_i(n) \in Z_b$. The first element of x_n is expressed as:

$$x_n^1 = \sum_{j=0}^{m-1} a_j(n) b^{-j-1} \tag{4.9}$$

For rest of the elements,

$$a_j(n) = \sum_{l=j}^{m-1} c_{lj} a_l(n) \bmod d \tag{4.10}$$

Then,

$$x_n^i = \sum_{j=0}^{m-1} a_j(n) d^{-j-1}, i \neq 1 \tag{4.11}$$

4.2 Distributional robustness analysis

ELV launcher model, a complex system, normally needs several seconds, or minutes (depending on configuration and phase of trajectory considered) for a single run of the simulation. Besides, the dimension of the uncertainty is typically large. Moreover, the aim is to carry out the analysis in a convincing manner within a limited computational budget. Though there are possibility for parallel processing on multi core machines, this option is not in the present plan and scope of the research work. Following the directions of industrial partners, the computational budget is fixed at 1000, i.e., the maximum number of allowed simulations for achieving the solution is 1000. From the nominal simulations itself, it was observed that the presence of wind contributes significantly to the aerodynamic load excursions. The original ELV model has 124 uncertainties, of which 6 of them are fixed to their nominal values (following discussions with ESA(European Space Agency) and the team members who create the ELV launcher model).

4.2.1 Objective functions

In present research problem, all the objective functions, which encapsulate the specific design performance measures, needs to be maximized (to reveal the extreme behavior associated with the system). Because the optimization algorithms can only deal with the minimization procedure, the objective function will be taken a negative sign when the optimization based validation is carried out. A general format of the worst-case

validation is defined as:

$$\begin{gathered} \max(f(\Delta)) \; or \; \min(-f(\Delta)) \\ \text{subject to } \Delta^- \leq \Delta \leq \Delta^+ \end{gathered} \tag{4.12}$$

where the Δ is the uncertainty set with 118 dimensions($\Delta = [\delta_1, \delta_2, ..., \delta_{118}]$), the Δ^- is the lower bound of the uncertainties and the Δ^+ is the upper bound of the uncertainties.

Six basic objective functions are considered [1], of which the main focus would be only given for the first objective function as it is of prime importance for the industry. The uncertainties are normalized, such that the lower bound for all δ_i is -1 and the upper bound for all δ_i is 1. The f_i is objective function or cost function
They are (Units given in square brackets):
(1) $f_1 = \max(Q_\alpha(Mach))$ - The maximum dynamic load over entire Mach, [Pa Deg]
(2) $f_2 = \max(D_{V\_Z}(t))$ - The maximum drift velocity of Z axis over the time, [m/s]
(3) $f_3 = \max(D_{P\_Z}(t))$ - The maximum drift position of Z axis over the time ,[m]
(4) $f_4 = \max(D_{V\_Y}(t))$ - The maximum drift velocity of Y axis over the time ,[m,s]
(5) $f_5 = \max(D_{P\_Y}(t))$ - The maximum drift position of Y axis over the time , [m]
(6) $f_6 = \max(AoA_T(t))$ - The maximum total angle of attack ,[Deg]

4.2.2 Results

For all efficient sampling based worst case validation, a computational budget fixed at 1000 simulation runs (following directions in [25]) have been considered. Four different sampling schemes have been considered. They are based on crude Monte Carlo method (Blue) and quasi Monte Carlo method using Faure (Green), Halton (Purple) and Sobol (Red) sequences. The worst case function values obtained from each sampling scheme vs. the six different cost functions are summarized in Table 4.1. Since the industrial focus in on the aerodynamic load response (Q_α) subject to the uncertain parameters, and moreover to save space and to give a comprehensive set of results, the analysis results are summarized mainly in three plots (related to only the cost function f_1,which is the maximum dynamic loads) such as i) convergence plot, i.e., the number of simulations

[1] In this chapter, only the results based on objective function f_1 are given. All other results in terms of other objective functions are given in the Appendix

vs. the maximum value for the cost function attained; ii) the worst case perturbation direction in a normalized interval; iii) the dynamic response plot, i.e., the Q_α vs. Mach plot. The Figure 4.1 is the convergence plot for cost function f_1 over different distribu-

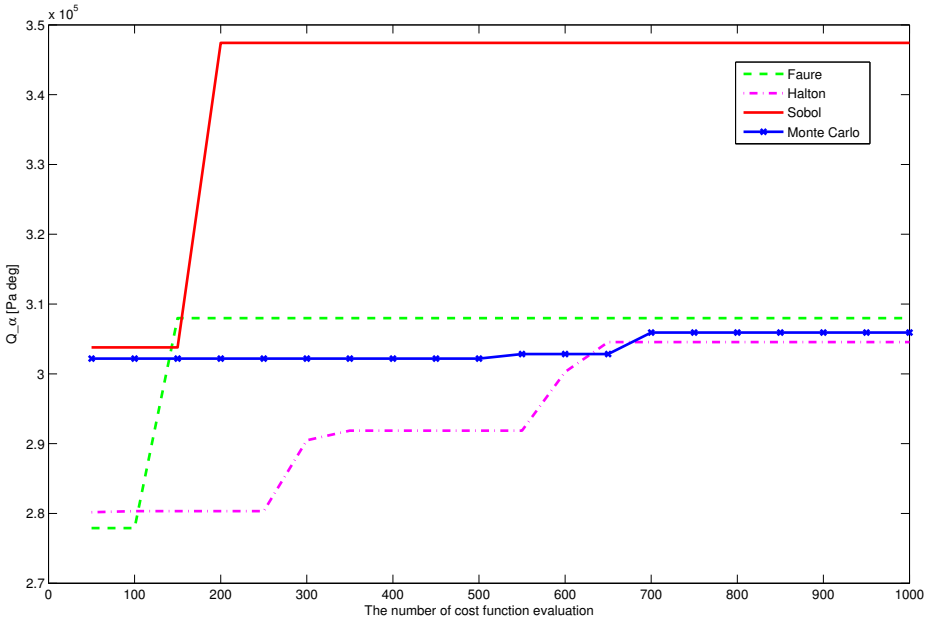
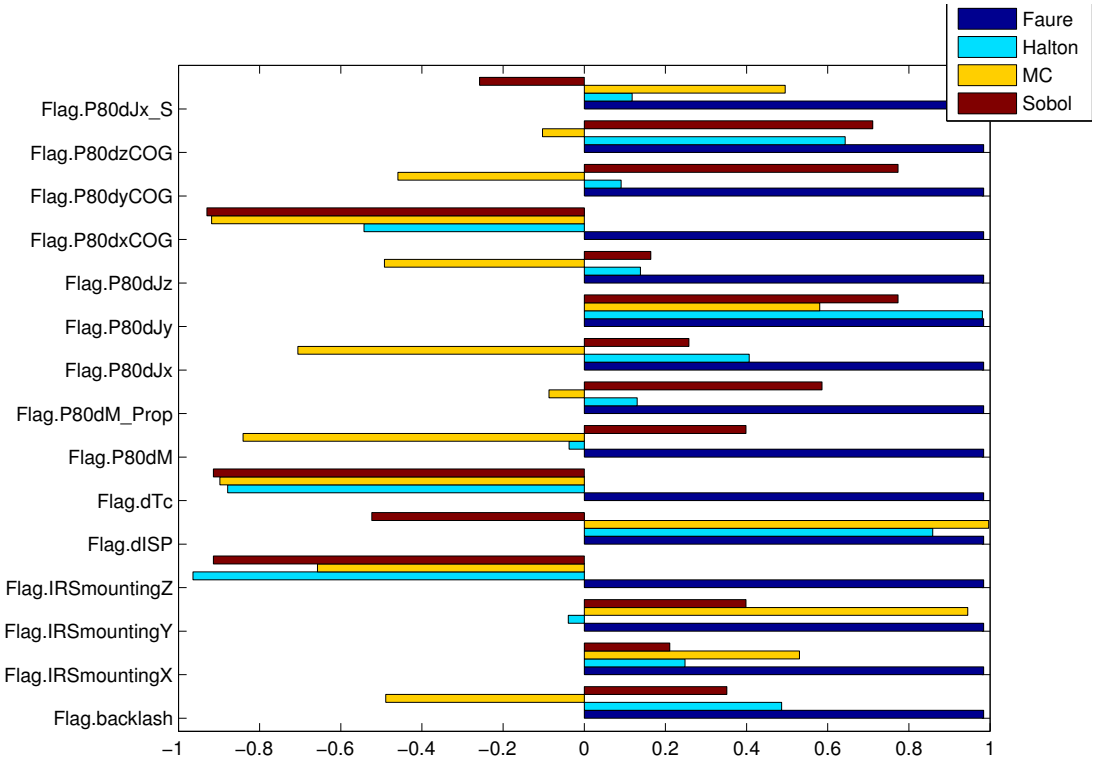


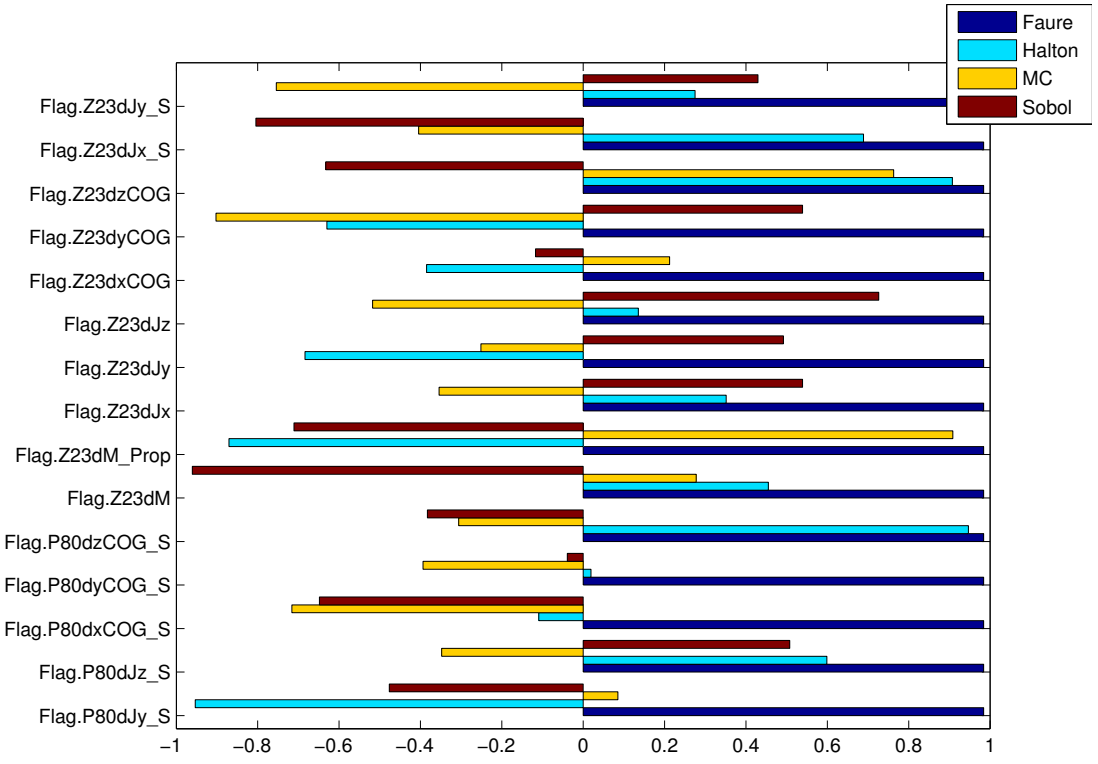
FIGURE 4.1: The convergence plot for f_1 based on different distribution method

tional methods. It is clear that, the Sobol sequence has a better convergence with the highest cost value (or objective value) over the other methods. When compared, Halton sequence seems to have the relatively poor performance than crude Monte Carlo method, however the difference is relatively small. Hence, the argument that in [11] 'quasi-Monte Carlo method would not loss effectiveness when compared to Monte Carlo' holds good in considered uncertainty analysis problems: the quasi-Monte Carlo has at least the similar performance as Monte Carlo for high dimensional uncertainty analysis problems with the launch vehicle models.

The worst case perturbation directions in a normalized interval are given in Figure 4.2,4.3,4.4 and 4.5. Note that, the uncertainty vector with the largest objective value, i.e., the worst case uncertain parameter, with respect to the quasi-Monte Carlo and Monte Carlo method is considered for the bar plot. From the perturbation bar plot, it can be observed that how far the variables are away from the nominal value. Those variables who have large perturbation values are highly possible to affect the cost value heavily. From this figure, all uncertainties in the worst-case of Faure sequence have large

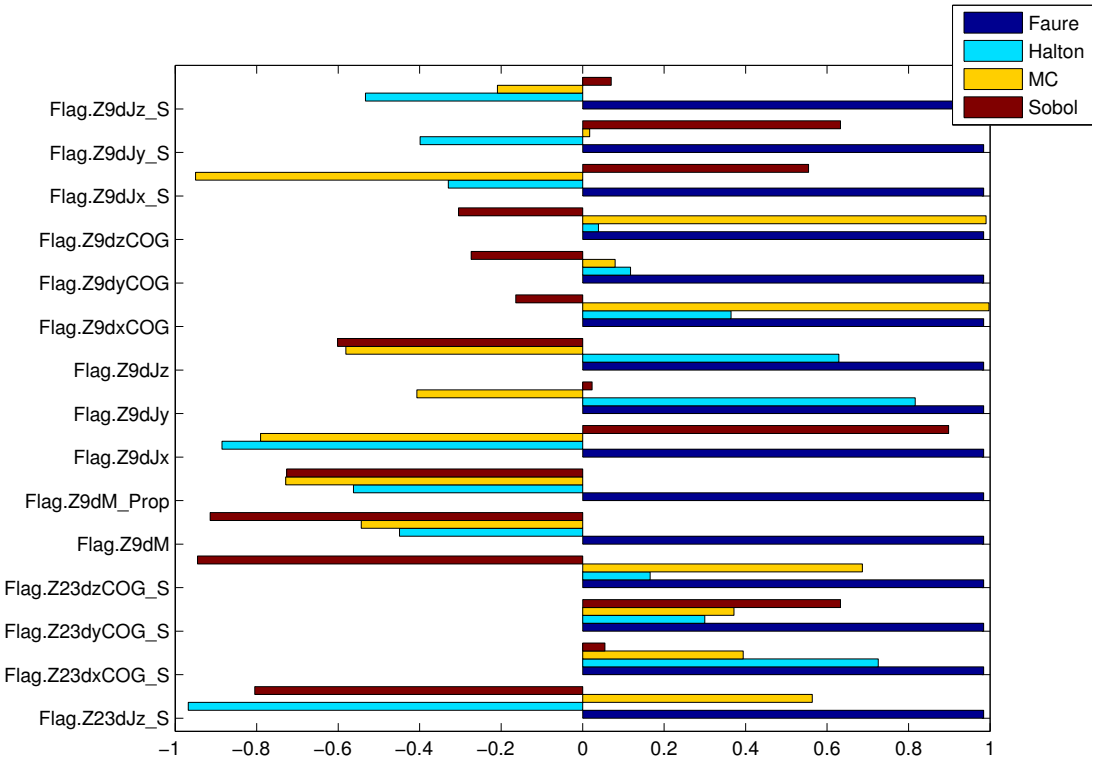


(A) Variables from 1 to 15 under different worst-case scenario

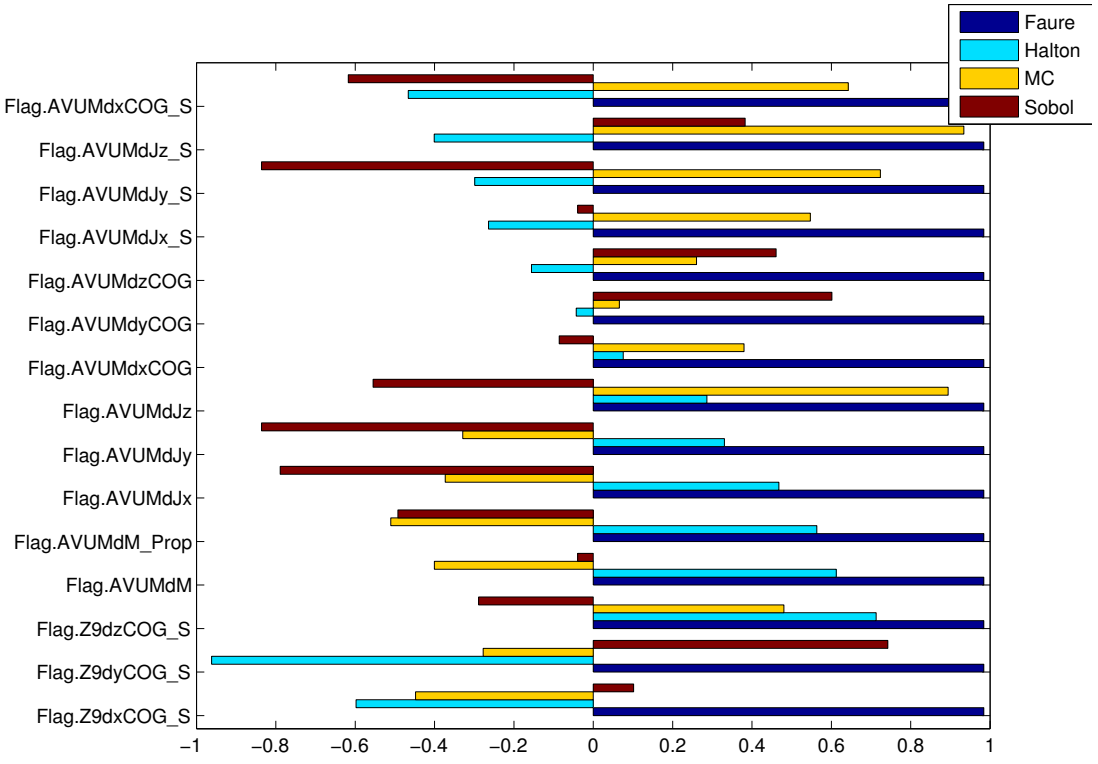


(B) Variables from 16 to 30 under different worst-case scenario

FIGURE 4.2: The worst case uncertainties (1 -30) for f_1

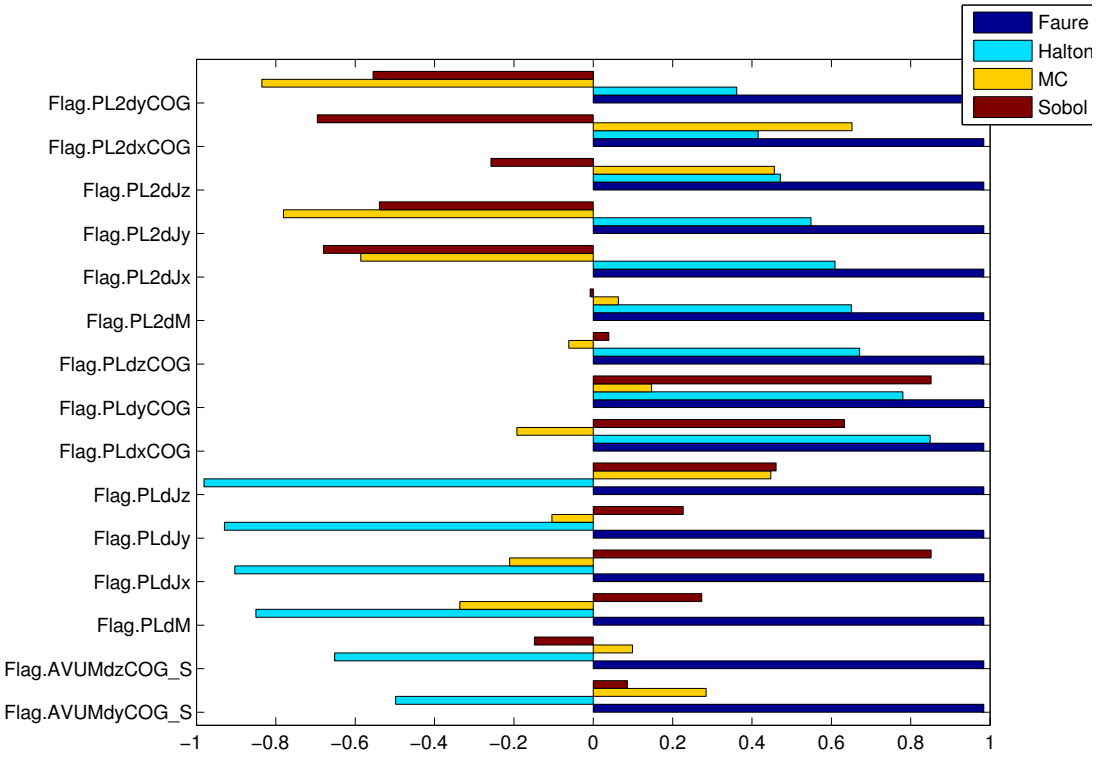


(A) Variables from 31 to 45 under different worst-case scenario

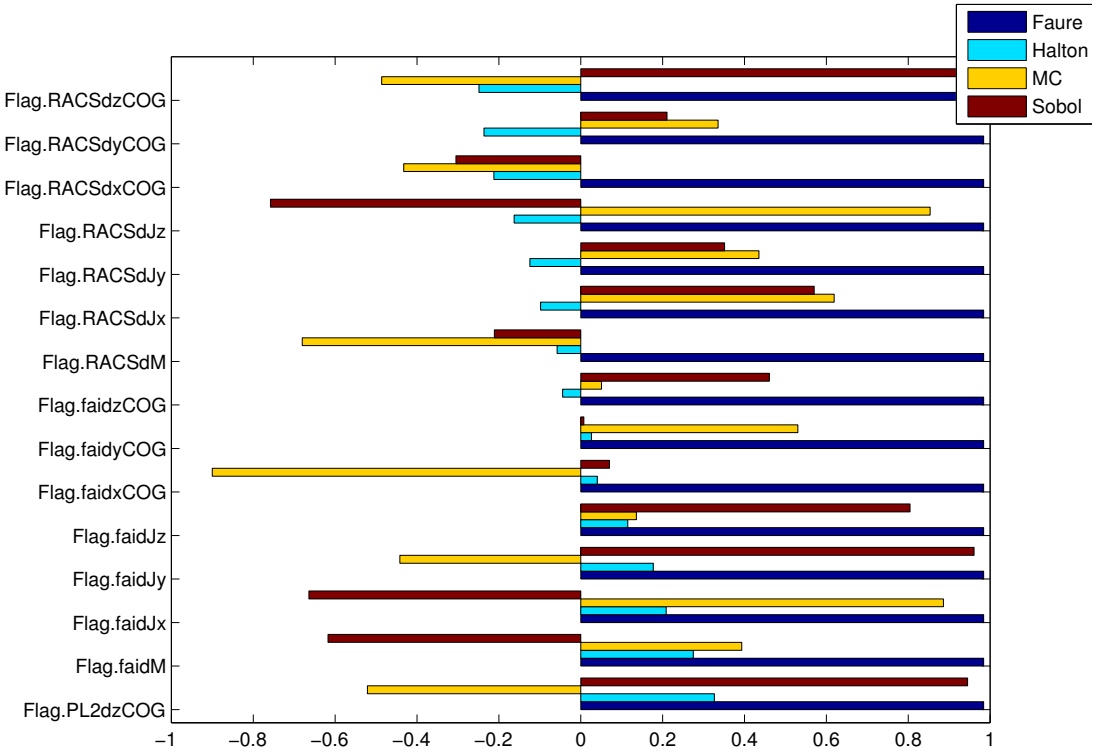


(B) Variables from 46 to 60 under different worst-case scenario

FIGURE 4.3: The worst case uncertainties (31 -60) for f_1

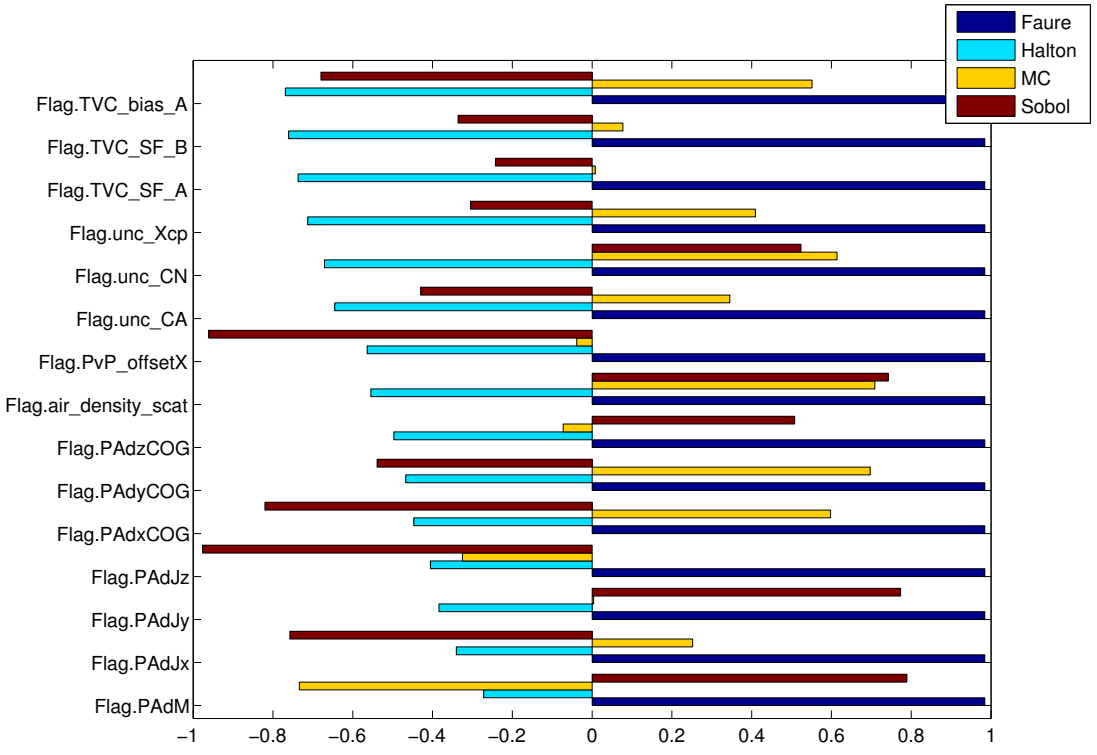


(A) Variables from 61 to 75 under different worst-case scenario

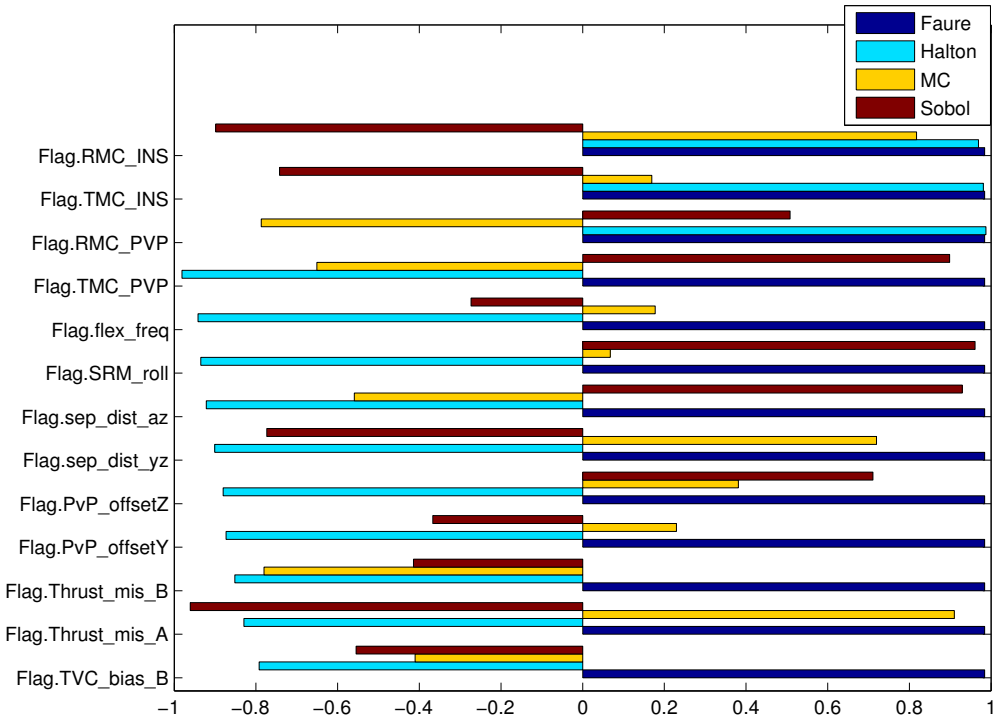


(B) Variables from 76 to 90 under different worst-case scenario

FIGURE 4.4: The worst case uncertainties (61 -90) for f_1



(A) Variables from 91 to 105 under different worst-case scenario



(B) Variables from 106 to 118 under different worst-case scenario

FIGURE 4.5: The worst case uncertainties (91 -118) for f_1

bias contrast to their nominal values and almost reach the upper bound 1. This means the worst value must be dominated by some large positive uncertainties. In the figure, some uncertainties in the four worst-cases do not have a particular pattern in direction or bias value, such as the variable '*Z23dJx_S*' in plot (B) in Figure 4.2. We can observe that the bias could either be small or larger and direction could either be negative or positive. On the other hand, some uncertainties, such as '*P80dJy*' in plot (A) of Figure 4.2, do have a recognized pattern that all uncertainties have similar bias with same direction. These uncertainties may have high probability in linear relationship between volume of bias and objective value. It also has a special pattern in the interests of investigate. Note the uncertainty 'dTc' in plot(A) of Figure 4.2, this uncertainty has large negative biases in the worst-case of Halton, MC and Sobol but large positive bias in the worst-case of Faure. There is a small group of such uncertainties that three biases are in the same direction with similar perturbation values but the remain one is in a different direction. These uncertainties may have similar property as the uncertainties have the four biases in the same direction and value. One example is that the variable 'dTc' is proved as one of the dominant variables in Chapter 6. One inference can be made in this case is that the negative bias in the worst-case of Faure sequence makes a 'negative effect' or 'damping effect' to objective value f_1 to decrease the cost value when the bias reach the upper bound. In this case, 12 uncertainties can be identified with pattern of interesting by following the principle mentioned above. Another important point to mention here is that the results using quasi Monte Carlo methods are reproducible when compared to Monte Carlo methods.

The Q_α vs. Mach plot gives the information that at what speed the VEGA lunch vehicle violates its load constraint, if any. The plot of Q_α vs. Mach number based on the worst-case of cost function f_1 in terms of different quasi Monte Carlo and Monte Carlo distribution methods is give in the Figure 4.6. The response line in black colour indicates the reference design aerodynamic load envelope w.r.t variation in Mach. From figure 4.6, it is obvious that the the value of Q_α based on different distribution methods exceed the constraint for a short span of Mach. These spans happen at different mach number of different methods, but all of them around the region the launch vehicle attains Mach number 2. The differences are due to the slight variations in the worst case uncertain parameters identified by the different methods as given in comparison figure **??**. It means that the dynamic load of VEGA launch vehicle is higher than its

designated value when the speed of the launcher is around Mach 2 operating region in the P80 phase. The constraint violation implies the VEGA launch vehicle suffers more pressure than its anticipated maximum pressure. This is of significant risk for the launch vehicle and hence there is high probability for a failure of mission. Note that the value of Q_α twice crossover the constraint for Sobol and Halton methods whereas once for Faure and Monte Carlo methods. For comprehensive reading of this thesis, several similar

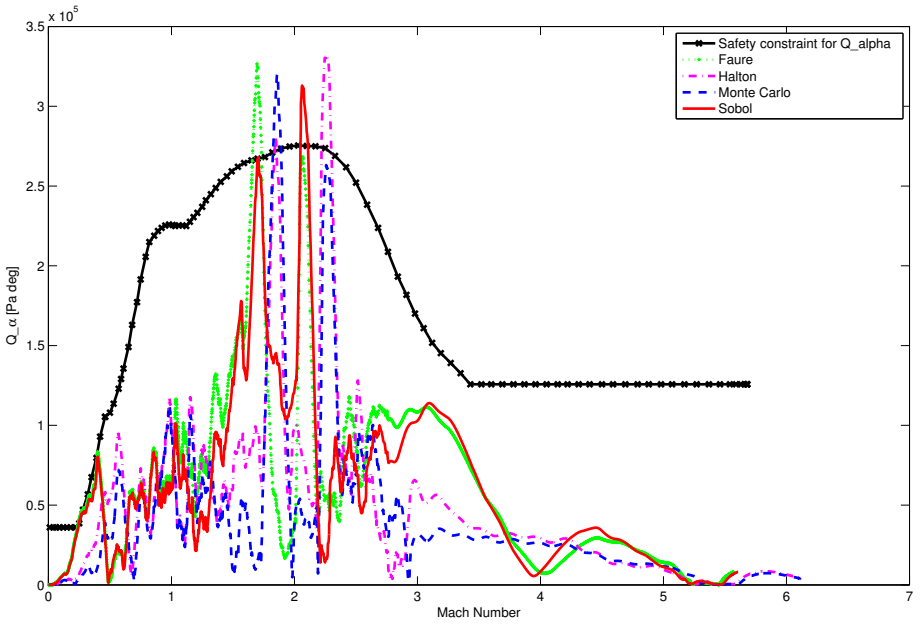


FIGURE 4.6: The Q_α plot in terms of the worst case from $f1$ by different distribution methods

figures (including convergence plot, perturbation bar plot and dynamic time response plot) for the cost function of f_2 to f_6 (details of cost function can be found in Chapter 4.2.1) are not given. However, to convey the main message put forward in the chapter, the worst case values based on different distribution methods for all the cost functions (within the fixed computational budget of 1000 simulations maximum) are given in the Table 4.1.

Computation Cost

As all the results are limited to 1000 simulations, an average computing time for each result is 372.4 minutes that is approximate 6.2 hours. The computation power and Matlab version is already introduced in Chapter 3.3.

Note that, to avoid the Monte Carlo method 'luckily' obtain a very large objective

Method	f1	f2	f3	f4	f5	f6
Monte Carlo	3.0592×10^5	23.3822	1003.4694	25.5220	961.7573	3.1390
Sobol	3.4742×10^5	20.6084	1076.1478	23.8545	1138.4139	3.1404
Halton	3.0455×10^5	25.5019	1073.8014.	24.0327	1121.7021	3.1397
Faure	3.0799×10^5	23.2363	1029.1031	23.4257	1147.3471	3.1369

TABLE 4.1: The worst case values based on different distribution method for cost function f_1 to f_6

value from a single worst case valuation (following guidance from [24, 25]), we precede an independent 25 runs of Monte Carlo based worst case validation. The worst case values for Monte Carlo method mentioned above is the closest one to the average of the 25 independent worst case values. This principle is applied to all worst case values obtained by Monte Carlo method in this thesis.

From the table, the Sobol sequence has a overall best performance , especially outstanding for function f_1 over other methods: about $10\% \sim 20\%$ larger than other methods . But the Sobol sequence has a much smaller worst-case value of objective function f_2 over other methods. For cost function f_5, the quasi-Monte Carlo based methods have a better performance than the Monte Carlo method. There is not too much difference in function f_6 based among these sampling methods. An observation in this regard is that the response surface of the objective function f_6 is too flat. This table reemphasize that the quasi-Monte Carlo at least has a similar performance to that of the Monte Carlo, and even much better for cost functions with this high dimensional problem.

Nevertheless, from the Table 4.1, we can guarantee the quasi-Monte Carlo won't be worse than the Monte Carlo method when the computation budget is limit. The quasi-Monte Carlo is good methodology for distributional uncertainty analysis for identifying potential worst case perturbations, and all of the worst case values given in the Table 4.1 exceed their design specified constraints. The results suggest there is requirement for further enhancing the robustness properties of the design, or rather introduce operational limitations. However, the issue of convergence of the quasi-Monte Carlo and its relation to the effective dimension still remains. This issue will be discussed more in the next chapter by carrying out the global sensitivity analysis for identifying the active uncertain parameter space, or key uncertain parameters or driving uncertain parameters for the specific problem. What we would then do is to check whether the worst case solutions

and directions obtained in this chapter is a fair representation of the identified active uncertain parameter space or driving uncertain parameter space.

4.3 Differential Evolution

The basic concept of DE and its application of robustness analysis in aerospace industry is introduced in Chapter 2.3. In this section, the analysis results based on DE are given.

4.3.1 Worst-case validation results

The objective functions in the optimization based worst-case validation are same as the functions used in distributional based worst-case validation. The configurations of the DE, DE with Sobol initialization and HDE algorithms are very important factors to affect the performance of the worst-case validation. All configurations for DE, DE with Sobol initialization and HDE algorithms are given in the tables 4.2 - 4.4 as followed.

Configuration	Value
Number of population	50
Number of iterations	20
Step-size(scaler factor) F for mutation $[0 \sim 1]$	0.8
Crossover probability ρ_c $[0 \sim 1]$	0.8

TABLE 4.2: The configurations for DE

Configuration	Value
Number of population	50
Number of iterations	20
Step-size(scaler factor) F for mutation $[0 \sim 1]$	0.04
Crossover probability ρ_c $[0 \sim 1]$	0.8

TABLE 4.3: The configurations for DE with Sobol initialization

Configuration	Value
Number of population	50
Number of iterations	20
Step-size(scaler factor) F for mutation $[0 \sim 1]$	0.8
Crossover probability ρ_c $[0 \sim 1]$	0.8
The number of function evaluation for local optimization	237

TABLE 4.4: The configurations for HDE

Choice of population and iterations

As the number of simulation is constrained to 1000, the multiplication of number of population and iterations is equal to 1000. The number of population theoretically can be chosen freely. However, a different choice could yield a different effect for the optimization. A larger number of population can explore a wider uncertainty space. More iterations leads a better worst-case in most cases because there is larger opportunity to drive the optimum value down to a lower objective surface(for minimization only) . So ideally, the number of population and iteration should both be as large as enough. However, constrained by the limited computation resource, it must have a trade-off choice to maximize the optimization performance. According to our experience and experiment conclusion from other researchers, [24, 25] for example, a 50 population size with 20 iterations is a good candidate for such a trade-off decision. As our research interest is not comparing the different trade-off choices that yield the best optimization results, we only give the robustness analysis results based on the configurations mentioned in the tables above.

Choice of step-size

Normally, for crude DE, the value of step-size is greater than 0.8 for a good performance based on our expertise and guidance in [20, 24, 25]. However, in Table 4.3, the step-size is very small, only 0.04. The reason is that the Sobol initialization provides a better uniform and even distribution. Although some 2D projections in high dimension may reveal a some cluster of points (for example, in Figure 2.2), our purpose is to disperse these cluster of points more widely but without much ruining the uniformity. Since the step-size determines how far the next generation will move, so a small value for DE with Sobol initialization is favoured. To support this argument, several numerical experiments with standard test case functions have been carried out. Based on the numerical

experiments, the empirical value for step-size for DE with Sobol initialization is smaller than 0.1. Nevertheless, this value is also coupled with the number of population.

For HDE, a extra configuration is called 'the number of function evaluation for local optimization'. The value of 237 in our experiment is assigned as the minimal requirement number of function evaluation based on the Matlab optimization function *fmincon* because of our limited computational budget. The 'number of function evaluation for local optimization' could increase the total number of function evaluations enormously. Based on the configurations in Table 4.4, the total number of function evaluations could be from 1000 to 5740, but forced stopping after 1250 simulations is introduced (just to be sure about completion of the local optimisation). The Table 4.5 gives the worst-case values based on the three optimization methods for cost function $f_1 \sim f_6$(details of cost function can be found in Chapter 4.2.1). From this table, the DE with Sobol

Method	f1	f2	f3	f4	f5	f6
DE	3.3318×10^5	22.3257	923.6494	26.5737	983.0093	3.1404
DE with Sobol	3.8773×10^5	29.4903	1163.9797	34.3820	1180.0562	3.1394
HDE	3.3033×10^5	22.6730	1051.2470.	23.4222	1046.5893	3.1399

TABLE 4.5: The worst case values based on different optimization methods for cost function f_1 to f_6

initialization has a much better performance overall. The advantage of the DE with Sobol initialization is obvious except the cost function f_6. It demonstrates that a better initialization of the DE is able to perform better in optimization. Comparing the results in Table 4.5 and 4.1, it can be seen that the optimization based worst-case validation has an overall better performance than the Monte Carlo and quasi Monte Carlo methods. Generally speaking, the optimization based method for worst-case validation can still be considered as a better choice than the distributional methods. Introducing the quasi Monte Carlo based initialization step in the optimization based analysis method offers significant improvement in obtaining the global solution swiftly.

Computation Cost

-DE and DE with Sobol: The number of simulation are both 1000. The average computing time for each result is 375.1 minutes that is approximate 6.3 hours. The computation power and Matlab version is already introduced in Chapter 3.3.

-HDE: As the number of simulation for HDE is unpredictable, the time taken for each experiment is varied. In our research, the HDE often switch on the local optimization. The number of simulations is normally between 2700 to 4500 approximately. Thus, the time takes for this method is roughly between 945 and 2100 minutes(between 15.8 hours to 35 hours). Note that the time spent in simulation is nonlinear to the number of simulations. This phenomenon is illustrated in Chapter 3.3.

The Figure 4.7 gives the convergence plots based on DE, HDE, DE with Sobol initialization and Monte Carlo method. From this figure, the convergence of DE with Sobol initialization can be seen to be significantly better than others. It dominates the maximum cost value after the first 100 function evaluations. After 300 function evaluations, the maximum cost value is already much larger than the maximum cost values obtained by all other methods. Note that the analysis in [25] was demanding the global solution within 300 function evaluation or so, due to major impediment on the requirement on the simulation time for each function evaluation. DE with Sobol initialization improves the performance of worst case validation significantly for aerodynamic load cost function f_1, which is of supreme interest to industry [20]. In summary, the Sobol sequence is a good strategy for population initialization for the DE algorithm in this case and builds confidence to be considered as the right candidate for the important problems as in [25]. The Figures 4.8,4.9,4.10 and 4.11 provides the directionalities of the uncertain parameter vector corresponding to the worst-cases obtained using the methods of DE, HDE and DE with Sobol initialization respectively.

The Q_α plot over the mach number based on the optimization methods are given in the Figure 4.12. From this figure, the DE and HDE exceed the constraint twice around the mach number 2. The DE with Sobol initialization exceed the constraints three times around mach number 0.3 and 2. Moreover, the larger area of constraint violation happens with DE with Sobol initialization. All Q_α plots suggests that the VEGA launch vehicle may suffer over dynamic load under these uncertain conditions. Especially for the condition obtained by DE with Sobol initialization, the larger area of violation (prolonged existence of performance limit violation) may be considered as higher risk for the launch vehicle in it's P80 phase.

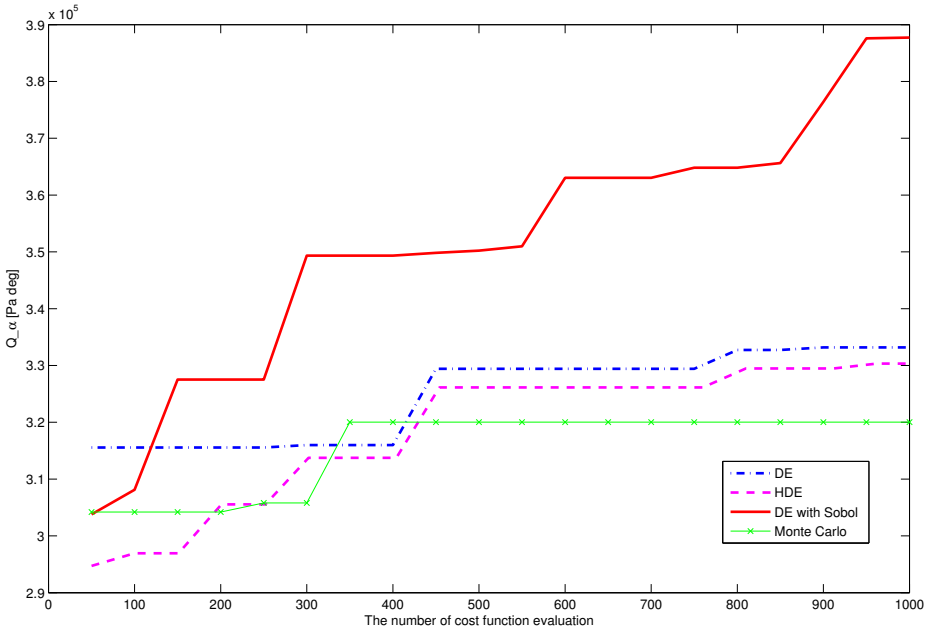
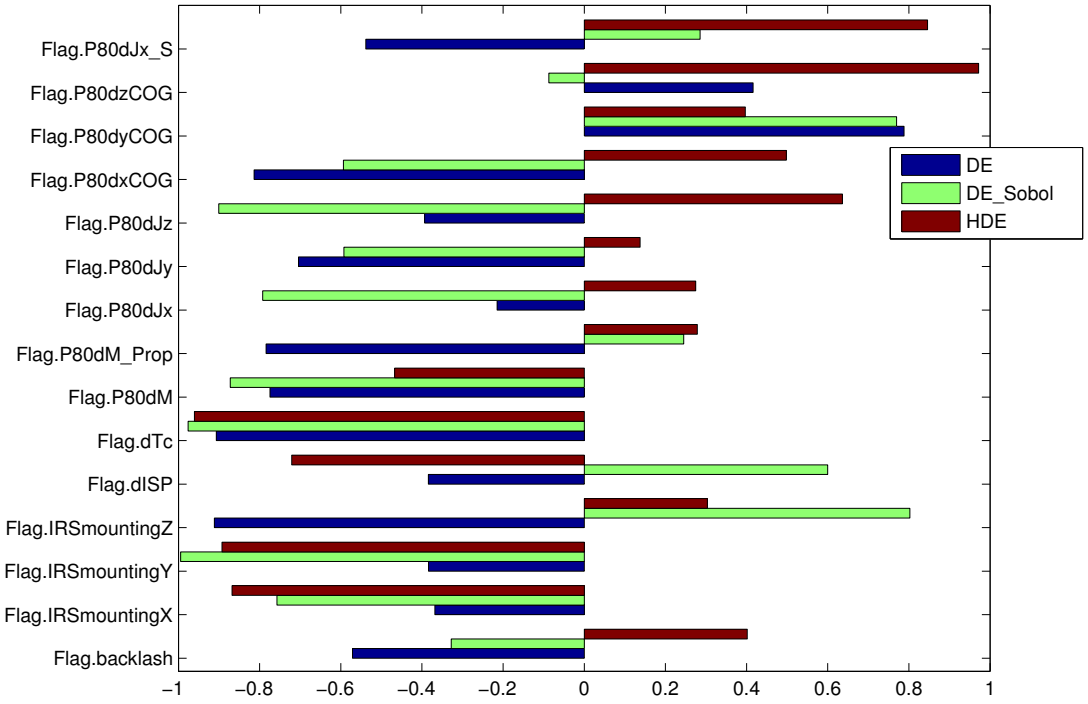


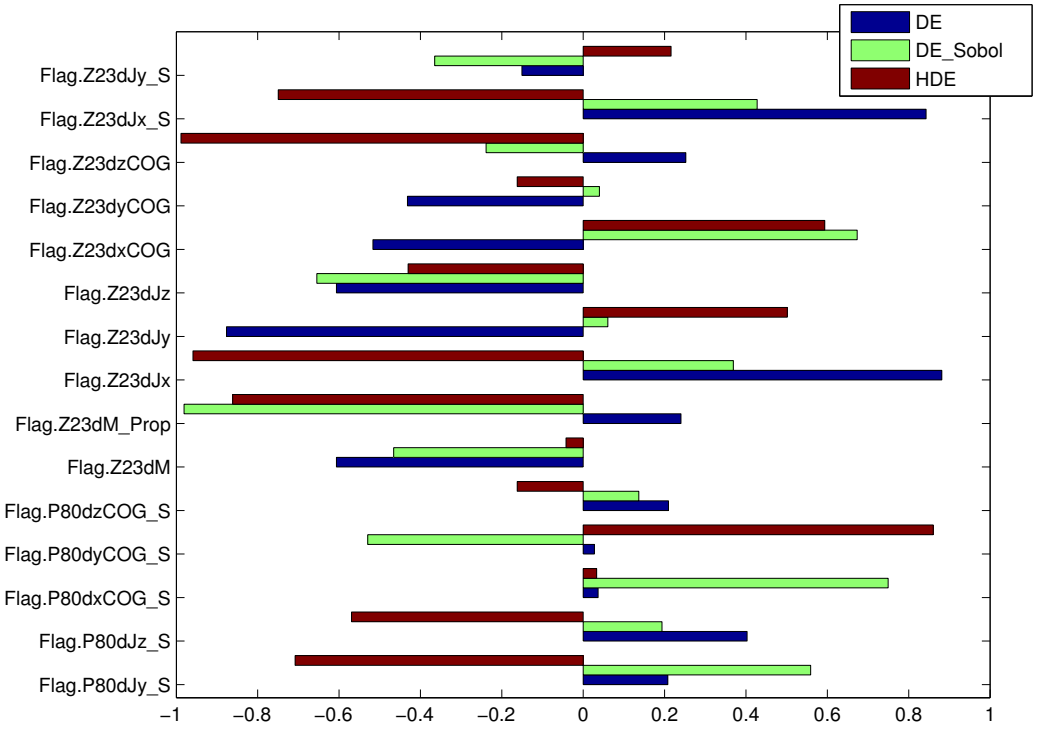
FIGURE 4.7: The convergence plot of Q_α

4.4 Short Summary

Introducing the quasi-Monte Carlo method is our main contribution in this chapter. Our experiment demonstrate that the quasi-Monte Carlo method in general outperform the Monte Carlo method in robustness analysis based on the six objective functions. Moreover, by introducing the quasi-Monte Carlo sequence to initialize the population for Differential Evolution, the DE obtain a much better optimization result over the DE without involving quasi-Monte Carlo sequence. These results show that the quasi-Monte Carlo method is a ideal candidate approach for robustness analysis indecently or cooperated with other optimization scheme dependently. From other point of view, a good distribution of samples is vital for robustness analysis.

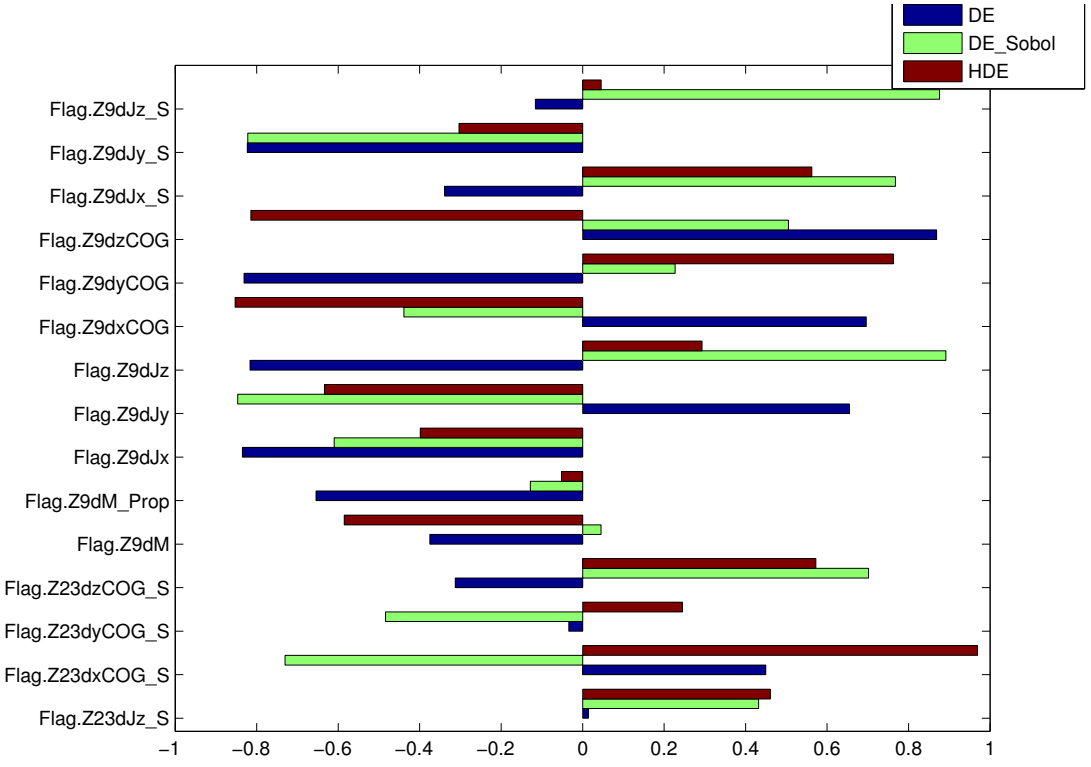


(A) Variables from 1 to 15 under different worst-case scenario

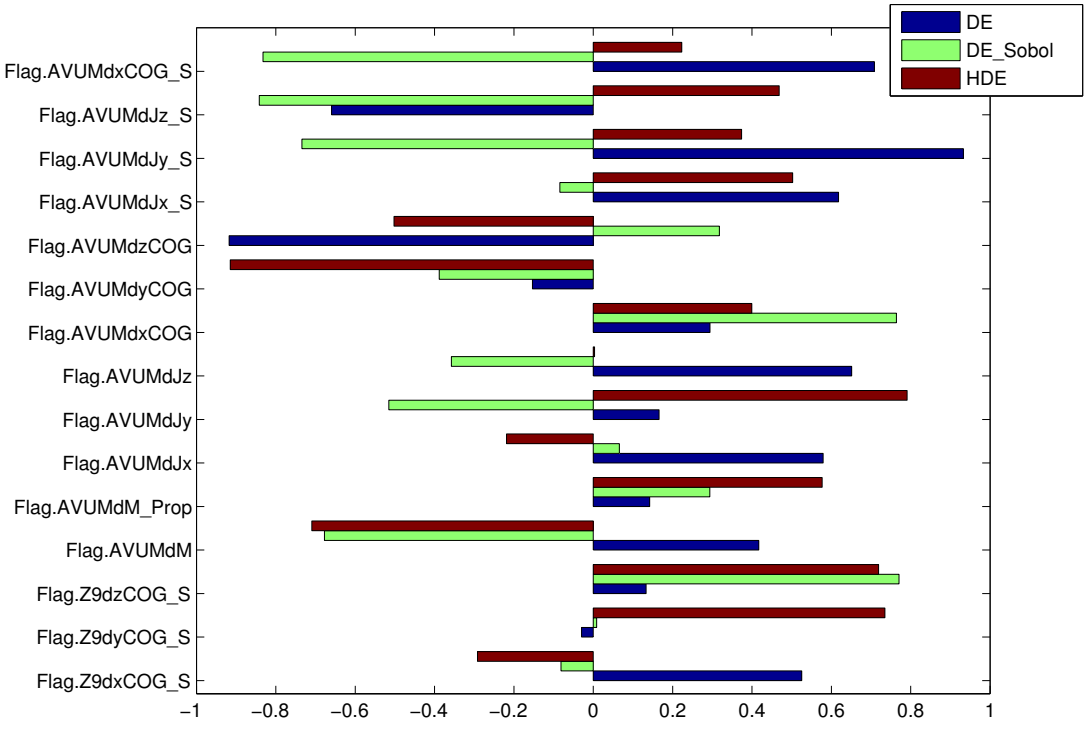


(B) Variables from 16 to 30 under different worst-case scenario

FIGURE 4.8: The worst case uncertainties (1 -30) for f_1

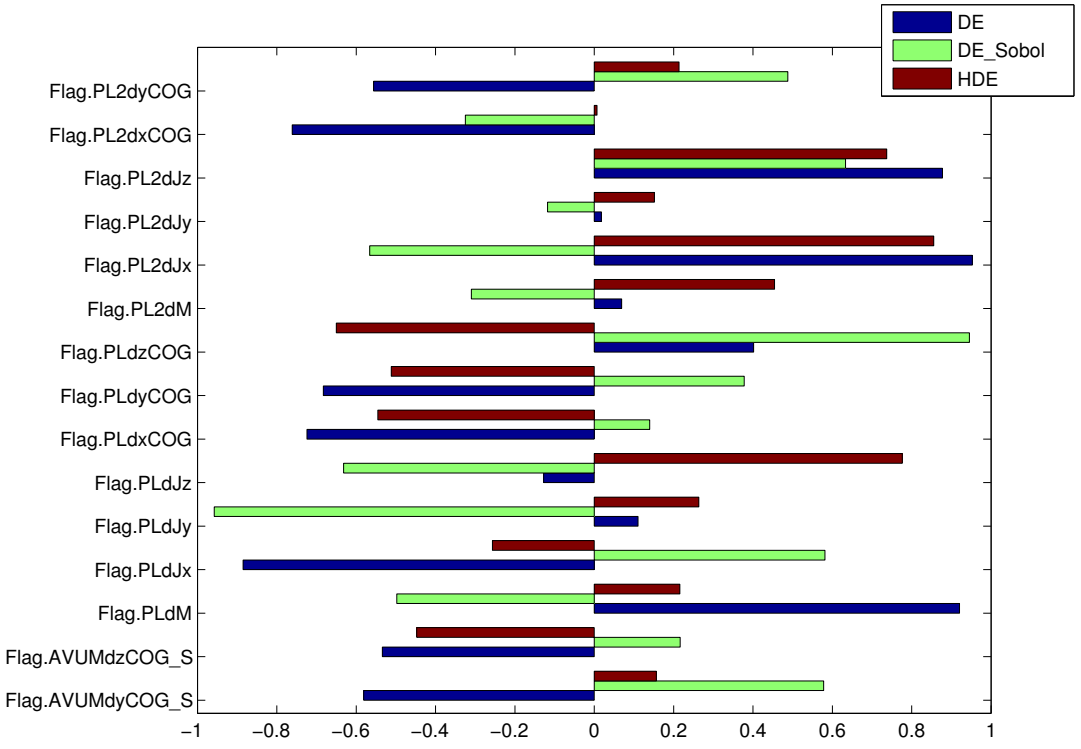


(A) Variables from 31 to 45 under different worst-case scenario

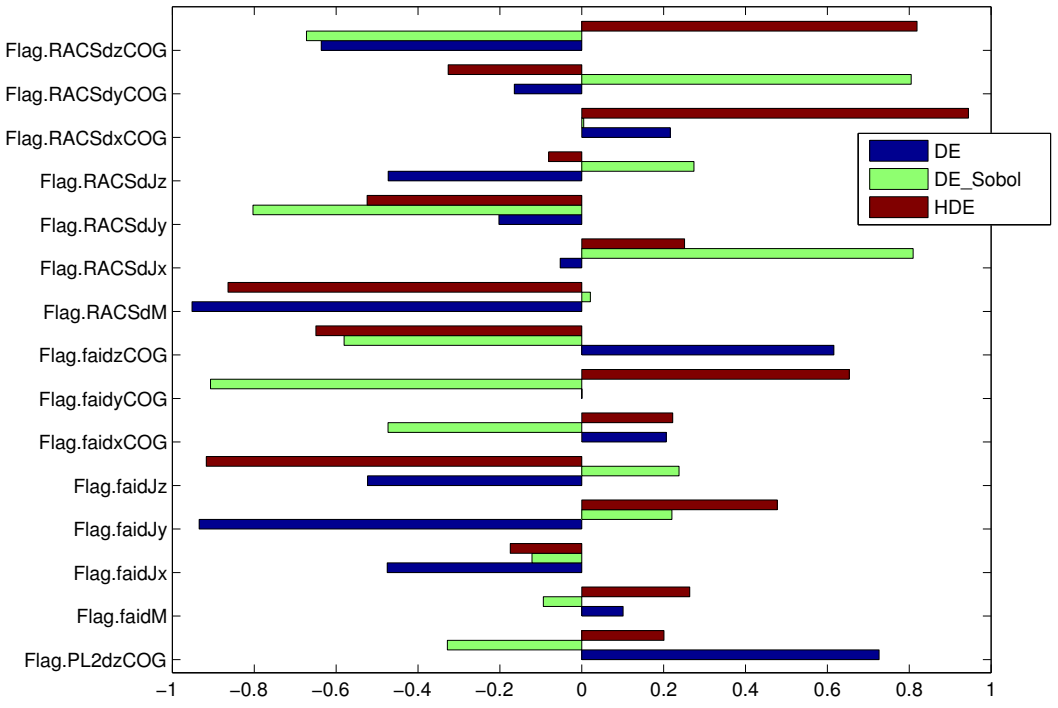


(B) Variables from 46 to 60 under different worst-case scenario

FIGURE 4.9: The worst case uncertainties (31 -60) for f_1

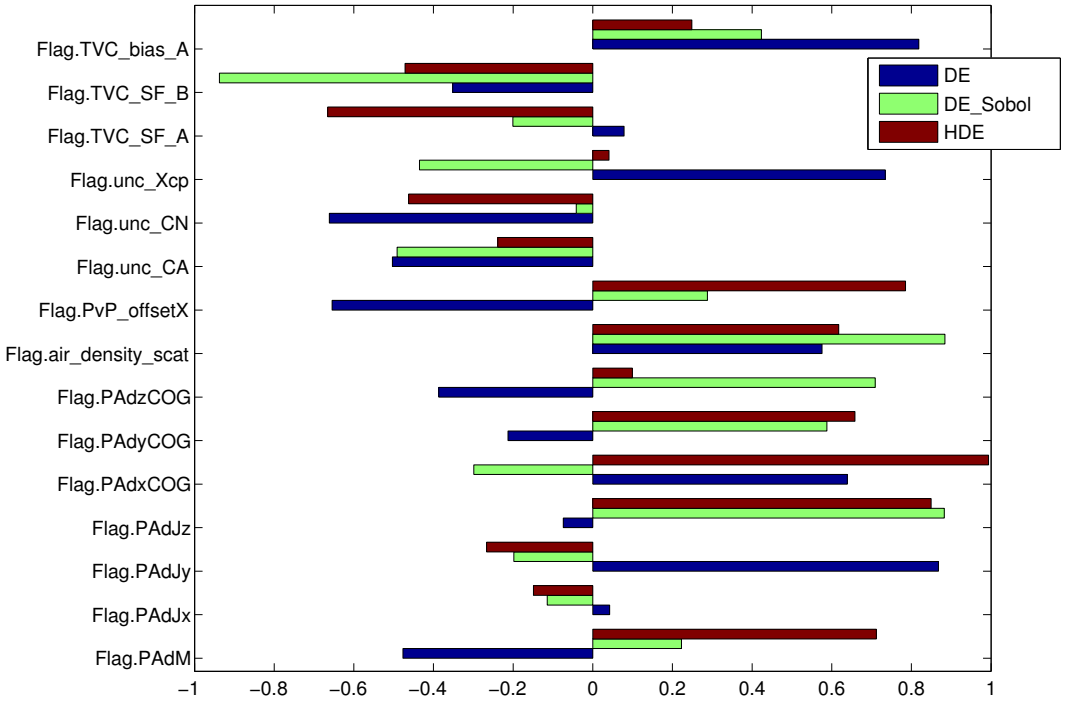


(A) Variables from 61 to 75 under different worst-case scenario

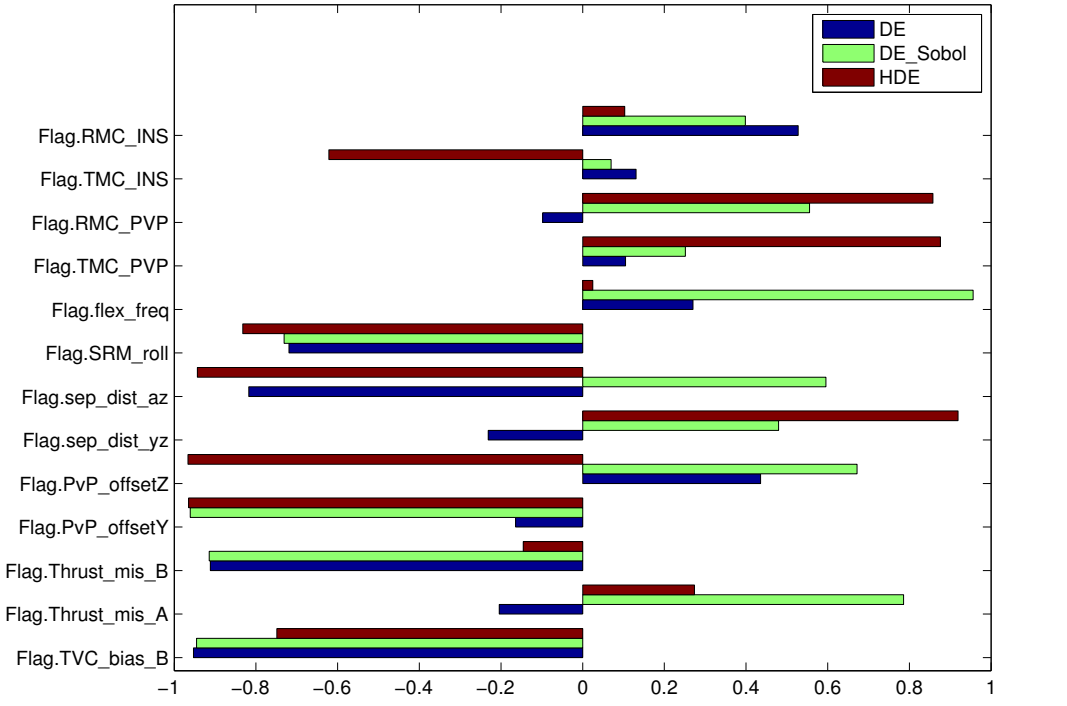


(B) Variables from 76 to 90 under different worst-case scenario

FIGURE 4.10: The worst case uncertainties (61 -90) for f_1



(A) Variables from 91 to 105 under different worst-case scenario



(B) Variables from 106 to 118 under different worst-case scenario

FIGURE 4.11: The worst case uncertainties (91 -118) for f_1

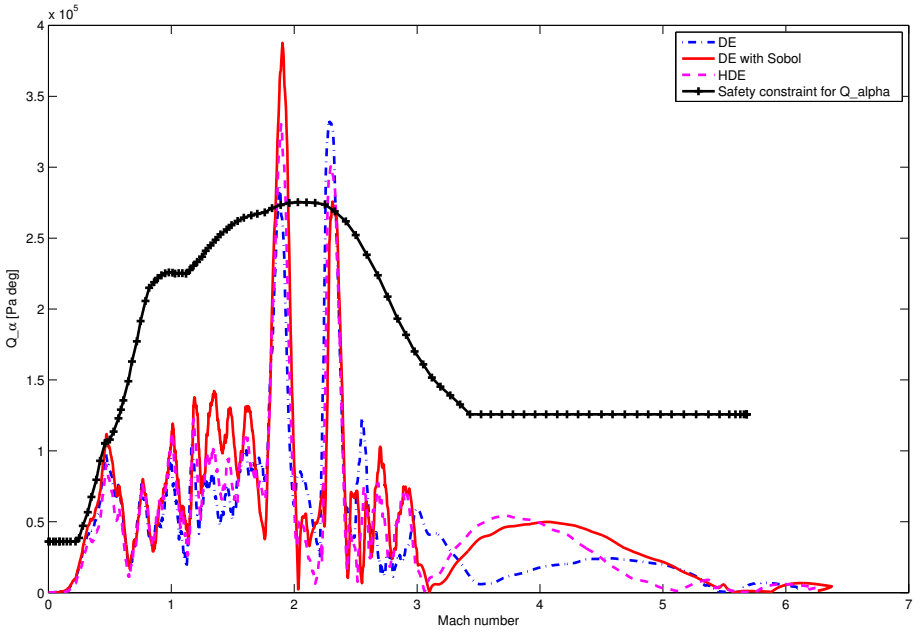


FIGURE 4.12: The Q_α plot over the mach number

Chapter 5

Gaussian Process based global optimization

In previous chapter, it was demonstrated that an efficient sampling strategy is potentially able to improve the performance of worst case validation for both sampling based methods and optimization based methods.

For a global optimization in terms of VEGA model, the number of function evaluations is limited by power of computation capability. The term 'efficient' is already explained above that it refers to obtaining an target worst case value in a lower computation cost or obtaining a larger worst case value by carrying out a certain number of function evaluations. Although an efficient sampling strategy is able to improve the worst-case value, it still may be failure to explore the entire uncertainty space deeply with a limited number of function evaluation. To address this issue, an challenge but a feasible way is to construct a objective value surface by collecting the date from a small number of evaluations. The surface needs to be able to be observed and accessed with little computational effort. There are many types of emulator could approximate the objective value surface, such as polynomial expression. But in this chapter, we introduce the Gaussian Process to build a statistic model. The reason to employ a statistic model to describe the objective value surface is due the high nonlinearity of its surface. One important feature for a statistic model is that the observations at a give condition of uncertainties would behave as a gaussian distribution. The key to using the statistic model to approximate the objective surface and optimize the objective problem is to

balance the error of model with the exploiting of the global optimum value. This chapter is going to show how we overcome the trade-off problem.The general concept of Gaussian Process is detailed in Chapter 2.4.

5.1 Model training procedure

Before carrying out the GP model training process, four most important matters needs to be discussed and investigated. The four matters are the choices of 'training points','mean function', 'covariance function' and 'prior information'. The quality of the GP model is depended on these four key factors.

5.1.1 Training points

The chosen of training points in Gaussian process is also important. These training points are not selected randomly. We hope the observations in terms of the training points can provide good information of uncertainty space. Thus, the training points ideally have a uniform distribution over the sampling space. In our experiment, we propose the quasi-Monte Carlo to generate the training points and obtain the corresponding observations. There is question rose here: how many training points are sufficient enough for the Gaussian process? It is easy to understand that the Gaussian process with more training points inevitable generate a higher quality meta-model. However, from the equation 2.46, the drawback of large number of training point is that the complexity and computational burden for the calculation of posterior distribution are significantly increasing. So our question becomes what is the trade-off number of training points for Gaussian process? To unswear this question, we investigate the trade-off point experimentally: Calculate the quality of the meta-model and computation time with different number of training points. The details and example of the strategy are given in the test case later.

5.1.2 Inference for mean function

A mean function in the Gaussian process is a scaler function and used to predict the mean value of the noise free output over the uncertainty space. The inference for mean

function is not exactly inference for the entire mean function 2.60, but is the inference for the regression functions $\phi(x)$ in equation 2.49. The reason that we call the 'inference for mean function' is that $\phi(x)$ is one of the important components in the mean function. The set of all regression functions are denoted as $\Phi(X)$ and $\Phi(X)$ are unknown hyperparameters. The inference of $\Phi(X)$ is normally based on some knowledge of the objective model.

5.1.3 Inference for covariance function

A covariance function is applied to calculate the covariance between two inputs X and X^*. For posterior distribution, it measure the covariance between the training points and predicted points. A good covariance function is able to improve the quality of meta-model. But the inference normally is depended on the knowledge of the objective model.

5.1.4 Inference for prior

The inference for prior ideally needs the expert knowledge of the objective model. In the paper [16, 17], the authors discuss some strategies and issues for the prior distribution. Normally, we expect the prior has zero mean function. So we week the prior distribution that is only depended on the unknown hyperparameter Σ_p^2 in our experiment. In function-space view, the Σ_p^2 is also written as σ_f in the covariance function and described as 'signal variance'.

5.1.5 Model training

As the training points should have a good coverage for the uncertain space, the Sobol sequence is employed in the test case for a uniformly distribution. But we have no idea how many training points are needed for the Gaussian process in the first place. An investigation for a proper number of training points with respect to an acceptable quality of the meta-model. So the first step for the investigation is to define the measurement of quality for the meta-model. The measurement that is introduced in the experiment

is called: prediction residuals. It is derived from mean squared error(MSE).
The prediction residuals is defined as followed:

$$Q_p(f, f_*) = 1 - \frac{\sum_{i=1}^{n}(f_i - f_{*i})^2}{\sum_{i=1}^{n}(\bar{f} - f_i)^2} \tag{5.1}$$

where f is the true observed value, f_* is the predicted value, n is the number of testing points, $\bar{f}$ is the empirical mean. Ideally, the value of Q_p is in range of [0,1] (will be explained later)and small value of Q_p stands for a poor quality model while large value means a good model. Normally, an acceptable quality of meta-model requires a value of Q_p is greater than 0.8. A large value of Q_p indicates that: (1) the most of predicted value f_{*i} is very closed to the true observed value f_i, so the noise in the model is very small; (2) although the noise is even considerable that makes the difference between the f_{*i} and f_i, the effects of noise compared to the variance of observation are very small and negligible. The advantage of the measurement of Q_p is that it neutralizes the effects of noise σ_n^2 and suggests a good mean and variance of the meta-model. This measurement is actually kind of 'vague'. It cannot measure the effect of noise so that the direct mapping from x_* to f_* may be very poor contrast the true observation. So a good quality measured by Q_p only means that the predicted mean and variance quite match the true observations.
Unlike the measurement of prediction residuals, the mean squared error is more straightforward and 'precisely'. The definition of mean squared error is given followed:

$$Q_m(f, f_*) = \frac{1}{n}\sum_{i=1}^{n}(f_*i - f_i)^2 \tag{5.2}$$

From the definition of MSE, the effects of noise is directly considered in the measurement of different between the predicted value and true observation. The quality of meta-model obtained by MSE presents the quality of the mapping quality from inputs to outputs. Thus, the value of Q_m could be any value that is equal or greater than zero. A small value of Q_m means a good quality of meta-model. However, as the value of Q_m could be non-negative value, it is not easy to compare the qualities obtained by Q_p and Q_m and not straightforward to make any statement about the quality according to the value of Q_m. For instance, the determinant of threshold of good quality with respect to the value

of Q_m is difficult. To address these issues, we derive the MSN to a modified version:

$$\begin{aligned} Q_{mn}(f, f_*) &= \frac{1}{n}\frac{\sum_{i=1}^{n}(f_{*i} - f_i)^2}{Var(f)} \\ &= \frac{\sum_{i=1}^{n}(f_{*i} - f_i)^2}{\sum_{i=1}^{n}(f_i - \bar{f})^2} \\ &= 1 - Q_p \end{aligned} \tag{5.3}$$

From this equation, assuming the meta-model is properly modeled, then the difference between f_{*i} and f_i is only dominated by the effects of noise. Thus, the variance of observations is guaranteed to be no smaller than the expected effects of noise. Then the range of Q_{mn} for a reasonable quality is [0, 1]. A smaller value of Q_{mn} also means a better quality of meta-model. Of course, a little improper modeled f_* with small effects of noise also are able to makes a small bias around f_i for each f_{*i} so that the value of Q_{mn} is till than 1. But we believe it is a reasonable quality as well, especially the subject model is a complex and nonlinear model. There is no necessary to worry about the effects of noise that makes the bias of predicted value with respect to the true observation if the value of Q_{mn} is smaller than 1. Because the true subjective model itself may preserve the nature of noise from the external. In Gaussian process, the hyperparameter σ_n^2 is exactly the parameter that is used to describe the effects of noise.

To generate the meta-model by Gaussian process, the inferences for mean and covariance function are also needed. In our experiments, the projection function $\phi(x)$ is a linear function and the covariance function is one of the *Mátern class* of covariance function. The *Mátern class* of covariance function is a production of exponential and a polynomial expression:

$$\begin{aligned} k(x, x^*) = \sigma_f^2 f_d(r_d) e^{(-r_d)},\ r_d = \sqrt{\frac{3}{\ell}(x - x^*)^T(x - x^*)} \\ ,\ f_d(r_d) = 1 + r_d \end{aligned} \tag{5.4}$$

The choices of projection function and covariance function are based on our tests and experience about the model. It is not easy to make a good inferences for the functions without any tests. There is also no guarantee that our choices for these functions are the best ones.

To make the Gaussian process is applicable in practice, the determination of the hyperparameters are necessary. The choices of hyperparamters that are involved in the mean

and covariance functions are variety and effect the quality of the meta-model. In our experiments, we propose the method called *Bayesian model selection* [19] to determine the hyperparameters. As we know the marginal distribution in the posterior distribution is a normalized constant and independent to the posterior distribution of weight w. If consider that the hyperparameters are also the conditions to the posterior distribution, then the marginal likelihood is extended to:

$$p(y|X,\theta) = \int p(y|X,\theta,w)p(w,\theta)\mathrm{d}w \tag{5.5}$$

where θ is the vector of hyperparameters. As the training points X and the corresponding observations y in the equation is known, the hyperparameters θ are able to be determined by *Maximum Likelihood* (ML) method. To address the integration issue in this equation, the numerical method is employed. However, the hyperparameters are not determined by directly implementing ML to solve the marginal likelihood in practice. Instead, the marginal like is extended to the log marginal likelihood:

$$\log p(y|X,\theta) = -\frac{1}{2}y^T K_y^{-1} y - \frac{1}{2}\log|K_y| - \frac{n}{2}\log 2\pi \tag{5.6}$$

where n is the number of training points, $K_y = K_f + \sigma_n^2 I$ is the covariance matrix for the observation y with noise, K_f is the covariance matrix of the noise-free latent f. Then, by maximizing the log marginal likelihood, the hyperparameters are able to be determined. The optimization algorithm implemented in the maximizing the log marginal likelihood is a local optimization algorithm and is called *fmincon* in Matlab. The details, such as the mathematic expression of this function, can be found in the Chapter 4.

5.2 Gaussian Process based optimization

The Gaussian statistic model is built to make a predictive distribution for a given point x. Suppose we are not sure about the value of $y(x)$ that is observation of objective function y. As $y(x)$ is not random but expensive to obtain by direct evaluating the objective function y, an alternative way then is required to identify whether a given point x makes any improvements of objective value by studying its distribution property. Thus, the key of Gaussian Process based optimization is to define a proper 'improvement function'

by involving mean and variance from the predictor to substitute the direct observing of $y(x)$ and find the expect improvement in objective value for any given point x.

There are many kinds of improvement functions proposed by other researchers [56–63]. In our research, an improvement function called 'Expected loss' which is introduced in the paper [56] is applied.

Supports there is only one objective function evaluation left to end the optimization and return the minimal objective value y_{min}. Denote the final observation from objective function evaluation is y under the condition of uncertainty x and the minimal objective value from previous evaluations $\eta = \min y_0$. Then the loss function is defined in a very simply way:

$$\lambda(y) = \begin{cases} y; & y < \eta \\ \eta; & y \geq \eta \end{cases}$$

So the minimal observation $\lambda = \min(y, \eta)$. However, as we mentioned above that the last observation $y(x)$ is unknown. We would like to find a condition of x that is assigned in the objective function and obtain an smaller observation y over the η in return without any objective function evaluation. By predicting the distribution from GP model over y, the expected loss under a selected condition of x can be defined as

$$\begin{aligned} V(x|I_0) &= \int_S \lambda(y) p(y|I_0) \mathrm{d}y \\ &= \eta \int_{\eta}^{\infty} N(y; m, C) \mathrm{d}y + \int_{-\infty}^{\eta} y N(y; m, C) \mathrm{d}y \\ &= \eta + (m - \eta)\Phi(\eta; m, C) - C N(\eta; m, C) \end{aligned} \tag{5.7}$$

where I_0 is context of the prior knowledge of mean and covariance functions; m denotes $m(y|I_0)$ and is the mean value of the selected point x from the prediction of GP model; C denotes $C(y|I_0)$ and is the covariance of the selected point x from the prediction of GP model; ϕ denotes the Gaussian cumulative distribution function and N denotes the Gaussian probability sensitivity function. This expected loss equation parentally is independent with the observation value y in future and only dependent on the mean and covariance value at the selected point x. Unlike the expected improvement functions defined in [57, 63] only consider the distributed area that is smaller than the current minimal value η, this expected loss function consider both effects from the distributed values who are below the current minimal value η and who are greater than η although

both improvement functions are quite similar. This expected loss function is also called Bayesian expected loss.

According to this expected loss function, the condition of uncertainty x with the lowest expected loss value is the global minimal. The value of expected loss will decrease while the mean value m decreases and covariance value C increases. This expected loss function provides both exploitation and exploration due to the property of m and C. The global minimal condition then will be assigned to the objective function $y(x)$ for the last function evaluation chance and be checked whether the last observation value y would be smaller than previous minimal value η.

The minimization problem is shifted from minimizing the objective function $y(x)$ to minimizing the expected loss function $V(x|I_0)$. To minimize the expected loss, the minimal value can be explored by investigating the gradient and Hessian matrix in terms of its analytics expression. But in practice with high dimensional problem, we use Monte Carlo method to carry out the minimization problem in this research.

If there are more than one objective function evaluations left, then minimization of expected loss are carried out before each function evaluation. However, we do not know what effects would be brought in when each decision is made of global optimal x after each function evaluation. Thus, we would like to update the information to the expected loss V after each function evaluation. In paper [56], the author updated the context information I_j (where $j = 1, 2, ..., n$, n is the total number of function evaluations) and the Gaussian pdf(probability density function) information to give the probability when the previous global minimal condition x_{j-1} is happened. These informations are cumulated to expected loss function $V(x|I_j)$for every function evaluations. In our research, the context information I_j is unchanged during the evaluations. So this information will not be updated. However, as the remaining total number of function evaluations would be very small (otherwise there is not necessary to use the Gaussian Process based optimization), we will update the observations and condition of uncertainties after each evaluation to improve the quality of the Gaussian statistic model because we won't use a large number of train set to obtain the Gaussian Process model by considering the expensive cost for each function evaluation.

The pseudocode for the Gaussian Process based optimization is given below. Although we take the effects from each decision made of each evaluation into account by following the concept in [56], we still need to exam that whether a different start point can

	Pseudocode of GP based optimization
1	**Begin**
2	**Input**: the training points x_0 and the corresponding observation y_0
3	**Find** $y_{best} = min(y_0)$. $x_{best} = arg\min_{x_0}(y_0)$
4	$X \leftarrow x_0,\ Y \leftarrow y_0$
5	$p\_dis = 1$
6	**Repeat**
7	**Find posterior :** $\theta \leftarrow arg\max_\theta(-\log p(\theta\|X,Y))$
8	$x_{new\_best} \leftarrow arg\max_x(EV(x\|X,Y,I,\theta) \times p_{dis})$
9	$y_{new\_best} \leftarrow y(x_{new\_best})$
10	**if** $y_{new\_best} < y_{best}$
11	**then** $y_{best} = y_{new\_best},\ x_{best} = x_{new\_best}$
12	$X \leftarrow [X\|x_{new\_best}],\ Y \leftarrow [Y\|y_{new\_best}]$
13	$p\_dis = p\_dis * N(y_{new\_best}; m(y_{new\_best}; \|I), C(y_{new\_best}\|I))$
14	**Until**(stopping criteria satisfied)
15	**return** x_{best} and y_{best}
16	**End**

affect the results of optimization (especially applied this method to VEGA launch vehicle model). This examination means the optimization scheme with multiply function evaluations left could be tracked into a local minimum according to the concerns from other authors, such as [57, 63] . Under this circumstance, if the number of function evaluation left for optimization is fixed to a small number, then a tradeoff decision is need to balance the number of start points with number of steps to forward in the GP based optimization. In order to improve the efficiency of optimization, normally the start points are selected by hand.

The scheme of the GP based optimization with multiply function evaluations left is described in the Figure 5.1 below.

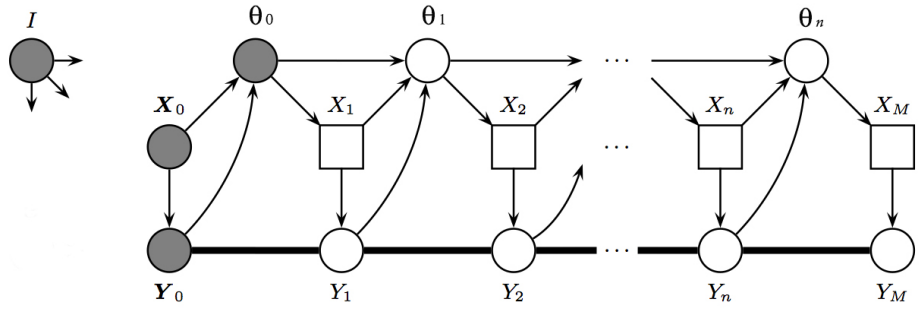


FIGURE 5.1: A GP based optimization scheme. The shaded nodes mean the informations are known.

5.3 VEGA launch objective function

In this thesis, only the objective function $f_1 : \max(Q_\alpha)$ is applied to GP based optimization. The reason is that the procedure to train a GP model for different objective function is same but may need much experience to understand basic statistic features of observations for each objective function in order to provide proper prior information and choose a suitable mean and covariance functions. The understanding process of each objective function is time consuming and may need assistance from experts' review.

The definition of the objective function f_1 can be found in previous chapter. Because there are 118 uncertainties involved in the function evaluations, training a good quality of GP model with a limited number of training points could be a great challenge. The relationship between the quality of GP model and the number of training potions is under survey in this thesis.

With all the 118 uncertainties , the complexity to get a good quality meta-model is very high. Although normally we would like to accept a statistic model with quality level that is over 0.8, in practices, the expected quality level for a complex system is much lower than this index. Then the GP model is said as an acceptable quality if an expected objective goal is achieved.

In this case, the inferences and sampling strategy are implemented as the methods and inferences mentioned above , the quality and time cost versus the different number of training points are given in the Figure 7.9 followed. After 1000 sampling points, the quality of the statistic model is improved very little by increasing the number of sampling points; on the other hand, the time cost for the model training is increased significantly. From this figure, it indicates that simply increasing the number of training points is not a effective solution to improve the quality of the meta-model. In our experiment, we tried 6000 training point for model generation. The quality we obtained is 0.72. This quality value is still lower than the minimum acceptable quality value. Moreover, the model training process takes approximate 2 hours to complete. There is also no way to keep increasing the number of training points due to the capability of the hardware. The Matlab would display an error message if the number of training points exceeds 8000 to show that the Matlab is out of memory. In this case, the meta-model applied to optimization is trained based on 500 training points.

With a selected number of training points, the next step for GP based optimization is to determine the number of function evaluations remaining for the optimization. In

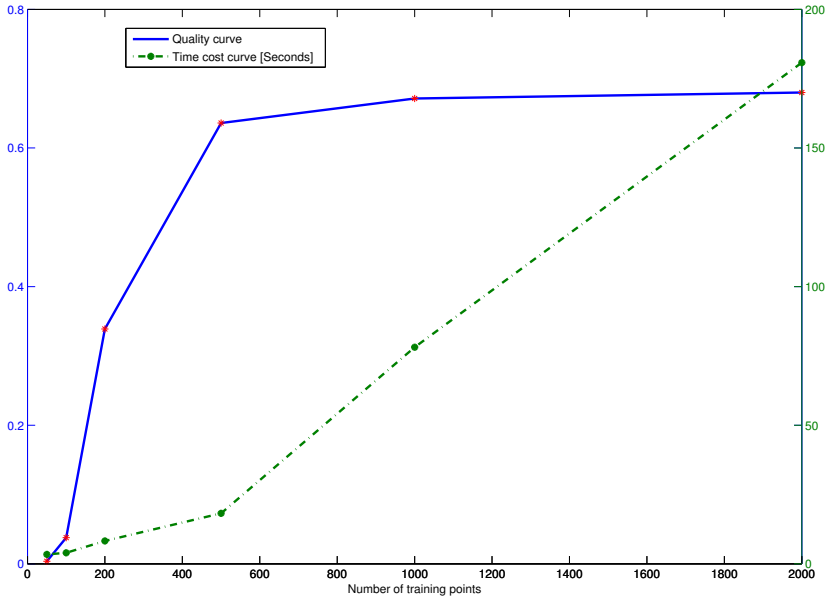


FIGURE 5.2: The quality and time cost for model training based on different number of training points

Chapter 4, the number of function evaluations for most of optimization cases are 1000. The purpose of the GP based optimization is to find a global minimum with a much small number of function evaluations. As we already have 500 function evaluations to generate a training data set, the number of function evaluations remaining for GP based optimization should not be a large value. In our research, we fixed this number for remaining evaluations is 50.

Next step, we need to make a trade-off decision about the number of start points and the number of function evaluations remaining under each start point. For example, if we decide to employ 5 start points, there are 10 function evaluations left for an optimization under each start point. Then the total number of function evaluations is $5 \times 10 = 50$. In this thesis, we propose four cases given below and would like to observe whether the number of start points would heavily affects results of optimization:

(1) 1 start point with 50 evaluations remaining under the single start point.

(2) 2 start points with 25 evaluations left under each start point.

(3) 5 start points with 10 evaluations left under each start point.

(4) 10 start points with 5 evaluations left under each start point.

For these start points, they are randomly generated by uniform random distribution. The Table 5.1 followed is the optimization results based on the four cases. Note that the

optimization is designed to find a global minimum. However, the objective function is designed to look for a maximum objective value. To address the issues , we add a negative sign to the objective function. Then the objective function is transferred to find a global minimum value. The tables and figures followed displays a positive value as we reassign a negative sign when we produce a post-process to generate the table and figures. From the table, the maximum objective values obtained under each case are content

Tests cases	Worst-case (Maximum dynamic loads)
Case 1 (1 start point)	3.2814×10^5
Case 2 (2 start points)	3.2123×10^5
Case 3 (5 start points)	3.3039×10^5
Case 4 (10 start points)	3.2755×10^5

TABLE 5.1: Worst-case validation GP model with different number of start points for objective function f_1(Maximum Dynamic loads)

because they achieve a larger objective value over the value obtained by sampling based methods with Monte Carlo, Faure sequence and Halton sequence sampling strategies. Although the maximum objective value obtain based GP optimization is a smaller than the maximum objective value obtained by sampling based method with Sobol sequence and by optimization method DE with Sobol initialization, it has little difference with the maximum objective values obtained based on DE and HDE in previous chapter. Thus, the GP based optimization in terms of the objective function f_1(which is the maximum dynamic load. Details can be found in Chapter 4.2.1) is demonstrated as a powerful tool that it needs much fewer computation cost but achieves a good optimization result.

Computation Cost

Because the number of additional simulation only set to 50, the average computation time only needs 14 minutes. Of cause, to make the GP model, it requires another 500 simulations. But both GP based optimization and GP based sensitivity analysis can depend on it.

A study of the choice of trade-off decision mentioned above is also needed by analyzing the evolution process for each optimization scheme under different start points and different number of evaluations of the four cases. Between each two cases in the table, the difference of the worst-case values is very small. The maximum difference is approximated 3%. The worst case results infer that the worst-case value based on a

different number of start won't be too much different. However, the results still cannot explain whether the GP optimization could be tracked into a local optimal value. The investigation of trade-off decision is given as followed.

For Case 1, the Figure 5.3 below describe the observations evaluated based on the selected condition of uncertainties after each expected loss function calculated during the evolution process of optimization. Meanwhile, the convergence of the entire optimization is also provided. In this figure, the red line is the convergence plot of the optimization

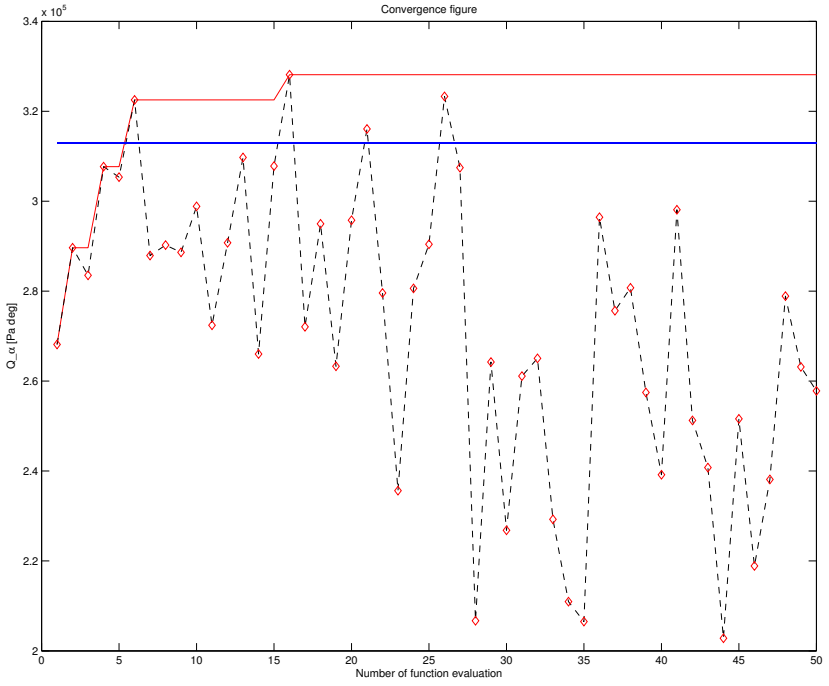


FIGURE 5.3: The convergence plot of the GP based optimization for Case 1

process. The black dash line with red diamond denotes the minimal observed value obtained after each expected loss function is calculated. The blue line is the maximum observed value from training date points. The global optimum value is converged after 15 function evaluations. A very fast convergence of optimum value happened in the first 7 evaluations. After 27 function evaluations, the new observations seem to be traced into a range of small value and are all smaller than the minimal value from the training date set.

Generally speaking, the optimization with such a few number of evaluations is very successfully by achieved a relatively large objective value although a fast convergence in the first 15 function evaluations makes the rest of evaluations with no efforts. The fast

convergence and the maximum objectives obtained in other 3 cases may infer that a large number of evaluations from a single start point is not quite necessary.

For Case 2, the convergence plots and the minimal observation after each expected loss calculated are given in the Figure 5.4 followed. In the figure, each optimization scheme

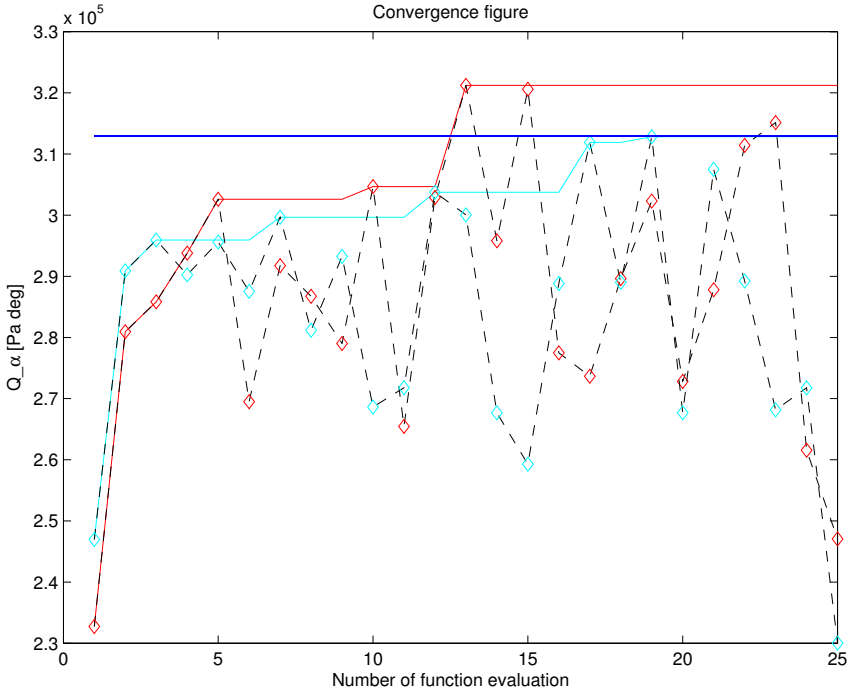


FIGURE 5.4: The convergence plots of the GP based optimization for Case 2

with different start points is coordinated with a specified color. The red solid line and the cyan solid line are the convergence plots of the GP based optimization scheme from different start points. The black dash line with red diamond and black dash line with cyan diamond denote the observations based on the chosen condition of uncertainties after each expected loss calculated from different start points. The blue solid line is the maximum observation from the training data set. The convergence of red solid line has a smaller observation at the start point. but it is converged very fast and reaches its global maximum value after 13 evaluations. In stead, the convergence of cyan solid line has a larger observed value at the start point. The convergence rate is also fast in the first a few evaluations. But its global maximum objective value is a little bit smaller than the maximum observation from the training data set. This means the GP based optimization from the cyan start point makes no effort at all. Thus, we can say that if a global maximum from a start point is smaller than the maximum observation from

training data set, then this optimization is failure of any improvement. Fortunately, the optimization from the red start points achieve a much larger objective value than the minimal observation from the training data set. For both optimization scheme from both different start points, the observations at both start points are very small but the maximums are converged very fast in the first 5 evaluations and quickly reach the global maximum around 15 evaluations. The observed values obtained after 20 evaluation are dropped very fast and suggest that a large number of evaluations from a single start points may be unnecessary.This phenomenon is quite similar as the Case 1. It seems that the maximum observation over the entire optimization could be obtained very fast but traced into this maximum value in the rest of evaluations and very difficult to jump out this 'local maximum'. This phenomenon is more obvious in Case 1.

For Case 3, there are only 10 evaluations for each optimization scheme with different 5 start points. According to the phenomenon summarized in previous Case 1 and Case 2, the convergence rates for each optimization scheme should be very fast and may only left a few number of observations without making any improvement in objective value after the global optimum is achieved. The convergence plots and all observations are given in the Figure 5.5 below. In this figure, one specific color represents an optimization

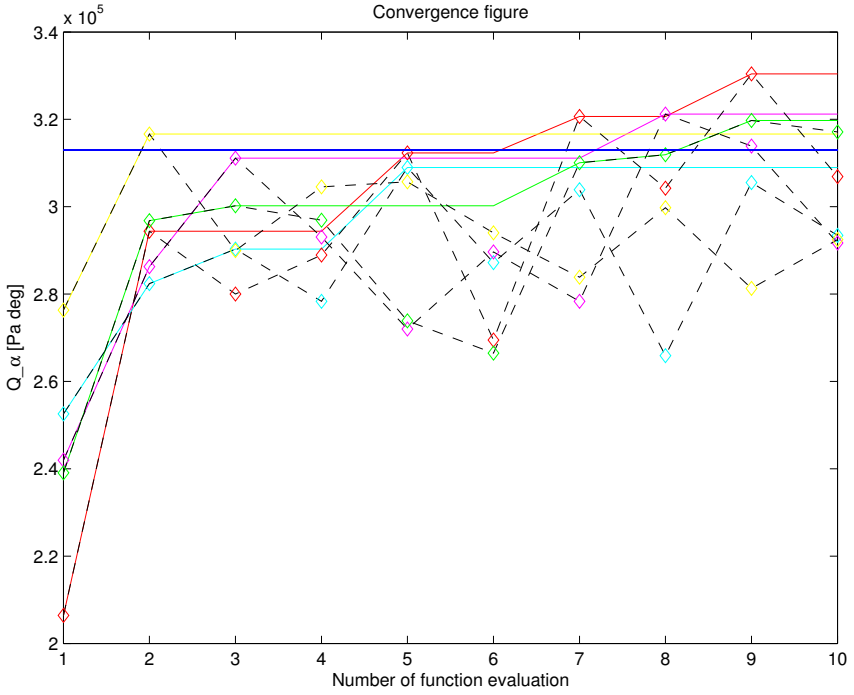
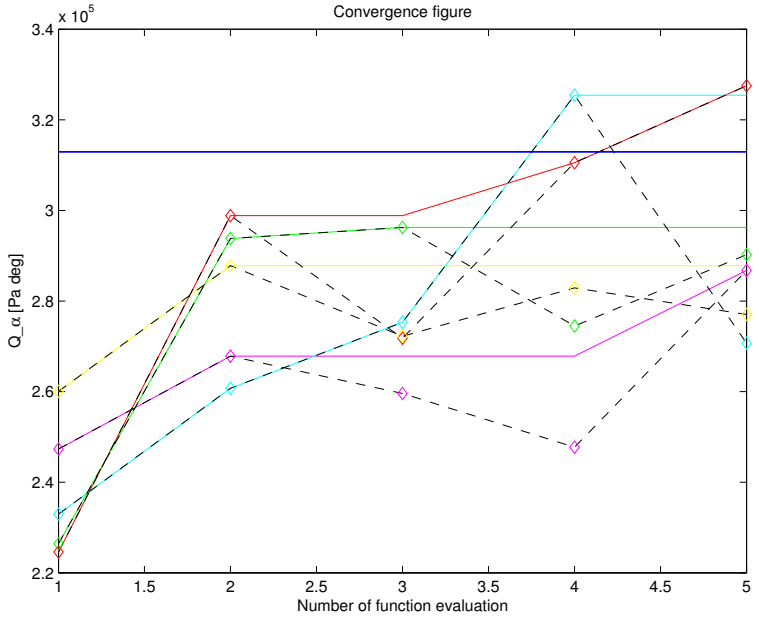


FIGURE 5.5: The convergence plots of the GP based optimization for Case 3

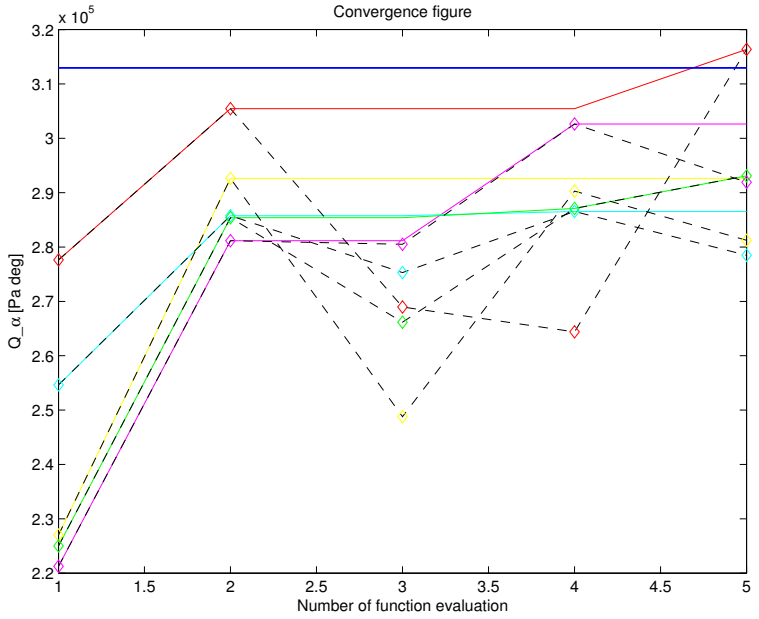
scheme as well. The solid lines with red, purple, cyan, yellow and green color are convergence plots in terms of different start points. The black dash lines with red, purple, cyan, yellow and green diamonds are observations after each expected loss calculated for every optimization schemes. In this case, four convergences behave as our expectation mentioned above that only a few number of observations left without making any improvement after the global maximum is achieved. The reason is the total number of function evaluations is small for each optimization scheme. So it is not obvious to tell whether the optimization schemes are tracked into a local maximum. The only exception is that the convergence with yellow color reaches its maximum value at the second observation. The optimization process is high likely traced into a local optimum point in the very beginning and unable to jump out the point even after another 8 evaluations. Overall, the convergence with red solid line obtains the largest objective value and keeps finding a new larger value during the optimization scheme although it has the smallest objective value at the start point.

For Case 4, it is difficult to put all 10 convergence plots in a single figure with friendly interface. Thus, these 10 convergence plots are equally assigned to two subfigures. Each subfigure has 5 plots. In this case, there is only 5 evaluations for each optimization scheme. By observing from previous three convergence figures, we can find that only one optimization scheme achieve a observed objective value larger than the maximum observation from the training data set after 5 function evaluations. The rest of optimization schemes obtain smaller observations over the maximum observation from the training data set after 5 evaluations. Thus, the only 5 function evaluations could possibly make the 10 optimization scheme obtain a smaller maximum value over the maximum observation from the training data set. The convergence figure is given in the Figure 5.6 below.

In each subfigure, one specific color represents an optimization scheme. But for every specific color, there are two optimization schemes represented by it. But these two optimization schemes are complete different and with different start points. The reason is that we do not have that many of colors that could distinguish every optimization schemes. In the two subfigures, the convergence plots are presented by red, purple, yellow, green and cyan solid line. The observations in each optimization are given with black dash line and diamond in specific color. In subfigure (A), the convergence rates for every optimization scheme are fast. Nevertheless, three of them obtain a much smaller



(A) Convergence plots for the first 5 optimization schemes



(B) Convergence plots for the other 5 optimization schemes

FIGURE 5.6: The convergence plots of GP based optimization for Case 4

observed objective value after 5 function evaluations than the maximum objective value in training data set. Similarly as the subfigure (A), four of the five optimization schemes obtain a observed objective value below the maximum observation from training data set after 5 evaluations in subfigure (B). In total, only 3 over the 10 optimization schemes achieve the maximum objective value after 5 evaluations that are larger than the maximum observation from training data set. In summary, the fast convergence rates for every optimization schemes lead no evidence to show that the optimization schemes are tracked into a local optimum. However, most of the optimization schemes are failure to make any improvement in objective value over the training data set. A highly possible reason is that the number of evaluation left for optimization is too small.

In conclusion based on the analysis for the four test cases, a GP based optimization scheme could be tracked into a local optimum with high possibility. Multiple start points for the GP based optimization are recommended in order to increase the chances to achieved a better optimum value. On the other hand, the number of evaluations left for the optimization doesn't have to be as large as possible. In this test example, the number of evaluations left for optimization is ideally between 10 to 20.

5.4 Short Summary

In this chapter, our main contribution is implement the Gaussian Process to train a meta-model for a probabilistic optimization scheme. We demonstrate that the GP based optimization can save a significant computation cost without loss too much accuracy. The Gaussian Process is seldom applied in aerospace industry. As our research subject is a complex nonlinear model what is treated as a 'black-box' model, GP is a ideal candidate for mapping the relationship between input and out with a probabilistic sense. This nature is used for probabilistic based optimization and the results also reveal its advantage.

Chapter 6

Variance based sensitivity analysis

Variance-based sensitive analysis is wildly used for a complicated nonlinear problem. In most cases, the nonlinear problem is not able to expressed by one or a few linear/nonlinear algebraic functions, so the differential-based calculation is unable to implement to identify the driven parameters. Variance based sensitivity, on the other hand, is a powerful tool to screen out the driven parameters for a nonlinear problem. The method of variance-based sensitivity analysis used in our research is called analysis of variance(ANOVA); the principle of ANOVA based sensitive analysis is already given in Chapter 2.5. In this chapter, the analysis results are presented in sequel.

6.1 Test Examples

6.1.1 Subjective model

The test subject is the VEGA launch vehicle. For a good distribution of the uncertainties by quasi-Monte Carlo simulation, we know from Chapter 2 that the ideal minimum number of points for a large dimensional case is 2^n, where the n denotes the number of dimensions. However, if all uncertain parameters are applied to the variance based sensitivity analysis, the recommended number of points is 2^{118}. The computation cost based on the recommended number of points is unaffordable for us. The approximate

time to carry out 2^{118} simulations is beyond measure. To verify whether or not the Sobol indices comprise a good tool for sensitivity analysis for the VEGA launch vehicle, we select 15 uncertainties out of 118 uncertainties.

The selected uncertainties are given in the following Table 6.1.

Category	**Variable name in VEGACONTROL**	**Description**
IRS	Flag.IRSmountingX	IRS Mounting Error w.r.t X Body Axis
	Flag.IRSmountingY	IRS Mounting Error w.r.t Y Body Axis
	Flag.IRSmountingZ	IRS Mounting Error w.r.t Z Body Axis
Thrust Parameters	Flag.dISP	Stage 1 impulse scattering
	Flag.dTc	Scattering on time burn
	Flag.SRM_roll	Scattering on P80 Roll Torque
Aerodynamics	Flag.disp_CA	Dispersion on Stage 1 Axial coefficient
	Flag.unc_CA	Uncertainty on Stage 1 Axial coefficient
	Flag.disp_CN	Dispersion on Stage 1 Normal Coefficient
	Flag.unc_CN	Uncertainty on Stage 1 Normal Coefficient
	Flag.disp_Xcp	Dispersion on Stage 1 Xcp
	Flag.unc_Xcp	Uncertainty on Stage 1 Xcp
Atmosphere	Flag.air_density_scat	Atmosphere Density
wind	Flag.h_wind	Synthetic wind gust altitude
	Flag.azimuth_wind_angle	Wind azimuth direction

TABLE 6.1: The description, category and variable name in VEGACONTROL of the 15 selected uncertainties

The 15 uncertainties are not selected at random, but based on our understanding of the VEGA launch vehicle. We believe that some of these uncertainties may potentially make a significant contribution to the output responses. According to the sensitivity analysis steps discussed in a previous section, the first step is to determine the uncertainty distribution. In the test case, the mean value(or nominal value) of the uncertainties are 0, the normalized upper bound is 1, and the lower bound is -1. These values are provided by an expert on the VEGA launch vehicle. The second step is to generate the sampling points. The sampling points in the test case are generated using the Sobol sequence. In order to obtain a good coverage, the number of sampling points is configured to be 2^{15}. This is the maximum number of evaluations that we can afford in our laboratory. The third step is to assign each condition of uncertainties (or each point) to the simulator of the VEGA launch vehicle and carry out the simulation. It takes approximately 5 months to complete the simulations to obtain the sensitivity indices. The fourth step is to present the uncertainty analysis results.The objective function f_1 in this chapter refers to the maximum dynamic load, which is detailed in Chapter 4.2.1.

6.1.2 Example 1: Maximum dynamic loads

The plot of cumulative distribution function is available in our first test example and is given in the Figure 6.1. This figure shows that the low probability of the value of dynamic loads is smaller than 2×10^5 and greater than 3.4×10^5. Thus, these conditions of uncertainties that make the objective values smaller than 2×10^5 or greater than 3.4×10^5 are considered rare events. At this point, two questions arise for the CDF plots: do we really need that many test points ($2^{15} = 32768$), and what effect does reducing the number of sampling points have on the CDF test plots?

The Figure 6.2 below demonstrates the CDF plots based on 1000,2500,5000,10000,20000 and 32768 sampling points that are generated by Sobol sequence. The colored curves present the CDF plots in terms of different number of sampling points. they are:

- Blue curve: 32768 sampling points
- Blue dash curve: 20000 sampling points
- Black curve: 10000 sampling points
- Black dash curve: 5000 sampling points

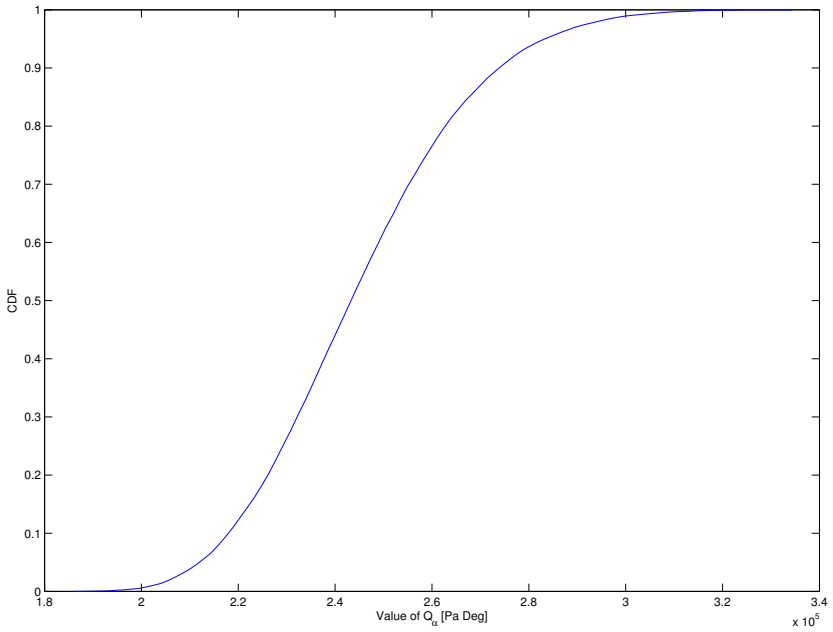


FIGURE 6.1: The cumulative distribution function of f_1 based on 15 uncertainties

- Red dash curve: 2500 sampling points

- Red curve: 1000 sampling points

From this figure, we find that the shapes of each plot are almost identical. The CDF plot based on 1000 sampling points (red solid line) is not that smooth compared to the other CDF plots. This figure, even for a small number of sampling points generated by the Sobol sequence, may also be able to approximately present the features of the distribution of the output response. According to the CDF plots, we will introduce the sensitivity indices based on different sampling sizes later. The final step is to present the sensitivity results, the scatter plots and the sensitivity indices, for our test case.

Computation Cost

As a Sobol sequence is a deterministic sampling method, there is no need to obtain the different number of sampling points respectively. Once the simulation results for the 32768 samples are obtained, the above-mentioned curves can be generated. The time spent on the computation to obtain the objective values for the 32768 samples is approximately 19062 min, or approximately 318 h. This is the reason we can only afford to solve the variance-based sensitivity analysis problem with 15 uncertain parameters.

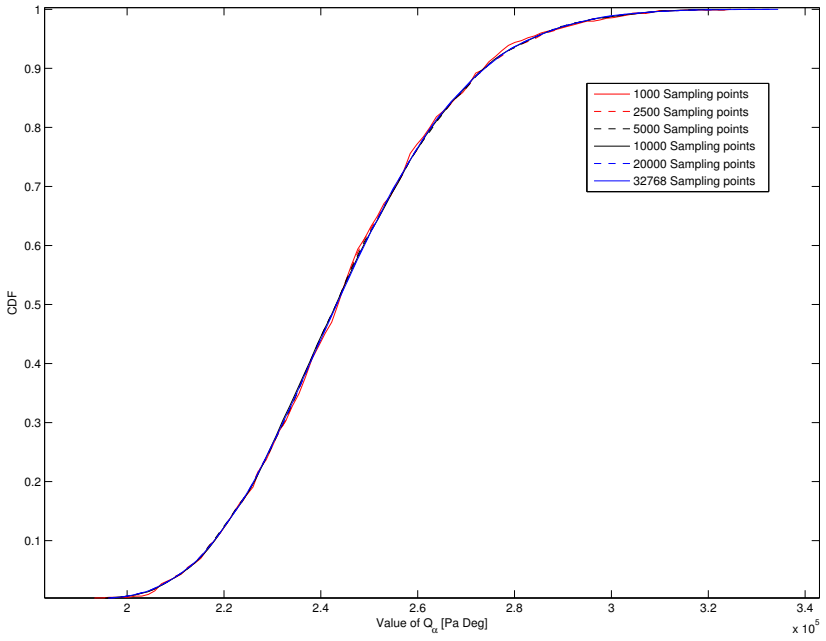
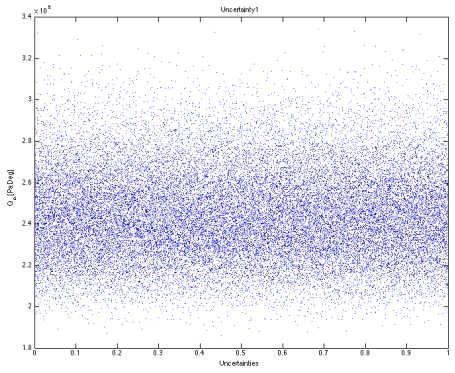


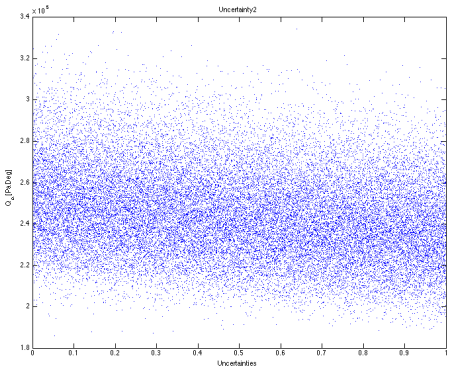
FIGURE 6.2: The cumulative distribution function of f_1 in terms of different sampling points

The scatter plot will reveal the linear or nonlinear relationship between the outputs (objective values) and the inputs (uncertainties). The relationship may give us some idea of how to reconstruct the distribution of the sampling points to improve the worst-case results. If the distribution pattern of the objective values is varied along with varying the uncertainty, then we can note that there is an obvious tendency toward this uncertainty. Figure 6.3, 6.4 and 6.5 shows the scatter plots for every single uncertainty (input) with respect to the maximum value of the dynamic loads (output).

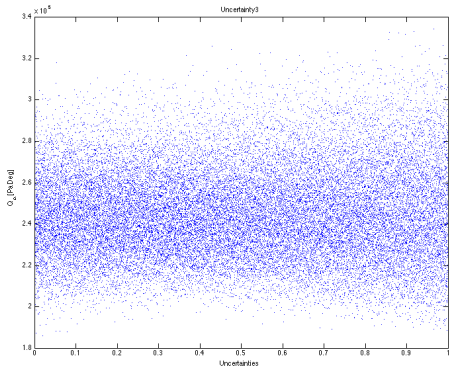
From the figure, the relationships of each uncertainty and the objective value are all nonlinear. However, the plots also suggest the tendency toward an objective value with respect to the uncertainty. Unlike other subplots, the plots (E) in Figure 6.3, (C) in Figure 6.4, and (C) in Figure 6.5 demonstrate an obvious tendency of the objective value when the value of the uncertainty varies. For plot (E) in Figure 6.3, the mean value of f_1 at a given value of the uncertainty increases via the nominal increase of the value of uncertainty. The maximum value then occurs at the value of the uncertainty close to 1. For plot (C) in Figure 6.4, the mean value of f_1 at a given value of the uncertainty decreases when the value of the uncertainty increases. The maximum objective value occurs at the value of uncertainty close to -1. For plot (C) in Figure 6.5, the tendency is



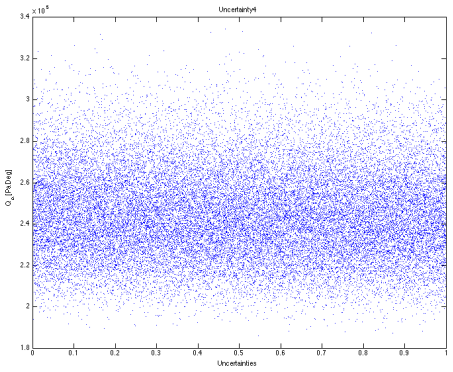
(A) IRSmountingX vs. f_1



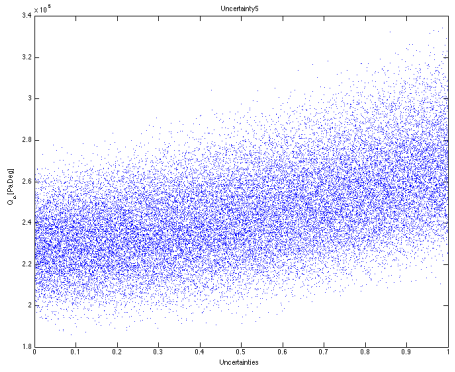
(B) IRSmountingY vs. f_1



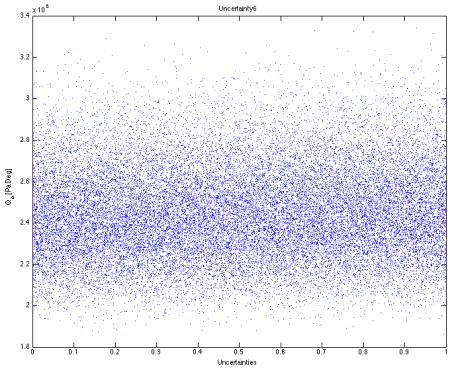
(C) IRSmountingZ vs. f_1



(D) dISP vs. f_1



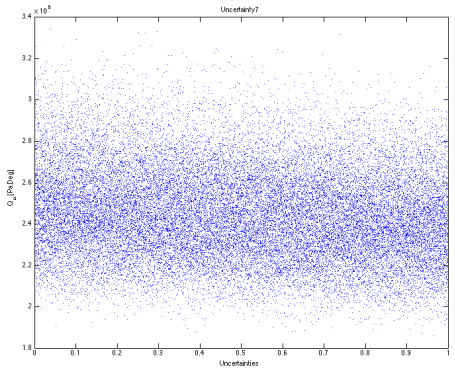
(E) dTc vs. f_1



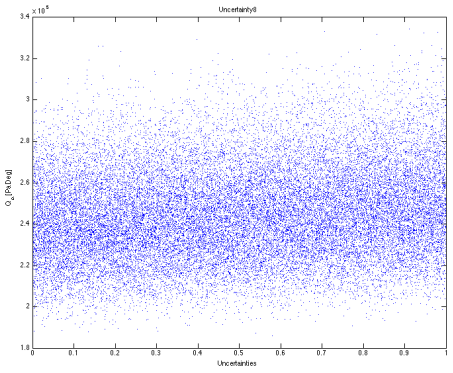
(F) disp_CA vs. f_1

FIGURE 6.3: The Scatter plots of f_1 - Part A

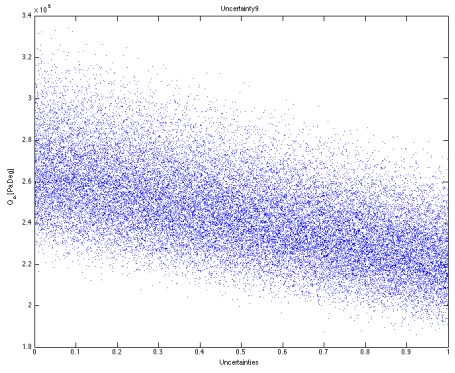
not as linear as plot (D) in Figure 6.3 and plot (C) in Figure 6.4. However, it is easy to determine that the mean and maximum values of f_1 occur when the value of the uncertainty is close to -1. These tendencies can help narrow down the range of distribution for the corresponding uncertainties. These uncertainties can also be generated using the importance sampling strategy, or just simply by using the Monte Carlo method. The dimension of the sampling space based on the Sobol sequence is then reduced.



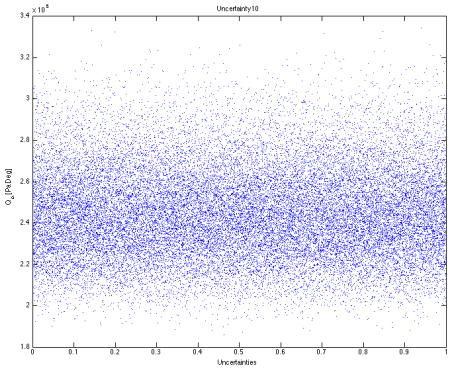
(A) disp_CN vs. f_1



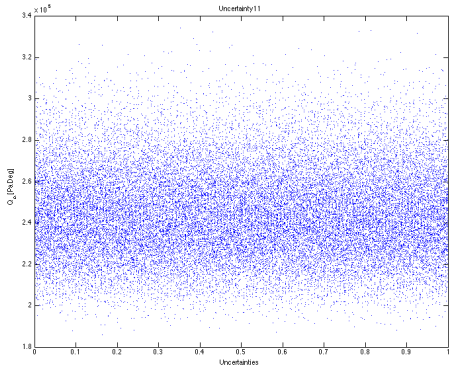
(B) disp_Xcp vs. f_1



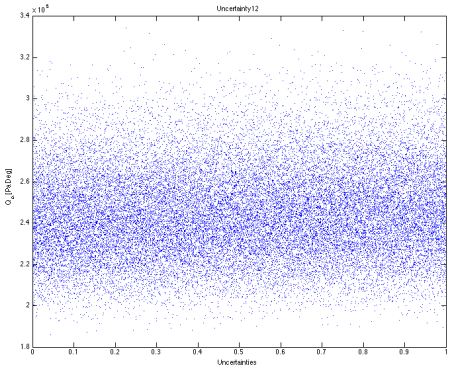
(C) air_density_scat vs. f_1



(D) h_wind vs. f_1



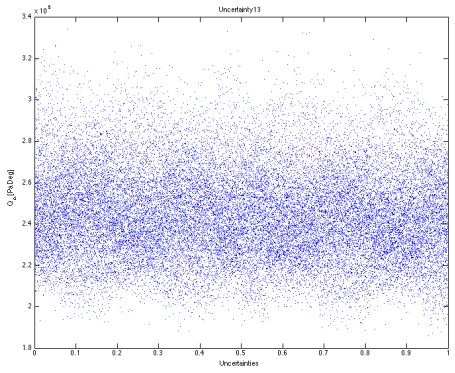
(E) azimuth_wind_angle vs. f_1



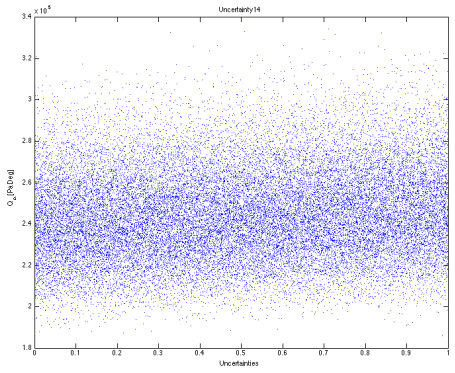
(F) unc_CA vs. f_1

FIGURE 6.4: The Scatter plots of f_1 - Part B

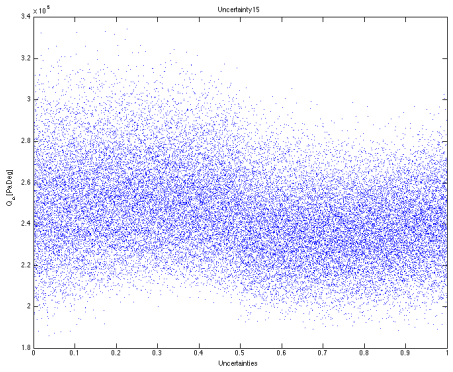
The Sobol indices for the objective function f_1 are given in the Table 6.2 followed. A larger value in this table suggests the greater importance of the corresponding uncertainty. Note that the values of the sensitivity index for the uncertainty'Flag.IRSmountingZ', 'Flag.h_wind' and 'Flag.azimuth_wind_angle' are all zeros. The zero value means these uncertainties make no contribution to the objective values, or that the contribution for the objective function is very small. The sensitivity indices should be non-negative in



(A) unc_CN vs. f_1



(B) unc_Xcp vs. f_1



(C) SRM_roll vs. f_1

FIGURE 6.5: The Scatter plots of f_1 - Part C

Variable Names	First order sensitivity index	Total sensitivity index
Flag.IRSmountingX	3.8427×10^{-4}	4.6756×10^{-4}
Flag.IRSmountingY	0.0865	0.0594
Flag.IRSmountingZ	0	0.0928
Flag.dISP	0.0036	0.0020
Flag.dTc	0.1154	0.3117
Flag.disp_CA	0.0035	4.5879×10^{-4}
Flag.disp_CN	0.0305	0.0257
Flag.disp_Xcp	0.0346	0.0203
Flag.air_density_scat	0.1760	0.3555
Flag.h_wind	0	0
Flag.azimuth_wind_angle	0	0
Flag.unc_CA	0.0089	0.0034
Flag.unc_CN	0.0032	0.0106
Flag.unc_Xcp	4.6509×10^{-5}	0.0116
Flag.SRM_roll	0.1	0. 2519

TABLE 6.2: The first order and total sensitivity index based on f_1

theory. However, a negative value may be obtained in practice. The probable reasons for this could be that the sampling size is not large enough, so that the sampling points do not have good coverage in the uncertainty space. If a negative value was obtained, normally we would treat this value as zero because the effect of the objective function from this uncertainty should be very small [64]. In our experiment, we obtained some negative sensitivity indices, such as the first-order sensitivity indices for 'Flag.IRSmountingZ', but with very low probability.

From the first-order sensitivity index, the uncertainties 'Flag.dTc' , 'Flag.air_density_scat', and 'Flag.SRM_roll' have larger sensitivity values than other uncertainties. These three uncertainties could make a great contribution to the objective value. From the total sensitivity analysis, only these three uncertainties have much larger sensitivity values compared to other uncertainties. These uncertainties are 'Flag.dTc' , 'Flag.air_density_scat', and 'Flag.SRM_roll' . The total sensitivity index of the uncertainty 'Flag.IRSmountingZ' is not quite as large, but it is considerable compared to its first-order sensitivity indices. This indicates that the contribution of the interaction between 'Flag.IRSmountingZ' and other uncertainties is large. Note that the total sensitivity indices should be greater than the first-order sensitivity indices in theory. However, due to the coverage issues of the uncertainty space, the total sensitivity indices for some uncertainties may be smaller than the first-order sensitivity indices in practice. This issue occurs in our experiment as well. The ways of addressing this issue could be (1) to enlarge the sampling size, or (2) change the sampling method. However, to carry out a variance-based sampling sensitivity analysis, the computational cost is very expensive. We can hardly afford another contrast test to redo the simulations and obtain the objective values based on a different sampling size or sampling method. Fortunately, the uncertainties that have larger first-order sensitivity indices than their total sensitivity indices are not considered the important variables, because the values of the first-order and total sensitivity indices are very small.

In summary, the weight of each uncertainty can be ranked depended on the first-order or total sensitivity index. To lower the redundancy of the sampling points, the first method allows for the less important uncertainties to be neglected in the simulation (normally, the less important uncertainties are fixed to the nominal value). Another method of

reducing the redundancy is to re-order the uncertainties sequence when sampling the points; this makes the more important uncertainties have a lower order/dimension in the sequence. This method does not exactly reduce the dimension of the problem. However, since the Sobol sequence gives an excellent distribution for lower dimensions, the reordered sequence has the potential to improve the performance of the worst-case validation. Thus, only the first method is actually able to reduce the dimension of the sampling space, but both methods are called dimension-reduction methods. A better worst-case validation result is expected when a dimension-reduction method is applied.

To understand whether a small number of sampling points can achieve similar sensitivity analysis results, we propose a small number of sampling points using a Sobol sequence to carry out the sensitivity analysis. In this test case, the number of sampling points is 1000. Table 6.3 lists the sensitivity indices based on 1000 sampling points. From this table, although the values of the Sobol indices are slightly different from the

Variable Names	First order sensitivity index	Total sensitivity index
Flag.IRSmountingX	2.0014×10^{-4}	0.0022
Flag.IRSmountingY	0.0700	0.0613
Flag.IRSmountingZ	0	0.0953
Flag.dISP	0.0090	0.0048
Flag.dTc	0.1068	0.3032
Flag.disp_CA	0.0027	0.0011
Flag.disp_CN	0.0203	0.0301
Flag.disp_Xcp	0.0198	0.0166
Flag.air_density_scat	0.1816	0.3294
Flag.h_wind	0	0
Flag.azimuth_wind_angle	0	0
Flag.unc_CA	0.0182	0.0126
Flag.unc_CN	0.0071	0.0182
Flag.unc_Xcp	0.0055	0.0072
Flag.SRM_roll	0.1069	0.2429

TABLE 6.3: The first order and total sensitivity index based on f_1

values given in Table 6.2, the exact same conclusion regarding the sensitivity analysis is made. Therefore, a small number of sampling points using a Sobol sequence is also able to identify the important variables.

However, whether the sensitivity analysis results are good or not is verified by a sampling-based worst-case validation test based on one of the dimension-reduction methods. We

propose use of both dimension-reduction methods mentioned above. The sampling points for the first worst-case validation test is generated by a Sobol sequence with 15 dimensions. The sequence of an uncertainty vector is reordered by the ranks. The ranks of the uncertainties are based on the value of the total sensitivity index. A larger value means a larger rank. The rank of uncertainties could be assigned based on the values of the first-order sensitivity indices in this case because the ranks wound not be vastly different, in contrast to ranks obtained using the total sensitivity indices. Thus, the reordered sequence of uncertainties is given in Table 6.4. For the second-worst-case

Variable Names	Order/ Rank
Flag.IRSmountingX	12
Flag.IRSmountingY	5
Flag.IRSmountingZ	4
Flag.dISP	11
Flag.dTc	2
Flag.disp_CA	13
Flag.disp_CN	6
Flag.disp_Xcp	7
Flag.air_density_scat	1
Flag.h_wind	14
Flag.azimuth_wind_angle	14
Flag.unc_CA	10
Flag.unc_CN	9
Flag.unc_Xcp	8
Flag.SRM_roll	3

TABLE 6.4: Ranks of uncertainties

validation, the first step is to determine which uncertainties are important. This step is actually not straightforward, and the principles to determine the threshold of the important uncertainty using the sensitivity indices is discussed later. To make this step easier, we halve the dimension of the uncertainties in our experiments. Thus, we only keep the uncertainties of rank not greater than 7. The value of those uncertainties with a rank above 7 is fixed to zero. The seven most important uncertainties are distributed by Sobol sequence. For the third-worst-case validation, the nonimportant uncertainties are neglected. In the test case, the uncertainties with a rank above 7 are fixed to zero. From Figures 6.3, 6.3 and 6.3, we know the linear tendency of the uncertainties 'Flag.dTc' and 'Flag.air_density_scat' and the nonlinear tendency of the uncertainty 'SRM_roll'. The upper and lower bounds of these uncertainties can be narrowed down to a small value. Instead of only using the Sobol sequece to generate the sampling points, we propose a

hybrid sampling strategy. These three uncertainties with identical linear and nonlinear tendencies are distributed using a Monte Carlo method and the other three uncertainties are distributed using a Sobol sequence. The Sobol sequence can provide an excellent distribution for the low-dimensional case, and the difference between the Monte Carlo and quasi-Monte Carlo methods will not be significant when the range of distribution is narrowed in this case. The number of sampling points for the three tests of worst-case validation is 1000.

The results of worst-case validation are given in Table 6.5.

Tests	Worst-case (Max Q_α [Pa Deg])
No application of dimension reduction method	3.2469×10^5
Test 1 (uncertainties sequence re-ordered)	3.1522×10^5
Test 2 (Sobol sequence for the 7 uncertainties)	3.3555×10^5
Test 3 (hybrid sampling for the 7 uncertainties)	3.3719×10^5

TABLE 6.5: Worst-case validation based on dimension reduction methods

From this table, the result of the first test indicates that the first dimension-reduction method (reordered sequence of uncertainties) is not able to improve the worst-case validation result. However, the results of the second and third tests demonstrate the power of the second dimension-reduction method. For the second dimension-reduction method, the neglected uncertainties do not affect the maximum objective value at all. Instead, without these redundant uncertainties, a larger objective value is achieved. Because the worst-case validation based on the Sobol sequence is efficient for the 15-dimension problem, a small improvement in maximum objective value is expected when the dimension of the problem is reduced to 7.

The sensitivity analysis successfully measures the weight of each uncertainty and provides the information for dimension reduction. The nonimportant uncertainties are free to be neglected. The value of the first-order or total sensitivity indices of an uncertainty decides whether the uncertainty is important or not. However, the boundary of a sensitivity index to classify the important and nonimportant uncertainty is not easy to determine. One principle by which the boundary can be determined is that in which the result of worst-case validation with the minimum number of uncertainties involved in the simulation is not smaller than the result of the worst-case validation when all uncertainties are involved. Another principle can follow the definitions of superposition

and truncation in terms of effective dimensions that are described in a previous chapter.

Not every sampling method (Halton, Faure, or Monte Carlo methods) is necessary to be applied to the verification process above. The aim is to verify that variance-based sensitivity analysis is applicable to the ELV launch vehicle. However, when all 118 uncertainties are involved in the sensitivity analysis that will be studied later, the four sampling methods need to be applied to the worst-case validation. According to the results of the sensitivity analysis, the variance-based sensitivity analysis successfully sorts out the important uncertainties and removes the redundant variables. Note that, compared to the worst-case validation results presented in Tables 4.1 and 6.5, the results based on the Sobol sequence with 15 uncertainties has a larger objective value (maximum dynamic load) than the result based on the Sobol sequence with all 118 uncertainties. The worst-case validation based on the Sobol sequence under seven uncertainties further improves the objective value. These results indicate there is much redundant information generated when 118 uncertainties are involved in the simulation, which significantly affects the objective value. In addition, the 15 selected uncertainties probably contain most of the dominant variables. Thus, if variance-based sensitivity analysis is applied to the ELV launch vehicle model with 118 uncertainties and some new dominant variables are identified, the sorted uncertainties with a high total sensitivity index value are expected to make the worst-case validation obtain a larger objective value.

6.1.3 Example 2: Drift of velocity on Z axis

For the cost functions f_2, the CDF plots with different number of simulation samples are given in the Figure 6.6 followed. From the figure, CDF plots based on the 1000,2500,5000,10000,20000 and 32768 sampling points are quite similar. It means that even for the a smaller number of sampling points, such as 1000 sampling points, the corresponding objective values also can present the statistic feature of the output distribution. The objective values that is smaller than 2 and greater than 12 are considered as rare events.

To reveal the relationship between inputs and outputs, the scatter plots for the objective function f_2 are given in the Figure 6.7, Figure 6.8 and 6.9 below.

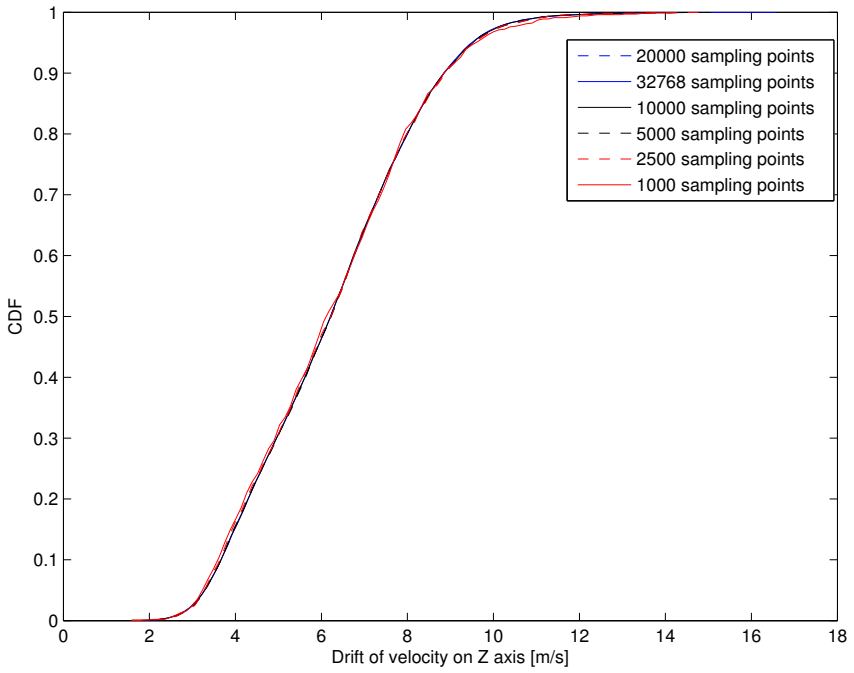
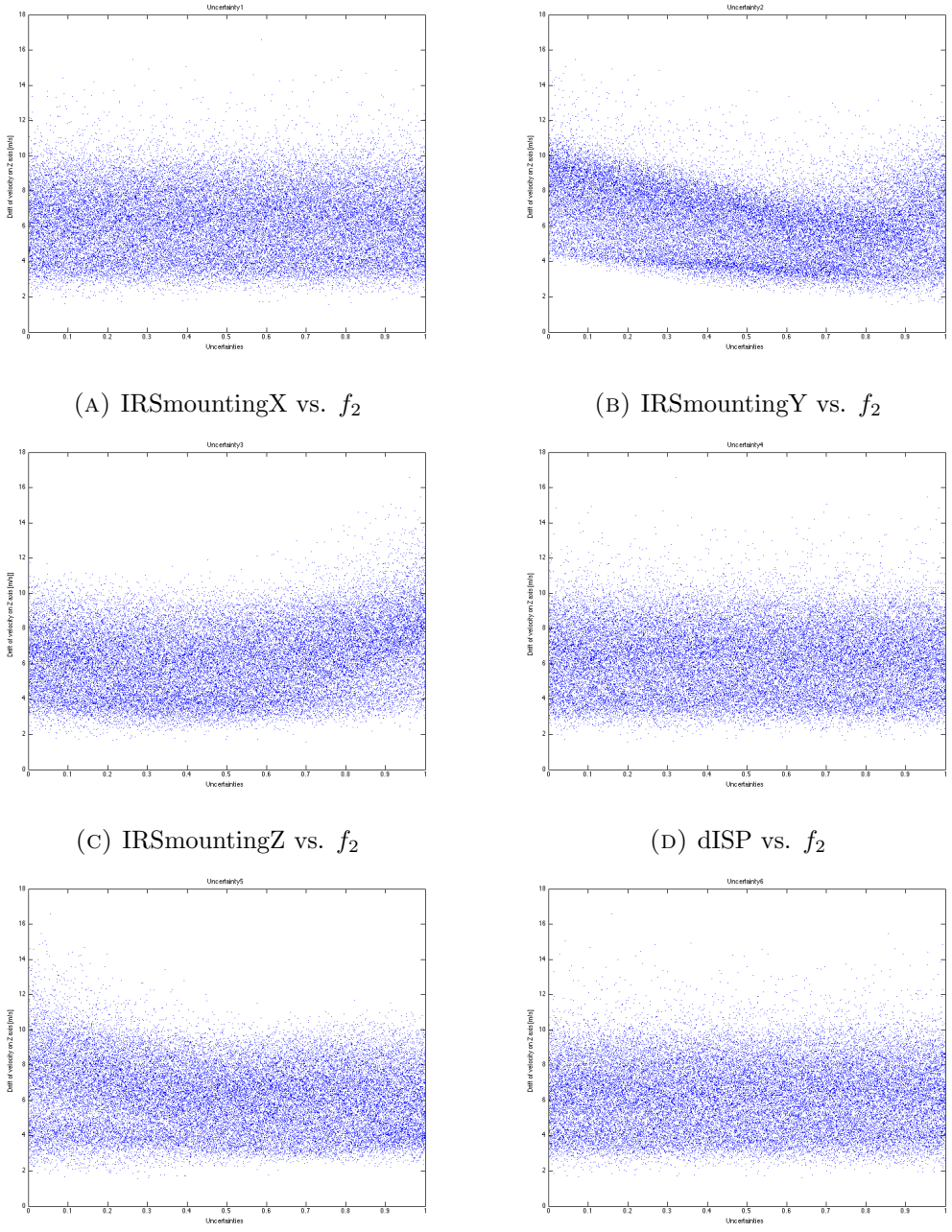


FIGURE 6.6: The cumulative distribution function of f_2 in terms of different sampling points

From this figure, all uncertainties are nonlinear to objective values. For plots (B),(C),(E) in Figure 6.7 and (C) in Figure 6.9, the linear and nonlinear tendencies can be identified, however, the linear tendencies are not strong. The tendencies are useful to narrowed down the range of distribution and help the worst-case validation obtains a larger objective value. From the plot (C), the tendency is nonlinear. The most largest values are scattered in which the value of uncertainty 'Flag.SRM_roll' is slight than its nominal value. From plots (B) and (E), the most largest objective values scattered in which the value of the corresponding uncertainties are around the lower bounds. From the plot (C), the most largest objective values scattered in which the value of uncertainty 'Flag.IRSmountingZ' around its upper bound.

The Sobol indices based on the objective function f_2 are given in the Table 6.6 followed. From the first order sensitivity index, only two uncertainties are considered as the important variables: 'Flag.IRSmountingY' and 'Flag.IRSmountingZ'. But from the total sensitivity index, four uncertainties can be considered as the important variables: 'Flag.IRSmountingY', 'Flag.IRSmountingZ', 'Flag.dTc' and 'Flag.SRM'. Two new uncertainties are identified as the important variable compared the identified important



(A) IRSmountingX vs. f_2

(B) IRSmountingY vs. f_2

(C) IRSmountingZ vs. f_2

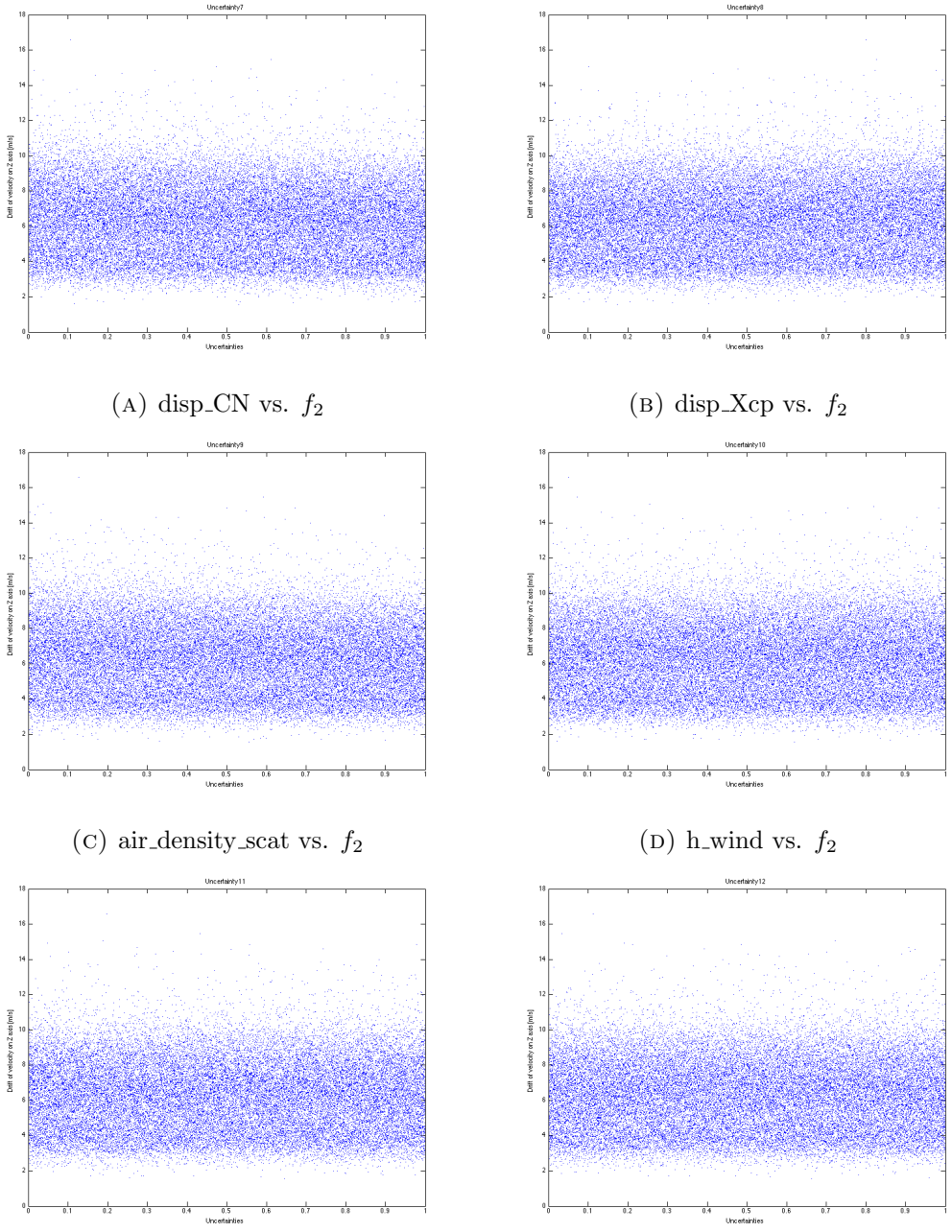
(D) dISP vs. f_2

(E) dTc vs. f_2

(F) disp_CA vs. f_2

FIGURE 6.7: The Scatter plots of f_2 - Part A

variables based on the first order sensitivity index. This suggests that these two uncertainties have strong interaction effect with other uncertainties. According to the total sensitivity index, the ranks to present the importance of an uncertainty are given in the Table 6.7 below. The rank of uncertainties is applied to the worst-case validation to verify whether the sensitivity analysis is successful or not. Similar as the verification process applied to f_1, three worst-case validations with different sampling strategies are



(A) disp_CN vs. f_2

(B) disp_Xcp vs. f_2

(C) air_density_scat vs. f_2

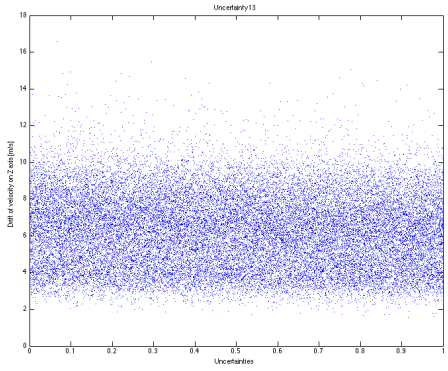
(D) h_wind vs. f_2

(E) azimuth_wind_angle vs. f_2

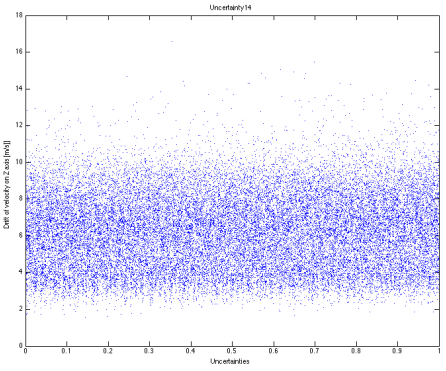
(F) unc_CA vs. f_2

FIGURE 6.8: The Scatter plots of f_2 - Part B

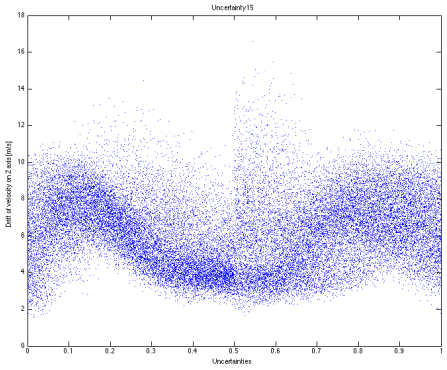
carried out. For the first case of worst-case validation, all of the 15 uncertainties are involved in the simulation with reordered sampling sequence. The reordered sequence is same as the rank of each uncertainty. For the second case of worst-case validation, we sort out 7 most important uncertainties and use the Sobol sequence to distribute the 7 uncertainties. The non-important uncertainties that have the rank above 7 are neglected from the simulation and fixed to zero. For the third case of worst-case validation, the



(A) unc_CN vs. f_2



(B) unc_Xcp vs. f_2



(C) SRM_roll vs. f_2

FIGURE 6.9: The Scatter plots of f_2 - Part C

Variable Names	First order sensitivity index	Total sensitivity index
Flag.IRSmountingX	9.6286×10^{-5}	1.1888×10^{-5}
Flag.IRSmountingY	0.1180	0.3745
Flag.IRSmountingZ	0.1918	0.2945
Flag.dISP	0.0034	0.0059
Flag.dTc	0.0207	0.3677
Flag.disp_CA	0.0012	0.0039
Flag.disp_CN	0.0087	0.0165
Flag.disp_Xcp	4.1661×10^{-4}	0.0073
Flag.air_density_scat	0.0031	0.0188
Flag.h_wind	0	0
Flag.azimuth_wind_angle	0	0
Flag.unc_CA	0.0012	0.0054
Flag.unc_CN	0.0036	0.0297
Flag.unc_Xcp	9.4434×10^{-4}	0.0183
Flag.SRM_roll	0.0206	0.6729

TABLE 6.6: The first order and total sensitivity index based on f_2

Variable Names	Order/ Rank
Flag.IRSmountingX	13
Flag.IRSmountingY	2
Flag.IRSmountingZ	4
Flag.dISP	10
Flag.dTc	3
Flag.disp_CA	12
Flag.disp_CN	8
Flag.disp_Xcp	9
Flag.air_density_scat	6
Flag.h_wind	14
Flag.azimuth_wind_angle	14
Flag.unc_CA	11
Flag.unc_CN	5
Flag.unc_Xcp	7
Flag.SRM_roll	1

TABLE 6.7: Re-ordered sequence

hybrid sampling strategy is employed. As the top four important uncertainties have an identified tendency, the lower and upper bounds of these four uncertainties can be narrowed down and sampled by Monte Carlo method. The remaining three important uncertainties are sampled by Sobol sequence because the Sobol sequence has an excellent distribution for low dimension problem. The non-important uncertainties are fixed to zero as the second test case.

The results of the three test cases are given in the Table 6.8 below. From the table,

Tests	Max drift of velocity(Z axis) [m/s]
No application of dimension reduction method	14.2137
Test 1 (uncertainties sequence re-ordered)	14.8575
Test 2 (Sobol sequence for the 7 uncertainties)	16.5794
Test 3 (hybrid sampling for the 7 uncertainties)	16.7033

TABLE 6.8: Worst-case validation based on dimension reduction methods for objective function f_2

all of three test cases obtain a larger maximum objective value compared the result of worst-case validation when none dimension reduction method applied. Especially the dimension reduction method that neglects the none important uncertainties achieve a much better performance according to the results of test 2 and 3. The variance based sensitivity analysis is successfully applied to the ELV launch vehicle model with 15 uncertainties. However, compared the results of worst-case validation in Table 4.1 when all

118 uncertainties are involved, the maximum objective value obtained by the 7 uncertainties are still much smaller than the maximum objective value when the 118 uncertainties are applied. It means that some highly important variables are not included in the 7 uncertainties. If the sensitivity analysis applied to the entire model with 118 uncertainties, some new dominant variables are highly expected to be identified.

6.1.4 Example 3: Drift of position on Z axis

For the cost function f_3, the CDF figure is firstly introduced. The CDF figure can be found in Figure 6.10 below. From the figure, the shape of CDF plots with different

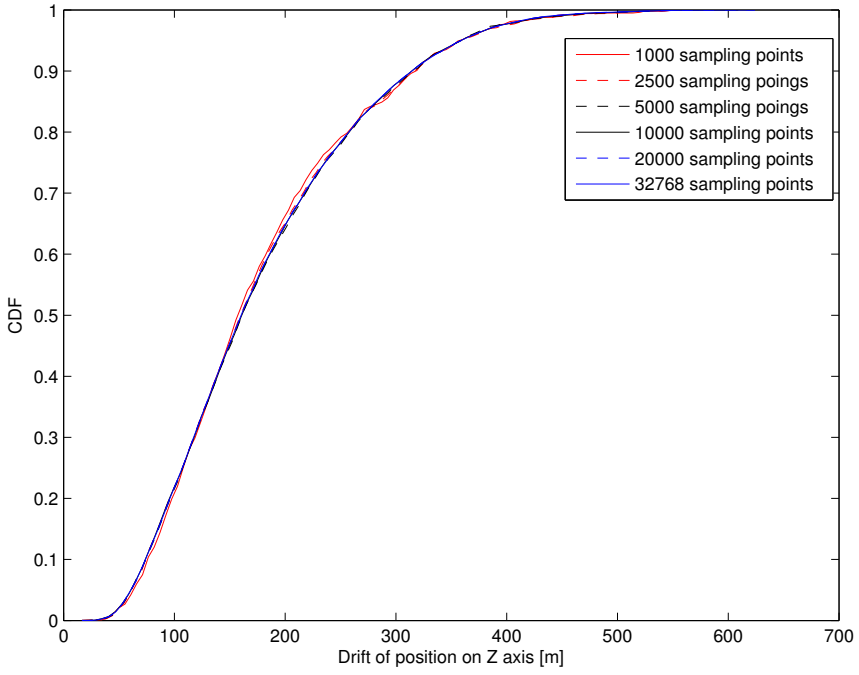
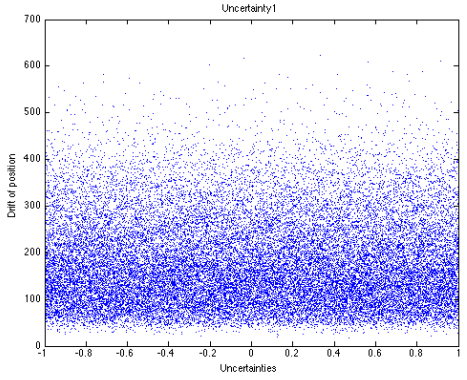


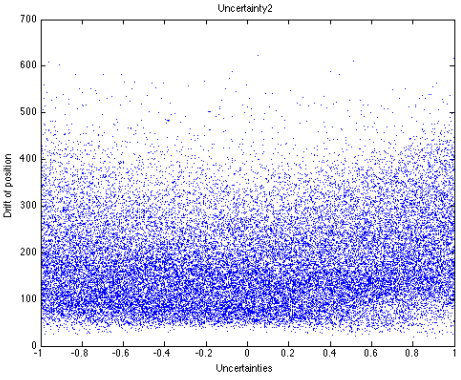
FIGURE 6.10: The cumulative distribution function of f_3 in terms of different sampling points

number of sampling points basically are very similar. A small number of sampling points also can present the statistic feature of the distribution of the objective values. The objective values that are greater than 500 and smaller than 50 can be considered as rare events.

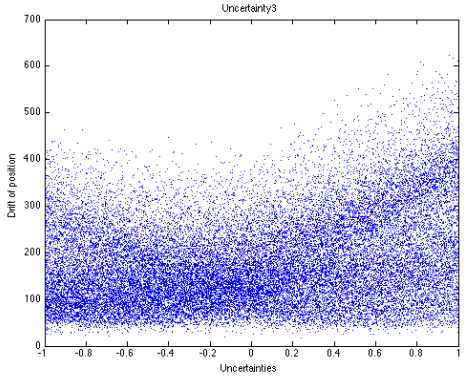
To understand the relationship between each single input and the output of the ELV launch vehicle, the scatter plots for each uncertainty are given in the Figure 6.11, Figure 6.12 and 6.13 followed.



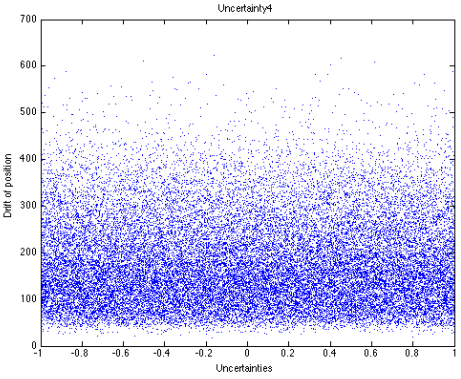
(A) IRSmountingX vs. f_3



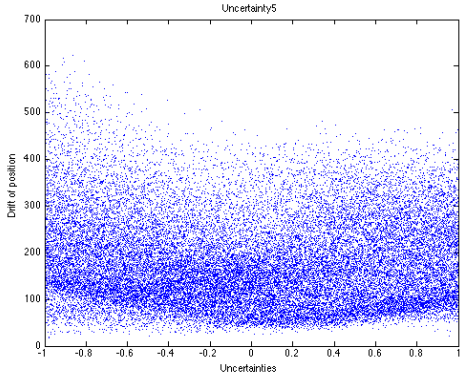
(B) IRSmountingY vs. f_3



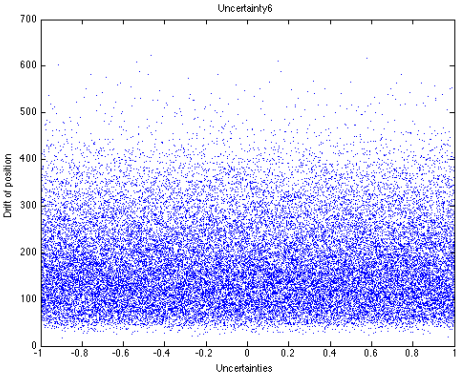
(C) IRSmountingZ vs. f_3



(D) dISP vs. f_3



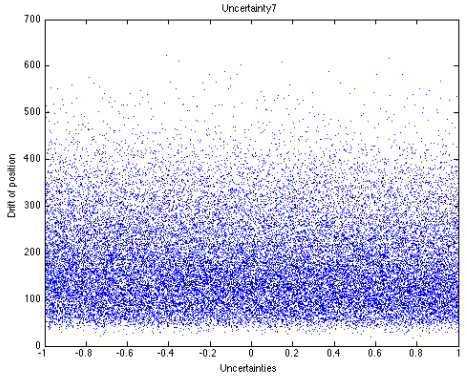
(E) dTc vs. f_3



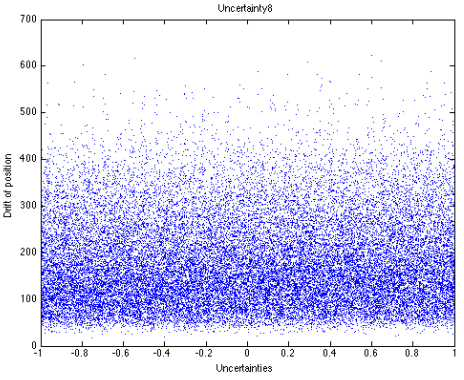
(F) disp_CA vs. f_3

FIGURE 6.11: The Scatter plots of f_3 - Part A

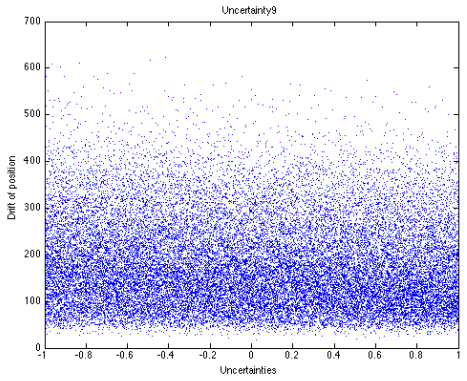
From the figure, all relationships between an uncertainty and objective are nonlinear. However, one linear tendency and two nonlinear tendencies are identified. For the plot (C) in Figure 6.11, the non-linear tendency is not that strong. The most largest values distribute around the uncertainty value 1. For the plot (E) in Figure 6.11, the linear tendency is not strong neither. But the most largest values are around the uncertainty value -1.For the plot (C) in Figure 6.13, the nonlinearity is obvious and the most largest values scatter around the nominal value 0 of the uncertainty. These tendencies can help



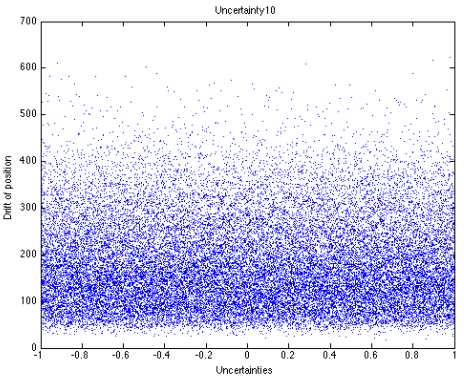
(A) disp_CN vs. f_3



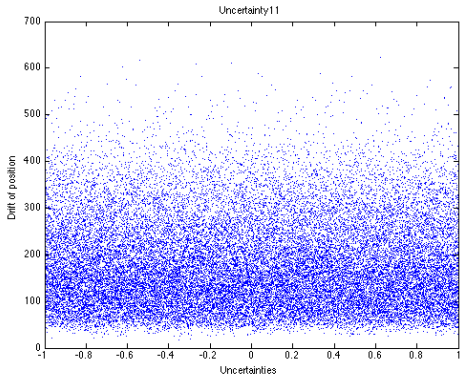
(B) disp_Xcp vs. f_3



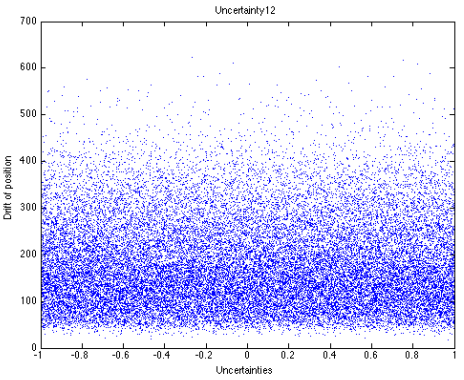
(C) air_density_scat vs. f_3



(D) h_wind vs. f_3



(E) azimuth_wind_angle vs. f_3

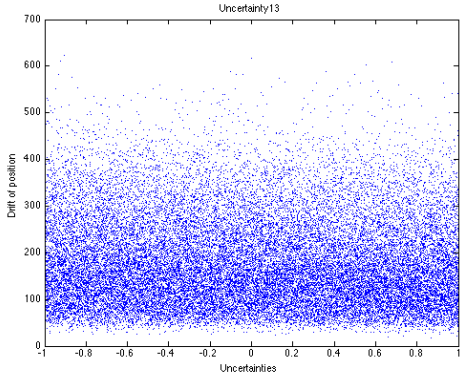


(F) unc_CA vs. f_3

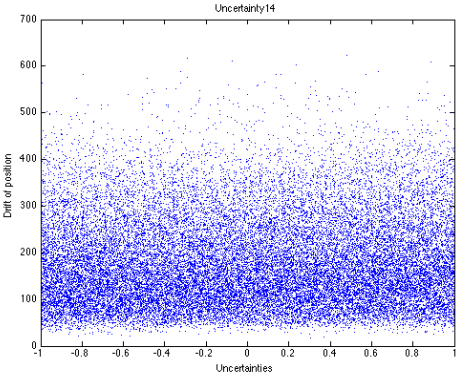
FIGURE 6.12: The Scatter plots of f_3 - Part B

to narrow down the range of the distributed samples if the corresponding uncertainties are identified as the important variables.

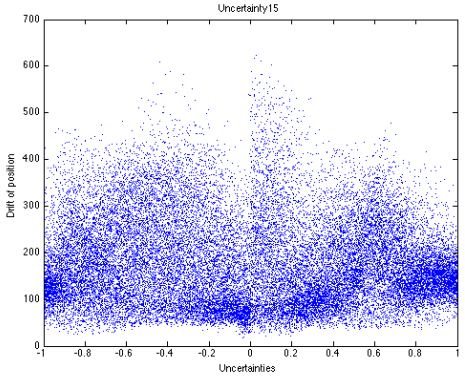
The followed Table 6.9 is the Sobol indices based on the objective function f_3. The first order sensitivity index and total sensitivity index for each uncertainty are provided. According to the first order sensitivity index of this table, only two most important uncertainties are identified(we consider an uncertainty with the value of first order sensitivity index above 0.1 is important variable): 'Flag. IRSmountingZ' and



(A) unc_CN vs. f_3



(B) unc_Xcp vs. f_3



(C) SRM_roll vs. f_3

FIGURE 6.13: The Scatter plots of f_3 - Part C

Variable Names	First order sensitivity index	Total sensitivity index
Flag.IRSmountingX	2.5755×10^{-5}	3.0547×10^{-4}
Flag.IRSmountingY	2.8325×10^{-4}	0.2356
Flag.IRSmountingZ	0.2567	0.5337
Flag.dISP	0.0019	0.0129
Flag.dTc	0.0792	0.4985
Flag.disp_CA	8.5510×10^{-5}	0.0071
Flag.disp_CN	0.0031	0.0269
Flag.disp_Xcp	8.5510×10^{-4}	0.0154
Flag.air_density_scat	0.0038	0.0298
Flag.h_wind	0	0
Flag.azimuth_wind_angle	0	0
Flag.unc_CA	0.0011	0.0111
Flag.unc_CN	0.0042	0.0452
Flag.unc_Xcp	4.0782×10^{-4}	0.0324
Flag.SRM_roll	0.1631	0.7592

TABLE 6.9: The first order and total sensitivity index based on f_3

'Falg.SRM_roll'. However, from the total sensitivity index, beside the important uncertainties identified by first order sensitivity index, two new important uncertainties are identified: 'Flag.IRSmountingY' and 'Flag.dTc'. Moreover, based on the difference between first order sensitivity index and total sensitivity index of the four uncertainties, it suggests the strong interaction effort between each important uncertainty and other uncertainties. According to the value of total sensitivity index, the rank of each uncertainty is given in the Table 6.10 followed.

Based on the rank of the uncertainties, the sequence of sampling points are able to

Variable Names	Order/ Rank
Flag.IRSmountingX	13
Flag.IRSmountingY	4
Flag.IRSmountingZ	2
Flag.dISP	10
Flag.dTc	3
Flag.disp_CA	12
Flag.disp_CN	8
Flag.disp_Xcp	9
Flag.air_density_scat	7
Flag.h_wind	14
Flag.azimuth_wind_angle	14
Flag.unc_CA	11
Flag.unc_CN	5
Flag.unc_Xcp	6
Flag.SRM_roll	1

TABLE 6.10: Re-ordered sequence based on the Sobol indices of f_3

re-ordered. A lower rank is in a lower order of dimension space. To verify whether the sensitivity analysis is successful to the objective function f_3, three worst-case validation test cases are applied with the two dimension reduction strategies. Similar as the verification process for f_1, three test cases are applied. For the first test case, the 15 uncertainties with re-ordered sequence of sampling points are involved in the simulation. For the second test case, the uncertainties with the rank above 7 are removed form the simulation (fixed to zero), the remaining uncertainties are sampled by reordered Sobol sequence. For the third test case, only the uncertainties with the rank under 7 are applied to the simulation and the others are fixed to zero. The three uncertainties with obvious tendencies are sampled by Monte Carlo method with narrowed upper and lower bounds. The other 4 uncertainties are sampled by re-ordered Sobol sequence.

The results of the worst-case validation of the three test cases are given in the Table 6.11

followed. From the table, the first test case with reordered sampling sequence to remove

Tests	Max drift of position(Z axis) [m]
No application of dimension reduction method	550.9015
Test 1 (uncertainties sequence re-ordered)	540.6460
Test 2 (Sobol sequence for the 7 uncertainties)	688.9805
Test 3 (hybrid sampling for the 7 uncertainties)	633.0633

TABLE 6.11: Worst-case validation based on dimension reduction methods for objective function f_3

the redundant information is not able to improve the performance of the worst case validation. It obtains a smaller value than the worst case validation results without any dimension reduction method. However, the second and third test cases by neglecting the non-important uncertainties significant improve the results of worst case validation. This time the hybrid sampling strategy doesn't obtain the largest value of worst case validation. The variance based sensitivity analysis is approved as an useful method to reduce the redundant information. However, compared the worst case validation result obtained by Sobol sequence with 118 uncertainties in the Table 4.1, the worst case validation result is still much smaller with only 15 uncertainties. This means some very important uncertainties are not included in the 15 uncertainties. When the sensitivity analysis is applied to the 118 uncertainties, some new important uncertainties are expected to be identified.

6.1.5 Example 4: Drift of velocity on Y axis

For the objective function f_4, the CDF plots are provided in the Figure 6.14 below with different number of sampling points. From the figure, the shapes of the CDF plots based

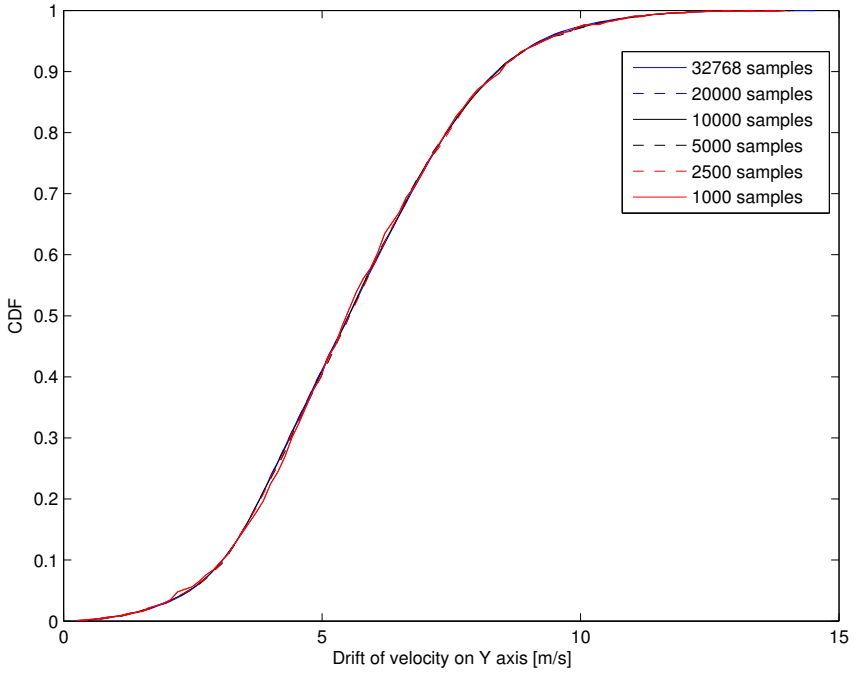
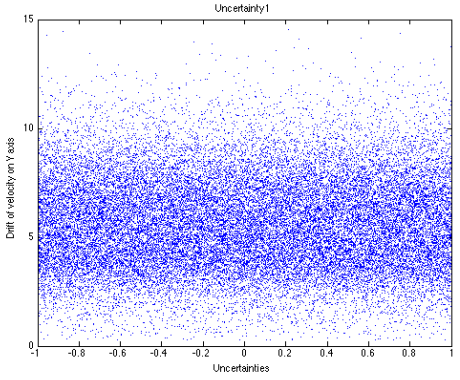


FIGURE 6.14: The cumulative distribution function of f_4 in terms of different sampling points

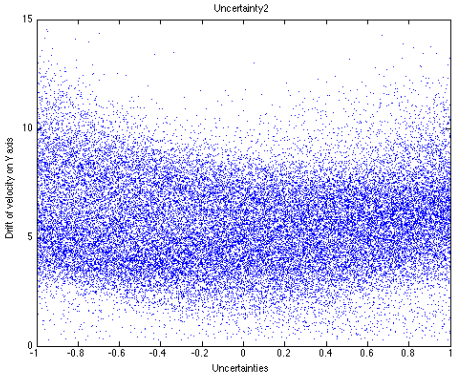
on the 1000, 2500, 5000,10000, 20000 and 32768 sampling points are almost same. Thus, a small number of sampling points, such as 1000 sampling points, are able to present the statistic feature of the distribution of the output. The objective values that are larger than 10 and smaller than 3 can be considered as the rare events.

To present the relationships between each uncertainty and the objective values, the scatter plots are given in the Figure 6.15, Figure 6.16 and Figure 6.17 below.

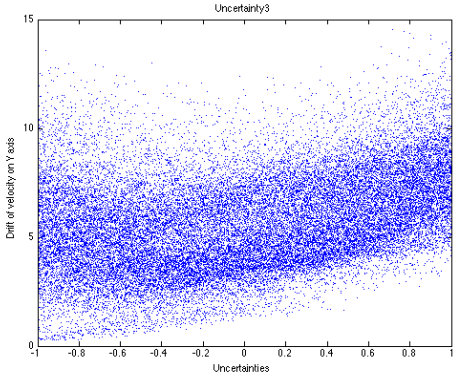
From the figure, all the relationships are nonlinear. However, for plot (B), (C), (E) in Figure 6.17 and (C) in Figure 6.17, the tendencies of the objective values are obviously and associated with value of uncertainties. For plot (B), the most largest values are distributed around the upper and lower bounds of the uncertainty 'Flag.IRSmountingY'. For plot (C) in Figure 6.15, the maximum objective is highly happened around the upper bound of the uncertainty 'IRSmountingZ' in general speaking. For plot (E), the most largest objective values are distributed around the lower bound of the uncertainty



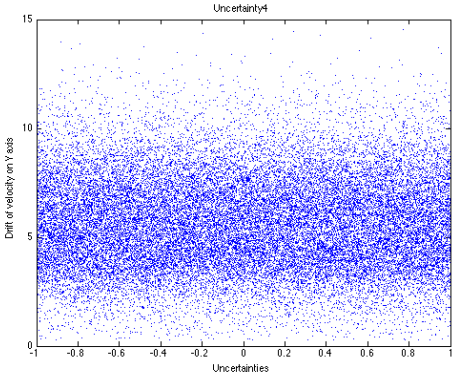
(A) IRSmountingX vs. f_4



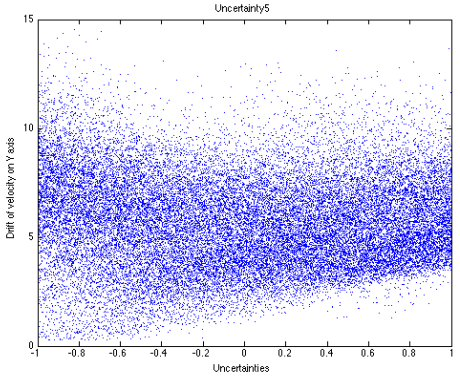
(B) IRSmountingY vs. f_4



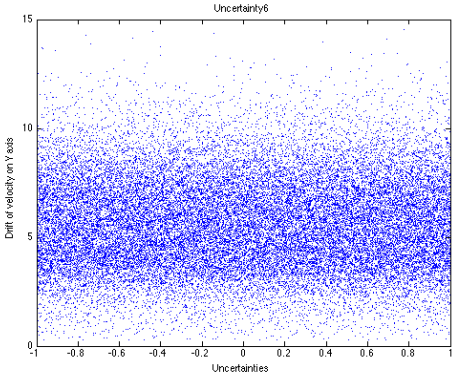
(C) IRSmountingZ vs. f_4



(D) dISP vs. f_4



(E) dTc vs. f_4

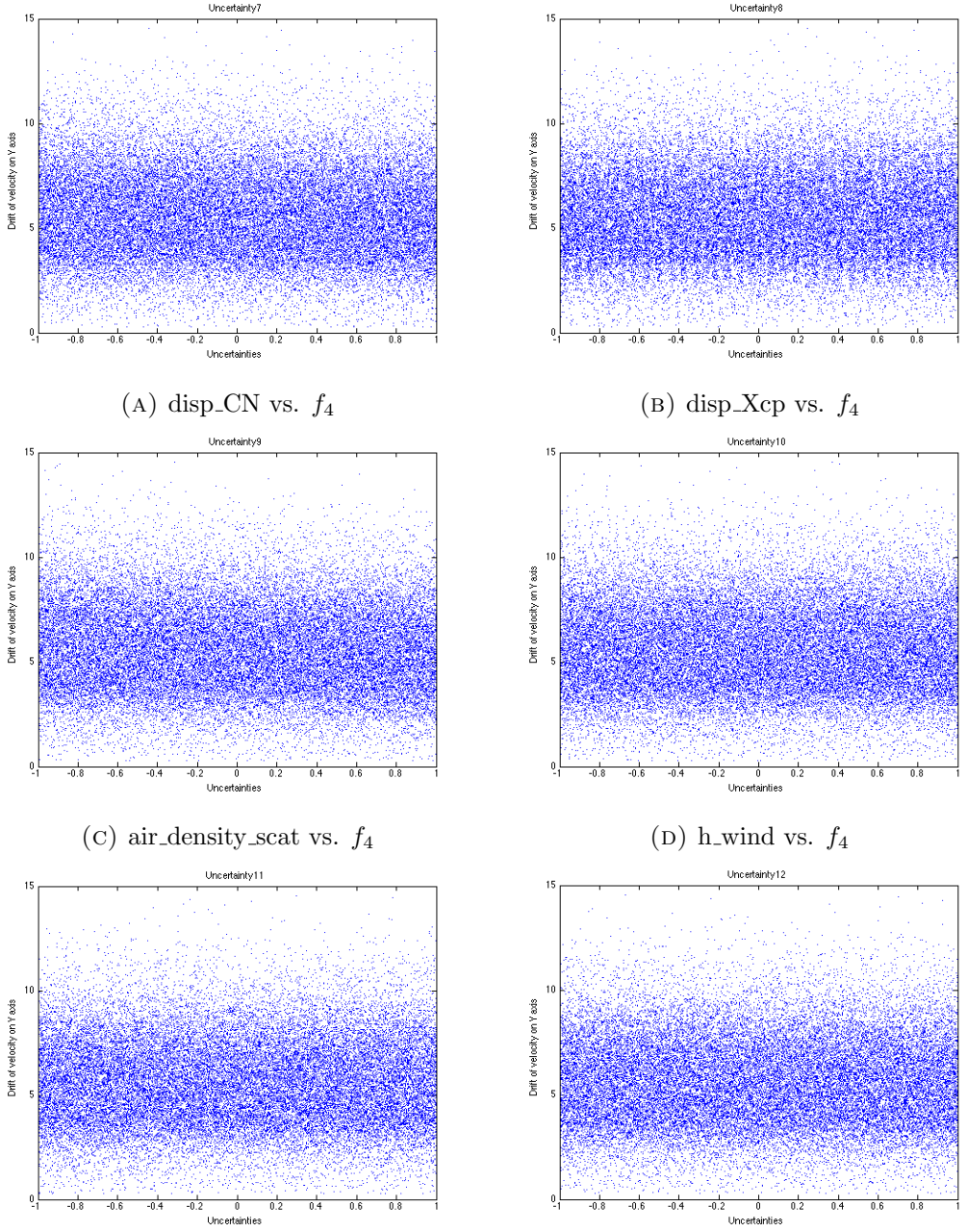


(F) disp_CA vs. f_4

FIGURE 6.15: The Scatter plots of f_4 - Part A

'Flag.dTc'. For plot (C) in Figure 6.17, the tendency is highly nonlinear. the most largest and smallest objective values are distributed around the nominal uncertainty value 0. These tendencies are usefully to narrow down the distribution ranges of these four uncertainties. If these uncertainties are identified as important variables, the narrowed down lower and upper bounds are able to applied to the worst case validation and potentially help to obtain a larger objective value.

The Sobol indices of the 15 uncertainties based on the objective function f_4 are given in



(A) disp_CN vs. f_4

(B) disp_Xcp vs. f_4

(C) air_density_scat vs. f_4

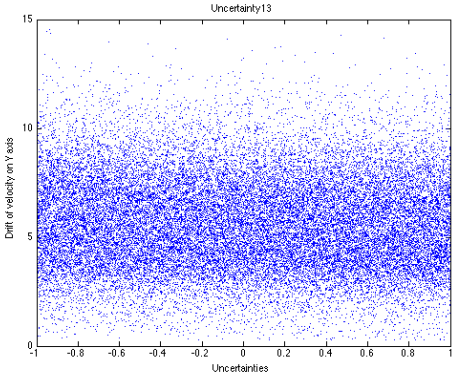
(D) h_wind vs. f_4

(E) azimuth_wind_angle vs. f_4

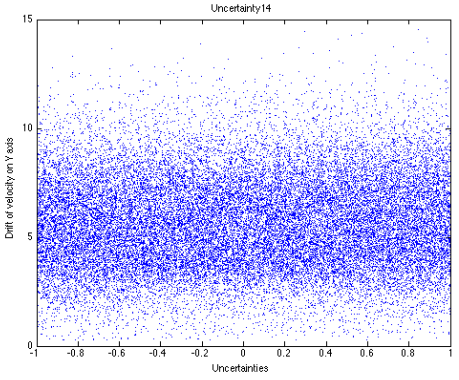
(F) unc_CA vs. f_4

FIGURE 6.16: The Scatter plots of f_4 - Part B

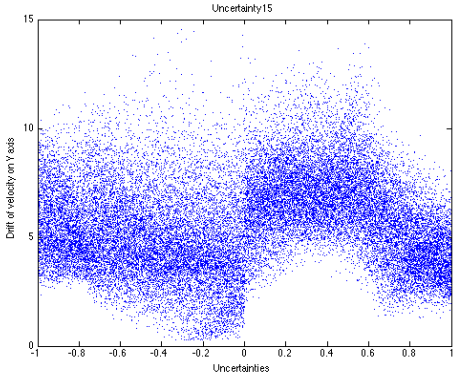
the Table 6.12 followed. From the first order sensitivity index of the Sobol indices. two important uncertainties can be indented: 'Flag.IRSmountingZ' and 'Flag.SRM'. Both uncertainties have a much larger value of first order sensitivity index compared to other uncertainties. From the total sensitivity index, besides the two important uncertainties that are identified by first order sensitivity index, another two uncertainties are identified as the important uncertainties: 'Flag.IRSmountingY' and 'Flag.dTc'. For the



(A) unc_CN vs. f_4



(B) unc_Xcp vs. f_4



(C) SRM_roll vs. f_4

FIGURE 6.17: The Scatter plots of f_4 - Part C

Variable Names	First order sensitivity index	Total sensitivity index
Flag.IRSmountingX	1.4456×10^{-4}	9.8108×10^{-5}
Flag.IRSmountingY	0.0595	0.2649
Flag.IRSmountingZ	0.3227	0.3491
Flag.dISP	3.2064×10^{-4}	0.0066
Flag.dTc	0.0525	0.3716
Flag.disp_CA	0.0027	8.3419×10^{-4}
Flag.disp_CN	0.0058	0.0123
Flag.disp_Xcp	0.0048	0.0083
Flag.air_density_scat	0.0021	0.0149
Flag.h_wind	0	0
Flag.azimuth_wind_angle	0	0
Flag.unc_CA	4.1773×10^{-4}	0.0053
Flag.unc_CN	0.0033	0.0257
Flag.unc_Xcp	2.8803×10^{-4}	0.0153
Flag.SRM_roll	0.3702	0.6877

TABLE 6.12: The first order and total sensitivity index based on f_4

uncertainties 'Flag.SRM', 'Flag.IRSmountingY' and 'Flag.dTc', the values of total sensitivity index are much larger than values of first order sensitivity index. It means these three uncertainties have strong interaction efforts with other uncertainties respectively. For the uncertainty 'Flag.IRSmountingZ', the difference between the first order index and total sensitivity index is very small. So the interaction effect of this uncertainty with other uncertainties are very small. According to the total sensitivity index, the ranks of each uncertainty are given in the Table 6.13 below. According to the ranks,

Variable Names	Order/ Rank
Flag.IRSmountingX	13
Flag.IRSmountingY	4
Flag.IRSmountingZ	3
Flag.dISP	10
Flag.dTc	2
Flag.disp_CA	12
Flag.disp_CN	8
Flag.disp_Xcp	9
Flag.air_density_scat	7
Flag.h_wind	14
Flag.azimuth_wind_angle	14
Flag.unc_CA	11
Flag.unc_CN	5
Flag.unc_Xcp	6
Flag.SRM_roll	1

TABLE 6.13: Re-ordered sequence based on the Sobol indices of f_4

the order of the sequence for the uncertainties generated by Sobol method is rearranged. The uncertainty with lower rank is in a lower dimension order in the Sobol sequence. The re-ordered sequence may be able to improve the performance of the worst case validation when we apply this re-ordered sequence for the first test case to examine the results of the variance based sensitivity analysis. For the second test case to examine the result of sensitivity analysis, the uncertainties that have the rank above 7 are neglected from the simulation and are fixed to 0. The remaining 7 uncertainties are sampled by Sobol method.For the third test case, 7 uncertainties are remained for the simulation as the second test case. These uncertainties are proposed by a hybrid sampling strategy. The four uncertainties with identified tendencies from the scatter plots are sampled by Monte Carlo method. The other three uncertainties are generated by Sobol sequence.
The results of the three tees cases are given in the Table 6.14 below. From the table, the first test case with re-ordered sampling sequence is not actually improve the result of

Tests	Max drift of velocity(Y axis) [m/s]
No application of dimension reduction method	13.9594
Test 1 (uncertainties sequence re-ordered)	12.7466
Test 2 (Sobol sequence for the 7 uncertainties)	15.1723
Test 3 (hybrid sampling for the 7 uncertainties)	15.7317

TABLE 6.14: Worst-case validation based on dimension reduction methods for objective function f_4

worst case validation. But the second and third case with only 7 selected uncertainties have much larger objective value. This means the variance based sensitivity analysis is applicable to the objective function f_4 of the ELV launch vehicle. However, the objective values obtained by the test cases are still smaller than the objective value when all the 118 uncertainties are involved in the worst case validation. Thus, some important uncertainties are not include in the 15 uncertainties. If the variance based sensitivity analysis is applied to the ELV launch vehicle model with 118 uncertainties, some new important variables are expected to be identified.

6.1.6 Example 5: Drift of position on Y axis

To analyze the sensitivity of the ELV launch vehicle for the objective function f_5, the CDF plots with different number of sampling points are given in the Figure 6.18 below. From the figure, all of the CDF plots are almost identical. This means a smaller number of sampling points, such as 1000 samples, are able to reveal the statistic feature of the distribution of the objective function f_5. The objective values that are larger than 400 and smaller than 50 are considered as the rare events.
The scatter plots of the objective function f_5 are given in the Figure 6.19, 6.20 and 6.21 followed.

The scatter plots is able to reveal the relationship between an uncertainty and the objective values. From this figure, all the relationships are nonlinear. However, there are four nonlinear tendencies are identified. For plots (B),(C) and (E) in Figure 6.19, the most objective values would become larger when the value of the corresponding uncertainty is close to 1 and -1. But the tendencies are not very strong, especially for the

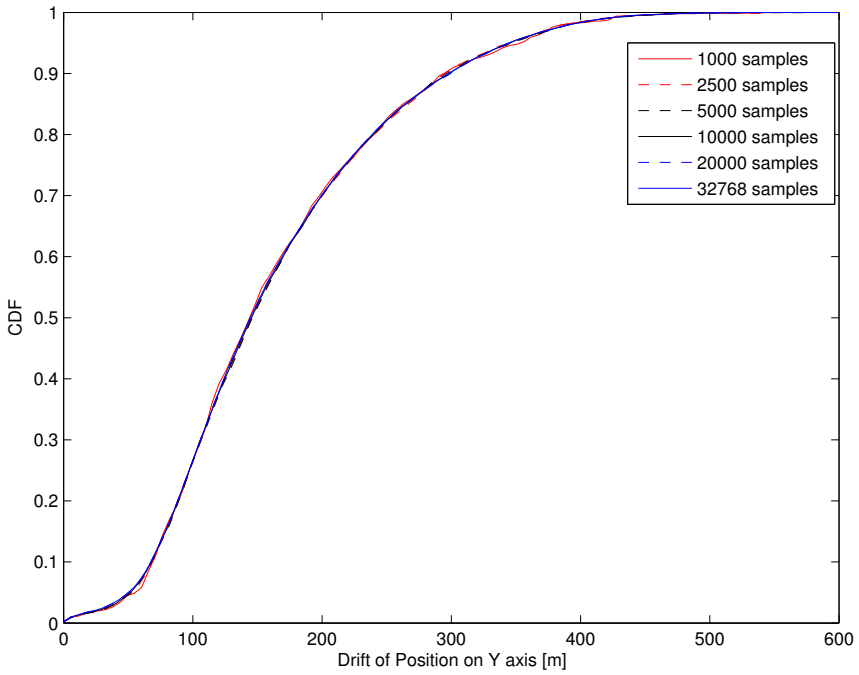
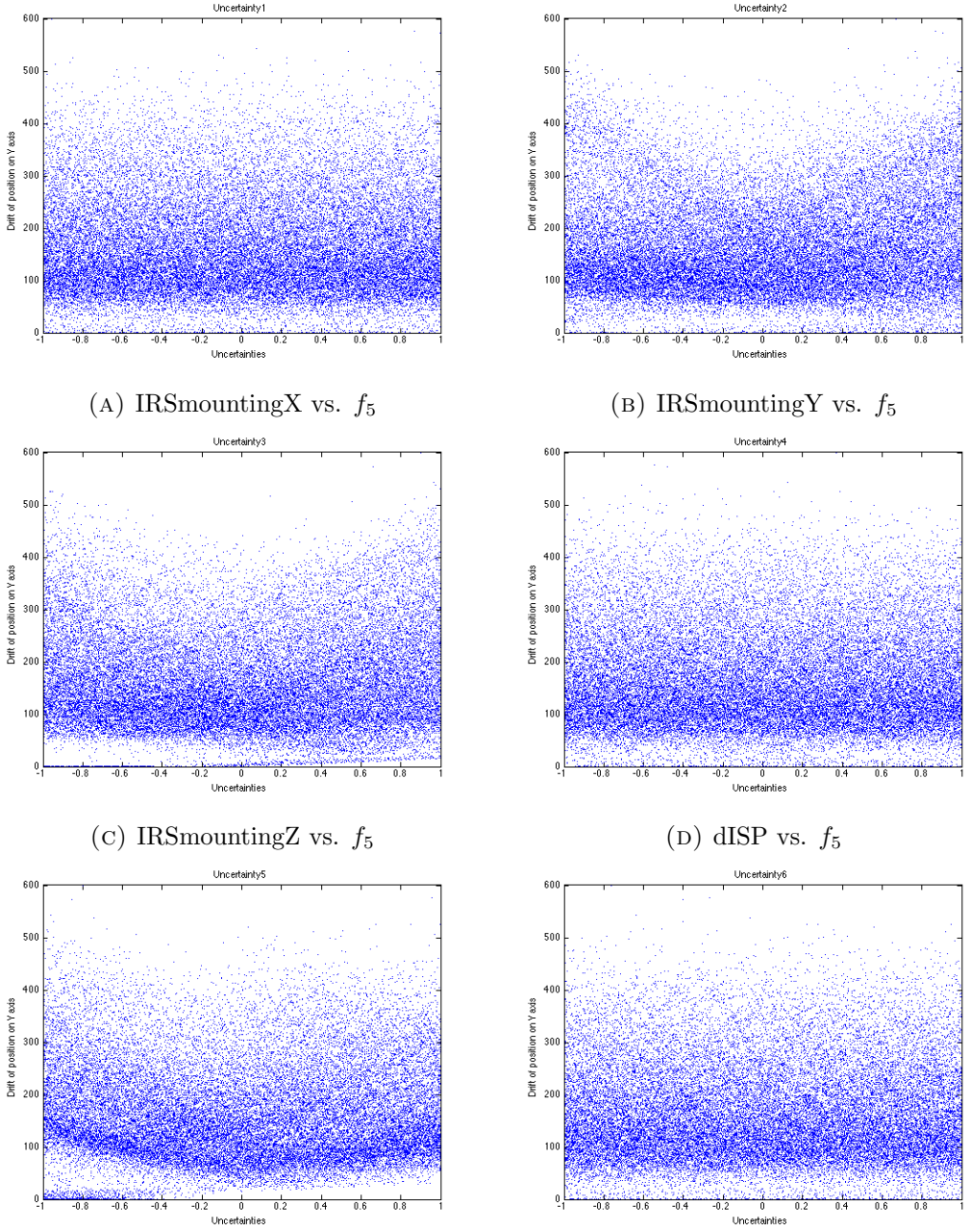


FIGURE 6.18: The cumulative distribution function of f_5 in terms of different sampling points

plots (B) and (E). Thus, there is necessary to apply these tendencies to narrow down the upper bounds and low bounds of the three uncertainties. For the plot (C) in Figure 6.21, the nonlinearity of the tendency is very strong. The most largest objective values are distributed when then value of uncertainty 'Flag.SRM_roll' is around -0.6 and 0.6. The distribution range of the uncertainty 'Flag.SRM_roll' can be narrowed down.

The followed Table 6.15 is the Sobol indices based on the objective function f_5. From the first order sensitivity index of this table, there is no uncertainty with a relatively large value (we define the large value as the value is greater than 0.1). Only the values of the uncertainties 'Flag.IRSmountingZ' and 'Flag.dTc' are close to 0.1. Unlike the results of the first order sensitivity index for the previous objective functions, there isn't any uncertainty can be apparently identified as the important variable. However, from the total sensitivity index, four uncertainties are able to be considered as the important variables: 'Flag.IRSmountingY','Flag.IRSmountingZ','Flag.dTc' and 'Flag.SRM_roll'. These four uncertainties have much larger value of total sensitivity index. Compared to the small values of the first order sensitivity index, the four uncertainties have a strong interacted efforts with other uncertainties. According to the total sensitivity index, the ranks are assigned to the uncertainties. An uncertainty with larger value has a smaller value of



(A) IRSmountingX vs. f_5

(B) IRSmountingY vs. f_5

(C) IRSmountingZ vs. f_5

(D) dISP vs. f_5

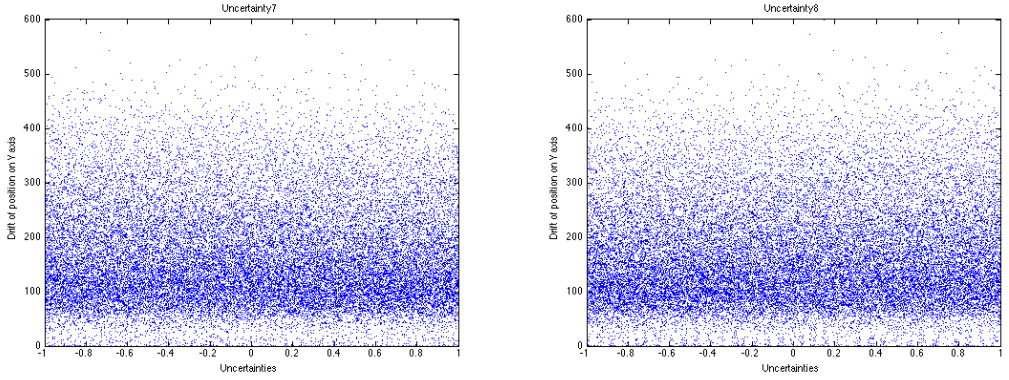
(E) dTc vs. f_5

(F) disp_CA vs. f_5

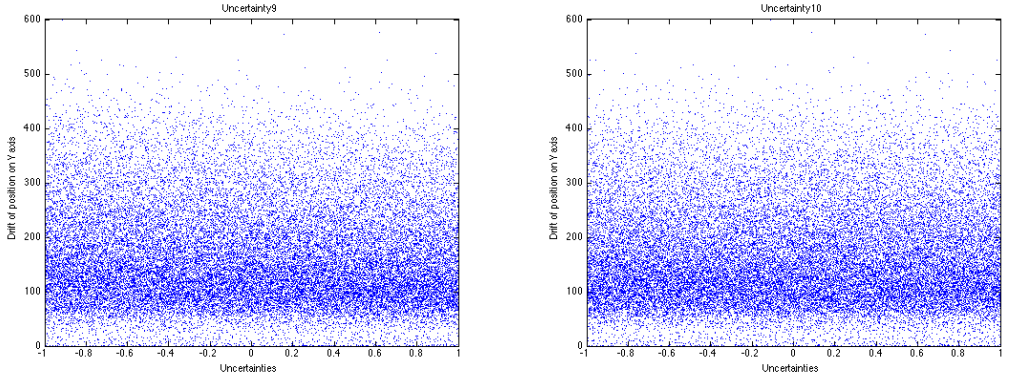
FIGURE 6.19: The Scatter plots of f_5 - Part A

rank.

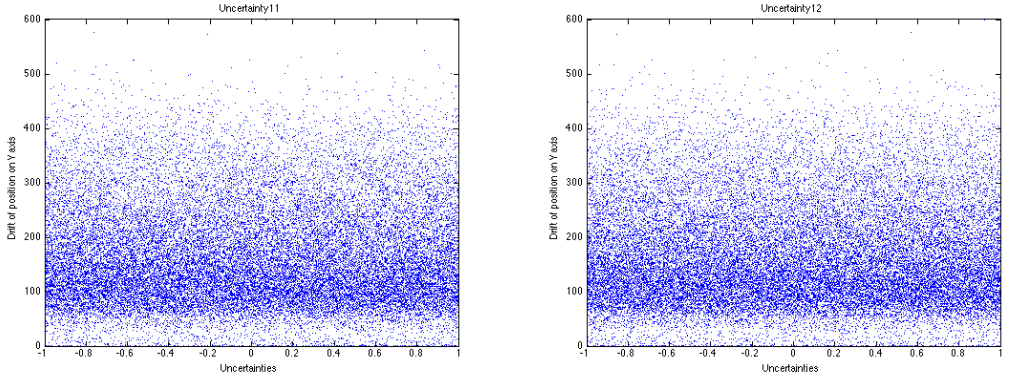
The Table 6.16 below gives the ranks for the uncertainties. Based on the ranks, the sequence of the uncertainties of the samples are reordered. An uncertainty with smaller rank is ordered in the lower dimension of the sequence. The reordered sequence is applied to the worst case validation and potential be able to improve the results of the worst-case. To verify the variance based sensitive analysis results based on the objective function f_5, three sampling based robustness test cases are employed. The first test case



(A) disp_CN vs. f_5 (B) disp_Xcp vs. f_5



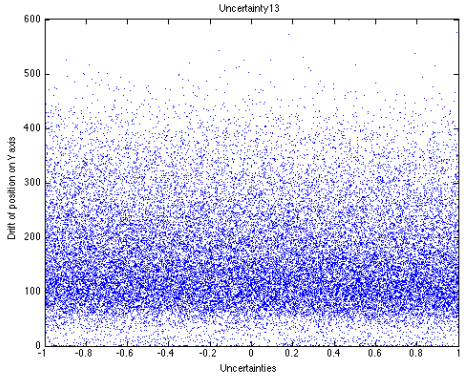
(C) air_density_scat vs. f_5 (D) h_wind vs. f_5



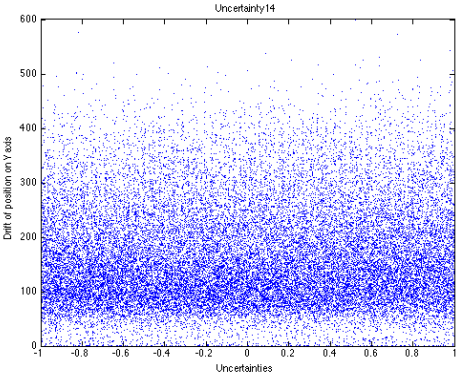
(E) azimuth_wind_angle vs. f_5 (F) unc_CA vs. f_5

FIGURE 6.20: The Scatter plots of f_5 - Part B

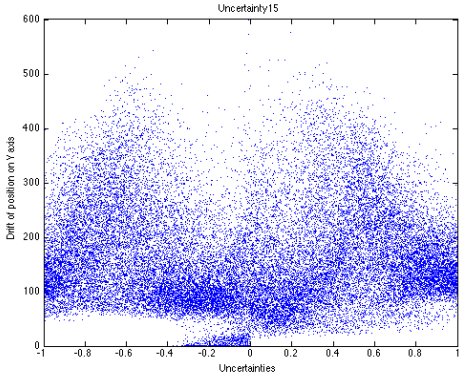
uses the reordered sequence of sampling points for the worst case validation. For the second test case, the uncertainties with the corresponding rank above 7 are neglected from the simulation. The remaining 7 uncertainties are sampled based on the reordered sequence by Sobol method. For the third test case, only 7 uncertainties are kept for the simulation as the second test case. A hybrid sampling strategy is applied. The uncertainties 'Flag.SRM_roll', 'Flag.SmountingY' and 'Flag.SmountingZ' are sampled by Monte Carlo method with narrowed down ranges and the remaining uncertainties are



(A) unc_CN vs. f_5



(B) unc_Xcp vs. f_5



(C) SRM_roll vs. f_5

FIGURE 6.21: The Scatter plots of f_5 - Part C

Variable Names	First order sensitivity index	Total sensitivity index
Flag.IRSmountingX	2.6470×10^{-4}	3.2443×10^{-4}
Flag.IRSmountingY	0.0587	0.3887
Flag.IRSmountingZ	0.0939	0.3221
Flag.dISP	8.1770×10^{-4}	0.01276
Flag.dTc	0.0975	0.5298
Flag.disp_CA	9.9756×10^{-4}	0.0052
Flag.disp_CN	0.0040	0.0249
Flag.disp_Xcp	0.0025	0.0150
Flag.air_density_scat	0.0014	0.0289
Flag.h_wind	0	0
Flag.azimuth_wind_angle	0	0
Flag.unc_CA	0.0022	0.0091
Flag.unc_CN	0.0040	0.0485
Flag.unc_Xcp	6.2553×10^{-4}	0.0295
Flag.SRM_roll	0.0191	0.8805

TABLE 6.15: The first order and total sensitivity index based on f_5

Variable Names	Order/ Rank
Flag.IRSmountingX	13
Flag.IRSmountingY	3
Flag.IRSmountingZ	4
Flag.dISP	10
Flag.dTc	2
Flag.disp_CA	12
Flag.disp_CN	8
Flag.disp_Xcp	9
Flag.air_density_scat	7
Flag.h_wind	14
Flag.azimuth_wind_angle	14
Flag.unc_CA	11
Flag.unc_CN	5
Flag.unc_Xcp	6
Flag.SRM_roll	1

TABLE 6.16: Re-ordered sequence based on the Sobol indices of f_5

sampled by Sobol sequence.

The results of the three test cases are given in the Table 6.17 below. From the table,

Tests	Max drift of position(Y axis) [m]
No application of dimension reduction method	542.7051
Test 1 (uncertainties sequence re-ordered)	531.3844
Test 2 (Sobol sequence for the 7 uncertainties)	681.3836
Test 3 (hybrid sampling for the 7 uncertainties)	587.7482

TABLE 6.17: Worst-case validation based on dimension reduction methods for objective function f_5

the worst case validation with the 15 reordered Sobol sequence doesn't obtain a larger maximum objective value. This means the Sobol sequence reordered method is not helpful in this case. However, the test case 2 and 3 with only 7 selected uncertainties have a larger worst-case value. The worst case validation without the redundant information improve the result of robustness analysis. The sensitivity analysis is applicable to the objective function f_5 of the ELV launch vehicle. Nevertheless, compared with the worst-case results in the Table 4.1, the maximum objective value obtained based on the 7 uncertainties are still much smaller. Thus, there exists some other important uncertainties that are not included in the 15 selected uncertainties. If applied the sensitivity analysis to Launch vehicle, new uncertainties are expected to be identified as the important variables.

6.1.7 Example 6: Total angle of attack

For the sensitivity analysis based on the last objective function f_6, the CDF plots are firstly introduced. The followed Figure 6.22 gives the CDF plots based on different number of sampling points. The shapes of the CDF plots with different number of

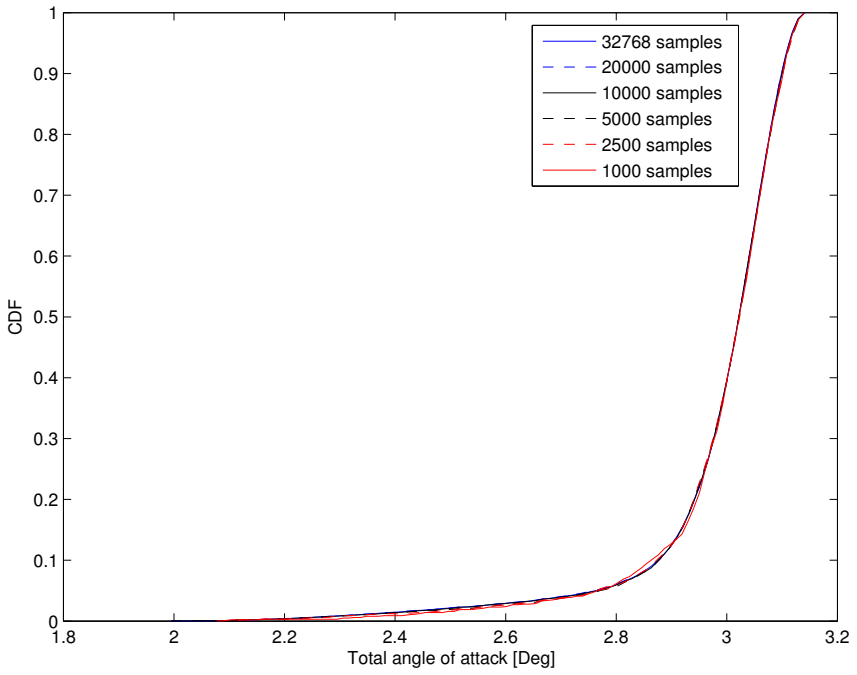
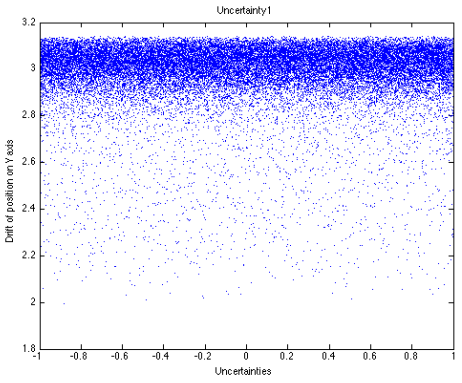


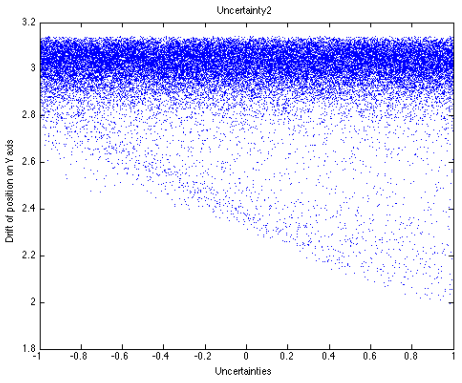
FIGURE 6.22: The cumulative distribution function of f_6 in terms of different sampling points

sampling points are very similar. This means a small number of sampling size is able to present the statistic feature of the distribution of objective values. Unlike the previous CDF figures based on other objective functions, in this figure, the most of objective values are distributed in a relatively narrowed range: from the value 2.8 to 3.1. The objective values with the value under 2.8 and above 3.1 are considered as the rare events. The scatter plots given in the followed Figure 6.23, Figure 6.24 and Figure 6.25 provide the relationships between every single uncertainty and objective values.

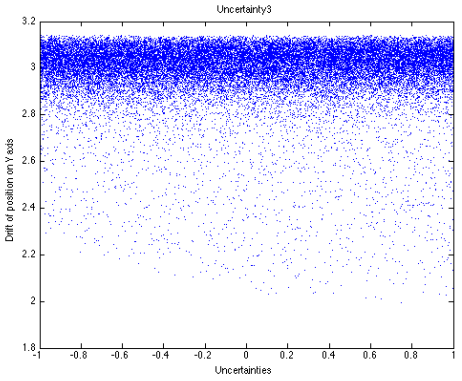
From the figure, all the relationships are nonlinear. Some linear and nonlinear tendencies are also can be identified, such as plots (B), (E) in Figure 6.23 and (C) in Figure 6.25.



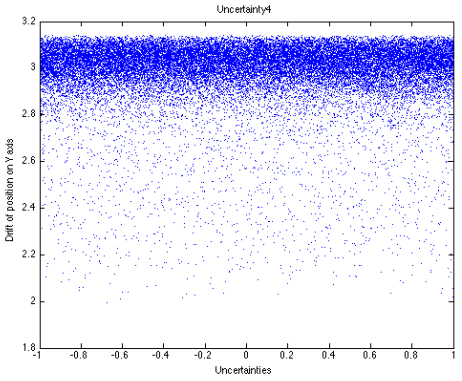
(A) IRSmountingX vs. f_6



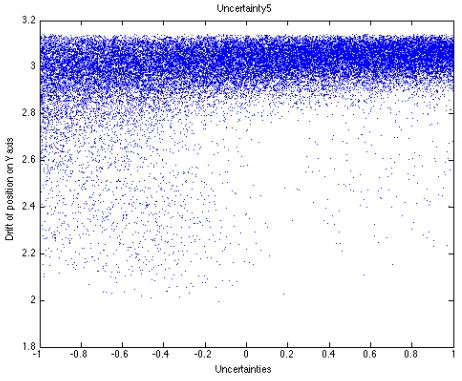
(B) IRSmountingY vs. f_6



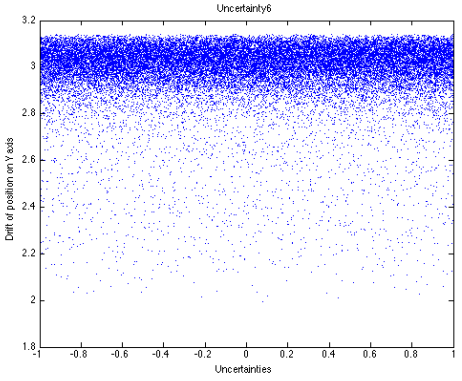
(C) IRSmountingZ vs. f_6



(D) dISP vs. f_6



(E) dTc vs. f_6

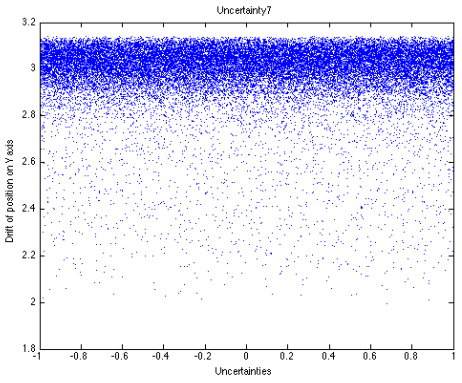


(F) disp_CA vs. f_6

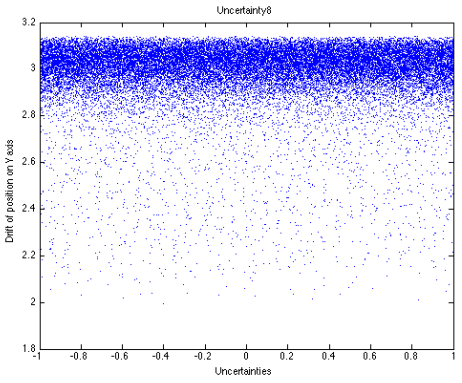
FIGURE 6.23: The Scatter plots of f_6 - Part A

However, the linearity and nonlinearity of the tendencies are not quite helpful. The reason is that the most largest objective values could be happened at any value of the uncertainties. In other words, the objective values are high possible non-sensitivity to a small number of uncertainties. If the uncertainties are equally important, no dimension reduction method is applicable.

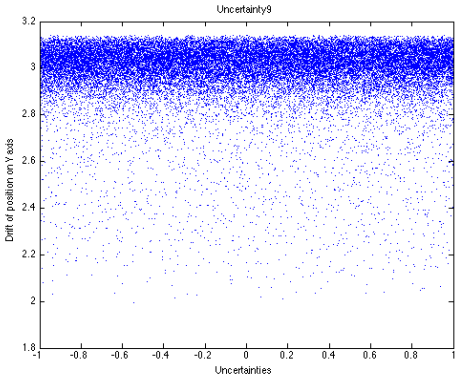
The Table 6.18 below is the Sobol indices for the objective function f_6. From the first order sensitivity index, only two uncertainties are considered as the important variables



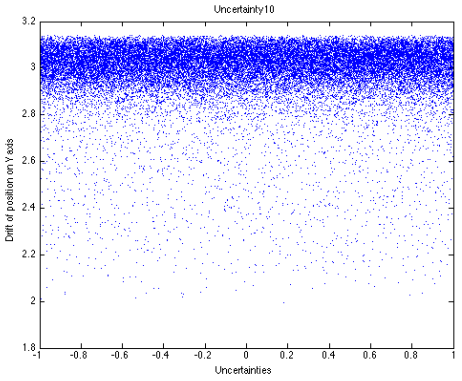
(A) disp_CN vs. f_6



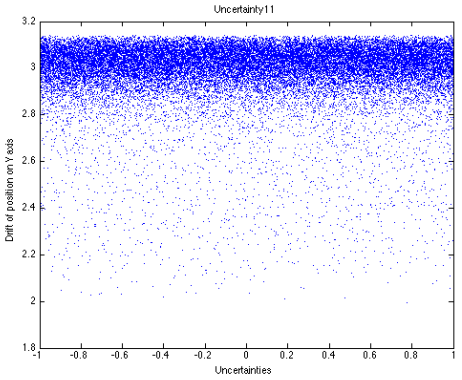
(B) disp_Xcp vs. f_6



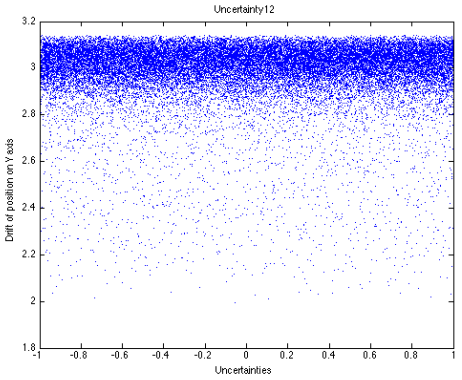
(C) air_density_scat vs. f_6



(D) h_wind vs. f_6



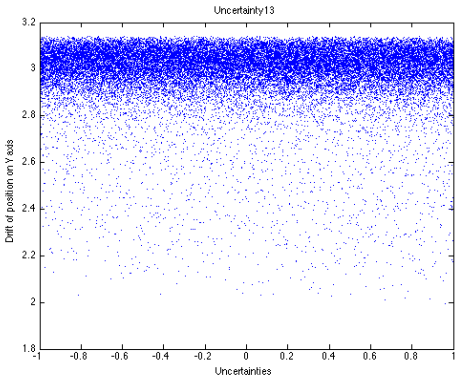
(E) azimuth_wind_angle vs. f_6



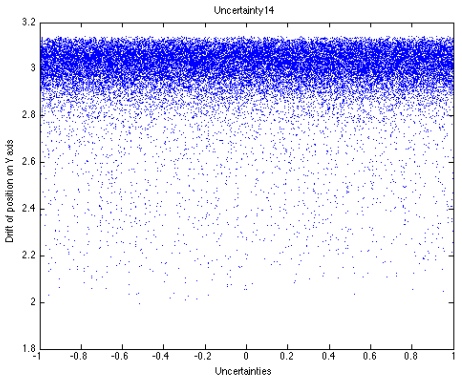
(F) unc_CA vs. f_6

FIGURE 6.24: The Scatter plots of f_6 - Part B

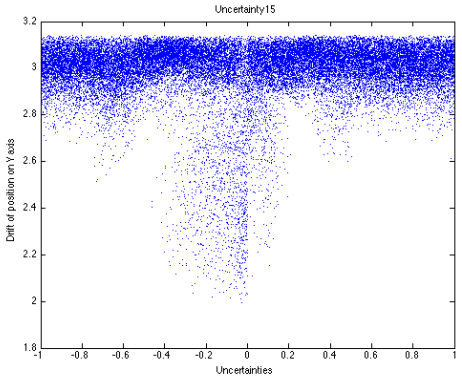
(with the value above 0.1): 'Flag.dISP' and 'Flag.SRM_roll'. However, from the total sensitivity index, there are 10 uncertainties can be identified as the important variables. It approves our inference made from the scatter plots above that the objective values is non-sensitivity to a small number of uncertainties. Nevertheless, according to total sensitivity index, two uncertainties have a much larger value compared to other uncertainties' value. If we redefine the condition of being an important variable, such as value of total sensitivity index is larger than 0.5, we can say there are only two important variables.



(A) unc_CN vs. f_6



(B) unc_Xcp vs. f_6



(C) SRM_roll vs. f_6

FIGURE 6.25: The Scatter plots of f_6 - Part C

Variable Names	First order sensitivity index	Total sensitivity index
Flag.IRSmountingX	0.0043	0.0045
Flag.IRSmountingY	0.1558	0.2028
Flag.IRSmountingZ	0.0074	0.3834
Flag.dISP	0.0703	0.1555
Flag.dTc	0.0966	0.8153
Flag.disp_CA	0.0356	0.0937
Flag.disp_CN	0.0246	0.1504
Flag.disp_Xcp	0.0284	0.1065
Flag.air_density_scat	0.0183	0.0379
Flag.h_wind	0	0
Flag.azimuth_wind_angle	0	0
Flag.unc_CA	0.0565	0.2172
Flag.unc_CN	0.0634	0.2002
Flag.unc_Xcp	0.0329	0.2394
Flag.SRM_roll	0.2164	0.7608

TABLE 6.18: The first order and total sensitivity index based on f_6

The assumption to redefine the condition of being an important uncertainty is examined by a worst case validation. Another assumption is that a set with small number of uncertainties from the 10 important variables could makes the worst case validation obtain a large objective value as well. This assumption is tested by another one worst case validations. The last assumption is that the result of the worst case validation with reduced dimensions is not likely to obtain a much larger value than the result of worst case validation when all 15 uncertainties are involved. In summary, there are 3 test cases in total. For test case 1, the worst case validation is assigned with the reordered Sobol sequence of the 10 important uncertainties. For test 2, only the largest two uncertainties 'Flag.SRM_roll' and 'Flag.dTc' are involved in the simulation. For test 3, only the uncertainties 'Flag.IRSmountingY','Flag.IRSmountingZ', 'Flag.unc_CA', 'Flag.unc_CN' and 'Flag.uc_Xcp' are involved in the simulation with reordered Sobol sequence. No tendencies information are available to the worst case validation. The rank of the uncertainties are given in the Table 6.19 below. For the reordered Sobol sequence, the

Variable Names	Order/ Rank
Flag.IRSmountingX	13
Flag.IRSmountingY	6
Flag.IRSmountingZ	3
Flag.dISP	8
Flag.dTc	1
Flag.disp_CA	11
Flag.disp_CN	9
Flag.disp_Xcp	10
Flag.air_density_scat	12
Flag.h_wind	14
Flag.azimuth_wind_angle	14
Flag.unc_CA	5
Flag.unc_CN	7
Flag.unc_Xcp	4
Flag.SRM_roll	2

TABLE 6.19: Re-ordered sequence based on the Sobol indices of f_6

uncertainty with lower value of rank is in the lower order sequence. The reordered sequence is applied to the worst case validation mentioned above.

The results of the three worst case validations are given in the Table 6.20 followed. From the table, the values of worst-case obtained by the three test cases is little difference and not exceed the worst-case value when the 15 uncertainties are involved. The test 2 with only 2 uncertainties also has a relatively large objective value. This means the two

Tests	Max total angle of attack [Deg]
No application of dimension reduction method	3.1406
Test 1 (Reordered sequence with 10 uncertainties)	3.1369
Test 2 (with the 2 uncertainties)	3.1400
Test 3 (with the 5 uncertainties)	3.1046

TABLE 6.20: Worst-case validation based on dimension reduction methods for objective function f_6

uncertainties are truly important and sensitivity analysis is quite useful. Nevertheless, the test 3 with 5 uncertainty has a little smaller value compared to other cases. The worst-case value of test 2 verify our assumption that the worst case validation with a small number of uncertainties can also achieve a large worst-case value. The absence of another 5 uncertainties does not significant affect the worst-case value. This means even a few number of important uncertainties are able to make the worst case validation obtain a large objective value.

6.2 Short summary

In this chapter, we propose variance-based sensitivity analysis to identify the driven parameters of a launch vehicle problem. The ranking results demonstrate that the five key parameters sorted out by this method exactly match our understanding of the importance of these 15 selected parameters. To more clearly observe the ranking results, we summarize all rankings with respect to the six objective functions in one place. The summary table is given in Table 6.21 , which can facilitate easy identification of the driving parameters overall.

Variable Names	f_1	f_2	f_3	f_4	f_5	f_6
Flag.IRSmountingX	12	13	13	13	13	13
Flag.IRSmountingY	5	2	4	4	3	6
Flag.IRSmountingZ	4	4	2	3	4	3
Flag.dISP	11	10	10	10	10	8
Flag.dTc	2	3	3	2	2	1
Flag.disp_CA	13	12	12	12	12	11
Flag.disp_CN	6	8	8	8	8	9
Flag.disp_Xcp	7	9	9	9	9	10
Flag.air_density_scat	1	6	7	7	7	12
Flag.h_wind	14	14	14	14	14	14
Flag.azimuth_wind_angle	14	14	14	14	14	14
Flag.unc_CA	10	11	11	11	11	5
Flag.unc_CN	9	5	5	5	5	7
Flag.unc_Xcp	8	7	6	6	6	4
Flag.SRM_roll	3	1	1	1	1	2

TABLE 6.21: Ranks of uncertainties in all objective functions

Chapter 7

Gaussian process for sensitivity analysis

The success of variance-based sensitivity analysis based on a launch vehicle (LV) model with 15 uncertainties offers strong support for the application of such analysis to a LV model with 118 uncertainties. However, if we exactly follow the procedure for variance-based sensitivity analysis for the LV model with 15 uncertainties mentioned above to study one with 118 uncertainties, the computational cost is too expensive. To address this issue, we propose an emulator to approximate the ELV launch vehicle model. The emulator should satisfy two conditions: (1) cheap computation and (2) high accuracy.

First, we tried to implement a method to generate a parametric model to accurately map the relationship between the inputs and outputs, such as generating a surrogate model. However, we realized that it is very difficult to quickly obtain a polynomial function with high accuracy to describe the mapping relationship due to the complexity and nonlinearity of an ELV launch vehicle. For example, when we considered a linear surrogate model (with only first-order uncertainties), the linear model fails to both make a good prediction and fit the training data. If we increased the order of the surrogate model, i.e., implement the second surrogate mode, we obtain a very good model to fit the training data, but it still has a poor prediction capability. To address these issues, a statistical model provides an alternative method of presenting the mapping relationship. Instead of directly finding a function of input variables to obtain accurate output, the statistical model approximates the features of the distribution of the outputs in terms

of the corresponding inputs. The distribution features normally include the probability density, mean value, and deviation of the outputs. Thus, we propose a Gaussian process to train the statistical model. These features are exactly the fundamental needs of the variance-based sensitivity analysis as defined.

7.1 Gaussian Process

The Gaussian process (GP) is introduced in Chapter 2 and the details of the training procedure are given in Chapter 5. Without introducing any new aspects of model training, the GP model training process for probability sensitivity analysis exactly follows the principle used to train the model for optimization. The only new concept in this chapter, then, is defining the approximation method used to estimate the variance of each uncertainty parameter. With knowledge of the GP model, the probability sensitivity analysis can be employed conveniently to screen out the dominant variables.

7.2 Probability sensitivity analysis

When the meta-model is trained using the Gaussian process and applied for the sensitivity analysis, this application is also called probability sensitivity analysis, since the Gaussian process itself is a collection of outputs of probability. Probability sensitivity analysis has studied by many researchers, such as [18, 48, 65]. The concept of probability sensitivity analysis is straightforward. However, its application for a complex system with a high dimension of uncertainties is not that easy. The proper inferences for the training points, mean functions, covariance function, main effect of an uncertainty, interaction effect of uncertainties, and variance of uncertainties are necessary to obtain good results. The analytical way of calculating the expected variance for sensitivity analysis is possible, but it is difficult and depends on the dimension of variables and on the inference for the mean and covariance functions. In Oakley's paper [18], a conditional meta-model is derived using the Gaussian process and predicts the posterior distribution in order to calculate the sensitivity indices in terms of some inferences. In other words, the meta-model itself is extended to a conditional posterior distribution. In Oakley's work, it is easy to obtain the posterior distribution of the main and interaction effects of a variable

in an analytical way, but the estimated sensitivity indices are ideally obtained using a numerical method. Actually, although Oakley mentioned that the numerical method is able to obtain the total sensitivity indices with low computation cost, the approximation for the integration of the conditional distribution is not straightforward, and the quality of total sensitivity indices is not good. In our experiment, we do not implement the conditional meta-model. The meta-model in our experiment is not conditional, but the predicted posterior distribution is. To avoid solving the multiple integration and inverse matrices, we propose a numerical way of addressing these issues with proper inferences and approximations. However, the drawback of the numerical method of approximating the integration is that it is difficult to obtain good total sensitivity indices.

In the following subsections, we briefly introduce some important concepts for training the model using a Gaussian process. The method of generating the training points and inferences for prior, mean, and covariance functions are same as the method and inferences used to train the GP model for optimization. The only difference is that we introduce an inference for variance.

7.2.1 Training points

The chosen of training points in Gaussian process is also important. These training points are not selected randomly. We hope the observations in terms of the training points can provide good information of uncertainty space. Thus, the training points ideally have a uniform distribution over the sampling space. In our experiment, we propose the quasi-Monte Carlo to generate the training points and obtain the corresponding observations. There is question rose here: how many training points are sufficient enough for the Gaussian process? It is easy to understand that the Gaussian process with more training points inevitable generate a higher quality meta-model. However, from the equation 2.46, the drawback of large number of training point is that the complexity and computational burden for the calculation of posterior distribution are significantly increasing. So our question becomes what is the trade-off number of training points for Gaussian process? To unswear this question, we investigate the trade-off point experimentally: Calculate the quality of the meta-model and computation time with different number of training points. The details and example of the strategy are given in the test case later.

7.2.2 Inference for mean function

A mean function in the Gaussian process is a scaler function and used to predict the mean value of the noise free output over the uncertainty space. The inference for mean function is not exactly inference for the entire mean function 2.60, but is the inference for the regression functions $\phi(x)$ in equation 2.49. The reason that we call the 'inference for mean function' is that $\phi(x)$ is one of the important components in the mean function. The set of all regression functions are denoted as $\Phi(X)$ and $\Phi(X)$ are unknown hyperparameters. The inference of $\Phi(X)$ is normally based on some knowledge of the objective model.

7.2.3 Inference for covariance function

A covariance function is applied to calculate the covariance between two inputs X and X^*. For posterior distribution, it measure the covariance between the training points and predicted points. A good covariance function is able to improve the quality of meta-model. But the inference normally is depended on the knowledge of the objective model.

7.2.4 Inference for prior

The inference for prior ideally needs the expert knowledge of the objective model. In the paper [16, 17], the authors discuss some strategies and issues for the prior distribution. Normally, we expect the prior has zero mean function. So we define the prior distribution that is only depended on the unknown hyperparameter Σ_p^2 in our experiment. In function-space view, the Σ_p^2 is also written as σ_f in the covariance function and described as 'signal variance'.

7.2.5 Inference for variances

From the selection of training points and inferences for mean function, covariance function and prior above, they are all related to the quality of the meta-model. In this

subsection, the inference for variances is implemented to calculate the sensitivity indices s_i and s_{Ti}.

Recall the definitions for main effects, interactions from last chapter, the inference for the main effects for a single variable is needed for the sensitivity analysis. The main effects for a given function $f(x)$ in a Bayesian model is expressed as:

$$E(f|x_p) = \int_{\chi_{-p}} f(x) \mathrm{d}G_{-p|p}(x_{-p}|x_p) \tag{7.1}$$

where the x_p denotes the uncertain variable p only, χ_{-p} is the entire uncertainty space except the variable p, $G_{-p|p}(x_{-p}|x_p)$ actually denotes a conditional distribution for the given variable x_p in the uncertainty space. An uncertainty vector x is composed by x_p and x_{-p}. In this thesis, we denote the posterior distribution for $f(x)$ is accompanied with a sign $*$. For instance, $f(x)_*$ or f_* means the prediction of $f(x)$

Thus, the expectation of the posterior distribution of $f(x)$ for the given x_p based on posterior mean in the equation 2.60 is expressed as:

$$\begin{aligned} E(f_*|x_p)] &= \int_{\chi_{-p}} K(X_*|X_p, X)[K(X,X) + \sigma_n^2 I]^{-1} y \ \mathrm{d}_{-p|p}(X_{-p}|X_p) \\ &= \int_{\chi_{-p}} \phi_* \Sigma_p \phi [\phi \Sigma_p \phi + \sigma_n^2 I]^{-1} y \ \mathrm{d}_{-p|p}(X_{-p}|X_p) \end{aligned} \tag{7.2}$$

where the $\phi = \phi(X)$, $\phi_* = \phi(x_*|X_p)$, y is the objective values of testing points. From the definition of the first order sensitivity indices, the inference for the variance of the main effects is expressed as:

$$\begin{aligned} Var_{X_p}[E_{X_{-p}}(f_*|X_p)] &= E_{X_p}[E_{X_{-p}}(f_*|X_p)^2] - E_{X_p}[E_{X_{-p}}(f_*|X_p)]^2 \\ &= E_{X_p}[E_{X_{-p}}(f_*|X_p)^2] - E(f_*)^2 \end{aligned} \tag{7.3}$$

The $E_{X_p}[E_{X_{-p}}(f_*|X_p)^2]$ in this equation can be extended to the expression followed by substituted the equation 7.2:

$$\begin{aligned} &E_{X_p}[E_{X_{-p}}(f_*|X_p)^2] \\ &= \int_{\chi_p} \int_{\chi_{-p}} \int_{\chi'_{-p}} \eta(x_*)\eta(x'_*) \mathrm{d}G_{-p|p}(X_{-p}|X_p) \mathrm{d}G_{-p|p}(X'_{-p}|X_p) \mathrm{d}G_p(X_p) \end{aligned} \tag{7.4}$$

where $\eta(x_*) = \phi_* \Sigma_p \phi [\phi \Sigma_p \phi + \sigma_n^2 I]^{-1} y$, the uncertainty set X is composed by the set X_p and its complementary set X_{-p}, the X'_{-p} and X_{-p} are in depended uncertainty sets but generated by same sampling method, the $G_p(X_p)$ is a marginal distribution of

variable x_p. Similar strategy for the estimated expectation of the conditional posterior distribution is also mentioned in [65].

The conditional posterior distribution of equation 7.4 is difficult to solve analytically because of the quadratic term in the equation. Fortunately, the computational cost is cheap to solve the equation by numerical method. In our experiment, we implement the quasi-Monte Carlo sampling strategy to approximate the integration of the equation 7.4. The estimated expectation of the conditional posterior distribution is given as:

$$\begin{aligned} E_{X_p}[E_{X_{-p}}(f_*|X_p)^2] &= \frac{1}{N}\sum_{i=1}^{N}\int_{\chi_{-p}}\int_{\chi'_{-p}}\eta(x_*)\eta(x'_*)\mathrm{d}G_{-p|p}(X_{-p}|x_{pi})\mathrm{d}G_{-p|p}(X'_{-p}|x_{pi}) \\ &= \frac{1}{N}\sum_{i=1}^{N}\eta(X_*|x_{pi})\eta(X'_*|x_{pi}) \end{aligned} \tag{7.5}$$

where X_p is a marginal distribution and its sampling set is defined as:

$$X_p = [x_{p1}, x_{p2}, ..., x_{pi}, ..., x_{pN}] \tag{7.6}$$

The distribution is also a uniform distribution; N is the dimension of the X_p; X_* and X'_* are two independent uncertainty sets under the condition of X_p. These two sets are applied to approximate the inter dual integrations in the above function. This approximation based on the independency of each uncertainties. The estimated expectation of the conditional posterior distribution then is easy to applied for the numerical calculation in practice. However, the quality of the estimated expectation is highly depended on the training points and inferences for the meta-model.

The posterior distribution of S_i seems difficult to obtain. Nevertheless, the $E(f_*)^2$ in equation 7.3 and $Var(f_*)$ is easy to obtain. Just simply divide the variance of expected conditional posterior distribution of equation 7.3 by $Var(f_*)$, the approximated first order sensitivity indices are achieved.

For the estimated total sensitivity indices, the most important part is to derive the variance of expected posterior distribution under the condition of X_{-p} : $Var_{X_{-p}}[E_{X_p}(f_*|X_{-p})]$. This variance of expected conditional posterior distribution has an exactly same format as the equation 7.3. Just simply switch the position of X_p and X_{-p}, the expression of

the $Var_{X_{-p}}[E_{X_p}(f_*|X_{-p})]$ is achieved:

$$\begin{aligned} Var_{X_{-p}}[E_{X_p}(f_*|X_{-p})] &= E_{X_{-p}}[E_{X_p}(f_*|X_{-p})^2] - E_{X_{-p}}[E_{X_p}(f_*|X_{-p})]^2 \\ &= E_{X_{-p}}[E_{X_p}(f_*|X_{-p})^2] - E(f_*)^2 \end{aligned} \tag{7.7}$$

Then the $E_{X_{-p}}[E_{X_p}(f_*|X_{-p})^2]$ is expressed as:

$$\begin{aligned} &E_{X_{-p}}[E_{X_p}(f_*|X_{-p})^2] \\ &= \int_{\chi_{-p}} \int_{\chi_p} \int_{\chi_p^{'}} \eta(x_*)\eta(x_*^{'}) \mathrm{d}G_{p|-p}(X_p|X_{-p}) \mathrm{d}G_{p|-p}(X_p^{'}|X_{-p}) \mathrm{d}G_{-p}(X_{-p}) \end{aligned} \tag{7.8}$$

To calculate the $E_{X_{-p}}[E_{X_p}(f_*|X_{-p})^2]$ numerically, we need to approximate the integration by discrete formate. Thus, the estimated expectation of the posterior distribution under the condition of X_{-p} is expressed as:

$$\begin{aligned} E_{X_{-p}}[E_{X_p}(f_*|X_{-p})^2] &= \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \int_{\chi_{-p}} \eta(x_*|X_{-p}, x_{pi}) \eta(x_*|X_{-p}, x_{pj}^{'}) \mathrm{d}G_{-p}(X_{-p}) \\ &= \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \eta(X_*|X_{-p}, x_{pi}) \eta(X_*|X_{-p}, x_{pj}^{'}) \end{aligned} \tag{7.9}$$

where the X_p and $X_p^{'}$ are two independent uniform sampling sets. This approximation is applicable only when every uncertainties are independent to each.

7.3 Example 1: with 15 dimensions

The test subject model in our experiment is the VEGA launch vehicle. The model is known as a complex nonlinear model from previous chapters. Thus, the true effects of each uncertainty are not able to be calculated analytically. One of several proper choices for analyzing the quality of probability sensitivity analysis in our case is to compare the probability sensitivity indices to the sensitivity indices obtained by a variance-based method, introduced in the previous chapter, based, in turn, on the case of 15 uncertainties. Therefore, the dimension of the VEGA launch vehicle is reduced to 15 manually, with the selected uncertainty parameters listed in Table 6.1 in the previous chapter. Another choice for performing the quality analysis of the probability sensitivity indices is to apply the most important variables that are sorted out by the probability sensitivity

indices for worst-case validation. The result of the worst-case validation is also a good indicator of the quality of the probability sensitivity analysis.

7.3.1 Model training

The GP model process used for the probability sensitivity analysis in this chapter is same as the model training process used for GP-based optimization mentioned in Chapter 5. Thus, the details of the model training process are not introduced here.

With the inferences and selected hyper-parameters, the meta-model can be obtained. As we will investigate the variation of time spent in the training process based on different numbers of training points, the number of iterations that are involved in the operation of maximizing the log marginal likelihood is also able to affect the time spent in the training process. In our experiment, we fixed this parameter at 300. To examine the quality of the model, the amount of testing data is 10000 in our case.
Figure 7.1 presents the quality measured by Q_p versus the different numbers of training points. The time that is spent on the calculation of quality is also given in the figure. It can be seen from this figure that when the number of training points is smaller than

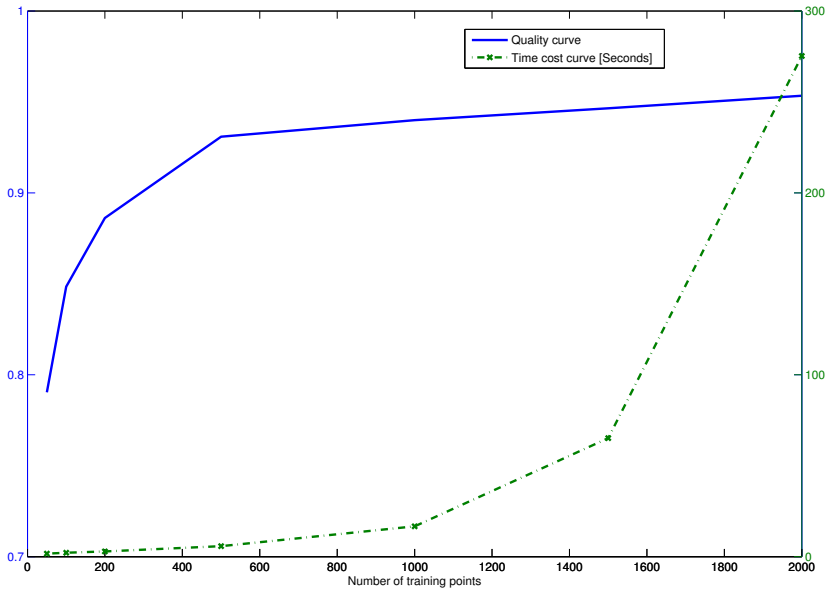


FIGURE 7.1: The quality and time cost for model training based on different number of training points

500, the quality of the training model is quickly improved when the number of training points is increased. However, when the number of training points is larger than 500, the quality improves, but at a slow rate. In contrast to the quality curve, the time-cost curve exhibits a very slow rate of increase before the number of samplings reaches 1000. However, then, after the number of training points is greater than 1000, the time spent on model training increases exponentially One aim of probability sensitivity analysis is that it expects a cheap computation cost to obtain the Sobol indices. Thus, an ideal trade-off number of training points for acceptable quality and affordable computational cost is between 500 and 1000. In this case, it was decided that 500 training points would be applied to the Gaussian process.

7.3.2 Results of probability sensitivity analysis

With selected number of training points and the inferences for mean and cost functions, the quality of Gaussian process measured by the equation 5.1 is :

$$Q_p(f, f_*) = 0.93089$$

This quality is satisfied by the minimum requirement of the meta-model. According to the meta-model with respect to this quality, the probability sensitivity indices are obtained and given in Table 7.1. The number of points in the estimated predicted posterior distribution is 10000. To calculate the probability sensitivity indices, we propose 10 independent experiments to obtain 10 results. The final results of the probability sensitivity indices are the mean of the 10 results for each uncertain variable.
Note that the cost function f_1 mentioned in this chapter refers to the maximum dynamic loads. The details of cost functions can be found in Chapter 4.2.1. From the the estimate of the first-order sensitivity indices $\hat{S}_i$ and the first-order sensitivity indices based on the variance-based sensitivity analysis in Table 6.2, the results of the first-order probability sensitivity indices are very good, although the two most important variables, 'Flag.dTc' and 'Flag.air_density_scat ' identified by S_i, are overweighted (the sum of first-order probability sensitivity indices for these two uncertainties are almost equal to 1. Theoretically, the sum of all first-order sensitivity indices are equal or smaller

Variable Names	$\hat{S}_i$	$\hat{S}_{Ti}$
Flag.IRSmountingX	0.0133	0.0027
Flag.IRSmountingY	0.0731	0.1759
Flag.IRSmountingZ	0.0030	0.0751
Flag.dISP	0.0051	0.0029
Flag.dTc	0.5329	0.8073
Flag.disp_CA	0.0035	0.0074
Flag.disp_CN	0.0308	0.0491
Flag.disp_Xcp	0.03309	0.1734
Flag.air_density_scat	0.4241	0.3946
Flag.h_wind	0	0
Flag.azimuth_wind_angle	0.0034	0
Flag.unc_CA	0.0007	0
Flag.unc_CN	0.0333	0.0645
Flag.unc_Xcp	0.0284	0
Flag.SRM_roll	0.0639	0.01725

TABLE 7.1: The probability sensitivity index based on f_1(The maximum dynamic loads)

than 1), and another important variable, 'Flag.SRM_roll', identified by S_i is slightly underestimated. The reason for the overweighted and underweighted variables could be diversity, but the actual reason is not easy to determine. For instance, the reason for the underweighted variable 'Flag.SRM_roll' could be that the linear projection function $\phi(x)$ is unable to accurately describe the nonlinear tendency of its scatter plot in Figures 6.3, 6.4 ,and 6.5; or it could be the quality of the meta-model. However, the results for the first-order probability are acceptable because the most important variables are identified and the effects of the ranks of each uncertainty used for dimension reduction are insignificant. However, from the total probability sensitivity indices, the results are not optimistic. The weight of the unimportant variable ' Flag.IRSmountingY' seems to be overestimated compared to the results obtained by variance-based sensitivity analysis, while the weight of variable 'Flag.SRM_roll' is heavily underestimated. Moreover, the value of the total sensitivity indices for the uncertainties 'Flag.unc_CA','Flag.h_wind', 'Flag.azimuth_wind_angle', and 'Flag.unc_Xcp' are negative. However, the values of the sensitivity indices are non-negative as mentioned in the previous chapter, and we must assign zero value to the total probability sensitivity indices of these uncertainties. Generally speaking, the quality of the total probability sensitivity indices is not as good as that of the first-order probability sensitivity indices. Thus, only the results of the first-order probability sensitivity indices are applied for dimension reduction in sequel.

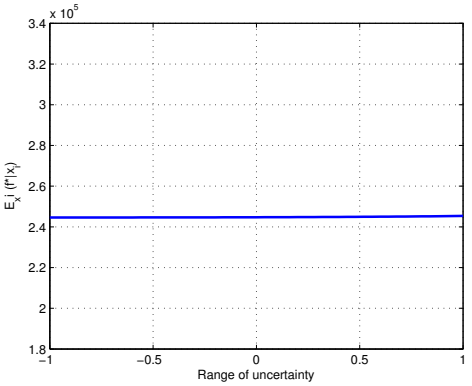
The ranks of each uncertainties based on the first order probability sensitivity indices are given in the Table 7.2 followed. The ranks are used to re-order the sequence of an uncertainty vector. From the variance-based sensitivity analysis discussed in the previous

Variable Names	Order/ Rank
Flag.IRSmountingX	9
Flag.IRSmountingY	3
Flag.IRSmountingZ	13
Flag.dISP	10
Flag.dTc	1
Flag.disp_CA	11
Flag.disp_CN	7
Flag.disp_Xcp	6
Flag.air_density_scat	2
Flag.h_wind	15
Flag.azimuth_wind_angle	12
Flag.unc_CA	14
Flag.unc_CN	5
Flag.unc_Xcp	8
Flag.SRM_roll	4

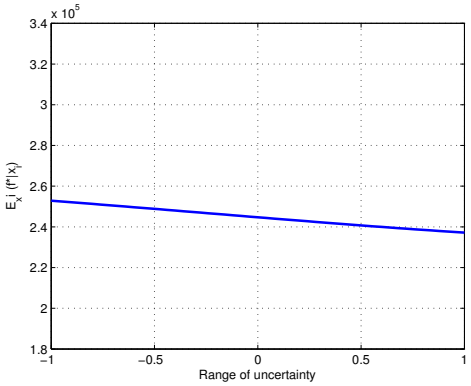
TABLE 7.2: Re-ordered sequence by ranks

chapter, we know that the scatter plots (Figure 6.3, 6.3, and 6.5) provide the 'tendency' information about an uncertainty. The 'tendency' can be treated as the expected objective values evaluated against the conditional variable x_i associated with a variance of noise. From this chapter, it is easy to understand that the concept of 'tendency' actually is the expected conditional posterior distribution. The expected conditional posterior distribution actually measures the ratio between an uncertainty and the predicted objective values. Thus, we can also provide a figure that contains the information about the 'tendency' for each uncertain variable. The plots of expected conditional posterior distribution $E_i(f_*|X_i)$ are given in Figures 7.2, 7.3, and 7.4 below.

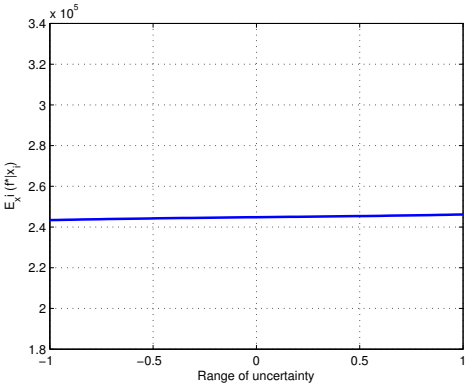
From the Figure 7.4, most of the expected conditional posterior distributions plotted against each uncertain parameter are very similar to the 'tendency.' The greatest difference is in the expected posterior distribution measured against the variable 'Flag.SRM_roll' linear, whereas the 'tendency' of this variable in the scatter plot is nonlinear. The most likely explanation for this difference is that the inferences for projection functions $\phi(x)$ are all linear and are not able to describe the nonlinear feature of the



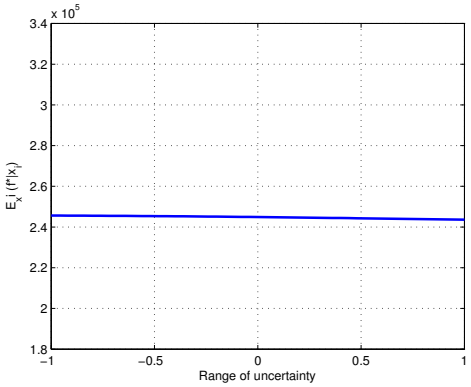
(A) IRSmountingX vs. f_{1*}



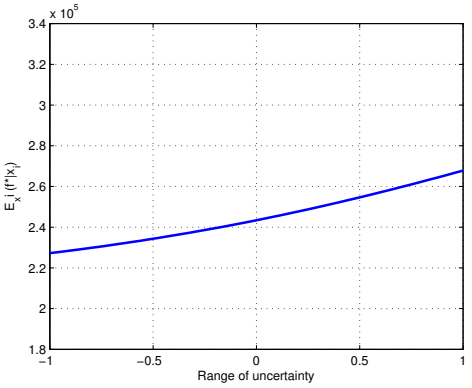
(B) IRSmountingY vs. f_{1*}



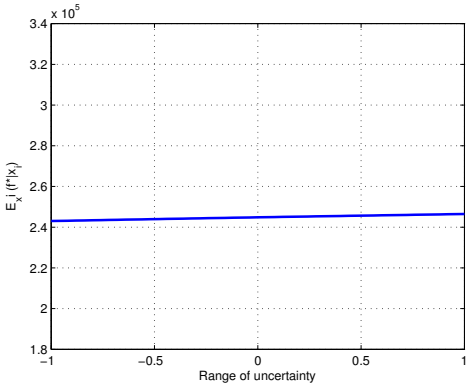
(C) IRSmountingZ vs. f_{1*}



(D) dISP vs. f_{1*}



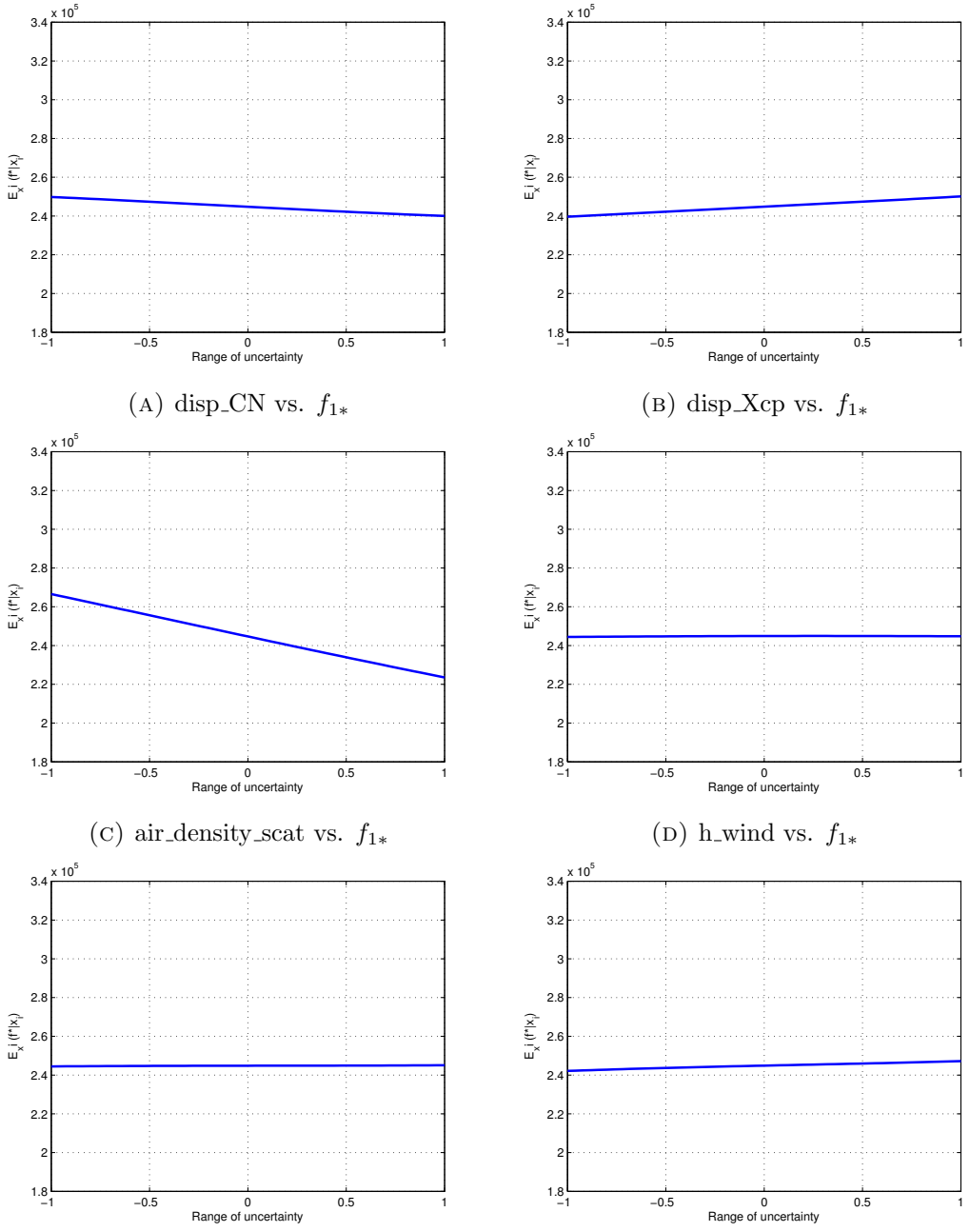
(E) dTc vs. f_{1*}



(F) disp_CA vs. f_{1*}

FIGURE 7.2: The expected posterior of $E_{X_i}(f_*|X_i)$ against the condition variable x_i of f_1 - Part A

variables. However, in practice, we can obtain the 'tendency' information for each variable by carrying out a relatively large set of sampling points. Then, the 'tendency' information can help the model trainer to examine the quality of the expected conditional posterior distribution of each variable. However, the difficulty of finding a proper projection function to describe the nonlinear 'tendency' is actually quite significant.



(A) disp_CN vs. f_{1*}

(B) disp_Xcp vs. f_{1*}

(C) air_density_scat vs. f_{1*}

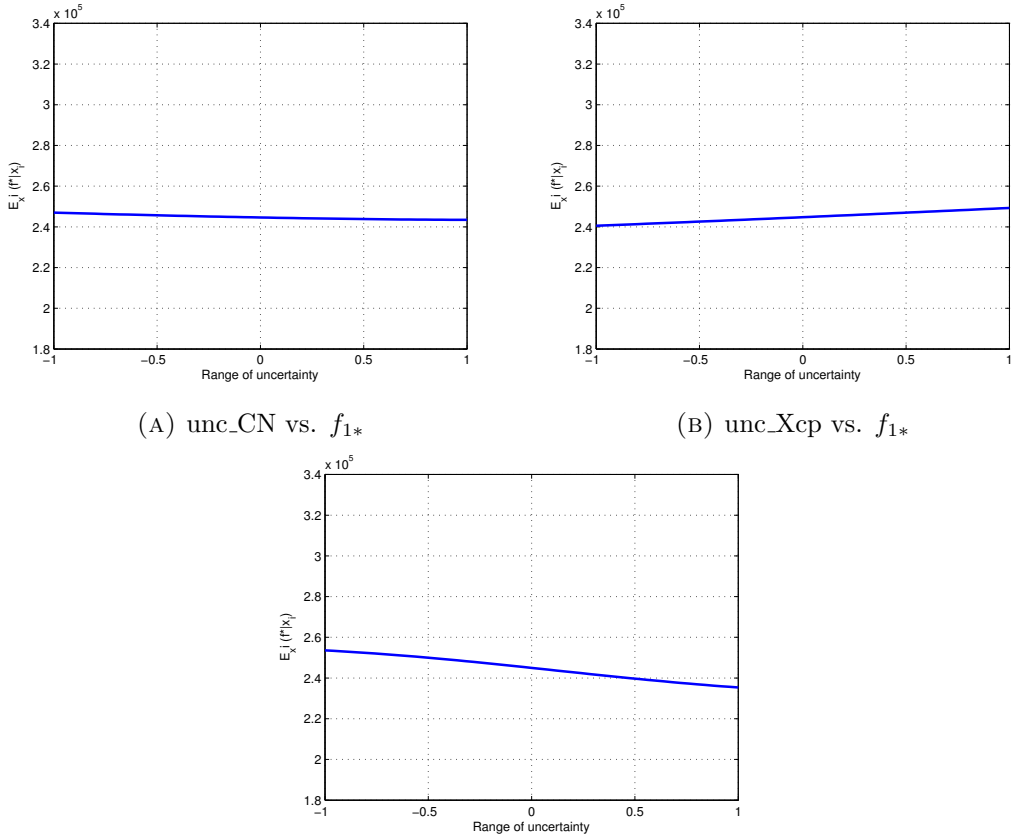
(D) h_wind vs. f_{1*}

(E) azimuth_wind_angle vs.f_{1*}

(F) unc_CA vs. f_{1*}

FIGURE 7.3: The expected posterior of $E_{X_i}(f_*|X_i)$ against the condition variable x_i of f_1 - Part B

To verify the results of the variance-based sensitivity analysis discussed in the previous chapter, the seven uncertain variables of ranks from 1 to 7 are selected as the 'important variables,' and are assigned to the worst-case validation. Although the number of important variables does not have to be seven, since we have not investigated this point further, our focus is elsewhere at this moment. To maintain consistency with the worst-case validation process discussed in the previous chapter, we keep the top seven



(A) unc_CN vs. f_{1*}

(B) unc_Xcp vs. f_{1*}

(C) SRM_roll vs. f_{1*}

FIGURE 7.4: The expected posterior of $E_{X_i}(f_*|X_i)$ against the condition variable x_i of f_1 - Part C

uncertainties with the highest values of the first-order probability sensitivity indices. In this case, two worst-case validations are carried out to verify the quality of the sensitivity indices. The first worst-case validation employs the seven uncertainties sampled using a Sobol sequence, whereas the second worst-case validation uses a hybrid sampling strategy (see the previous chapter for details) to sample the seven uncertainties. Note that the large ratios obtained from the expected conditional posterior distribution are considered in the sampling strategy to narrow down the sampling range of the corresponding uncertainties. In this case, only the variables 'Flag.dTc' and 'Flag.air_density_scat' exhibit the obvious large ratio.

The results of the two worst-case validations with a reduced dimension of uncertainties are given in Table 7.3. The results of the two worst case validations with reduced dimension of uncertainties are given in the Table 7.3 followed. From the Table 7.3, the result of test 1 obtains a slightly smaller worst-case value than the result obtained using no dimension-reduction method, whereas the result of test 2 has a slightly larger worst-case

Tests	Worst-case (Max Q_α [Pa Deg])
No application of dimension reduction method	3.2469×10^5
Test 1 (Sobol sequence for the 7 uncertainties)	3.2242×10^5
Test 2 (hybrid sampling for the 7 uncertainties)	3.2809×10^5

TABLE 7.3: Worst-case validation based on dimension reduction methods for objective function f_1

value than the result involving no dimension-reduction method. These results indicate that the neglected variables do not significantly affect the objective function. The probability sensitivity indices provide good information about the weight of the variables, and the distribution properties of the original ELV launch vehicle model is successfully reserved by the statistical meta-model based on 15 uncertainties.

7.4 Example 2: with expanded dimension

As the probability sensitivity analysis is demonstrated as a feasible method for dimension reduction, we would like to implement this method for sensitivity analysis based on the a larger number of uncertainties. The driven variables identified from the 15 uncertainties case could have a smaller value of sensitivity indices, but the value are still relative large. Due to the expanded dimension, the difficultly and complexity of training the meta-model by Gaussian process are increased inevitably. To study a probability sensitivity analysis with a larger dimension of uncertainties, we do not directly introduce the entire 118 uncertainties but expand the dimension of uncertainties step by step. The reasons are: (1) Training the statistic model with large dimensions becomes much more complicated; (2) A small number of expanded uncertainties has fewer chance to completely change the weigh of the uncertainties in sensitivity indices. As we do not have the empirical weight of the expanded uncertainties, it would be much easier to verify the quality of the sensitivity analysis with the large dimension of uncertainties. In this example, the number of dimension of uncertainties is doubled compared to example 1. The uncertainties involved in the simulation is given is the Table 7.4 followed.

Category	**Variable name in VEGACONTROL**	**Description**
IRS	Flag.IRSmountingX	IRS Mounting Error w.r.t X Body Axis
	Flag.IRSmountingY	IRS Mounting Error w.r.t Y Body Axis
	Flag.IRSmountingZ	IRS Mounting Error w.r.t Z Body Axis
Thrust Parameters	Flag.dISP	Stage 1 impulse scattering
	Flag.dTc	Scattering on time burn
	Flag.SRM_roll	Scattering on P80 Roll Torque
Aerodynamics	Flag.disp_CA	Dispersion on Stage 1 Axial coefficient
	Flag.unc_CA	Uncertainty on Stage 1 Axial coefficient
	Flag.disp_CN	Dispersion on Stage 1 Normal Coefficient
	Flag.unc_CN	Uncertainty on Stage 1 Normal Coefficient
	Flag.disp_Xcp	Dispersion on Stage 1 Xcp
	Flag.unc_Xcp	Uncertainty on Stage 1 Xcp
Atmosphere	Flag.air_density_scat	Atmosphere Density
wind	Flag.h_wind	Synthetic wind gust altitude
	Flag.azimuth_wind_angle	Wind azimuth direction
Bending modes	Flag.flex_freq	Scattering on Bending frequencies
	Flag.TMC_PVP	Scattering on Translation at Pivot Point
	Flag.RMC_PVP	Scattering on Rotation at Pivot Point
	Flag.TMC_INS	Scattering on Translation at INS
	Flag.RMC_INS	Scattering on Rotation at INS

Continued on next page

Table 7.4 – *Continued from previous page*

Category	**Variable name in VEGACONTROL**	**Description**
Thrust offset and misalignment	Flag.TVC_SF_A	Scattering on TVC gain Lane A
	Flag.TVC_bias_A	Scattering on TVC Lane A
	Flag.TVC_SF_B	Scattering on TVC gain Lane B
	Flag.TVC_bias_B	Scattering on TVC Lane B
	Flag.Thrust_misA	Scattering on thrust misalignment first lane
	Flag.Thrust_misB	Scattering on thrust misalignment second lane
	Flag.PvP_offsetX	Scattering on thrust offset in X
	Flag.PvP_offsetY	Scattering on thrust offset in Y
	Flag.PvP_offsetZ	Scattering on thrust offset in Z
Others	Flag.backlash	Scattering on backlash

TABLE 7.4: The description, category and variable name in VEGACONTROL of the 30 selected uncertainties

The inferences for mean function, covariance function and prior are exactly same as the test case with 15 uncertainties. In the meantime, the training points are also sampled by Sobol sequence. The optimal combination of hyperparameters are obtained by employing the local optimization algorithm to find the minimal negative marginal likelihood as well. The quality of the meta-model is measured by the equation 5.1. Due to the increased dimension of uncertainties, more training points are needed in order to obtain a good quality level compared to the test case 1 with only 15 uncertainties. The quality and time cost versus the different number of training points are given in the Figure 7.5 below. From the figure, the quality of the statistic model grows very quickly in the first 500 training points. In the meanwhile, the time cost of training the model is largely increased when the number of training points exceeds 500. Thus, a good candidate number of training points is 500. The quality of the model with the 500 training points reaches the acceptable level and the times cost is affordable as well. However, we have to

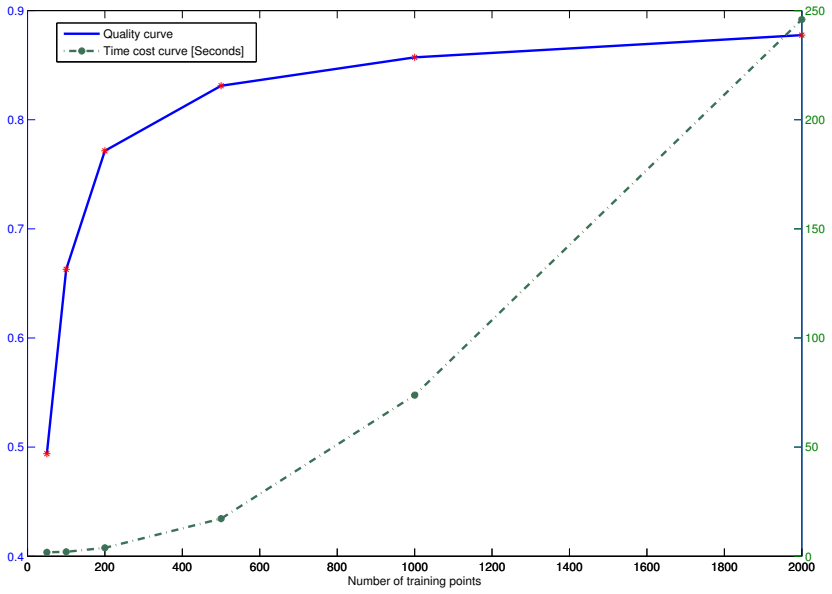


FIGURE 7.5: The quality and time cost for model training based on different number of training points

notice that the qualities for all candidate number of training points with 30 uncertainties are smaller than the qualities with same candidate number of training points when only 15 uncertainties are involved in the simulation in a certain degree. It shows the expanded uncertainties obviously raise the complexity level of training a statistic model. If there was no better inferences for Gaussian Process or distribution for training points, the quality of the meta-model with the entire 118 uncertainties based on the current inferences and sampling strategy is highly possible to drop below the acceptable quality level (quality value is smaller than 0.8). In this case the quality of the meta-model with 500 training points is:

$$Q_p(f, f_*) = 0.8311$$

According to this quality of statistic mode, the first order of probability sensitivity indices is given in the Table 7.5 followed. As the total probability sensitivity indices with 15 uncertainties is not in acceptable accuracy, the total probability sensitivity indices with 30 uncertainties is not expected to achieve a good results. So the total probability sensitivity indices is abandoned from the table. The drawback of abandoning the total probability sensitivity indices is that the correlation between the uncertainties are unknown. Some uncertainties with low main effects but strong interaction effect other uncertainties are not able to be identified as driven variables. Because the same

Variable Names	$\hat{S}_i$
Flag.IRSmountingX	0.0017
Flag.IRSmountingY	0.0703
Flag.IRSmountingZ	1.4224×10^{-4}
Flag.dISP	0.0331
Flag.dTc	0.4485
Flag.disp_CA	0.0231
Flag.disp_CN	0.0245
Flag.disp_Xcp	0.0411
Flag.air_density_scat	0.4306
Flag.h_wind	0.0028
Flag.azimuth_wind_angle	0.0020
Flag.unc_CA	0.0123
Flag.unc_CN	0.0105
Flag.unc_Xcp	0.0031
Flag.SRM_roll	0.1056
Flag.flex_freq	0.0276
Flag.TMC_PVP	0.0109
Flag.RMC_PVP	0.0026
Flag.TMC_INS	0.0094
Flag.RMC_INS	0.0364
Flag.TVC_SF_A	0.0201
Flag.TVC_bias_A	0.0006
Flag.TVC_SF_B	0.0288
Flag.TVC_bias_B	0.0336
Flag.Thrust_misA	0
Flag.Thrust_misB	0.0102
Flag.PvP_offsetX	0.0063
Flag.PvP_offsetY	0
Flag.PvP_offsetZ	0
Flag.backlash	0.0084

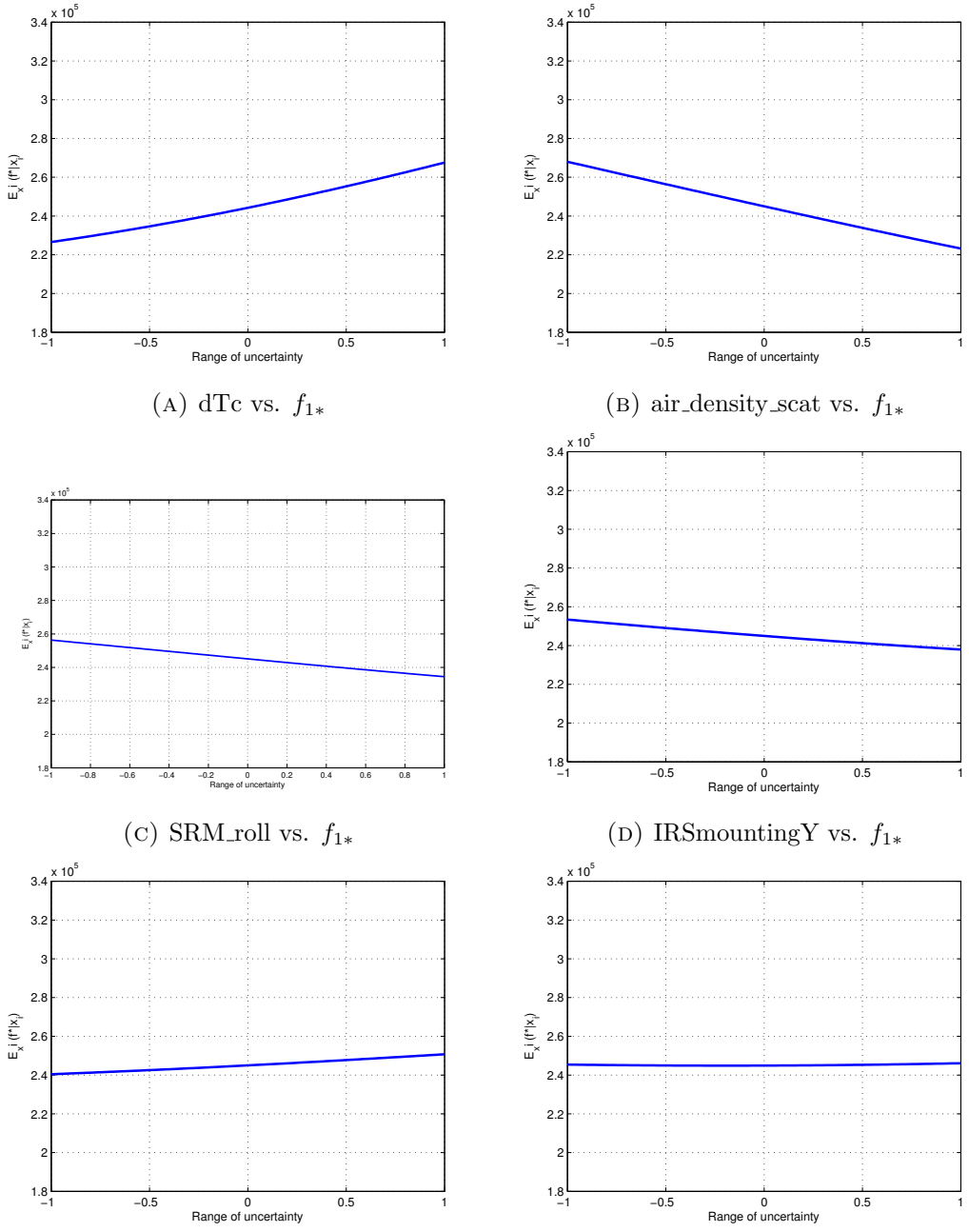
TABLE 7.5: The probability sensitivity index based on f_1 with expanded uncertainties

inference and sampling strategy is employed in this case, all the approximation error is reserved. From the table, the two variables 'Flag.dTc' and 'Flag.air_density_scat' are still over weighted. But it doesn't affect the results of sensitivity analysis. Generally speaking, the driven variables that are identified based on the 15 uncertainties case still have a relative large value of first order sensitivity indices. This indicates the results of sensitivity analysis based on the current quality of meta-model is acceptable and should be able to explain the importance of each uncertainty of its main effect to the objective function(maximum dynamic loads). The rank of each uncertainty is given in the Table 7.6 below. The rank of each uncertainties are used to re-order the

Variable Names	Rank/Order
Flag.IRSmountingX	25
Flag.IRSmountingY	4
Flag.IRSmountingZ	27
Flag.dISP	8
Flag.dTc	1
Flag.disp_CA	12
Flag.disp_CN	11
Flag.disp_Xcp	5
Flag.air_density_scat	2
Flag.h_wind	22
Flag.azimuth_wind_angle	24
Flag.unc_CA	14
Flag.unc_CN	16
Flag.unc_Xcp	21
Flag.SRM_roll	3
Flag.flex_freq	10
Flag.TMC_PVP	15
Flag.RMC_PVP	23
Flag.TMC_INS	18
Flag.RMC_INS	6
Flag.TVC_SF_A	13
Flag.TVC_bias_A	26
Flag.TVC_SF_B	9
Flag.TVC_bias_B	7
Flag.Thrust_misA	28
Flag.Thrust_misB	17
Flag.PvP_offsetX	20
Flag.PvP_offsetY	28
Flag.PvP_offsetZ	28
Flag.backlash	19

TABLE 7.6: The rank of each variable based on f_1 with expanded uncertainties

sequence of sampling points. From the table of tank, we notice that three uncertainties in the set of expanded uncertainties are ranked below 10. To further study the these uncertainties, the uncertainties with the rank no greater than 10 are selected and involved in the simulation in this case. Moreover, the expected posterior distributions against each uncertainty are able to provide the 'tendency' information, we will investigate the expected conditional posterior distribution for the selected 10 uncertainties. However, to keep the consistent with previous test case, 15 plots of expected conditional poster ion distribution are given in the Figure 7.6 , Figure 7.7 and Figure 7.8 followed.



(A) dTc vs. f_{1*}

(B) air_density_scat vs. f_{1*}

(C) SRM_roll vs. f_{1*}

(D) IRSmountingY vs. f_{1*}

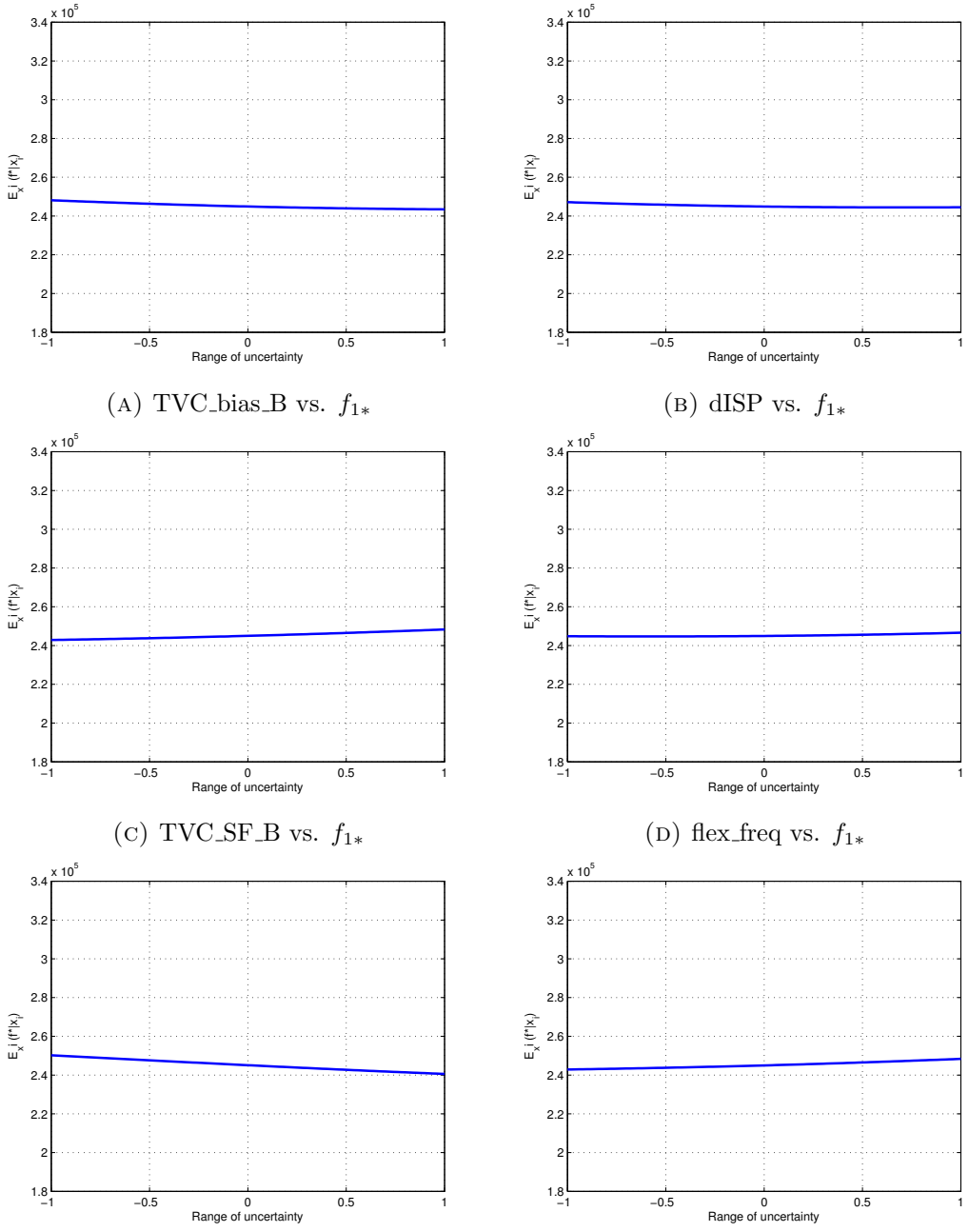
(E) disp_Xcp vs. f_{1*}

(F) RMC_INS vs. f_{1*}

FIGURE 7.6: The expected posterior of $E_{X_i}(f_*|X_i)$ against the selected variable x_i of f_1 - Part A

From the this figure, the uncertainties 'Flag.dTc' and 'Flag.air_density_scat' have obvious increased and decreased ratios respectively. The ratios of uncertainties 'Flag.SRM_roll' and 'Flag. IRSmountingY' are not significant, but are still considerable. These linear increased or decreased information can be applied to the simulation to narrow down the range of distribution.

To verify the quality of the probability sensitivity analysis, the worst case validation is



(A) TVC_bias_B vs. f_{1*}

(B) dISP vs. f_{1*}

(C) TVC_SF_B vs. f_{1*}

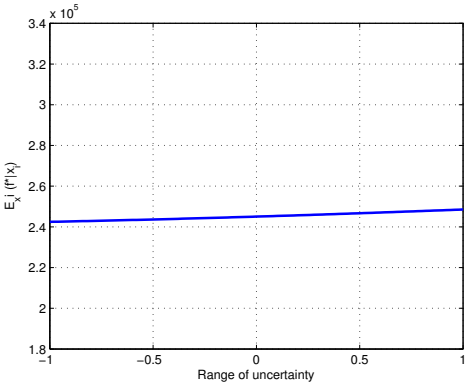
(D) flex_freq vs. f_{1*}

(E) disp_CN vs.f_{1*}

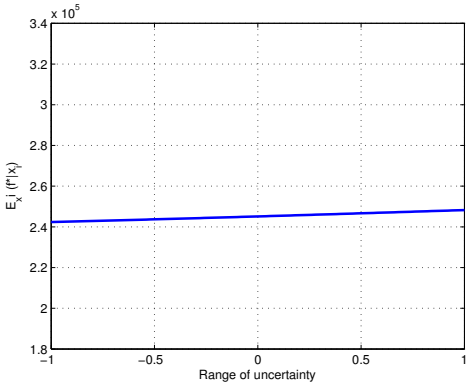
(F) disp_CA vs. f_{1*}

FIGURE 7.7: The expected posterior of $E_{X_i}(f_*|X_i)$ against the selected variable x_i of f_1 - Part B

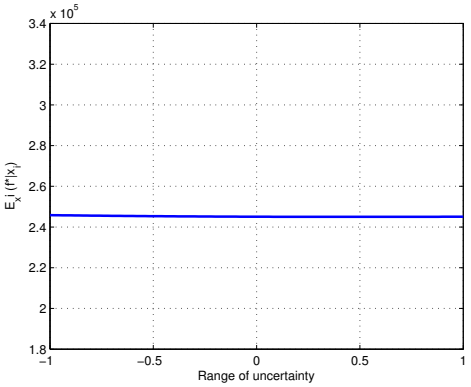
employed based on the selected driven variables. In this case, two sampling strategies are implemented to sample the 10 driven uncertainties: Sobol sequence and hybrid sampling. The two sampling strategies are exactly same as the strategies used in previous test case. The results of worst case validation based on these two sampling strategies are given in the Table 7.7 below. From the table, the results of worst case validation based on the reduced dimension have little difference with the result of worst case validation based



(A) TVC_SF_A vs. f_{1*}



(B) unc_CA vs. f_{1*}



(C) TMC_PVP vs. f_{1*}

FIGURE 7.8: The expected posterior of $E_{X_i}(f_*|X_i)$ against the selected variable x_i of f_1 - Part C

Tests	Worst-case (Max Q_α [Pa Deg])
No application of dimension reduction method	3.3658×10^5
Test 1 (Sobol sequence for the 10 uncertainties)	3.3416×10^5
Test 2 (hybrid sampling for the 10 uncertainties)	3.3090×10^5

TABLE 7.7: Worst-case validation based on dimension reduction methods for objective function f_1

on the entire 30 uncertainties. This means the neglected uncertainties have little effects to the objective function. The 10 selected variables can be considered as the driven variables. The probability sensitivity analysis is successfully identify the important variables and the quality of the probability first order sensitivity indices is acceptable. Note that the results obtained in Table 7.7 are larger than the results in Table 7.3 based on the 15 uncertainties for each individual case of worst case validation. This indicates that there exists some driven variables in the set of expanded uncertainties and these uncertainties are successfully identified by the probability sensitivity analysis.

7.5 Example 3: with all 118 uncertainties

When all the 118 uncertainties are involved in the simulation , the complexity to get a high quality meta-model is increased significantly. When the same inferences and sampling strategy are applied as the previous examples, the quality of the meta-model based on the number of training points is expected to be much lower. If the value of quality of the meta-model is below the acceptable value, the probability first order sensitivity indices may not be able to give a good weight of each variable.

In this case, the same inferences and sampling strategy are implemented as well, the quality and time cost versus the different number of training points are given in the Figure 7.9 followed. After 1000 sampling points, the quality of the statistic model is

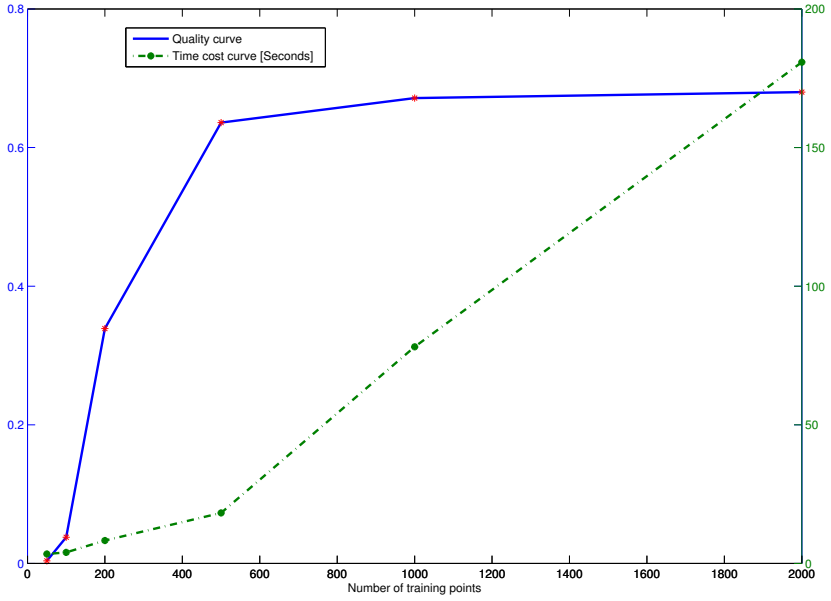


FIGURE 7.9: The quality and time cost for model training based on different number of training points

improved very little by increasing the number of sampling points; on the other hand, the time cost for the model training is increased significantly. From this figure, it indicates that simply increasing the number of training points is not a effective solution to improve the quality of the meta-model. In our experiment, we tried 6000 training point for model generation. The quality we obtained is 0.72. This quality value is still lower than the minimum acceptable quality value. Moreover, the model training process takes approximate 2 hours to complete. There is also no way to keep increasing the number

of training points due to the capability of the hardware. The Matabl would display an error message if the number of training points exceeds 8000 to show that the Matlab is out of memory. In this case, the meta-model applied to sensitivity analysis is trained based on 1000 training points. The corresponding quality of this model is:

$$Q_p(f, f_*) = 0.67144$$

As this quality of model doesn't satisfy the minimum quality level, the results of probability sensitivity analysis may not be very good. However, whether this model would be rejected or not is depended on the results of sensitivity indices. The Table 7.8 below gives the probability first order sensitivity indices in terms of the meta-model with above quality. To keep a good reading interface, only the selected fifteen variables in the first test case are given in the table. This is enough for us to analyze the quality of model. From this table above, the uncertainties 'Flag.disp_CN',

Variable Names	$\hat{S}_i$
Flag.IRSmountingX	0.0014
Flag.IRSmountingY	0.0441
Flag.IRSmountingZ	0.0013
Flag.dISP	0.0025
Flag.dTc	0.4329
Flag.disp_CA	0.0023
Flag.disp_CN	0.0039
Flag.disp_Xcp	0.0037
Flag.air_density_scat	0.4811
Flag.h_wind	4.6356×10^{-4}
Flag.azimuth_wind_angle	0.0023
Flag.unc_CA	0.0077
Flag.unc_CN	0.0084
Flag.unc_Xcp	0.0210
Flag.SRM_roll	0.0533

TABLE 7.8: The probability sensitivity index for the selected 15 uncertainties based on f_1 with 118 uncertain parameters

'Flag.disp_Xcp' and 'Flag.SRM_roll ' are likely to be underestimated of weight compared to the sensitivity indices obtained based on the variance based method. The uncertainties 'Flag.dTc', 'Flag.air_density_scat', 'Flag.unc_CN' and 'Flag.unc_Xcp' seem to be overestimated. Among these seven uncertainties, the ranks of 'Flag.disp_CN', 'Flag.disp_Xcp', 'Flag.unc_CN' and 'Flag.unc_Xcp' are heavily affected. However, the uncertainties

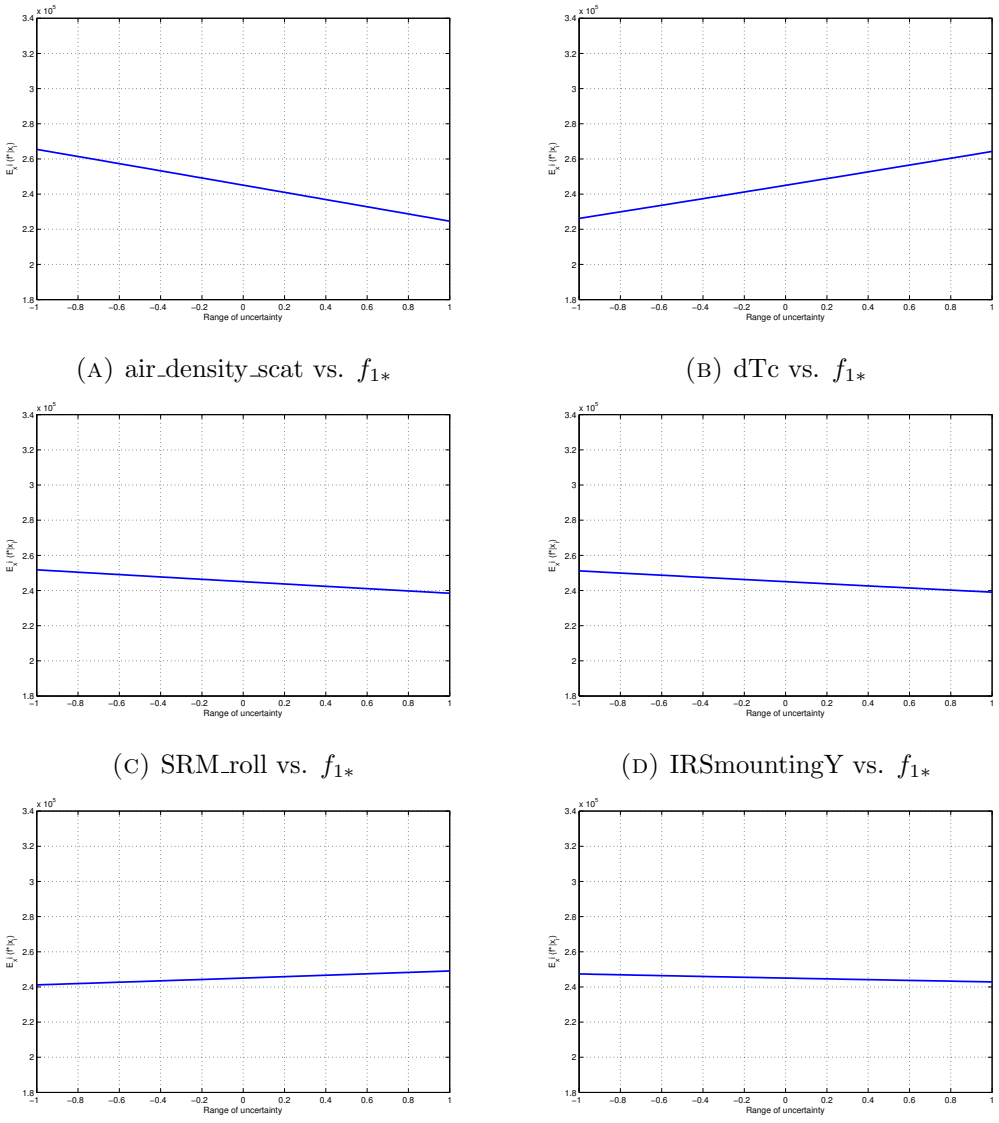
'Flag.IRSmountingY', 'Flag.dTc', 'Flag.air_density_scat' and 'Flag.SRM_roll' are still in the top ranks. Although the quality of the statistic model doesn't reach the acceptable quality level, but the most important variables are also identified. Thus, the worst case valuation would be able to give the answer that whether the result of probability sensitivity indices is acceptable or not. Before the worst case validation is carried out, we would like to identify the top 15 ranks of uncertainties over the entire 118 uncertain parameters. The Table 7.9 followed lists the weights and ranks for the top 15 uncertainties. Although the uncertainties 'Flag.unc_Xcp' and 'Flag.unc_CN' in this table are

Variable Names	$\hat{S}_i$	Rank
Flag.air_density_scat	0.4811	1
Flag.dTc	0.4329	2
Flag.SRM_roll	0.0533	3
Flag.IRSmountingY	0.0441	4
Flag.unc_Xcp	0.0210	5
Flag.unc_CN	0.0084	6
Flag.TVC_SF_A	0.083	7
Flag.RACSdxCOG	0.0080	8
Flag.unc_CA	0.0077	9
Flag.Z9dJy	0.0077	9
Flag.TVC_SF_B	0.0052	10
Flag.Z9dyCOG	0.0050	11
Flag.Z23dxCOG	0.0045	12
Flag.disp_CN	0.0039	13
Flag.disp_Xcp	0.0037	14

TABLE 7.9: The probability sensitivity index for the 15 top ranks of uncertainties based on f_1 with 118 uncertainties

highly possible overestimated and should not be identified as important uncertainties, the worst case validation would still employ these top 15 driven variables. The expected conditional posterior distribution $E_i(f_*|X_i)$ is given in the Figure 7.11 , Figure **??** and Figure 7.12 below.

From the figure, only the expected probability posterior plots on the condition of 'Flag.dTc' and 'air_density_scat' have a obvious 'tendency' pattern with a relatively large slope. Thus, the range of these two uncertainties can be narrowed down. For the worst case validation, since the entire 118 uncertainties are considered in the sensitivity analysis, five test cases are employed with different sampling strategies. Beside the two strategies (Sobol and hybrid sampling) used in previous test examples, the Monte Carlo



(A) air_density_scat vs. f_{1*}

(B) dTc vs. f_{1*}

(C) SRM_roll vs. f_{1*}

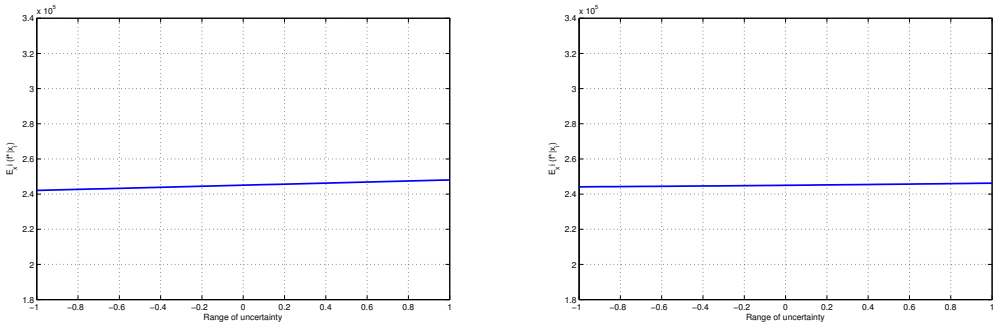
(D) IRSmountingY vs. f_{1*}

(E) unc_Xcp vs. f_{1*}

(F) unc_CN vs. f_{1*}

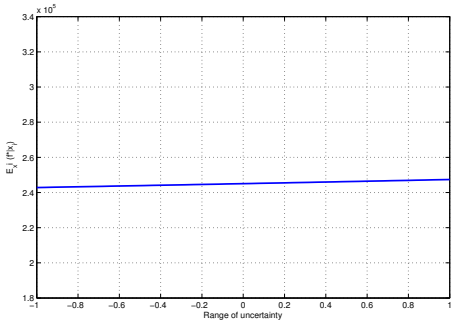
FIGURE 7.10: The expected posterior of $E_{X_i}(f_*|X_i)$ against the condition variable x_i of f_1 - Part A

method is also Implemented in this case. The extra cases are applied to compare the efficiency with each sampling method and keep consistent with the worst case validation described in Chapter 3 when all uncertainties are involved in the simulation. The Table 7.10 below contains the results of worst case validation. From the table above, the worst cases of the maximum dynamic loads during the ascending phase of ELV launch vehicle based on different sampling methods with the reduced dimensional space are larger than the worst cases obtained when all 118 uncertainties are involved in the simulation although some actually non-important variables are also identified as driven variables and assigned to the simulation. Especially, for the results obtained by Sobol sequence,

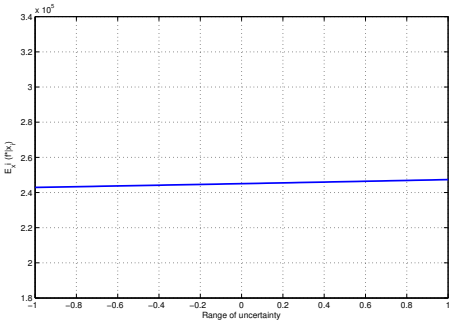


(A) TVC_SF_A vs. f_{1*}

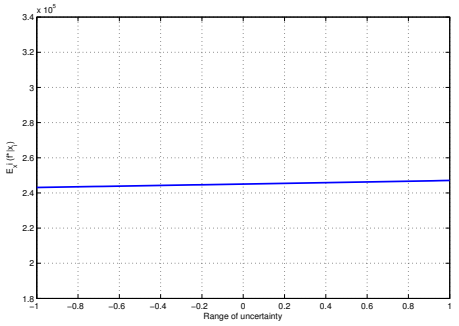
(B) RACSdxCOG vs. f_{1*}



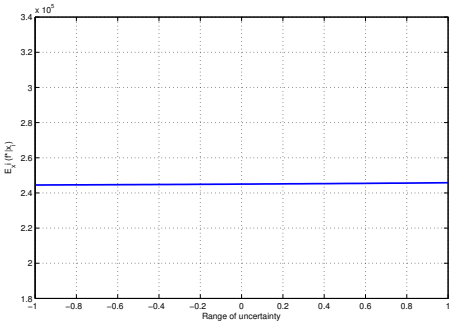
(C) unc_CA vs. f_{1*}



(D) Z9dJy vs. f_{1*}



(E) TVC_SF_B vs.f_{1*}

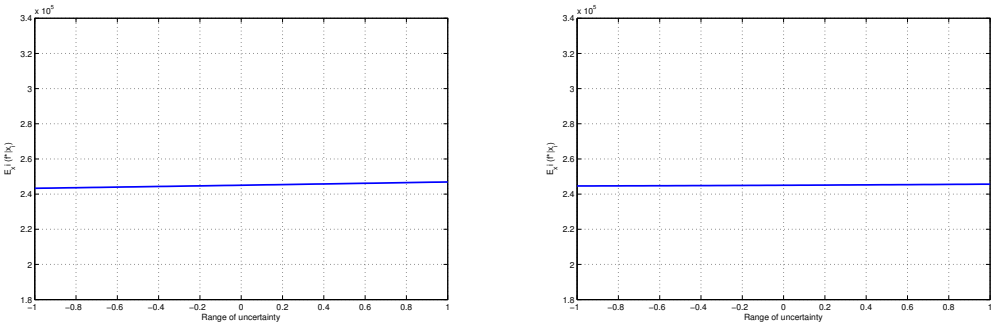


(F) Z9dyCOG vs. f_{1*}

FIGURE 7.11: The expected posterior of $E_{X_i}(f_*|X_i)$ against the condition variable x_i of f_1 - Part A

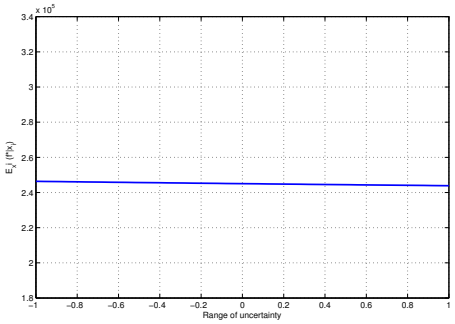
the improvement of worst case value is about 10 percentage. The efficiency of the sampling strategies are improved because of the reduced dimensions. Generally speaking, the quasi-Monte Carlo still has a better efficiency in worst case validation compared to Monte Carlo method in terms of the reduced dimensions. Nevertheless, the difference between the Monte Carlo and quasi-Monte Carlo methods of worst case values is still small based on the objective f_1 (maximum dynamic loads).

Because the quality of the meta-model trained by the 118 uncertainties doesn't satisfy the acceptable level, a larger worst case value is expected if the quality of the statistical model is improved. On the other hand, due to the results of the five worst case



(A) Z23dxCOG vs. f_{1*}

(B) disp_CN vs. f_{1*}



(C) disp_Xcp vs. f_{1*}

FIGURE 7.12: The expected posterior of $E_{X_i}(f_*|X_i)$ against the condition variable x_i of f_1 - Part C

Tests	Worst-case (Max Q_α [Pa Deg])
Monte Carlo method with 118 uncertainties	3.2002×10^5
Sobol sequence with 118 uncertainties	3.1297×10^5
Halton sequence with 118 uncertainties	3.3168×10^5
Faure sequence with 118 uncertainties	3.2658×10^5
Test 1 (Sobol sequence with the 15 uncertainties)	3.4910×10^5
Test 2 (Hybrid method with the 15 uncertainties)	3.4382×10^5
Test 3 (Monte Carlo method with the 15 uncertainties)	3.3836×10^5
Test 4 (Halton sequence with the 15 uncertainties)	3.3699×10^5
Test 5 (Faure sequence with the 15 uncertainties)	3.4474×10^5

TABLE 7.10: Worst-case validation based on dimension reduction methods for objective function f_1

validations are good and reasonable, the quality of the probability sensitivity analysis is acceptable.

From the three test cases, although the ranks of uncertainties are varied in each case, the uncertainties 'Flag.dTc', 'Flag.air_density_scat', 'Flag.SRM_roll', and 'Flag.IRSmountingY'

are always in the top four ranks. This phenomenon suggests that these four uncertainties are actually the most driven variables and are less sensitive to the varying of the quality of the meta-model compared to other uncertainties. The probability sensitivity analysis based on the statistical model trained using the Gaussian process successfully identifies the most weighted variables in general and facilitates the worst-case validation more efficiently. Moreover, the quasi-Monte Carlo method still has a slight advantage over the Monte Carlo method based on the objective function f_1. Although we expect a much more efficient sampling based on the quasi-Monte Carlo method in lower dimensions compared to the Monte Carlo method, the efficiency still could be negatively affected by the surface of the objective function. According to the worst-case results based on 118 uncertainties for all objective functions listed in Table 4.1, we believe the quasi-Monte Carlo method would behave much more efficiently than the Monte Carlo method based on the objective functions f_2, f_3, f_4, and f_5 if the probability sensitivity analysis is applied to screen out the driven uncertainties for each objective function. However, the time left for the research is limited and not sufficient for completing all of the sensitivity analysis based on different objective functions. To train the meta-model for other objective functions, only the same process used to train the meta-model based on f_1 needs to be followed. The only difficulty and difference are in the matters of making inferences for mean and covariance functions. The rest of the sensitivity analysis work could be left to future research. Nevertheless, the probability sensitivity analysis is verified as an efficient and powerful tool.

7.6 Short Summary

In this chapter, we deduced a novel inference for estimating the probabilistic-based first-order and total sensitivity indices. The results show that the first-order sensitivity index successfully identified the key parameters. On the other hand, the total sensitivity index is not good enough for identification. Nevertheless, our approach can be used to solve the problem that variance-based sensitivity analysis is barely able to handle. The estimation of sensitivity indices is based on 118 uncertain parameters rather than 15. Moreover, we only need roughly 1/40 of the computing time to obtain the results that variance-based sensitivity analysis requires.

Chapter 8

Conclusions and Future works

8.1 Conclusion

The robustness of the VEGA launch vehicle is assessed by the Monte Carlo, quasi-Monte Carlo, and optimization-based methods. Although all of these methods are able to find a worst-case value that violates its constraint in terms of different objective functions, the efficiency of each method is different.

From the Chapter 3, the efficiencies of Monte Carlo and quasi-Monte Carlo based worst case validation for a complex system are compared in the first place. From the theoretical definitions, the quasi-Monte Carlo should be more efficient than the Monte Carlo method for worst case validation, especially when the dimension of uncertainties is small or the number of sampling points is small. The results of worst case validation based on these two method with respect to different objective functions demonstrate that the quasi-Monte Carlo indeed obtains a larger worst-case value contrast to the worst-case values obtained by Monte Carlo method 'in general speaking'. However, we have to note that none of the three quasi-random sequences (Sobol, Halton and Faure sequence) is able to guarantee the efficiency for all objective functions over the Monte Carlo method. Monte Carlo method even obtains a larger mean worst-case value over worst-case value obtained by some of quasi-random sequences in many cases, although we normally can identify one or more quasi-random sequences obtain a larger worst-case value than the worst-case values obtained by Monte Carlo method in most circumstances. This is also the reason why we use the phrase 'in general speaking' above. In practice, both Monte

Carlo and quasi-Monte Carlo method are ideally applied for worst case validation In parallel, then a larger/smaller worst-case value then is guaranteed than only a single method is implemented.

The optimization based worst case validation is often applied to compare the efficiency to the Monte Carlo method. In this thesis, the optimization based worst case validation is also introduced. However, the purpose is not to exhaustively implement many of commonly used global optimization methods. We provide a modified DE for worst case validation and verify any improvement of efficiency. The generation of initial population of modified DE is replaced by Sobol method rather than the Monte Carlo method. This idea is originated from the study of quasi-Monte Carlo as the quasi-Monte Carlo method provides a better uniformity distribution over the Monte Carlo method. We believe a good distribution of initial population of DE would be able to improve the efficiency of worst case validation when the number of sampling points is small. Unlike the original DE, the step-size in the configuration parameters of optimization is recommended to select a small value. The reason is a large value of step-size would ruin the uniformity of Sobol sequence. Moreover, a small value of step-size is potentially able to address the issues that a bad 2-D projection happens between two variables in high dimensions. The results of worst-case validation based on the modified DE are much larger than the worst-case values obtained by conventional DE, Monte Carlo and quasi-Monte Carlo methods. This means the modified DE has a better convergence with high efficiency.

We also investigate the effect of the wind for the robustness analysis. The worst-case results show that the absence of wind greatly reduces the maximum dynamic loads, but not apparently affects other objectives. It indicates that the effects of wind must be considered in the simulation.

From the concept of effective dimension, the quasi-Monte Carlo based worst case validation should be more efficiency with a subset of uncertainties which are most important to drive the objective functions. In Chapter 4, the variance-based sensitivity analysis is approved as a good method to identify the driven variables in a complex system without any understanding of the mathematical expression of the model. As the real sensitivity indices of these variables are unknown, one of possible method is to employ the driven variables for worst case validation. An ideal worst-case value should not be much smaller than the worst-case value without any dimension reduced. However, the original purpose is to carry out the sensitivity analysis for the entire 118 uncertainties. The test

case actually only has 15 uncertainties. We realized that we cannot afford such a large computation cost for the entire 118 uncertainties. Thus, we selected 15 uncertainties. These 15 uncertainties are not randomly selected, but based on our understanding of the dynamic model that these uncertainties highly possible drives the dynamic loads of the launch vehicle. The worst-case results based on these 15 uncertainties and the driven variables demonstrate our inference. Inevitably, the worst-case value based on the 15 uncertainties for other objective functions are heavily affected and much smaller than the worst-case value when all 118 uncertainties are involved. Due to the expensive computation of variance based sensitivity analysis for the 118 uncertainties, it drives us to propose an cheap computation for sensitivity analysis.

So an emulator for the launch vehicle model is a possible solution to reduce the computational burden. In the early research stage, we tried a few methods to train a polynomial model for the Launch vehicle model. However, all the polynomial models are unable to give a good mapping between inputs and outputs. We realized that a polynomial model may be very difficult to be a high quality emulator for a complex nonlinear system. So our focus shift to the statistic model. The reason why a statistic model is also potentially available for sensitivity analysis is that we know the variance based sensitivity analysis only needs the informations of variance and mean value of the outputs. These informations are exactly a statistic model provided. In Chapter 5, we successfully trained a statistic model based on the objective function f_1 (Maximum dynamic loads over the time) with the selected 15 uncertainties. The emulator provides a good posterior distribution and the first order sensitivity indices are successfully estimated with a good quality. However, the quality of total sensitivity indices is not optimistic. The potential reasons are that: (1) the quality of model is not high enough;(2) and the accuracy of approximation for the multiple integration to calculate the expectation of posterior distribution is not very good.

The purpose of probability sensitivity analysis is to obtain a statistic model to identify the driven variables for the ELV launch vehicle model based on the 118 uncertainties. Unfortunately, we are not able to obtain an acceptable quality of Gaussian process model at present based on the objective f_1. The best quality we achieved so far is 0.74. An unqualified statistic model, especially with a high dimension of uncertainties, brings a huge approximate error for the expectation of posterior distribution. The difficulties of getting an acceptable statistic model with such a high dimension of uncertainties are

that (1) the number of training points are not sufficiently and properly cover the uncertainty space; (2) an optimized set of hyperparameters are difficult to obtain with high quality; (3) and good inferences for the mean and covariance functions are not easy to make. These difficulties are actually not easy to address. For examples: (1) for the first difficulty, it is impossible to simply increase the number of training points because the Matlab will stop the simulation and an error message pops up to show that the memory of Matlab is exceeded when the number of training points exceeds 8000; (2) for the second difficulty, a global optimization algorithm is ideally to employed to find a set of hyperparameters rather than a local optimization algorithm to avoid a local optimal solution. Moreover, the number of iteration needs to be increased which means the time spending on the optimization is also heavily increased. So the results of probability sensitivity analysis based on the 118 uncertainties given in the thesis have a considerable error. Nevertheless, the results of worst case validation based on the driven variables identified by the probability sensitivity analysis is good and acceptable. Although the error brought by the meta-model in probability sensitivity analysis leads a improper rank for many uncertainties, the four most driven uncertainties are till remained in the top four ranks. In the future research, more efforts are needed in order to obtain an acceptable meta-model, including a better understanding of the launch model and the properties of different mean and covariance functions. Moreover, the probability analysis for other objective functions are able to carry out if necessary by repeating the process of probability sensitivity analysis based on objective function f_1.

8.2 Summary of contributions

The detailed contributions of this work are presented in Chapter 1.3. To summarize, in our research we have proposed the use of the quasi-Monte Carlo method to perform robustness analysis in the aerospace industry. We demonstrated that the quasi-Monte Carlo method is superior to the Monte Carlo method that is currently in wide use in that industry. We also showed that using the quasi-Monte Carlo method to generate the initial population for different evolutions also significantly improves the performance of robustness analysis.

We also introduced the Gaussian process (GP) for probabilistic optimization in order to further reduce the computational burden. We showed that GP-based optimization can reduce computing time remarkably without losing too much accuracy.

Variance-based sensitivity analysis is a method of identifying the driven parameters of a complex nonlinear model. However, its drawback is its heavy computational cost. In our research, we proposed a novel strategy for computing the probabilistic sensitivity indices based on the GP model. The results show that our method can successfully sort out the key parameters of the first-order sensitivity indices.

8.3 Future Works

In this thesis, the meta-model generated by the Gaussian process employs only the linear functions for the inference of polynomial functions $\Phi(x)$[1]. This linear inference may not, however, be able to sufficiently express the nonlinear relationship between inputs and outputs.Thus, refining the quality of the meta-model is a goal of future work.

Moreover, for the current GP-based sensitivity analysis, we sorted out the driven parameters using the first-order sensitivity indices. Unfortunately, these indices do not contain information on the interaction between each uncertain parameter (no higher-order terms). The ideal sensitivity analysis results should be based on total sensitivity analysis. In the current research, the results obtained by our inference for total sensitivity analysis is not excellent, and, therefore, an improved inference for total sensitivity analysis is a desired goal of future research in pursuit of more excellent results.

[1]Details can be found in Chapter 2.4

Appendix A

Results of robustness analysis based on other objective functions

A.1 Other recommend objective functions

The f is objective function or cost function

They are:

(1) $f_1 = \max(Q_\alpha(Mach))$ - The maximum dynamic load over the speed (Mach), [Pa Deg]

(2) $f_2 = \max(D_{V\_Z}(t))$ - The maximum drift velocity of Z axis over the time, [m/s]

(3) $f_3 = \max(D_{P\_Z}(t))$ - The maximum drift position of Z axis over the time ,[m]

(4) $f_4 = \max(D_{V\_Y}(t))$ - The maximum drift velocity of Y axis over the time ,[m,s]

(5) $f_5 = \max(D_{P\_Y}(t))$ - The maximum drift position of Y axis over the time , [m]

(6) $f_6 = \max(AoA_T(t))$ - The maximum total angle of attack ,[Deg]

There are also 18 extended objective functions in terms of the 6 objective functions above by cooperating with a function metric. Three types of function metrics are introduced in our research. They are:

Metric type 1 (we will use Metric 1 in later sections) is defined as:

$$\text{Metric 1} = \begin{cases} \sum(y(i) - y\_ref(i) | y(i) - y\_ref(i) > 0) & \text{if } \exists y(i) - y\_ref(i) > 0 \\ \sum y(i) - y\_ref(i) & \text{if } \nexists y(i) - y\_ref(i) > 0 \end{cases}$$

where y is a discrete output of an objective over the time or speed, $y_{\_ref}$ is the reference trajectory(also called constraint in our research) of that output over the time or speed. Metric 1 is actually measures the summation of difference between the output of an objective and reference trajectory of that output over the time or speed. However, we only consider the positive difference if any positive difference exists; on the other hand, all negative difference are considered if no positive difference exists.

In our research, the y can be:

$y_1 = Q_\alpha(Mach)$

$y_2 = D_{V\_Z}(t)$

$y_3 = D_{P\_Z}(t)$

$y_4 = D_{V\_Y}(t)$

$y_5 = D_{P\_Y}(t)$

$y_6 = AoA_T(t)$

For all of the extended objective functions, we aim to find the maximum objective value. For instance, the objective function based on Metric 1 for $Q_\alpha(Mach)$ is:

$$\text{Metric 1} = \max \begin{cases} \sum(Q_\alpha - Q_{\alpha\_ref} | Q_\alpha - Q_{\alpha\_ref} > 0) & \text{if } \exists Q_\alpha - Q_{\alpha\_ref} > 0 \\ \sum Q_\alpha - Q_{\alpha\_ref} & \text{if } \nexists Q_\alpha - Q_{\alpha\_ref} > 0 \end{cases}$$

The Metric type 2 is defined as:

$$\text{Metric 2} = \max \begin{cases} \sqrt{\sum[(Q_\alpha - Q_{\alpha\_ref} | Q_\alpha - Q_{\alpha\_ref} > 0)]^2} & \text{if } \exists Q_\alpha - Q_{\alpha\_ref} > 0 \\ -\sqrt{\sum[Q_\alpha - Q_{\alpha\_ref}]^2} & \text{if } \nexists Q_\alpha - Q_{\alpha\_ref} > 0 \end{cases}$$

The Metric 2 suggests that only each individual positive difference is considered if any positive difference is confirmed; otherwise, all the negative difference are considered by assigned with a negative sign ahead.

Similarly, the Metric type 3 only considers the maximum difference and is defined as:

$$\text{Metric 3} = max(y - y_{\_ref}) \tag{A.1}$$

Based on the equation A.1, the remaining 17 objective functions can be defined by following this principle. The reason why we would like to introduce these extended objective functions is that we want to investigate the violations happened during the entire simulation.

A.2 Results

The Table A.1 gives the results of worst case validation based on Metric 1, Metric 2 and Metric 3 for Q_α . Under each Metric, the worst values in terms of different sampling methods that are mentioned above. Note that the sampling sequences are exactly same based on Halton, Sobol and Faure method for all kind of cost functions because they are deterministic sampling methods. From this table, it is clear that the differences

Q_α	Metric 1	Metric 2	Metric 3
Monte Carlo	3.6819×10^5	9.3038×10^4	3.2519×10^4
Sobol	1.4329×10^6	2.5548×10^5	7.3935×10^4
Halton	4.9686×10^5	1.2030×10^5	3.5292×10^4
Faure	5.7623×10^5	1.1765×10^5	3.2336×10^4

TABLE A.1: The worst case values based on different distribution methods for cost function metric $Metric$ 1 to $Metric$ 3 in terms of $Q_\alpha(mach)$

of worst-case value between different sampling methods in terms of Metric 1, Metric 2 and Metric 3 are obvious. These differences suggest that the surface of the objective functions are not flat and the strength of each sampling method could be demonstrated by its worst-case value. In practices, Neither quasi-Monte Carlo and Monte Carlo can guarantee the achievement of largest worst-case value based on the different objective functions among those sampling methods. However, in this case, the quasi-Monte Carlo method demonstrates that it has a better worst-case value over the Monte Carlo method in general. Especially, the Worst-case values achieved by Sobol sequence is much larger than any other method based on the three extended objective functions. This results shown in this table suggest that the worst case results obtained by quasi-Monte Carlo at least won't be much worse than the results obtained by Monte Carlo method again. By following the same analysis procedure for f_1 above, we would like to plot the convergence figures for each cost function based on different distribution methods to reveal some additional information.

The Figure A.1 is the convergence plot for Q_α in terms of the Metric 1. From the figure, the Sobol sequence has the overall best performance of the convergence over other methods and a good candidate for population initialization of DE in next section. Faure sequence has a better cost value between the sampling number 100 and 150. The eventual cost value is much difference between the Sobol sequence and other methods.

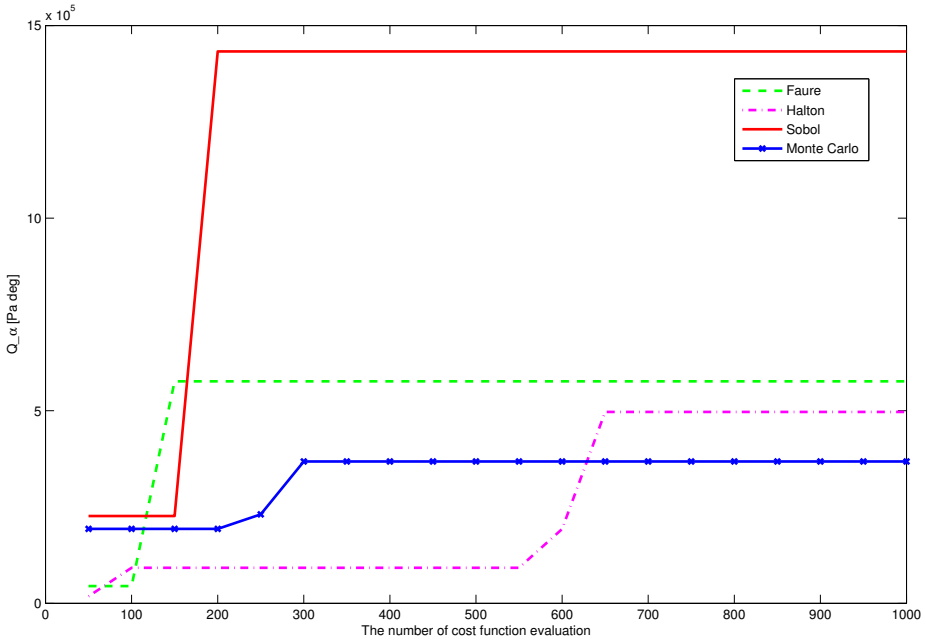


FIGURE A.1: The convergence plot for Q_α based on Metric 1

However, there is no direct information to present that this objective values is rated to the robustness of the VEGA launch model because of the definition of the objective function. So it is no idea whether the launch vehicle is stable or not under these worst-case conditions of variables. However, according the definition of Metric 1, we know that it measures the sum of the difference between the output response and the reference trajectory over the time. A positive summation only happens when the output response is over the reference trajectory at some mach number. In this case, the objective values are all positive which suggests the violation is happened during the ascending mission in P80 stage of the ELV launch vehicle under the worst-case situation.

So the worst-case results can be expected to suggest one of the following two completely different situations:

(1) The value of Q_α at some time span (could be the entire time span)is much higher than the reference value if a objective values are positive.

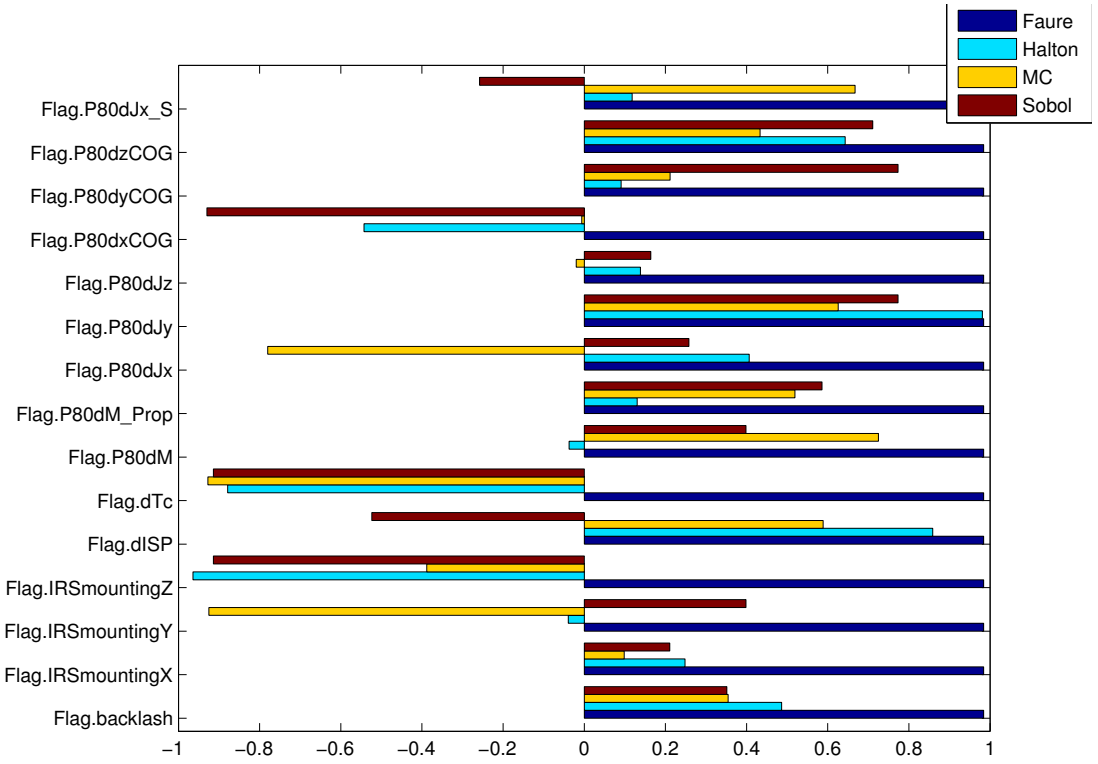
(2) The value of Q_α at the entire time span is much smaller than the reference value if a objective values are negative.

The convergence plot is able to provide the information about the worst-cases that belong to which situation. In fact, the Metric 1 is interested in finding a condition of uncertainties that makes a large violation happened for a relatively long time period.

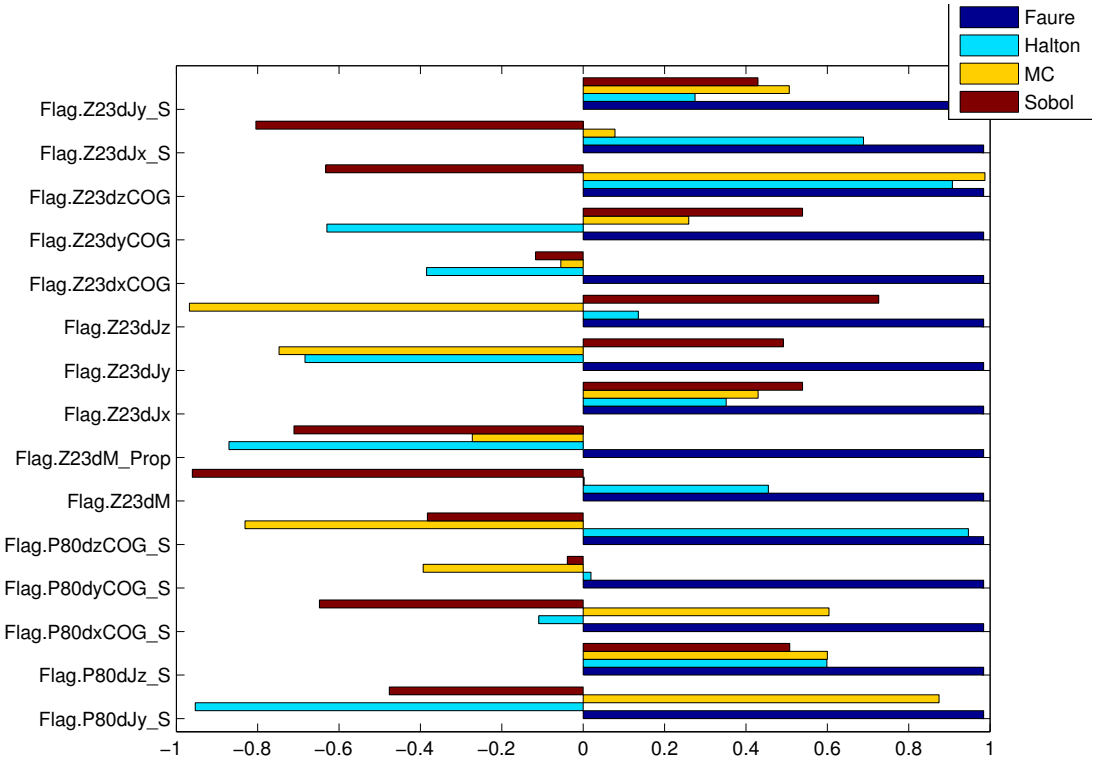
Thus, two circumstances are greatly contributed to a large objective value: large volume of violation and a long lasting time of violation. The way to find out which circumstance plays a much more vital effects for the objectives can be analyzed by observing the Q_α plots over the mach number. For robustness analysis, the first situation mentioned above is our expectation. We would like to find a condition of the uncertainties that makes the Q_α violates its constraint for a long time. This means the VEGA launcher may exceed the load constraint for a longer time period of risk.

The perturbation bar plot of the uncertainties in terms of the worst-cases is given in the Figure A.2. Except condition of worst-case uncertainty vector can be observed from the bar plot, there is few extra information that could be inferred from this figure. Based on the plots inn the Figure 4.2 and A.2, the biases for each uncertainties in the worst-case of Faure, Halton and Sobol sequences are identical. it means the condition of worst-case uncertainty vectors are exact same under the two different objective functions based on Sobol, Halton and Faure sampling methods. Moreover, compared the biases of Faure and Sobol sequence in this case, a large bias of some some uncertainty may plays a dumpling effects for the objective functions because the worst-case value obtained by Sobol sequence is much higher than the worst-case value obtained by Faure sequence. Similarly as Figure 4.2, we can identify 12 uncertainties with pattern of interesting according to their biases.

According to the analysis of the convergence plot, the Q_α output response over the mach number in Figure A.6 gives the answer that how much contribution is made by the two circumstances 'large volume of violation' and 'long lasting time of violation' based on worst-case results for each sampling method. From the figure, the values of Q_α based on all type of distribution methods are above the reference trajectory over a period. These violations match the conclusions made by analyzing the objective values above. In the figure, a large violation is occurred but only lasts for a short period under each worst-case result. This phenomena infers that the circumstance 'Large volume of violation' affects the objective function most in this case. So the VEGA launcher only suffers a shot time period of risk under a specific range of mach number during the ascending phase. Especially for the Q_α plot of Monte Carlo method, although the worst-case results doesn't have the largest Q_α value over the all sampling points, it has the largest

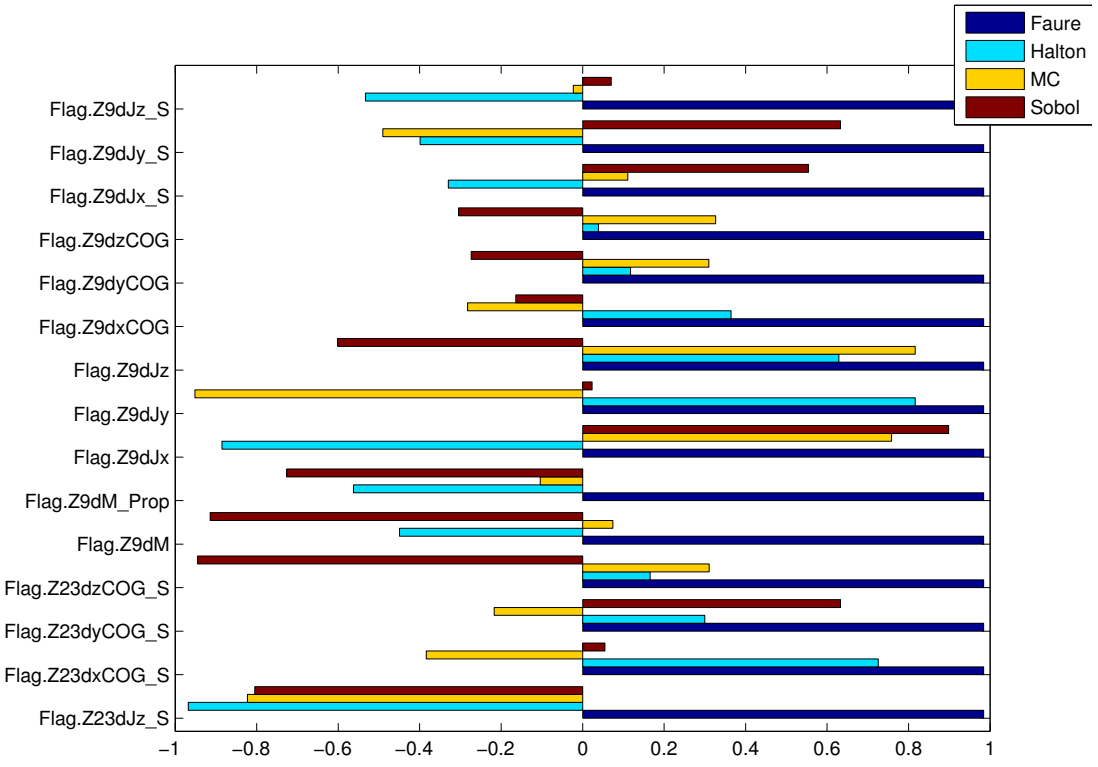


(A) Variables from 1 to 15 under different worst-case scenario

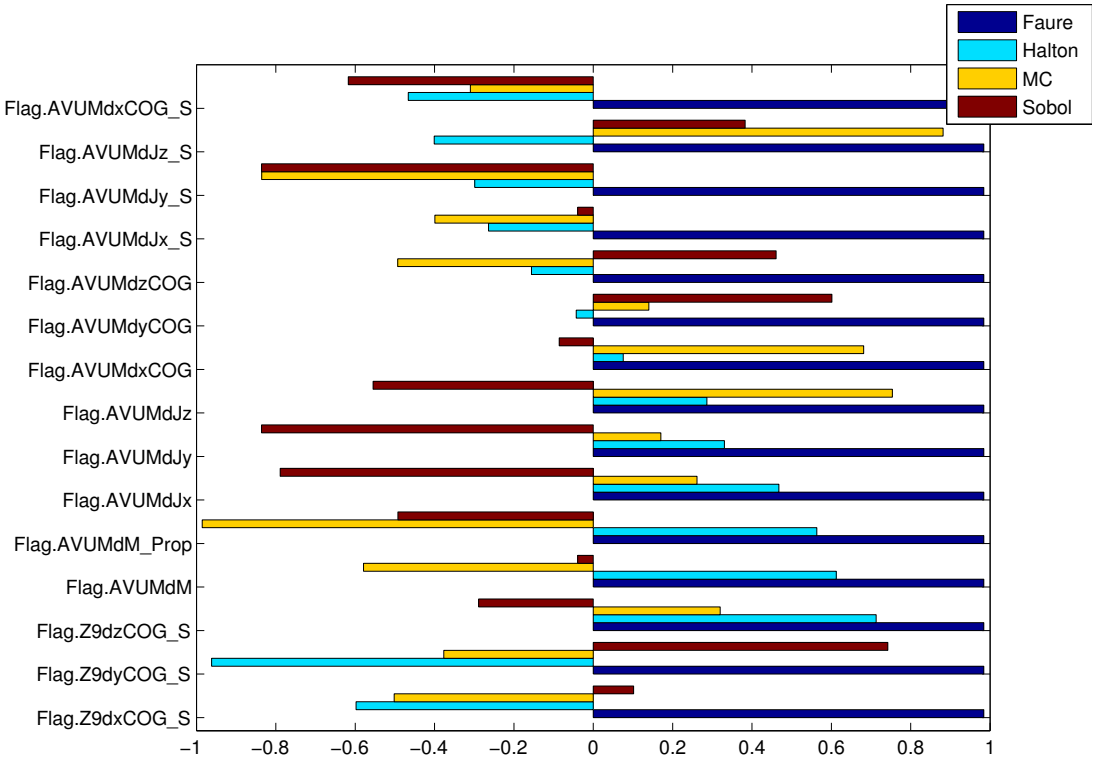


(B) Variables from 16 to 30 under different worst-case scenario

FIGURE A.2: The worst case uncertainties (1 -30) for f_1 in terms of Metric 1

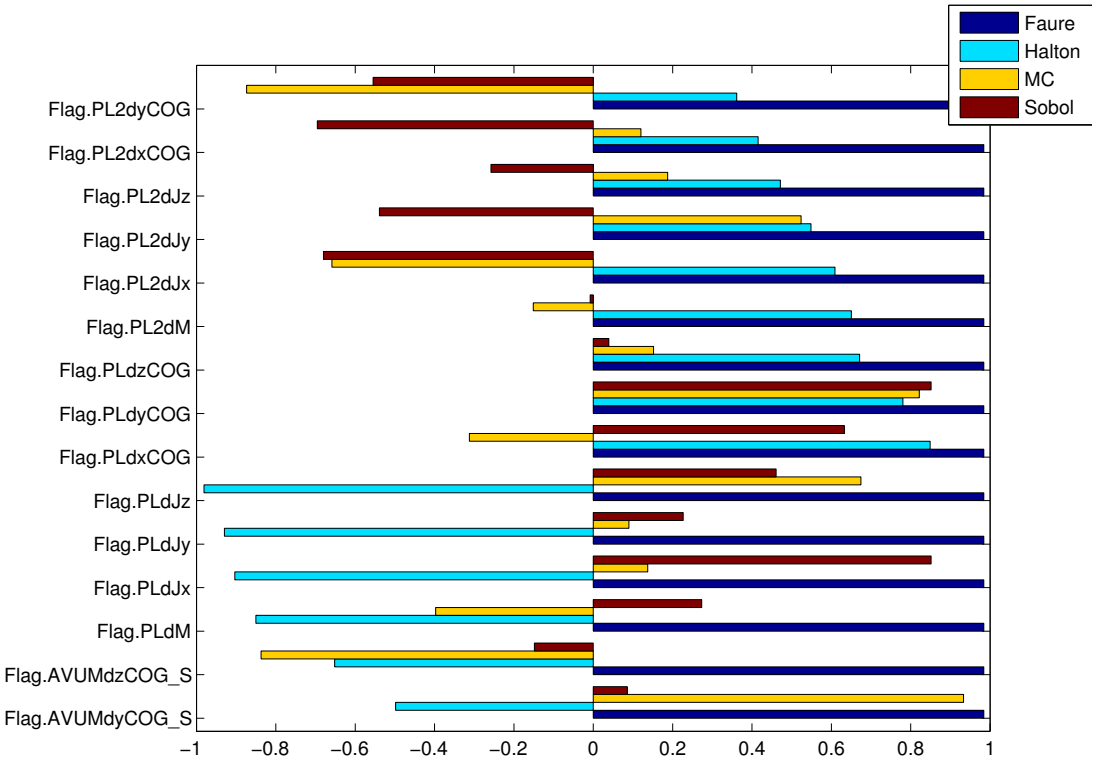


(A) Variables from 31 to 45 under different worst-case scenario

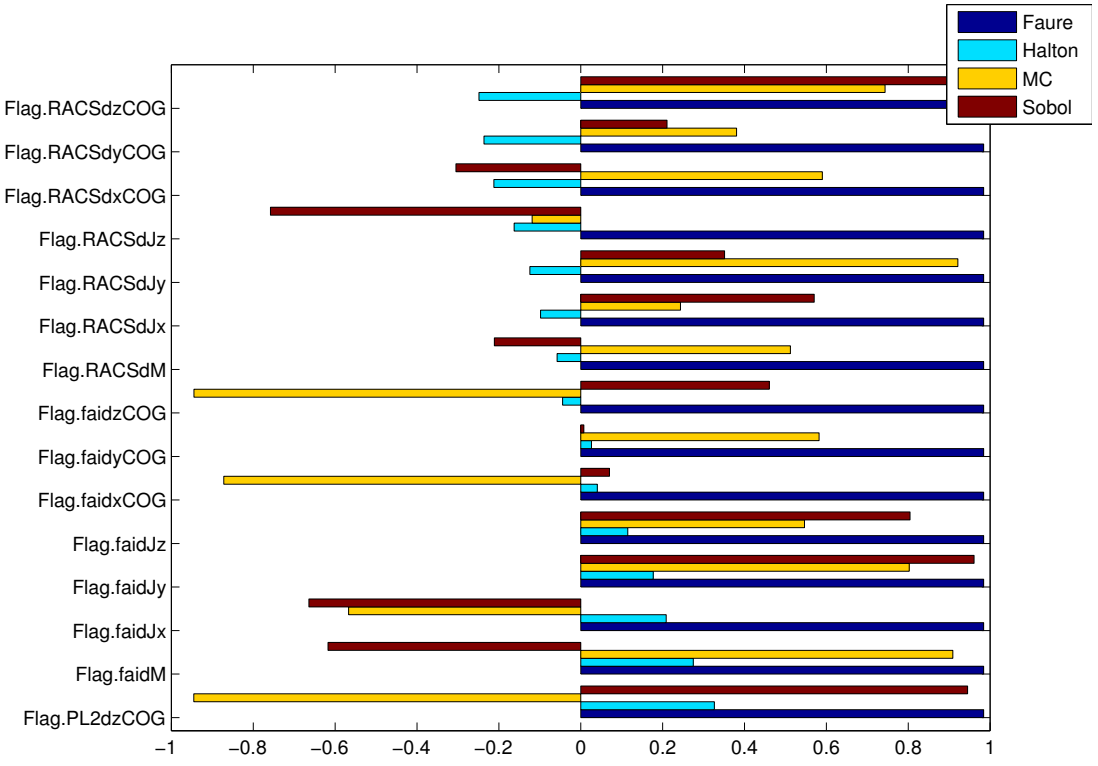


(B) Variables from 46 to 60 under different worst-case scenario

FIGURE A.3: The worst case uncertainties (31 -60) for f_1 in terms of Metric 1

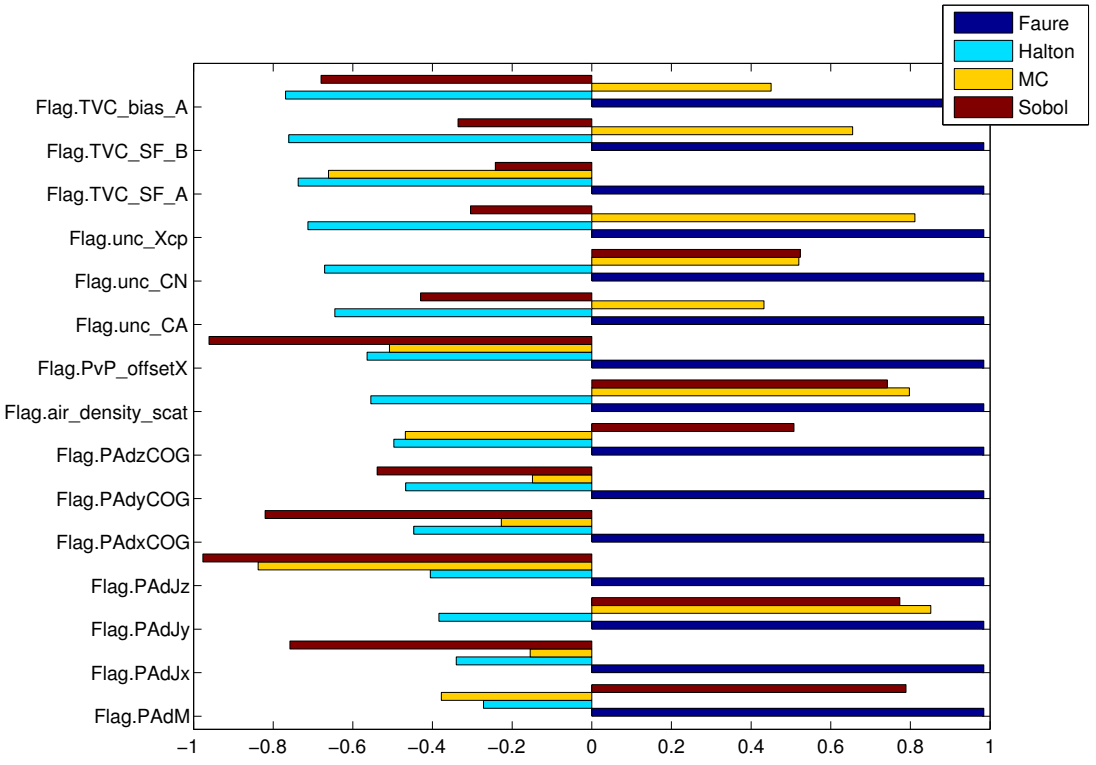


(A) Variables from 61 to 75 under different worst-case scenario

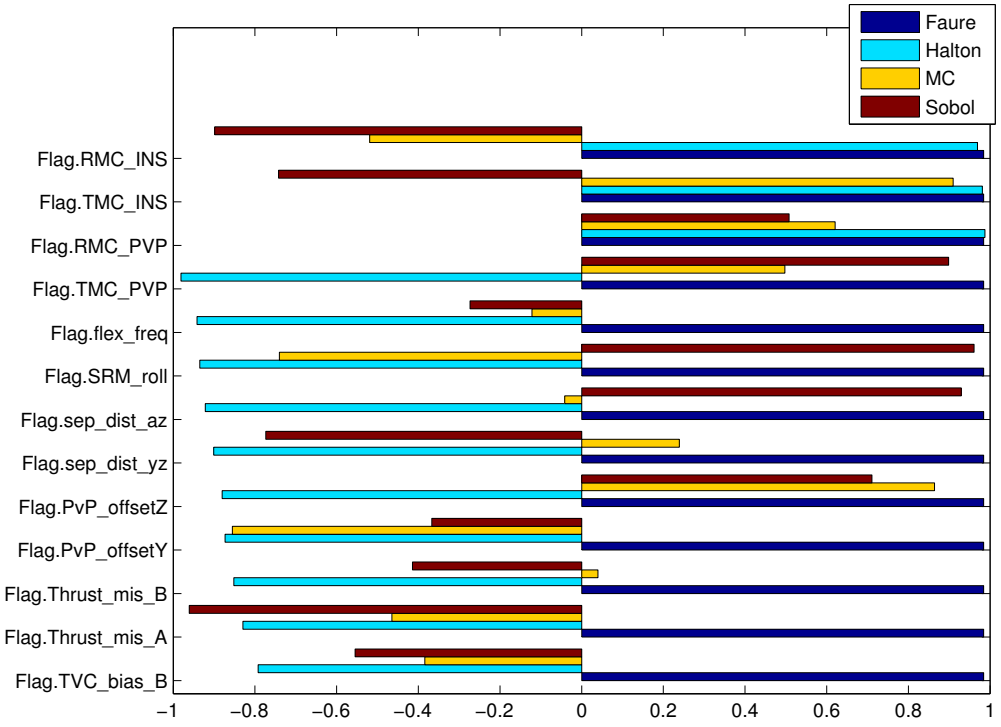


(B) Variables from 76 to 90 under different worst-case scenario

FIGURE A.4: The worst case uncertainties (61 -90) for f_1 in terms of Metric 1



(A) Variables from 91 to 105 under different worst-case scenario



(B) Variables from 106 to 118 under different worst-case scenario

FIGURE A.5: The worst case uncertainties (91 -118) for f_1 in terms of Metric 1

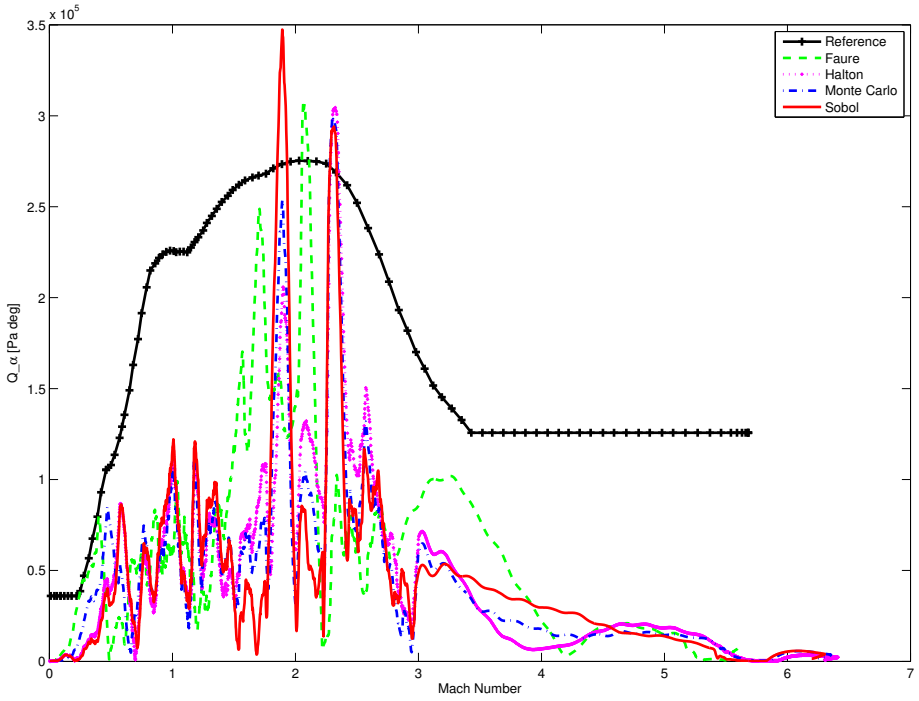


FIGURE A.6: The Q_α plot of the worst case in terms of Metric 1 by different distribution methods

area that exceeds the constraint.

For other five criteria such as $AoA_T(t)$, the details of analysis for the worst-case results obtained based on the Metric 1 method is not provided in order to retain a friendly reading interface of this thesis. The analysis procedure for the remaining five criteria in terms of Metric 1 is exactly same as the procedure introduced in this case.

To analyze the worst-case results obtained based on the cost function constructed by the method of Metric 2 in terms of the criteria y_1, we just follow the steps of the analysis for the worst-case results based on the cost function constructed by Metric 1 in terms of y_1. The convergence plot, perturbation bar plot and the Q_α response plot are provided respectively. It measures the root of the accumulative squared difference between the output response and reference trajectory. Similar as Metric 1, the Metric 2 also consider the violation prior. Only the violation part will be considered if the output response exceeds the reference trajectory at some time being whereas a negative sign is assigned if no violation happens during the entire simulation. The objective function is applied for

maximization only. The difference between the Metric 1 and Metric 2 is that the Metric 2 weakens the effects made by a 'impulse-shape' output response in a short time window for an objective function. A constant high volume of violation with a long lasting time is more welcome by Metric 2 than Metric 1.

Similar as Metric 1, the worst-case results can also be expected to suggest one of the followed two situations:
(1) The value of Q_α at some time span (could be the entire time span)is much higher than the reference value if a positive objective value exists.
(2) The value of Q_α at the entire time span is much smaller than the reference value if only the negative objective exists.
The convergence figure is provided in Figure A.7. From the figure, the convergence

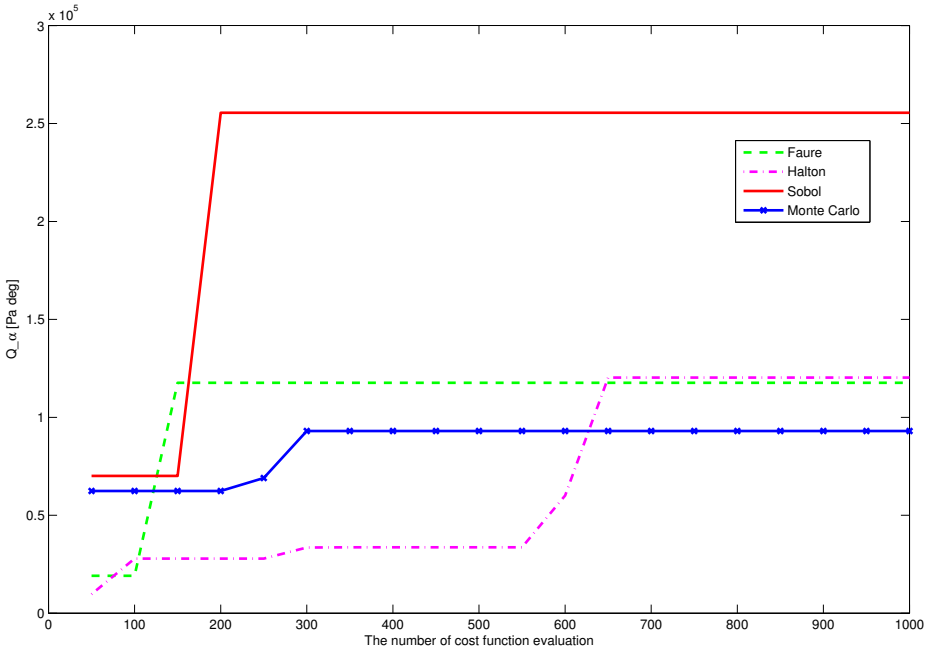


FIGURE A.7: The convergence plot for Q_α based on Metric 2

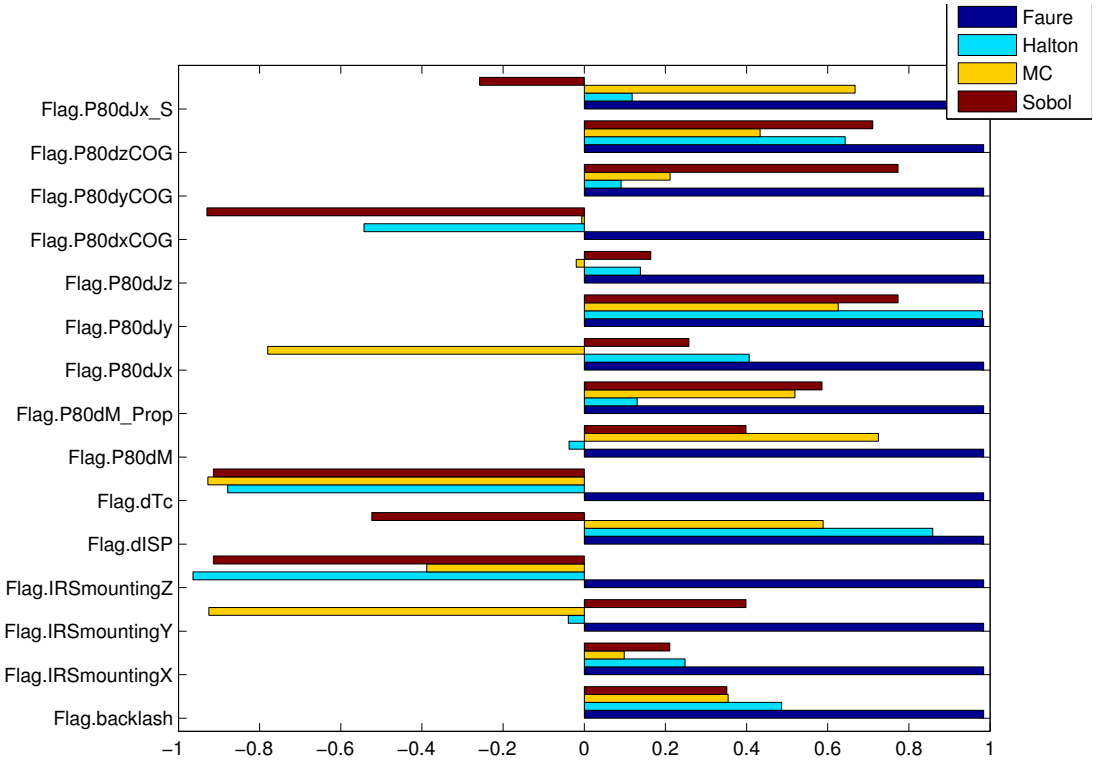
based on Sobol sequence has the best overall performance. The maximum cost value after the 150 evaluations of Sobol sequence is much larger than cost values based on other methods. It also has a higher objective value in first 100 evaluations till the Faure sequence takes the first place of largest objective value between 100 and 150 evaluations. Thus, the Sobol is also ideally to initialize the population for DE in next chapter based on cost function of Q_α in terms of Metric 2. The Monte Carlo method is also performed

not very well in this case. In summary, according to the convergence figure, the difference between the Monte Carlo method and quasi-Monte Carlo is obvious. Similar as Metric 1, the cost function constructed by Metric 2 is also able to reveal any constraint violation. A large positive cost value suggests that the value of Q_α is much higher than the constraint over a time span. The weights of two circumstances 'large volume of violation' and 'long time lasting of violation' are still unknown until by observing the Q_α over the mach number which will be given later.

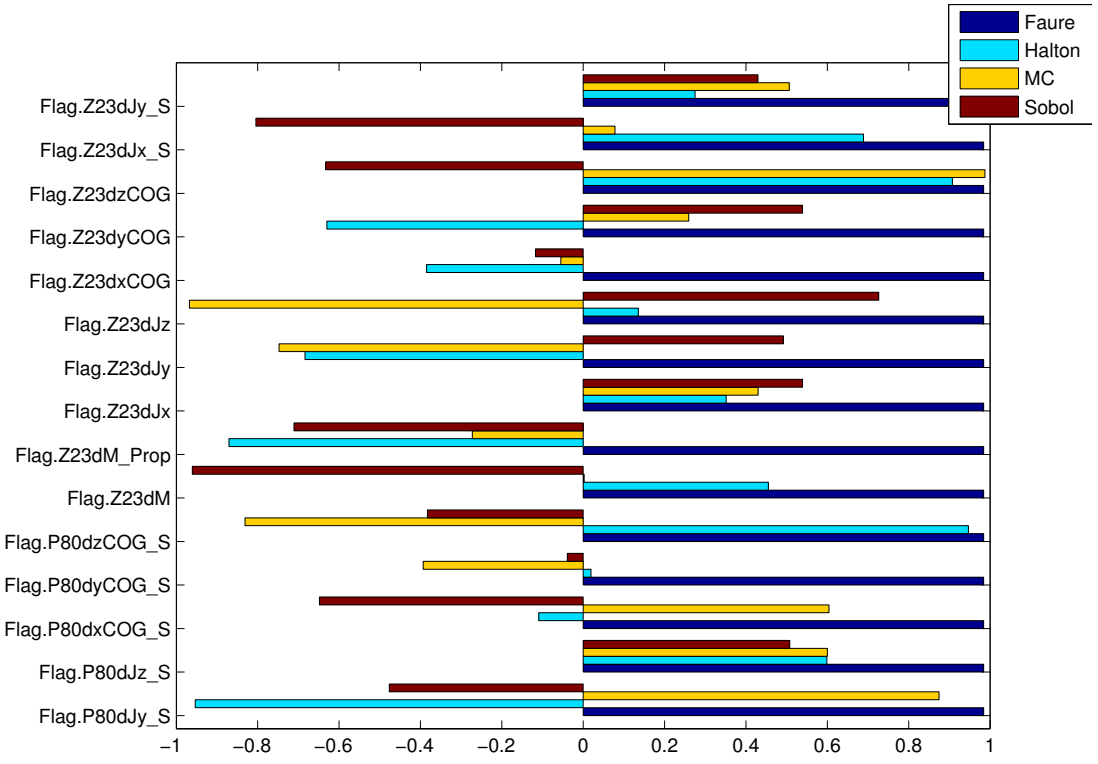
The figure followed is the bar plot of the uncertainties perturbation.

In Figure A.8, the biases of each uncertainty under the four worst-cases are exactly same as the perturbations in Figure A.2. It means the worst-case values based on Metric 1 and Metric 2 of the four sampling methods have same condition of uncertainties. This results indicates that the circumstance 'large volume of violation' still retain a vital position in contribution of the largest objective value.
The Q_α figure is given below. There is not doubt that the Q_α plots in Figure A.12 and A.6 are identical because the worst-case is under the same condition of uncertainties. This is not a bad at all even the worst-case results obtained by Metric 1 and Metric 1 are exactly same. The reason is the Metric 2 still find out a condition of uncertainties that makes the VEGA launch vehicle under a risk during the ascending phase. The results in this case infer that the weight of circumstance 'large volume of violation' is too large and it makes great contribution to the objective value even we try to reduce the effects made by a large violation in a short time span.

The analysis for the results of worst case validation based on Q_α by employing the Metric 3 method still follows the same processes of the two cases mentioned above. The Metric 3 is actually looking for a maximum difference between the output response and reference trajectory.
Similar as Metric 1 and 2, the worst-case results can also be expected to suggest one of the followed two situations:
(1) The value of Q_α at a time point is higher than the reference value if the objective value is positive.
(2) The value of Q_α at the entire time span is smaller than the reference value if the objective value is negative.

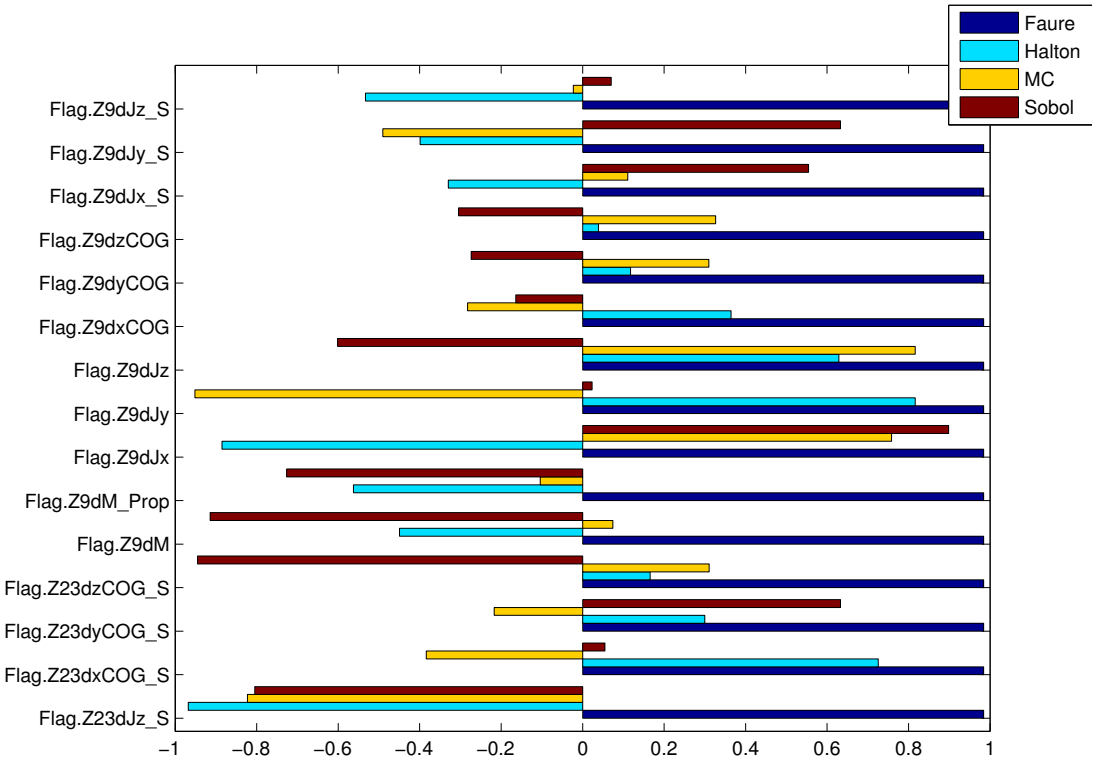


(A) Variables from 1 to 15 under different worst-case scenario

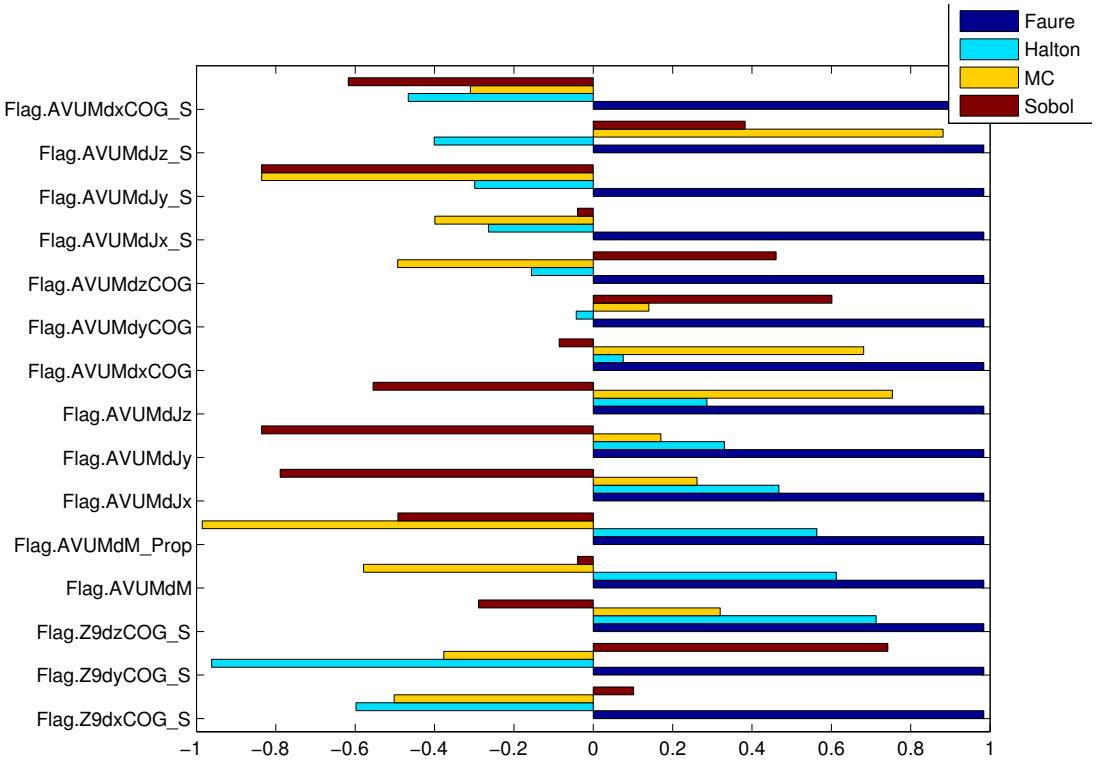


(B) Variables from 16 to 30 under different worst-case scenario

FIGURE A.8: The worst case uncertainties (1 -30) for f_1 in terms of Metric 2

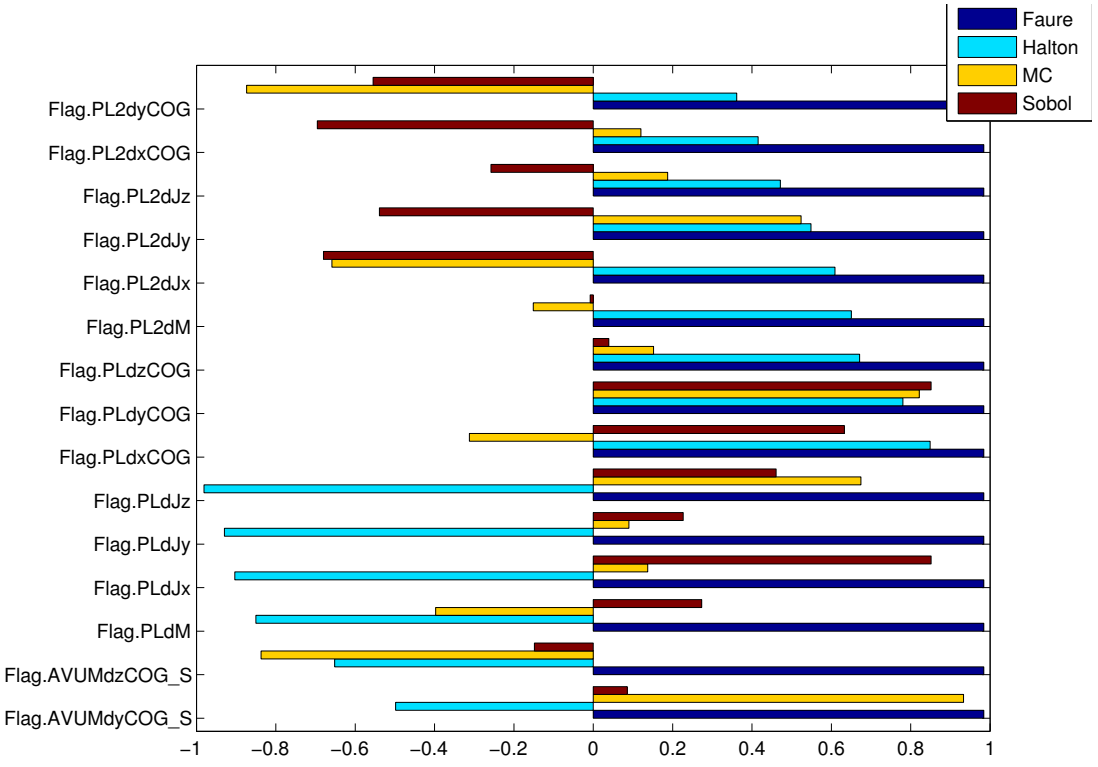


(A) Variables from 31 to 45 under different worst-case scenario

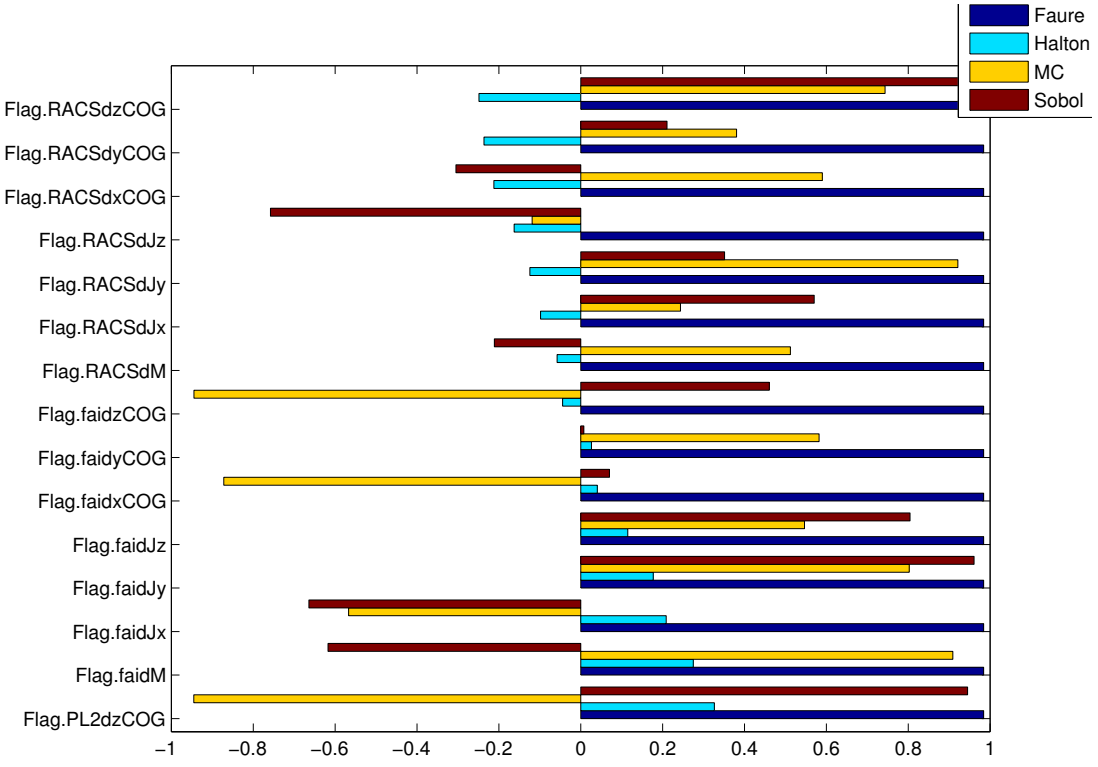


(B) Variables from 46 to 60 under different worst-case scenario

FIGURE A.9: The worst case uncertainties (31 -60) for f_1 in terms of Metric 2

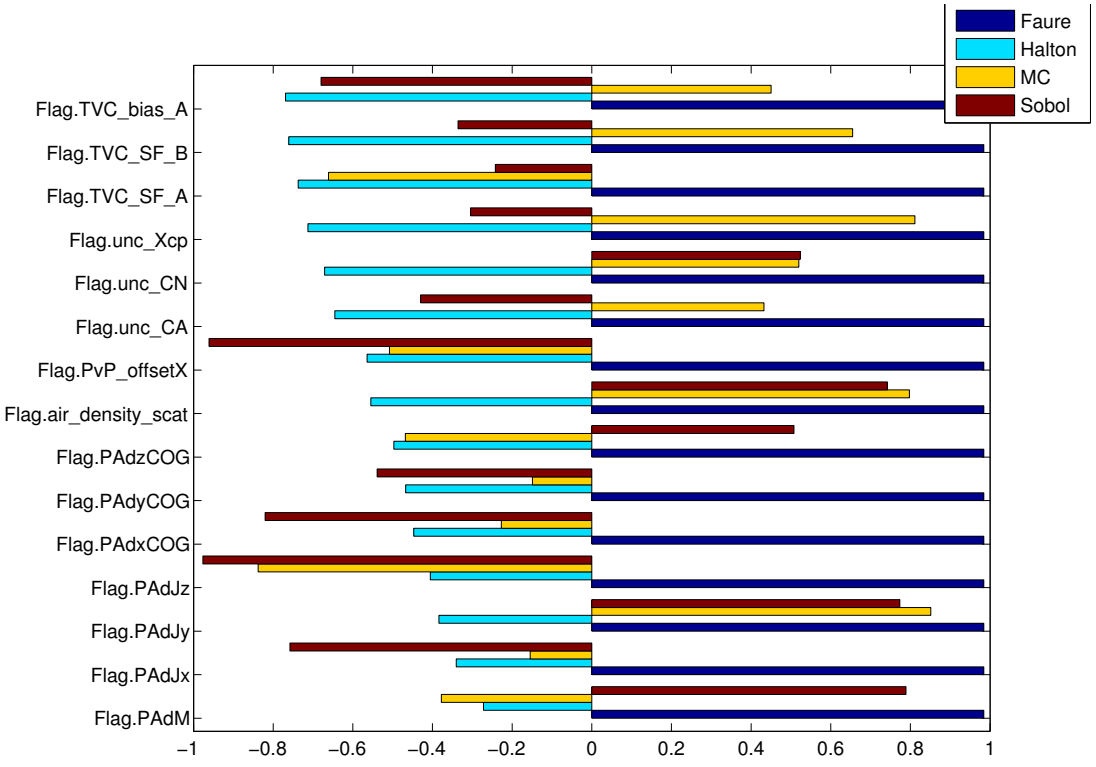


(A) Variables from 61 to 75 under different worst-case scenario

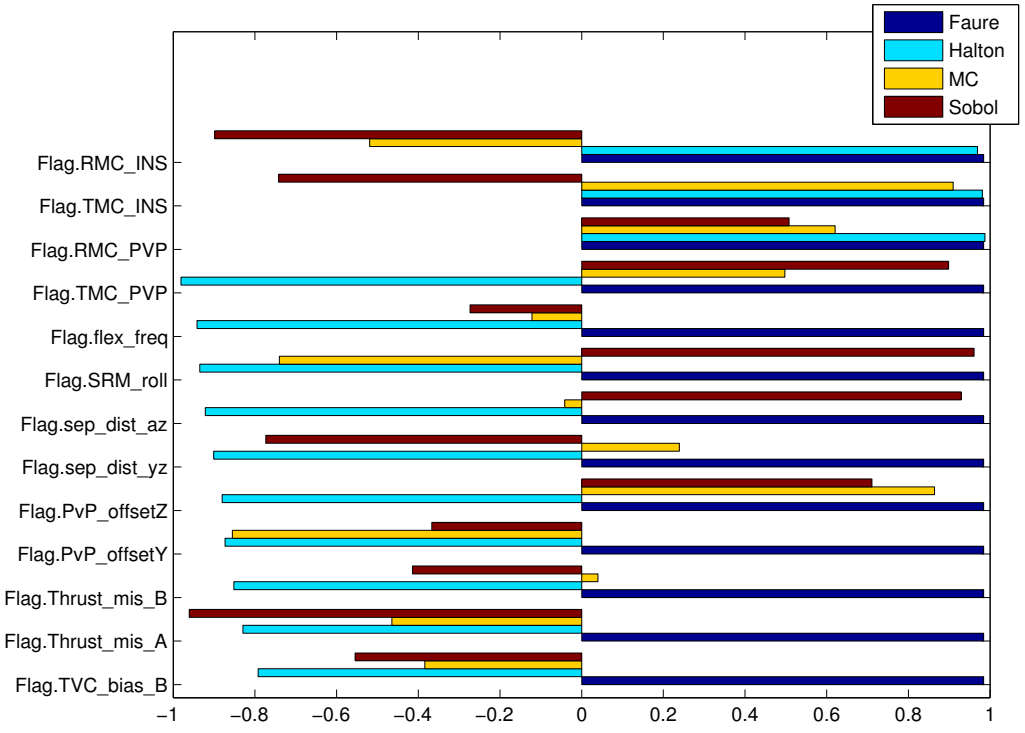


(B) Variables from 76 to 90 under different worst-case scenario

FIGURE A.10: The worst case uncertainties (61 -90) for f_1 in terms of Metric 2



(A) Variables from 91 to 105 under different worst-case scenario



(B) Variables from 106 to 118 under different worst-case scenario

FIGURE A.11: The worst case uncertainties (91 -118) for f_1 in terms of Metric 2

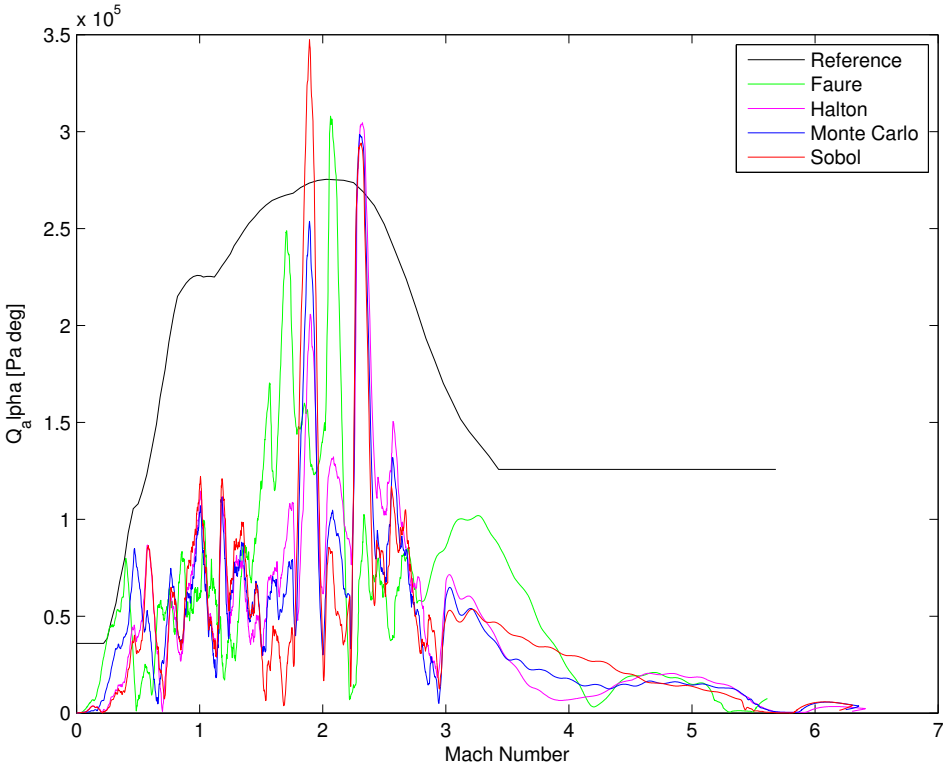


FIGURE A.12: The Q_α plot of the worst-case in terms of Metric 2 by different distribution methods

Note that, the Metric 3 only consider the maximum difference at a single time point, rather than a time span in Metric 1 and 2. When the reference trajectory is not straight line, the point(condition of uncertainties) with a maximum Q_α value over all of the evaluations doesn't have to be the point of maximum difference. The maximum difference could happens at any time spot. The maximum difference could also mean the volume of the violation over the constraints is the maximum. The maximum violation is potentially more dangerous for the VEGA launch vehicle in the P80 phase. On the other hand, if the reference trajectory is a straight line, the point with the maximum Q_α over the entire evaluations must be the point with the maximum difference.

Firstly, the convergence plot is given in Figure A.13 From this figure, the Sobol sequence has the overall best performance. The difference of maximum objective value between the Sobol sequence and other sampling methods are obvious. The sobol sequence is a better choice to generate the initial population for DE in the next section. The positive objective during the evaluations suggests that the violation is existed.

The next figure is the bar plot of the uncertainties and is shown in Figure A.14.

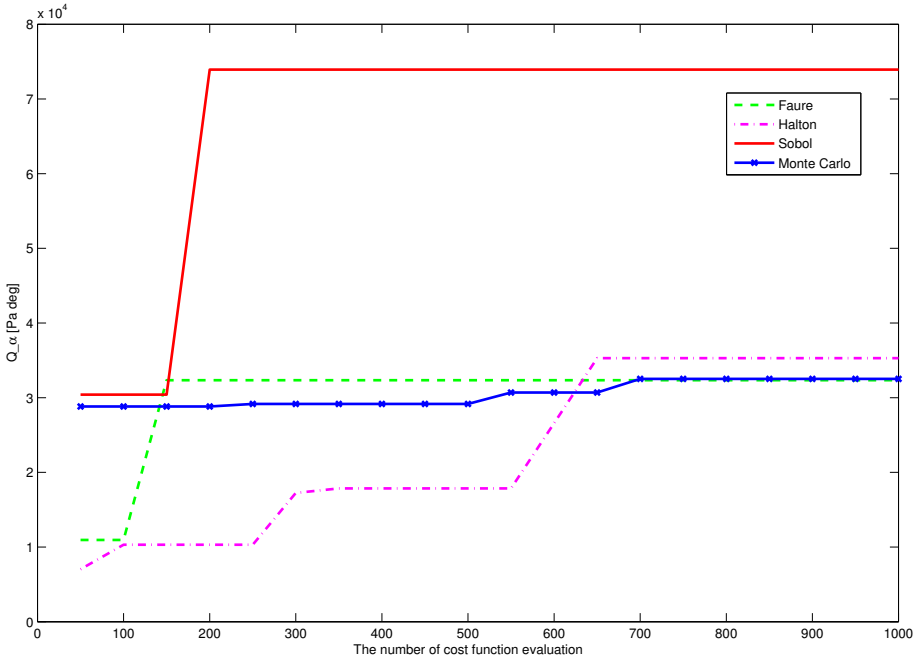
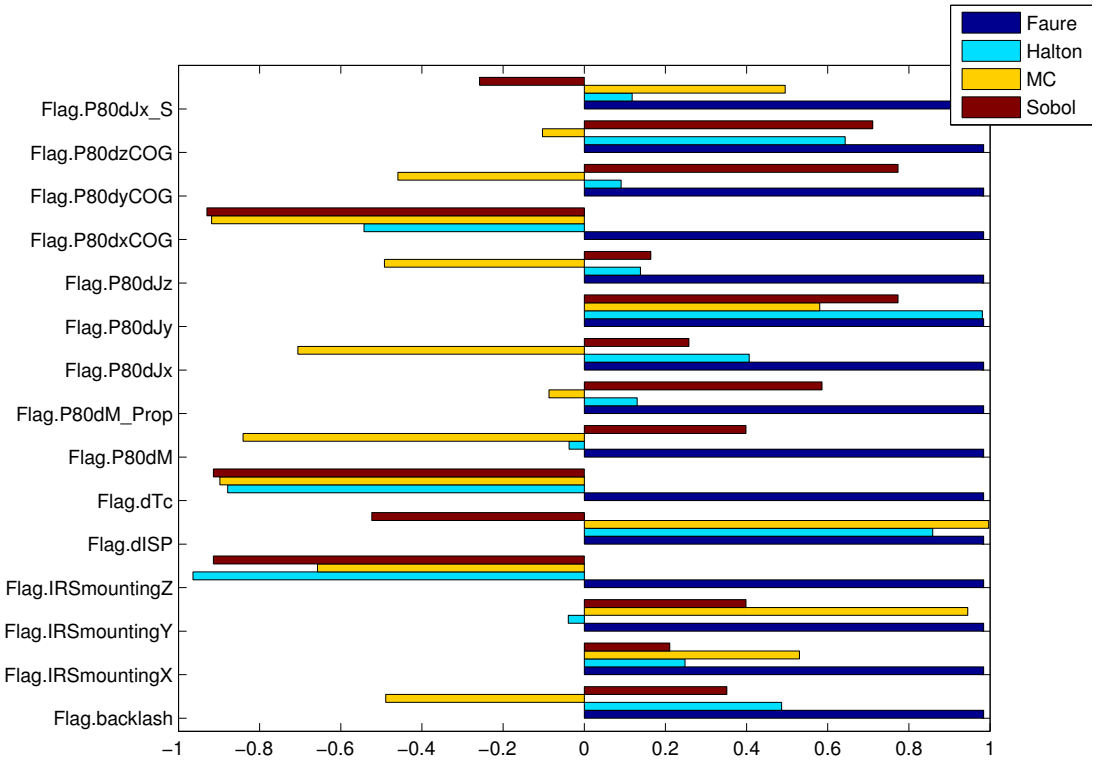


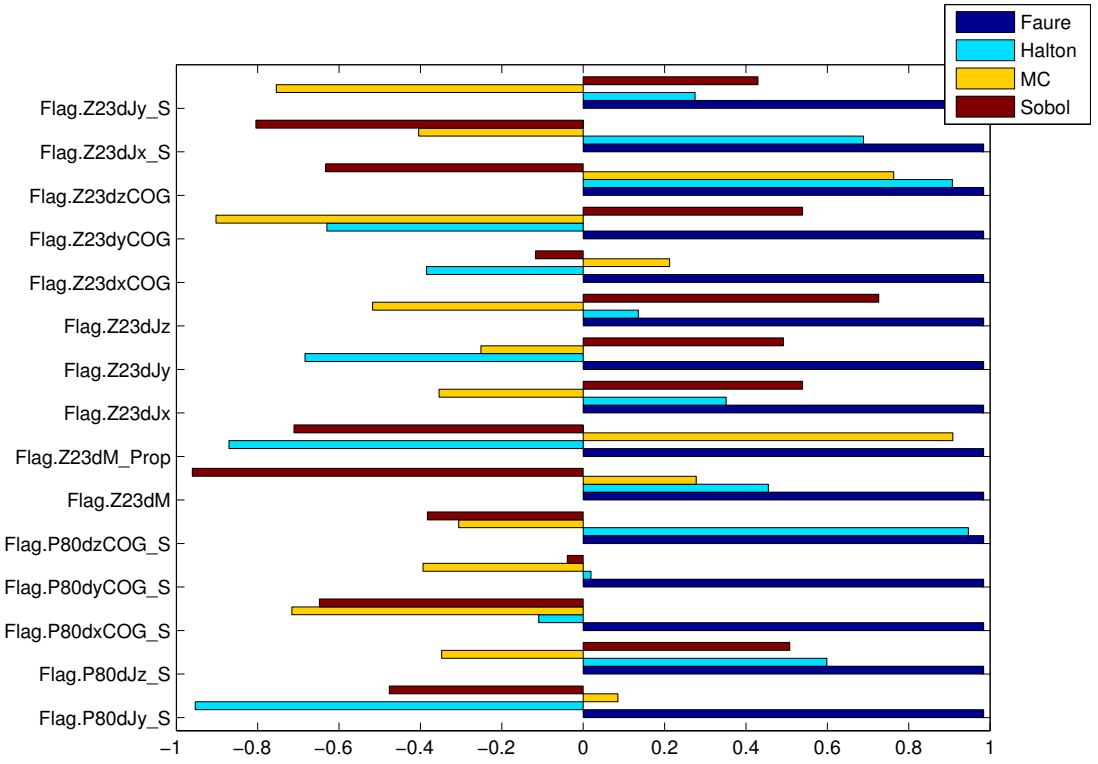
FIGURE A.13: The convergence plot for Q_α based on Metric 3

The bar plots in this figure is exactly same as the plots in the Figure **??**. This means the condition of uncertainties of the worst-case is same. It also suggests that the point with the maximum Q_α value over the entire evaluations is same as the point with maximum difference between the Q_α and its reference trajectory.
The Q_α plots based on Metric 3 is given in Figure A.18. From the worst-case results based on the f_1 and y_1 by employing Metric 1,2 and 3, the same condition of uncertainties are obtained for Sobol, Faure and Halton sequence. As each objective function represents a different risk situation, these conditions of uncertainties should be really be carefully to investigate in order to avoid any risk during the ascending phase.

For the remaining objective criteria, only the tables of cost values are given. The plots for output response are not provided for a friend reading interface. The Table A.2 presents the objective values $D_{V\_Z}(t)$ (The drift of velocity in Z axis)based on the different distribution methods. From the table, the Sobol sequence has an overall best performance. But the advantage of the quasi-Monte Carlo is obvious but not extraordinary. Especially the difference of the cost value for Metric 2 is very small. The remaining tables that contains the objective values in terms of $D_{P\_Z}(t)$,$D_{V\_Y}(t)$, $D_{P\_Y}(t)$ and $AoA_T(t)$ are given below.

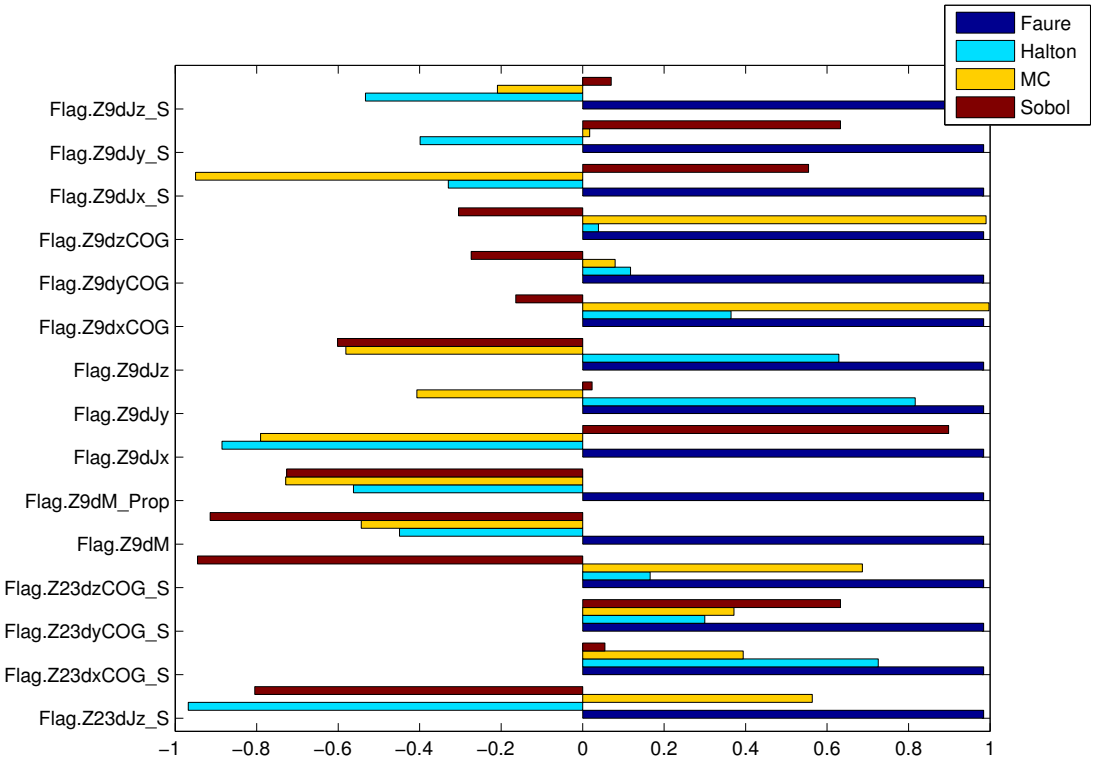


(A) Variables from 1 to 15 under different worst-case scenario

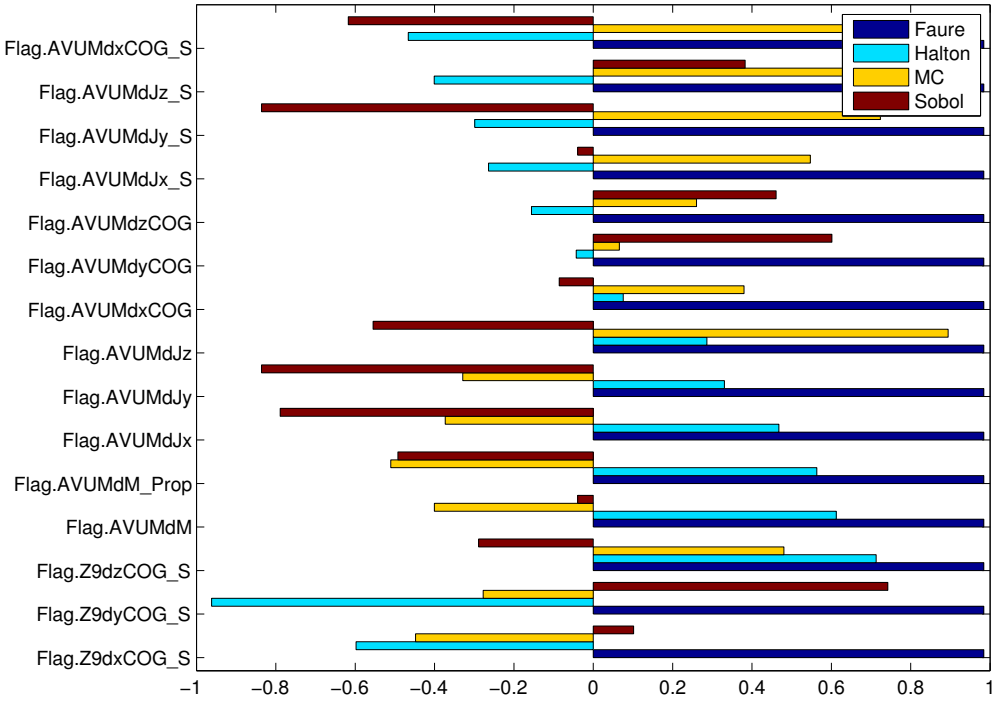


(B) Variables from 16 to 30 under different worst-case scenario

FIGURE A.14: The worst case uncertainties (1 -30) for f_1 in terms of Metric 3

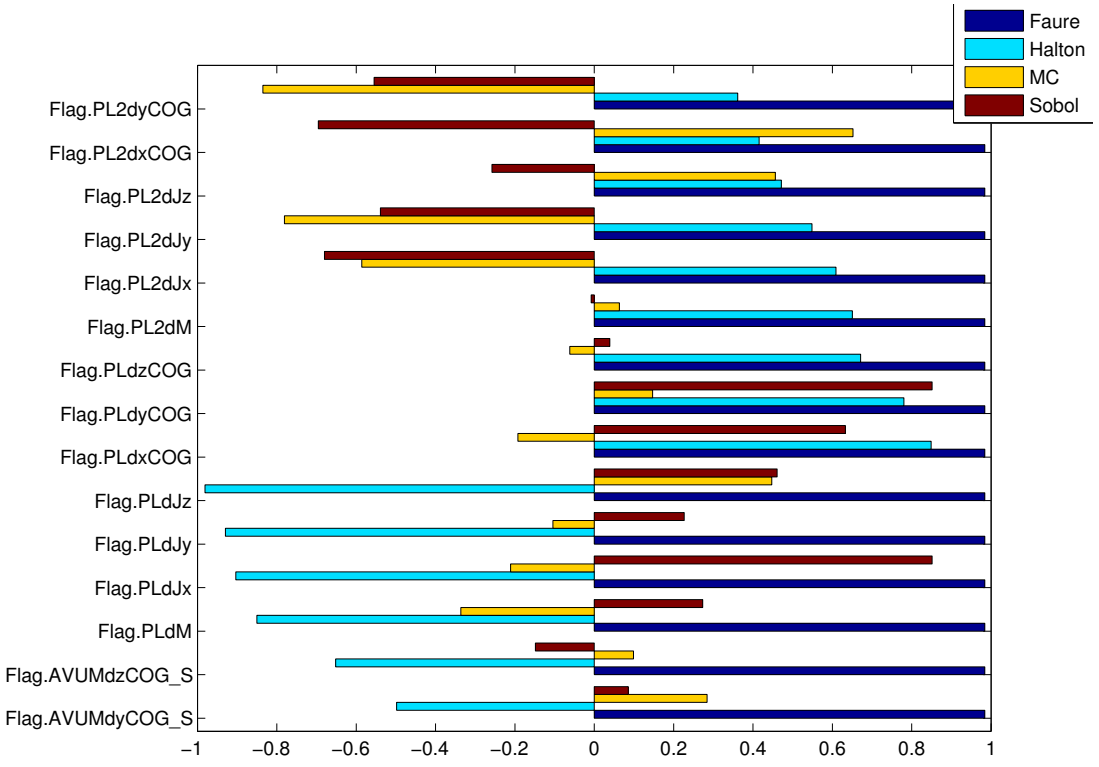


(A) Variables from 31 to 45 under different worst-case scenario

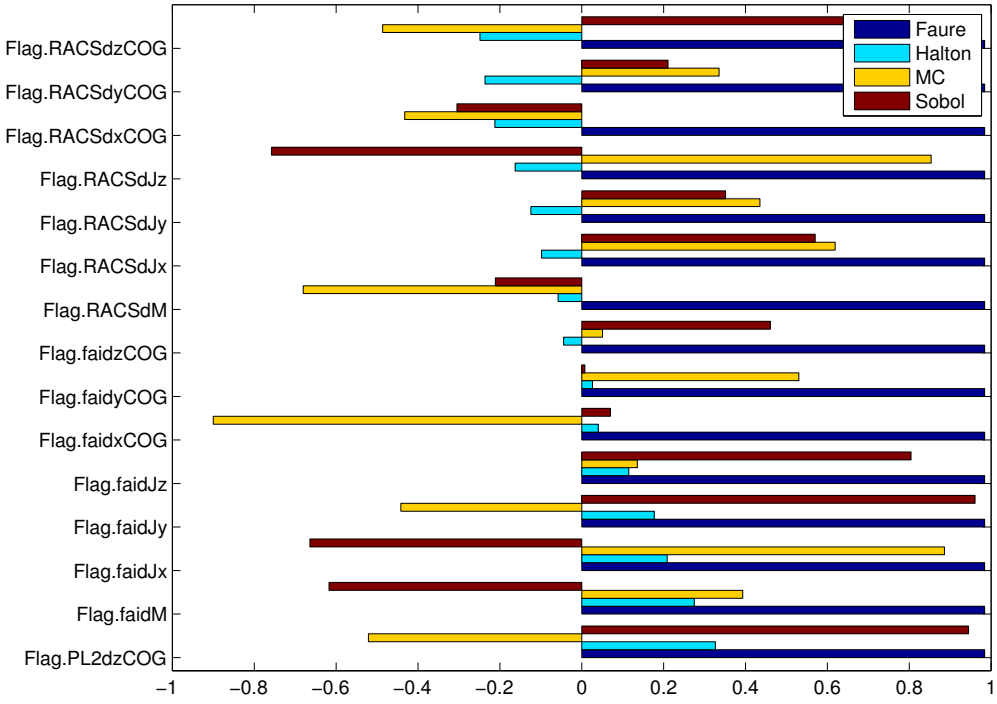


(B) Variables from 46 to 60 under different worst-case scenario

FIGURE A.15: The worst case uncertainties (31 -60) for f_1 in terms of Metric 3

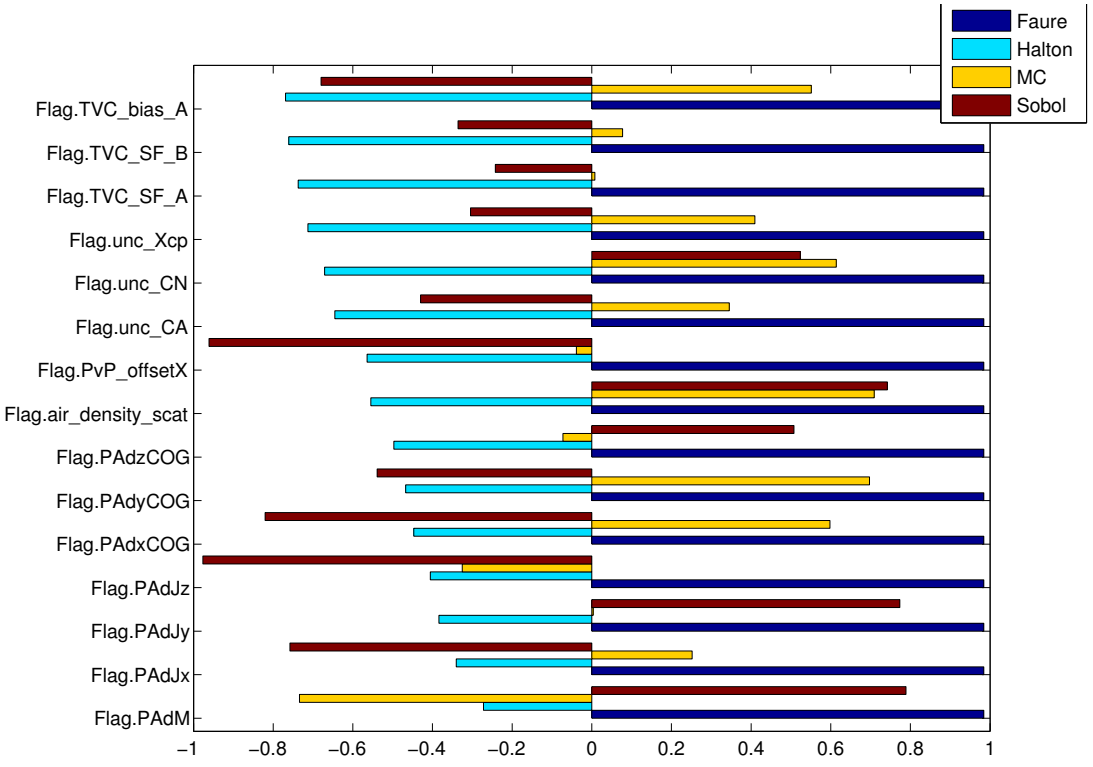


(A) Variables from 61 to 75 under different worst-case scenario

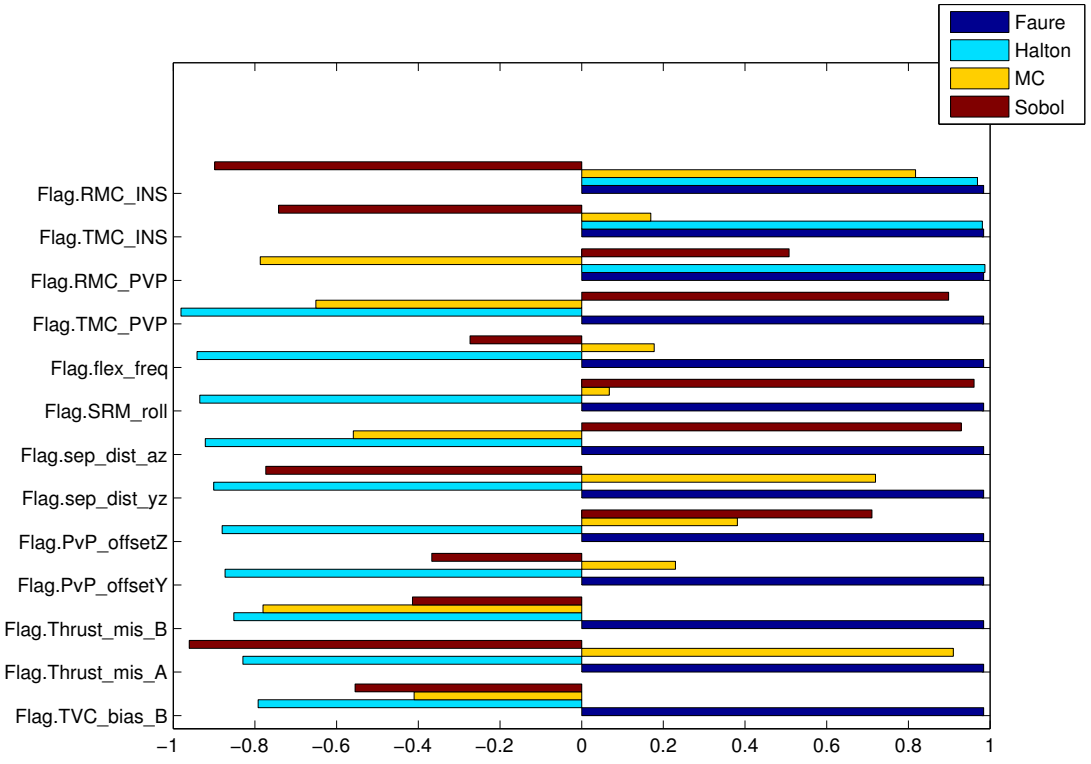


(B) Variables from 76 to 90 under different worst-case scenario

FIGURE A.16: The worst case uncertainties (61 -90) for f_1 in terms of Metric 3



(A) Variables from 91 to 105 under different worst-case scenario



(B) Variables from 106 to 118 under different worst-case scenario

FIGURE A.17: The worst case uncertainties (91 -118) for f_1 in terms of Metric 3

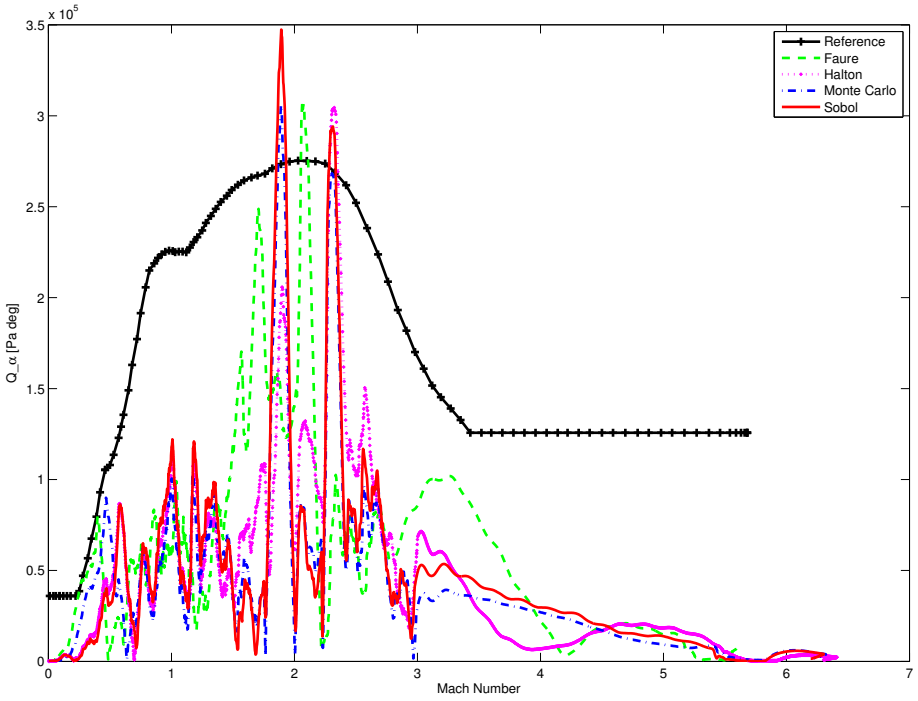


FIGURE A.18: The Q_α plot of the worst-case in terms of Metric 3 by different distribution methods

$D_{V\_Z}(t)$	Metric 1	Metric 2	Metric 3
MC	6.5858×10^4	1.1904×10^3	36.8863
Sobol	6.7640×10^4	1.2265×10^3	37.9781
Halton	6.2259×10^4	1.1147×10^3	41.7372
Faure	6.5842×10^4	1.1931×10^3	32.3711

TABLE A.2: The worst case values based on different distribution methods for cost function metric $Metric$ 1 to $Metric$ 3 in terms of $D_{V\_Z}(t)$

The Table A.3 presents the cost values in terms of the objective criterion $D_{P\_Z}(t)$ (The position drift in Z axis). The Table A.4 gives the cost values in terms of the objective criterion $D_{V\_Y}(t)$(Velocity drift in Y axis). The Table A.5 gives the cost value in terms of the objective criterion $D_{P\_Y}(t)$(Drift of position in Y axis). The Table A.6 presents the cost values in terms of the objective criterion $AoA_T(t)$. According to the worst-case results in the Table [4.1 $\sim$ A.6], the quasi-Monte Carlo demonstrate its effectiveness is at least similar as the crude Monte Carlo for the VEGA launch vehicle worst-case validation with the high dimension of uncertainties. For all of objective functions, such as f_1,f_3 and f_5, the quasi-Monte Carlo has a much larger worst-case value than the Monte

$D_{P\_Z}(t)$	Metric 1	Metric 2	Metric 3
MC	2.2243×10^6	4.0377×10^4	1.3490×10^3
Sobol	2.1732×10^6	3.9602×10^4	1.3493×10^3
Halton	2.0573×10^6	3.6758×10^4	1.6295×10^3
Faure	2.0545×10^6	3.7039×10^4	1.2179×10^3

TABLE A.3: The worst case values based on different distribution methods for cost function metric $Metric$ 1 to $Metric$ 3 in terms of $D_{P\_Z}(t)$

$D_{V\_Y}(t)$	Metric 1	Metric 2	Metric 3
MC	7.0524×10^4	1.2847×10^3	37.2564
Sobol	6.5656×10^4	1.2216×10^3	36.4732
Halton	6.3046×10^4	1.1422×10^3	41.7374
Faure	7.7164×10^4	1.3866×10^3	34.6496

TABLE A.4: The worst case values based on different distribution methods for cost function metric $Metric$1 to $Metric$ 3 in terms of $D_{V\_Y}(t)$

$D_{P\_Y}(t)$	Metric 1	Metric 2	Metric 3
MC	2.1811×10^6	3.9771×10^4	1.4375×10^3
Sobol	2.0774×10^6	3.8075×10^4	1.3230×10^3
Halton	2.0288×10^6	3.6399×10^4	1.6026×10^3
Faure	2.2993×10^6	4.1653×10^4	1.3258×10^3

TABLE A.5: The worst case values based on different distribution methods for cost function metric $Metric$1 to $Metric$3 in terms of $D_{P\_Y}(t)$

$AoA_T(t)$	Metric 1	Metric 2	Metric 3
MC	6.0353×10^3	109.3784	2.9997
Sobol	6.0779×10^3	110.3587	2.9995
Halton	6.0244×10^3	109.6962	2.9999
Faure	5.8920×10^3	106.3448	2.9994

TABLE A.6: The worst case values based on different distribution methods for cost function metric $Metric$ 1 to $Metric$ 3 in terms of $AoA_T(t)$

Carlo. Although the difference of the worst-case value is small for most objective criteria, the quasi-Monte Carlo obtains a better worst-case value than the value obtained by Monte Carlo overall. Only there is a few cases that the Monte Carlo method obtains a slightly better worst-case value than some methods of the quasi-Monte Carlo. Generally speaking, the quasi-Monte Carlo has a better effectiveness for robustness analysis than the crude Monte Carlo.

Bibliography

[1] Bong Wie. *Space vehicle dynamics and control*. American Institute of Aeronautics and Astronautics, 2008.

[2] M.Bartolimi. Object:vegacontrol user manual. *ESA technical note*, 2013.

[3] John M Hanson and Bernard B Beard. Applying monte carlo simulation to launch vehicle design and requirements verification. *Journal of Spacecraft and Rockets*, 49 (1):136–144, 2012.

[4] John M Hanson and Charlie E Hall. Learning about ares i from monte carlo simulation. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, pages 18–21, 2008.

[5] David Thipphavong. Accelerated monte carlo simulation for safety analysis of the advanced airspace concept. *system*, 3:5, 2010.

[6] Luis G Crespo, Daniel P Giesy, and Sean P Kenny. Robustness analysis and robust design of uncertain systems. *AIAA journal*, 46(2):388–396, 2008.

[7] Pierre L?Ecuyer. Quasi-monte carlo methods with applications in finance. *Finance and Stochastics*, 13(3):307–349, 2009.

[8] Junyi Lin and Xiaoqun Wang. New brownian bridge construction in quasi-monte carlo methods for computational finance. *Journal of Complexity*, 24(2):109–133, 2008.

[9] Xiaoqun Wang and Ian H Sloan. Quasi-monte carlo methods in financial engineering: An equivalence principle and dimension reduction. *Operations Research*, 59 (1):80–95, 2011.

[10] Maxime Dion and Pierre L'Ecuyer. American option pricing with randomized quasi-monte carlo simulations. In *Simulation Conference (WSC), Proceedings of the 2010 Winter*, pages 2705–2720. IEEE, 2010.

[11] Russel E Caflisch. Monte carlo and quasi-monte carlo methods. *Acta numerica*, 7: 1–49, 1998.

[12] John H Halton. On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerische Mathematik*, 2(1):84–90, 1960.

[13] William J Morokoff and Russel E Caflisch. Quasi-monte carlo integration. *Journal of computational physics*, 122(2):218–230, 1995.

[14] Harald Niederreiter. Quasi-monte carlo methods for multidimensional numerical integration. In *Numerical Integration III*, pages 157–171. Springer, 1988.

[15] Xiaoqun Wang and Kai-Tai Fang. The effective dimension and quasi-monte carlo integration. *Journal of Complexity*, 19(2):101–124, 2003.

[16] Marc C Kennedy and Anthony O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3): 425–464, 2001.

[17] Jeremy Oakley. Eliciting gaussian process priors for complex computer codes. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 51(1):81–97, 2002.

[18] Jeremy E Oakley and Anthony O'Hagan. Probabilistic sensitivity analysis of complex models: a bayesian approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 66(3):751–769, 2004.

[19] Carl Edward Rasmussen. Gaussian processes for machine learning. 2006.

[20] Andrés Marcos, Hector Garcia De Marina, Valerio Mantini, Christophe Roux, and Samir Bennani. Optimization-based worst-case analysis of a launcher during the atmospheric ascent phase.

[21] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

[22] K. Price R. Storn. Differential evolution - a simple and efficient heuristic for golbal optimisation over continunous space. *Journal of Global Optimisation*, 11:341–359, 1997.

[23] S Oudin, G Puyou, G Ferreres, and P Mouyon. Design and worst-case validation of a longitudinal adaptive flight control law: a practical approach. 2013.

[24] Prathyush P Menon, Emmanuel Prempain, Ian Postlethwaite, Declan Bates, and Samir Bennani. Nonlinear worst-case analysis of an lpv controller for approach-phase of a re-entry vehicle. In *AIAA GNC*, 2009.

[25] Wenfei Wang, Prathyush Menon, Declan Bates, and Samir Bennani. Verification and validation of attitude and orbit control systems for flexible satellites. In *AIAA Guidance, Navigation, and Control Conference*, page 5953, 2009.

[26] Jorge E Hurtado and Diego A Alvarez. A method for enhancing computational efficiency in monte carlo calculation of failure probabilities by exploiting form results. *Computers & Structures*, 117:95–104, 2013.

[27] Rémi Arnaud and Fabrice Poirion. Stochastic annealing optimization of uncertain aeroelastic system. *Aerospace Science and Technology*, 2014.

[28] William Feller. *An Introduction to Probability Theory and Its Applications: Volume One*. John Wiley & Sons, 1950.

[29] Peter Jäckel and Russ Bubley. *Monte Carlo methods in finance*. J. Wiley, 2002.

[30] Harald Niederreiter. *Random number generation and quasi-Monte Carlo methods*, volume 63. SIAM, 1992.

[31] Harald Niederreiter. Point sets and sequences with small discrepancy. *Monatshefte für Mathematik*, 104(4):273–337, 1987.

[32] John M Hammersley. Monte carlo methods for solving multivariable problems. *Annals of the New York Academy of Sciences*, 86(3):844–874, 1960.

[33] Ilya M Sobol. On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Computational Mathematics and Mathematical Physics*, 7(4): 86–112, 1967.

[34] Henri Faure. Discrépance de suites associées à un système de numération (en dimension s). *Acta Arithmetica*, 41(4):337–351, 1982.

[35] Bradley Moskowitz and Russel E Caflisch. Smoothness and dimension reduction in quasi-monte carlo methods. *Mathematical and Computer Modelling*, 23(8):37–54, 1996.

[36] Sergei Kucherenko, Balazs Feil, Nilay Shah, and Wolfgang Mauntz. The identification of model effective dimensions using global sensitivity analysis. *Reliability Engineering & System Safety*, 96(4):440–449, 2011.

[37] E Yücesan, CH Chen, JL Snowdon, and JM Charnes. Enhanced quasi-monte carlo methods with dimension reduction. 2003.

[38] Russel E Caflisch, William J Morokoff, and Art B Owen. *Valuation of mortgage backed securities using Brownian bridges to reduce effective dimension*. Department of Mathematics, University of California, Los Angeles, 1997.

[39] Thomas Mannchen, Declan G Bates, and Ian Postlethwaite. Modeling and computing worst-case uncertainty combinations for flight control systems analysis. *Journal of guidance, control, and dynamics*, 25(6):1029–1039, 2002.

[40] Vedat Toğan and Ayşe T Daloğlu. An improved genetic algorithm with initial population strategy and self-adaptive member grouping. *Computers & Structures*, 86(11):1204–1218, 2008.

[41] P Victer Paul, A Ramalingam, R Baskaran, P Dhavachelvan, K Vivekanandan, and R Subramanian. A new population seeding technique for permutation-coded genetic algorithm: Service transfer approach. *Journal of Computational Science*, 5 (2):277–297, 2014.

[42] Pedro A Diaz-Gomez and Dean F Hougen. Empirical study: Initial population diversity and genetic algorithm performance. *Artificial Intelligence and Pattern Recognition*, 2007:334–341, 2007.

[43] Andrea Saltelli, Karen Chan, E Marian Scott, et al. *Sensitivity analysis*, volume 1. Wiley New York, 2000.

[44] Jon C Helton, Jay Dean Johnson, Cedric J Sallaberry, and Curt B Storlie. Survey of sampling-based methods for uncertainty and sensitivity analysis. *Reliability Engineering & System Safety*, 91(10):1175–1209, 2006.

[45] JC Helton and FJ Davis. Illustration of sampling-based methods for uncertainty and sensitivity analysis. *Risk Analysis*, 22(3):591–622, 2002.

[46] Il'ya Meerovich Sobol'. On sensitivity estimation for nonlinear mathematical models. *Matematicheskoe Modelirovanie*, 2(1):112–118, 1990.

[47] Andrea Saltelli, Paola Annoni, Ivano Azzini, Francesca Campolongo, Marco Ratto, and Stefano Tarantola. Variance based sensitivity analysis of model output. design and estimator for the total sensitivity index. *Computer Physics Communications*, 181(2):259–270, 2010.

[48] Curtis B Storlie, Laura P Swiler, Jon C Helton, and Cedric J Sallaberry. Implementation and evaluation of nonparametric regression procedures for sensitivity analysis of computationally demanding models. *Reliability Engineering & System Safety*, 94(11):1735–1763, 2009.

[49] Qiao Ge, Biagio Ciuffo, and Monica Menendez. Combining screening and metamodel-based methods: An efficient sequential approach for the sensitivity analysis of model outputs. *Reliability Engineering & System Safety*, 134:334–344, 2015.

[50] Con J Doolan. Hypersonic missile performance and sensitivity analysis. *Journal of Spacecraft and Rockets*, 44(1):81–87, 2007.

[51] Andrea Saltelli. Making best use of model evaluations to compute sensitivity indices. *Computer Physics Communications*, 145(2):280–297, 2002.

[52] S Tarantola, D Gatelli, SS Kucherenko, W Mauntz, et al. Estimating the approximation error when fixing unessential factors in global sensitivity analysis. *Reliability Engineering & System Safety*, 92(7):957–960, 2007.

[53] Ilya M Sobol. Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates. *Mathematics and computers in simulation*, 55(1):271–280, 2001.

[54] John H Blakelock. *Automatic control of aircraft and missiles*. John Wiley & Sons, 1991.

[55] Paul Bratley and Bennett L Fox. Algorithm 659: Implementing sobol's quasirandom sequence generator. *ACM Transactions on Mathematical Software (TOMS)*, 14(1): 88–100, 1988.

[56] Michael A Osborne, Roman Garnett, and Stephen J Roberts. Gaussian processes for global optimization. In *3rd international conference on learning and intelligent optimization (LION3)*, pages 1–15, 2009.

[57] Marcus Frean and Phillip Boyle. Using gaussian processes to optimize expensive functions. In *AI 2008: Advances in Artificial Intelligence*, pages 258–267. Springer, 2008.

[58] Dirk Büche, Nicol N Schraudolph, and Petros Koumoutsakos. Accelerating evolutionary algorithms with gaussian process fitness function models. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 35(2): 183–194, 2005.

[59] Daniel J Lizotte, Tao Wang, Michael H Bowling, and Dale Schuurmans. Automatic gait optimization with gaussian process regression. In *IJCAI*, volume 7, pages 944–949, 2007.

[60] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias W Seeger. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *Information Theory, IEEE Transactions on*, 58(5):3250–3265, 2012.

[61] Robert B Gramacy and Nicholas G Polson. Particle learning of gaussian process models for sequential design and optimization. *Journal of Computational and Graphical Statistics*, 20(1), 2011.

[62] Jin Yuan, Kesheng Wang, Tao Yu, and Minglun Fang. Reliable multi-objective optimization of high-speed wedm process based on gaussian process regression. *International Journal of Machine Tools and Manufacture*, 48(1):47–60, 2008.

[63] Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4): 455–492, 1998.

[64] GEB Archer, A Saltelli, and IM Sobol. Sensitivity measures, anova-like techniques and the use of bootstrap. *Journal of Statistical Computation and Simulation*, 58 (2):99–120, 1997.

[65] Amandine Marrel, Bertrand Iooss, Beatrice Laurent, and Olivier Roustant. Calculations of sobol indices for the gaussian process metamodel. *Reliability Engineering & System Safety*, 94(3):742–751, 2009.