Sensitivity Analysis and Uncertainty Integration for System Diagnosis and Prognosis

By
Chenzhao Li

Dissertation
Submitted to the Faculty of the
Graduate School of Vanderbilt University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY
in

Civil Engineering
December 2016
Nashville, Tennessee

Approved:
Sankaran Mahadevan, Ph.D
Douglas Adams, Ph.D.
Prodyot Basu, Ph.D.
Caglar Oskay, Ph.D.
Liping Wang, Ph.D.

To my mother Xiuying Shen, my father Qingguo Li, and my wife Junjie Zhao

ACKNOWLEDGEMENTS

First and foremost, I express my greatest gratitude to my advisor Prof. Sankaran Mahadevan, for his expert professional advice and guidance throughout my Ph.D. program at Vanderbilt University. His constant encouragement has been a great source of motivation not only in my academic career but also in my personal life. I am extremely indebted to him, for all that I have learned during my graduate education, and look forward to continuing this relationship for the rest of my professional career.

I would like to thank my committee members, Professor Douglas Adams, Professor Prodyot Basu, Professor Caglar Oskay, and Dr. Liping Wang, for their kind support and wise advice for my research and career development.

I would like to thank our research collaborators from Sandia National Laboratories, including Dr. Angel Urbina, Dr. Josh Mullins, and Dr. Kenneth Hu, for sharing their simulation codes and all the valuable discussion. I also would like to our research collaborators from Dr. Liping Wang's Probabilistics Lab in GE Global Research, including Dr. You Ling, Dr. Isaac Asher, Dr. Felipe Viana, Dr. Serio Choze and Dr. Youngwon Shin, for all the valuable discussions and giving me the opportunity of the internship in GE Global Research in 2015 summer.

I am also grateful to work with so many brilliant people at Vanderbilt University. In particular, I am very lucky to have Dr. Shankar Sankararaman and Dr. You Ling as my mentors as I started my research. Both of them guided me to the entrance of the research field and gave me detailed and even hands-on help. I also thank my teammates from Prof. Mahadevan's group, including Dr. Zhen Hu, Dr. Chen Liang (working in Ford now), Dr. Josh Mullins (working in Sandia National Laboratories now), Saideep Nannapaneni, Erin DeCarlo, Ghina Absi, Guowei Cai, Harshini Devathi, Nath Paromita, Kyle Neal, Eric Vanderhorn, Yanqing Bao, Dan Ao, and Xiaoge Zhang, for all the teamwork in coding, brainstorming in the whiteboard, and presentations in our group meeting. I would also thank the visiting Scholars from Southwest University in China, including Pr. Yong Deng, Xinyang Deng, Xiaoyan Su, and Peida Xu, for their new ideas brought to our group and the collaboration in producing papers.

In addition, I would like to thank all the other friends in Nashville. My deep appreciations to Dr. Tong Hui, Shuhai Zhang, Xiang Zhang, Jingjing Bu, Yiyuan Zhao, Xujie Si, and many others. My life couldn't have been so colorful without sharing my time with them.

Most importantly, I would like to thank the most important people in my life: my parents and my wife. This dissertation would not have been possibly done without their priceless support and love.

The research in this dissertation was supported by funding from two sources: 1) Sandia National Laboratories through contract no. BG-7732 (Technical Monitor: Dr. Angel Urbina); and 2) US Air Force Research Laboratory under Contract no. FA8650-14-D-2443 (Project Monitors: Douglas Henderson, Pamela Coburn, and Eric Teugel, at the US Air Force Research Laboratory, Wright Pattern Air Force Base). Some numerical studies were conducted using the Advanced Computing Center for Research and Education (ACCRE) at Vanderbilt University. All of these sources of support are gratefully acknowledged.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

1.1 Overview

Predicting the response of a system of interest at unknown input conditions is one primary task in engineering. This task involves many major activities, such as physics modeling, uncertainty quantification, statistical inference, probabilistic analysis, sensitivity analysis, etc. Predicting the system response is not simply propagating the input through the computational model of the system, since various uncertainty sources are involved in the prediction, including input uncertainty, model discrepancy, model parameter uncertainty, surrogate model uncertainty, measurement error, etc. All of these uncertainty sources can be categorized into two types: the aleatory uncertainty caused by natural variability that cannot be reduced, and the epistemic uncertainty caused by lack of knowledge that can be reduced by collecting more information.

It is desirable to reduce the epistemic uncertainty by using available information such as experimental data, thus reducing the uncertainty propagated to the prediction, so that the prediction can be more accurate. This activity is related to several topics, including 1) optimization of the data collection effort within limited resources, 2) model calibration to reduce the epistemic uncertainty with available test results, and 3) model validation to evaluate the quality of the calibrated model. The third step is necessary to guarantee that the reduced uncertainty is converging to the true value of the quantity of interest, instead of biasing away from it.

Computational efficiency in model calibration is another concern in the system response prediction. Two strategies are possible the improve this efficiency: 1) reduce the dimension of model calibration by identifying and fixing non-important uncertainty sources, and 2) develop more efficient and scalable inference algorithms. The first strategy can be realized by sensitivity analysis. However, the sensitivity analysis considering both aleatory and epistemic uncertainty sources is not well-established [1–3], and the computational efficiency of the existing sensitivity analysis algorithms is not satisfactory. New developments in sensitivity analysis will be one objective in this dissertation. The second strategy depends on the mathematical tool used for model calibration, and this dissertation selects the Bayesian network (BN). While efficient analytical inference algorithms (either approximate or exact) for the BN with discrete variables have been well-established in the literature, the inference in BN with continuous variables is still challenging if the BN is nonlinear and/or non-Gaussian, and this will be another objective in this dissertation.

Another concern in system response prediction is uncertainty integration, which includes two challenges. First, the various uncertainty sources in system response prediction are usually correlated, thus integration across these uncertainty sources are required. Second, the results from model calibration and model validation need to be integrated, especially when alternative results for the same quantity of interest are present. This dissertation also aims to contribute to solving these two challenges.

In the rest of this chapter, Section 1.2 proposes the research objectives in the dissertation based on the introduction above, then Section 1.3 illustrates the organization of this dissertation section by section.

1.2 Research Objectives

The overall goal of the proposed research is to develop a versatile and efficient framework for system response prediction under aleatory and epistemic uncertainty. In this research, both time independent and time dependent systems are considered. Various uncertainty integration techniques are utilized, including model calibration, model validation, sensitivity analysis, Bayesian network, etc. The innovations of the proposed research are mainly related to global sensitivity analysis and Bayesian network.

Five objectives are pursued to achieve the overall goal. Since sensitivity analysis contributes to reducing the dimension of our prediction challenge prior to other activities, the first task is to develop a computational framework to compute the sensitivity indices that quantify the relative contributions of various aleatory/epistemic uncertainty sources towards the system response prediction uncertainty, where both random variable input and time series input should be considered.

The second objective is the system response prediction of time independent systems. This objective is straightforward if adequate amounts of system test data are available. However, challenges emerge if 1) test data from the system of interest may not be available so the prediction relies on the data from component or sub-system tests; 2) the test budget is limited thus an optimum allocation of test resources is needed.

The third objective is the system response prediction for time-dependent systems. In this case, the evolution of the state variables of the system need to be tracked, thus the resultant prediction also varies over time.

Global sensitivity analysis (GSA) is heavily used in earlier objectives. However, computational efficiency is always a bottle-neck to use GSA in uncertainty integration. Therefore, the fourth objective is to propose a new efficient algorithm to compute the sensitivity index.

Beside GSA, another foundation mathematical tool of this research is the Bayesian network (BN). Thus the fifth objective is to improve the uncertainty reduction efficacy and the computational efficiency of the BN. The uncertainty reduction efficacy is measured by whether after inclusion in the BN, the observation data are effective in reducing the uncertainty of the state variables via Bayesian inference. And the computational efficiency refers to the reduction in the time cost of the Bayesian inference, in both static Bayesian and dynamic Bayesian networks.

1.3 Organization of the Dissertation

The subsequent chapters of this dissertation will be devoted to the objectives proposed above.

Chapter 2 provides an introduction to the tools and methods for system response prediction considered in this research, including 1) Bayesian network, 2) Bayesian inference basics, 3) Bayesian inference algorithms, 4) various uncertainty sources in system response prediction, 5) model calibration and model validation, 6) global sensitivity analysis, and 7) auxiliary variable method.

Chapter 3 develops a novel computational framework to compute the Sobol' sensitivity indices that quantify the relative contributions of various uncertainty sources towards the system response prediction uncertainty. The proposed framework is developed for two types of model inputs: random variable input and time series input and both aleatory and epistemic uncertainty sources are considered. A novel controlled-seed computational technique based on pseudo-random number generation is proposed to efficiently represent the natural variability in the time series input. This controlled-seed method significantly accelerates the Sobol' indices computation under time series input and makes it computationally affordable.

Chapter 4 addresses the system response prediction for a complex multi-level problem. In this problem, the lack of data at the system level makes it impossible to conduct model calibration directly. So system model parameters are estimated using tests at lower levels of complexity which share the same model parameters with the system. The results of calibration, validation, and the proposed sensitivity-based relevance analysis are integrated into a roll-up method to predict the system output.

Chapter 5 aims to achieve “robust” test resource allocation, which means that the system response prediction is insensitive to the variability in test outcomes, therefore, consistent predictions can be achieved under different test outcomes. It is concluded that this objective can be achieved if the contribution of model parameter uncertainty in the synthetic data can be maximized. Global sensitivity analysis (Sobol’ index) is used to assess this contribution, and to formulate an optimization problem to achieve the desired consistent prediction.

Chapter 6 extends the discussion on system response prediction in Chapter 4 and Chapter 5 to time dependent systems, where the concept of dynamic Bayesian network (DBN) is used. The DBN integrates physics models and various aleatory (random) and epistemic (lack of knowledge) uncertainty sources in crack growth prediction. A modification to the DBN structure, which does not affect the diagnosis results but reduces time cost significantly, is also proposed. By using particle filter as the Bayesian inference algorithm for the DBN, the proposed approach handles both discrete and continuous variables of various distribution types, and non-linear relationships between nodes.

Sobol’ index is a prominent methodology in the global sensitivity analysis, thus Chapter 7 proposes a new algorithm to calculate the first-order Sobol’ index. The proposed algorithm is capable of computing the first-order index if only input-output samples are available but the underlying model is unavailable, and its computational cost is not proportional to the dimension of the model inputs. In addition, the proposed method can also estimate the first-order index with correlated model inputs. Considering that the first-order index is the desired metric to rank model inputs but current methods can only handle independent model inputs, the proposed algorithm contributes to filling this gap.

Chapter 8 extends the usage of global sensitivity analysis (GSA) from deterministic model to stochastic model, i.e., Bayesian network. The proposed method aims to calculate the Sobol' sensitivity index of a node with respect to the node of interest. Before collecting observations, the proposed algorithm can predict the uncertainty reduction of the node of interest purely using the prior distribution samples, thus providing quantitative guidance for effective observation and updating.

The inference is one key objective of a Bayesian network, and Chapter 9 proposes an efficient approximate inference algorithm for a continuous Bayesian network. A network collapsing technique is proposed to convert a multi-layer BN to an equivalent simple two-layer BN so that the unscented Kalman filter can be applied to the collapsed BN and the posterior distributions of state variables can be obtained analytically. For dynamic BN, the proposed method is also able to propagate the state variables to the next time step analytically using the unscented transform, based on the assumption that the posterior distributions of state variables are Gaussian. Thus the proposed method achieves a very fast approximate solution, making it particularly suitable for dynamic BN where inference and uncertainty propagation are required over many time steps.

CHAPTER 2

BACKGROUND CONCEPTS AND METHODS

2.1 Introduction to Bayesian Network

During the past 30 years, the Bayesian network (BN) has become a key method for representation and reasoning under uncertainty in the fields of engineering [4,5], machine learning [6,7], artificial intelligence [8,9], etc. BN is a directed acyclic graph (DAG) model which means that all the nodes are connected by directed edges and along the directions of these edges we cannot find a cycle with the same node as the starting and ending node. An example of a DAG model is given in Figure 2.1.

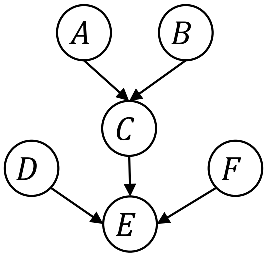


Figure 2.1 DAG model example

In a BN, random variables are denoted by nodes (vertices) and their dependence relationships are denoted by directed edges (arcs). An edge indicates the conditional dependence of the down-stream child node on the upstream parent node(s). This dependence is described mathematically by a conditional probability distribution (CPD), which can be as simple as a small table, or as complex as a stochastic model. The entire BN represents the joint distribution of the random variables. Denote the random variables in a BN as $\boldsymbol{X} = \{X_1, X_2, \dots, X_n\}$. Based on the chain rule in probability theory, the joint distribution of $\boldsymbol{X}$ is

$$p(\boldsymbol{X}) = \prod_{i=1}^{n} p\left(X_i|\mathrm{Pa}_{X_i}\right) \tag{2.1}$$

where $p\left(X_i|\mathrm{Pa}_{X_i}\right)$ denotes the CPD of X_i and Pa_{X_i} denotes the parent nodes of X_i. Note that $p\left(X_i|\mathrm{Pa}_{X_i}\right) = p(X_i)$ if X_i does not have any parent node, and X_i is a root node. If Figure 2.1 is considered as a BN, its root nodes are A, B, D and E. Based on Eq. (2.1), the joint distribution of this BN is

$$p(A, B, C, D, E, F) = p(A)p(B)p(C|A, B)p(D)p(F)p(E|D, F) \tag{2.2}$$

BN can take different types of random variables as nodes, including discrete and continuous variables of different distribution types. A BN with discrete variables only is called a discrete BN, and a BN with continuous variables only is called a continuous BN. A BN with both discrete and continuous variables is called a hybrid BN.

The BN explained above refers to a “static” Bayesian network for a time-independent system. To track a time-dependent system whose states evolve over time, the concept of BN is extended to a dynamic Bayesian network (DBN), which can be considered as a series of static BNs, one for each time instant, with additional edges connecting the state variables in adjacent time instants. One example of DBN is shown in

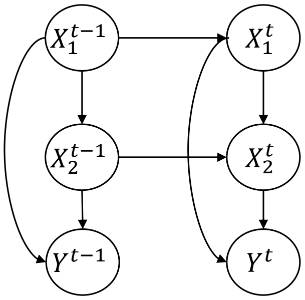


Figure 2.2 DBN example

The DBN follows first-order Markov assumption, so that:

1. The state variables of the BN at time t depend only on the state variables of the BN at time $t-1$, and this dependence and the underlying CPDs are generally assumed to be time-invariant [10];
2. The observable variable $\boldsymbol{Y}^t$ at time t only depends on the state variable $\boldsymbol{X}^t$ at the same time instant.

The following expressions and equations can be derived from this first-order Markov assumption:

$$\begin{gathered}\boldsymbol{X}^t \perp \boldsymbol{y}^{1:t-1}\,|\boldsymbol{X}^{t-1} \Rightarrow p(\boldsymbol{X}^t|\boldsymbol{y}^{1:t-1},\boldsymbol{X}^{t-1}) = p(\boldsymbol{X}^t|\boldsymbol{X}^{t-1}) \\ \boldsymbol{y}^t \perp \boldsymbol{y}^{1:t-1}|\boldsymbol{X}^t \Rightarrow p(\boldsymbol{y}^t\,|\boldsymbol{X}^t,\boldsymbol{y}^{1:t-1}) = p(\boldsymbol{y}^t|\boldsymbol{X}^t)\end{gathered} \tag{2.3}$$

In Eq. (2.3), the symbol “⊥” means “independent of”, so that the first formula in Eq. (2.3) denotes that $\boldsymbol{X}^t$ is independent of $\boldsymbol{y}^{1:t-1}$ at a given value of $\boldsymbol{X}^{t-1}$; and the second formula denotes that $\boldsymbol{y}^t$ is independent of $\boldsymbol{y}^{1:t-1}$ at a given value of $\boldsymbol{X}^t$.

In this research, Bayesian network is the main methodology for uncertainty integration, diagnosis, and prognosis. Another main methodology is global sensitivity analysis, which will be introduced in Section 2.6.

2.2 Bayesian Inference Basics

Based on the earlier discussion in Section 2.1, we can denote a BN as $\langle\langle \boldsymbol{V},\boldsymbol{E}\rangle,\boldsymbol{P}\rangle$, where $\boldsymbol{V} = \{\boldsymbol{X},\boldsymbol{Y}\}$ is the vector of nodes (random variables); $\boldsymbol{X}$ denotes the state variables to be inferred and $\boldsymbol{Y}$ denotes the observable variables; $\boldsymbol{E}$ represents the directed edges; and $\boldsymbol{P}$ denotes the CPDs for the edges in $\boldsymbol{E}$.

The research on BN includes two main topics: inference and learning. Inference aims to estimate the posterior distribution of the state variables based on the prior distribution of BN and evidence. Usually, this evidence is the observation $\boldsymbol{y}$ of nodes $\boldsymbol{Y}$, thus the inference is to calculate the posterior probability distribution $p(\boldsymbol{X}|\boldsymbol{Y}=\boldsymbol{y})$. In contrast, learning aims to construct the DAG and estimate the CPD for each edge based on the data of the random variables; thus learning calculates $\boldsymbol{E}$ and $\boldsymbol{P}$. This research focuses on inference, i.e., calculating $p(\boldsymbol{X}|\boldsymbol{Y}=\boldsymbol{y})$. The inference is based on Bayes' theorem:

$$p(\boldsymbol{X}|\boldsymbol{Y}=\boldsymbol{y}) \propto p(\boldsymbol{X})p(\boldsymbol{Y}=\boldsymbol{y}|\boldsymbol{X}) \tag{2.4}$$

where $p(\boldsymbol{X})$ and $p(\boldsymbol{X}|\boldsymbol{Y}=\boldsymbol{y})$ are the prior and posterior distributions of state variables $\boldsymbol{X}$, and $p(\boldsymbol{Y}=\boldsymbol{y}|\boldsymbol{X})$ is the likelihood function of $\boldsymbol{X}$. The likelihood function can be understood as the probability to observe $\boldsymbol{Y}=\boldsymbol{y}$ at given value of $\boldsymbol{X}$, so that it is a function of $\boldsymbol{X}$ and we denote it as $L(\boldsymbol{X})$. Assume that $\boldsymbol{Y}=\{Y_1,Y_2,\dots,Y_m\}$ and correspondingly $\boldsymbol{y}=\{y_1,y_2,\dots,y_m\}$, then the expression of the likelihood function is:

$$L(\boldsymbol{X}) = \prod_{i=1}^{m} p\left(Y_i = y_i|\mathrm{Pa}_{Y_i}\right) \tag{2.5}$$

where $\mathrm{Pa}_{Y_i} \in \boldsymbol{X}$ is the parents nodes of Y_i and $p\left(Y_i = y_i|\mathrm{Pa}_{Y_i}\right)$ is the PDF value at $Y_i = y_i$ of the CPD for Y_i. It is easy to see that $p\left(Y_i = y_i|\mathrm{Pa}_{Y_i}\right)$ is a function of Pa_{Y_i}, and the product is a function of X due to $\mathrm{Pa}_{Y_i} \in \boldsymbol{X}$. Note that if Y_i has no parent node, its corresponding term reduces to $p(Y_i = y_i)$, which is the PDF value of the prior distribution of Y_i at the location of y_i, and it is simply a constant.

Note that Eq. (2.5) is for data obtained in a single experiment. In the case of data from multiple independent experiments, the entire likelihood function will be the product of the likelihood function for each single experiment.

In a dynamic Bayesian network (DBN), inference estimates the probability $p(\boldsymbol{X}^t|\boldsymbol{y}^{1:t}\ )$, i.e., the posterior distribution of the state variables in the current time instant given observations in the past and current time instants. The inference in a DBN is a recursive process across time instants. Using Eq. (2.3) and Bayes' theorem in Eq. (2.4), if $\boldsymbol{X}^t$ are continuous variables we have

$$\begin{aligned} & p(\boldsymbol{X}^t|\boldsymbol{y}^{1:t}) \propto p(\boldsymbol{X}^t|\boldsymbol{y}^{1:t-1})p(\boldsymbol{y}^t|\boldsymbol{X}^t,\boldsymbol{y}^{1:t-1}) \\ & = \left[\iint p(\boldsymbol{X}^t|\boldsymbol{y}^{1:t-1},\boldsymbol{X}^{t-1})p(\boldsymbol{X}^{t-1}|\boldsymbol{y}^{1:t-1})\mathrm{d}\boldsymbol{X}^{t-1}\right]p(\boldsymbol{y}^t|\boldsymbol{X}^t) \\ & = \left[\iint p(\boldsymbol{X}^t|\boldsymbol{X}^{t-1})p(\boldsymbol{X}^{t-1}|\boldsymbol{y}^{1:t-1})\,\mathrm{d}\boldsymbol{X}^{t-1}\right]p(\boldsymbol{y}^t|\boldsymbol{X}^t) \end{aligned} \tag{2.6}$$

In Eq. (2.6), $p(\boldsymbol{y}^t|\boldsymbol{X}^t,\boldsymbol{y}^{1:t-1})$ is replaced by $p(\boldsymbol{y}^t|\boldsymbol{X}^t)$ based on the second formula of Eq. (2.3); and $p(\boldsymbol{X}_t|\boldsymbol{y}_{1:t-1},\boldsymbol{X}_{t-1})$ is replaced by $p(\boldsymbol{X}^t|\boldsymbol{X}^{t-1})$ based on the first formula of Eq. (2.3). Then Eq. (2.6) can be rewritten as $p(\boldsymbol{X}^t|\boldsymbol{y}^{1:t}) \propto [\int p(\boldsymbol{X}^t|\boldsymbol{X}^{t-1})p(\boldsymbol{X}^{t-1}|\boldsymbol{y}^{1:t-1})\,\mathrm{d}\boldsymbol{X}^{t-1}]p(\boldsymbol{y}^t|\boldsymbol{X}^t)$, where the terms on the right-hand side indicate two components in estimating $p(\boldsymbol{X}^t|\boldsymbol{y}^{1:t})$:

1. Propagate the posterior distribution $p(\boldsymbol{X}^{t-1}|\boldsymbol{y}^{1:t-1})$ obtained at time $t-1$ through the transient CPD $p(\boldsymbol{X}^t|\boldsymbol{X}^{t-1})$ and marginalize over $\boldsymbol{X}^{t-1}$ to construct the prior distribution $p(\boldsymbol{X}^t|\boldsymbol{y}^{1:t-1})$ at time t;
2. Calculate the likelihood function $p(\boldsymbol{y}^t|\boldsymbol{X}^t)$ based on Eq. (2.5), which only utilizes the observation at time t.

These two components also imply that the state variables and observations at earlier time instants can be neglected once the prior distribution $p(\boldsymbol{X}^t|\boldsymbol{y}^{1:t-1})$ at time t is constructed. This process is repeated for the BN in each time instant in order to track the evolution of the state variables over time.

Note that if $\boldsymbol{X}^t$ are discrete variables, Eq. (2.6) will be re-derived as $p(\boldsymbol{X}^t|\boldsymbol{y}^{1:t}) \propto [\sum_{\boldsymbol{X}^{t-1}} p(\boldsymbol{X}^t|\boldsymbol{X}^{t-1})p(\boldsymbol{X}^{t-1}|\boldsymbol{y}^{1:t-1})]p(\boldsymbol{y}^t|\boldsymbol{X}^t)$. The implication of the two components in the previous paragraph is still valid.

In Eq. (2.4) for static BN and Eq. (2.6) for DBN, the product of the prior distribution and the likelihood function is only proportional to but not equal to the posterior distribution. Thus a specific inference algorithm, either exact or approximate, is required to calculate the PDF/PMF value of the posterior distribution or generate random samples representing the posterior distribution. Fast, analytical inference algorithms for static/dynamic BN with discrete variables have been well-developed in the literature, but the current algorithms for static/dynamic BN with continuous variables are either time-consuming or restricted to specific CPDs and/or BN topology. A literature review of inference algorithms is provided below in Section 2.3.

2.3 Bayesian Inference Algorithms

2.3.1 Static BN

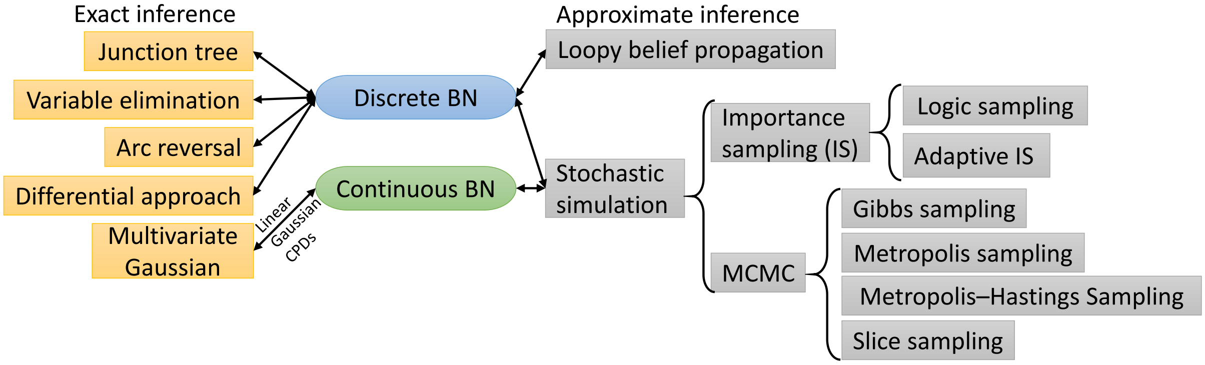


Figure 2.3 Class of inference algorithms for static BN

Exact and approximate inference algorithms for static BN have been developed in the literature, as shown in Figure 2.3. For a static BN with discrete variables, exact inference is always possible and available algorithms include the most popular Junction tree algorithm [11], the variable elimination algorithm [12], the arc reversal method [13], the differential approach [14], etc.

However, exact inference is computationally prohibitive for large networks, thus approximate inference algorithms such as loopy belief propagation [15] have been developed to improve the computational efficiency.

For a static BN with continuous variables, if all the root nodes (i.e., nodes without parents) have Gaussian distributions and all the edges from parent nodes $\boldsymbol{U} \in \mathbb{R}^{N_U}$ to child node $V \in \boldsymbol{V}$ are linear Gaussian CPDs such that $p(V|\boldsymbol{U}) \sim N(\boldsymbol{W}_V U + \boldsymbol{\mu}_V, \sigma_V^2)$ where matrix $\boldsymbol{W}_V \in \mathbb{R}^{N_U \times N_U}$ and vector $\boldsymbol{\mu}_V \in \mathbb{R}^{N_U}$ and variance $\sigma_V^2 \in \mathbb{R}$ have been predefined, then the joint distribution of $\boldsymbol{V}$ is multivariate Gaussian. Inference $p(\boldsymbol{X}|\boldsymbol{Y} = \boldsymbol{y})$ for this static BN is simply a conditional Gaussian distribution and the exact solution can be found in Ref. [16].

A more general static BN will have non-Gaussian variables, thus a sampling-based approximate inference algorithm (referred to here as stochastic simulation) is needed. This is a family of algorithms categorized into importance sampling (IS) and Markov Chain Monte Carlo (MCMC) methods. The major difference between these two categories is that the IS generates samples independently from an importance function in one shot, while the MCMC methods generate samples sequentially thus the next sample depends on the current sample. IS has several variants including 1) the logic sampling algorithm [17] where the importance function is the prior distribution of BN; and 2) the adaptive importance sampling algorithm [18,19] where the importance function is optimized adaptively. Note that these stochastic simulation algorithms are also applicable for a static BN with discrete variables.

As shown in Figure 2.3, usually the stochastic simulation algorithms are the only choice for a static BN with continuous non-Gaussian variables. These sample-based methods are computationally expensive for large networks. In this research, a more efficient inference algorithm will be proposed in Chapter 9.

2.3.2 Dynamic Bayesian Network (DBN)

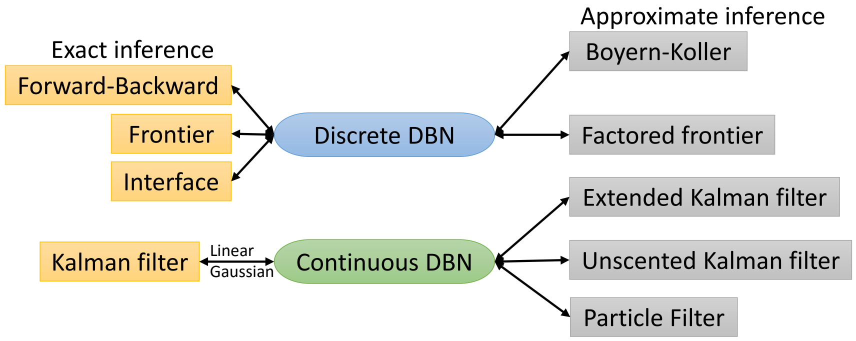


Figure 2.4 Class of inference algorithms for DBN

Exact and approximate inference algorithms for the DBN have been developed in the literature, as shown in Figure 2.4. For the DBN with discrete variables, exact inference is always possible and available algorithms include the forward-backwards algorithm [20] and the frontier algorithm [21], etc. As shown in Eq. (2.6), the inference at time t of the DBN is not related to earlier state variables and observations once the prior distribution of $\boldsymbol{X}^t$ is constructed, and the subsequent step is the inference for the BN at time t, which is static. Thus the exact inference algorithms for static BN can be extended to DBN. Murphy [22] proposed the interface algorithm by extending the junction tree algorithm to the inference of DBN with discrete variables. Approximate inference algorithms for the DBN with discrete variables have been developed to improve computational efficiency, including the loopy belief propagation algorithm , the Boyern-Koller algorithm [23], and the factored frontier algorithm [24].

The particle filter is a generic approximate algorithm for dynamic Bayesian networks. The particle filter is also named “survival of the fittest”, where a particle with higher weight (defined based on likelihood) is prone to be replicated and a particle with lower weight is prone to be dropped. The particle filter is applicable to both discrete and continuously DBNs, and has no limit on the DBN topology and CPD formats. The main concern on particle filter is computational cost. A DBN has more nodes requires more particles to cover the sampling space of the state variables, thus increases the computational cost. Details of particle filter will be introduced in Chapter 6, where it is used for uncertainty integration in time-dependent structural health diagnosis/prognosis.

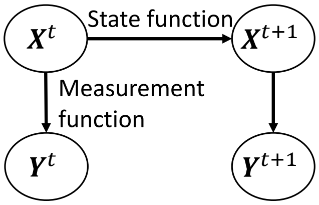


Figure 2.5 Underlying DBN of Kalman filter

In contrast to particle filter, Kalman filter and extended Kalman filter and unscented Kalman filter are analytical algorithms thus they are more efficient. Note that the three types of Kalman filters above are NOT proposed for DBN but for a dynamic system which can be depicted by the state function and measurement function. Kalman filter [25] gives exact inference for a *linear Gaussian* dynamic system, while the extended Kalman filter or unscented Kalman filter are designed when the state function and/or the measurement function are non-linear, still with Gaussian variables. But this dynamic system has an underlying DBN as shown in Figure 2.5. This DBN has two layers: Layer 1 is for state variables $\boldsymbol{X}^t$ and Layer 2 is for observation variables $\boldsymbol{Z}^t$. Theoretically, the three types of Kalman filters are applicable for any DBN if it has the topology in Figure 2.5 so that the CPDs from $\boldsymbol{X}^t$ to $\boldsymbol{Z}^t$ can be represented by a measurement function and the CPDs from $\boldsymbol{X}^{t-1}$ to $\boldsymbol{X}^t$ can be represented by a state function. The basic Kalman filter is adequate if both the state function and measurement function are linear and the noise terms are zero-mean Gaussian variables; otherwise the extended Kalman filter or unscented Kalman filter is required.

One contribution of this research is to extend the unscented Kalman filter to be an inference algorithm for Bayesian networks of more complex topology (more than two layers), as shown in

Chapter 9. A brief introduction to the basic Kalman filter and the unscented Kalman filter can be found in Section 9.1, and the proposed inference algorithm will be illustrated in Section 9.3.

2.4 Uncertainty Sources in System Response Prediction

In order to predict the response of a system, we usually describe the system by a computational model in the format of $Y = F(\boldsymbol{\theta}_m; \boldsymbol{X})$ where Y is the system response to be predicted; $\boldsymbol{\theta}_m$ is the vector of unknown model parameters; and $\boldsymbol{X}$ is the vector of model inputs. In the ideal case where the model perfectly represents the underlying physics and the values of $\boldsymbol{\theta}_m$ and $\boldsymbol{X}$ are known, the system response can be easily obtained by a functional evaluation of the computational model. However, various uncertainty sources arise in a real system, making the response prediction more complex. And these uncertainty sources can be categorized into irreducible aleatory uncertainty due to natural variability and reducible epistemic uncertainty due to lack of knowledge.

First, usually there is discrepancy between the model prediction by $F(\boldsymbol{\theta}_m; \boldsymbol{X})$ and the true physics, due to two types of errors [26,27]: 1) the numerical errors in solving the mathematical model (such as discretization, truncation and round-off errors); and 2) model form error. Often the estimates of these errors are also uncertain (epistemic); therefore the model output is uncertain. For example, if the mathematical model is a differential equation and $F(\boldsymbol{\theta}_m; \boldsymbol{X})$ solves it using numerical discretization (e.g., finite element, finite difference), then the discretization error $\epsilon_h(\boldsymbol{X})$ at a given model input is deterministic for a given value of the input [28]; however some implementations use Gaussian process (GP) models [29,30] to capture the uncertainty in estimating ϵ_h. The model discrepancy $\delta(\boldsymbol{X})$ is the difference between the computational model and the real system. The model error $\delta(\boldsymbol{X})$ can be modeled using different formulations [31], which introduces more parameters. Kennedy and O'Hagan [32] represent it by a GP model, so the model discrepancy is also stochastic at a given model input. In some studies the discrepancy term

$\delta(\boldsymbol{X})$ includes the numerical errors, and in some studies it refers only to model form error, after accounting for numerical errors that are estimated separately.

Second, to promote computational efficiency, often the computational model is replaced by a surrogate model $S(\boldsymbol{\theta}_m; \boldsymbol{X})$. This surrogate model brings additional uncertainty in the prediction, due to limited training points. Several options such as polynomial response surface [33], polynomial chaos expansion [34], Gaussian process (GP) model [30,35] etc. are available. This research uses the GP surrogate model [35]. The output of the GP model at a given input is a Gaussian distribution, which represents the surrogate model prediction uncertainty. Considering the discretization error, model form error, and surrogate model uncertainty, a general expression of the corrected system response prediction may be written as

$$Y = S(\boldsymbol{\theta}_m; \boldsymbol{X}) + \epsilon_h(\boldsymbol{X}) + \delta(\boldsymbol{X}) \tag{2.7}$$

where the prediction Y is stochastic due to the uncertainty in the three terms on the right hand side, even at a fixed value of $\boldsymbol{\theta}_m$ and $\boldsymbol{X}$.

In addition, extra uncertainty sources arise in characterizing $\boldsymbol{\theta}_m$ and $\boldsymbol{X}$. The model parameters $\boldsymbol{\theta}_m$ have fixed but unknown values, thus there is epistemic uncertainty (lack of knowledge) regarding $\boldsymbol{\theta}_m$.

If a model input X is a random variable, its natural variability can be represented by a probability distribution with distribution parameters $\boldsymbol{\theta}_X$. If only limited observations of X are available, there is uncertainty in the distribution type and distribution parameters. This uncertainty is also referred as statistical uncertainty [36] or second-order uncertainty [37]. Therefore, the uncertainty in model input X has two components: aleatory natural variability and epistemic uncertainty regarding distribution type and distribution parameters.

If a model input X is not a random variable but a time series, the prediction of system response Y requires the values of X over all time steps. Two types of time domain methods have been developed to model the time series input using observed data: 1) cycle counting methods, including the rainflow counting method [38] and the Markov chain method [39]; and 2) random process methods, such as the autoregressive moving average (ARMA) model [40]. This research chooses the ARMA model and therefore a brief introduction to ARMA is given in Section 3.2.

Table 2.1. Uncertainty sources in system response prediction

Uncertainty type	Symbol	Uncertainty source	Category
Solution approximation	$S(\boldsymbol{\theta}_m; \boldsymbol{X})$	Surrogate model	Epistemic
Solution approximation	$\epsilon_h(\boldsymbol{X})$	Discretization error	Epistemic
Model form error	$\delta(\boldsymbol{X})$	Model discrepancy	Epistemic
Model parameter	$\boldsymbol{\theta}_m$	Model parameter uncertainty	Epistemic
Random variable input	$\boldsymbol{\theta}_X$	Distribution parameter uncertainty	Epistemic
	X given $\boldsymbol{\theta}_X$	Input natural variability	Aleatory
Time series input by ARMA model	$\bar{X}, \boldsymbol{\phi}, \boldsymbol{\theta}, \sigma_\epsilon$	Model parameter uncertainty	Epistemic
	ϵ_t	Input natural variability	Aleatory

The uncertainty sources discussed above are listed in Table 2.1. The natural variability in the model input is aleatory; all other sources are epistemic.

2.5 Model Calibration and Model Validation

2.5.1 Model Calibration

Section 2.4 gave a generic formula for system response prediction in Eq. (2.7). The main challenge in using this formula for prediction is that the values of the model parameters $\boldsymbol{\theta}_m$ and other parameters are unknown, thus model calibration is needed. Model calibration aims to adjust these calibration parameters so that the agreement between model prediction and experimental data is maximized [41]. Techniques of model calibration includes least squared error, maximum likelihood estimation, maximum a posteriori, etc. This research uses Bayesian inference as the model calibration technique, as shown in the following brief introduction.

Model calibration requires experimental data. Consider the case of random variable input. Usually, the model input $\boldsymbol{X}$ and output Y can be observed in each test, thus forms the pairwise input-output data. Experimental data brings another uncertainty of measurement error. And the relationship between the experimental data Z and the corrected model prediction as:

$$Z = S(\boldsymbol{\theta}_m; \boldsymbol{X}) + \delta(\boldsymbol{X}) + \epsilon_m \tag{2.8}$$

where ϵ_m is the measurement error in the output observation, and ϵ_m is usually assumed to be Gaussian distribution $N(0, \sigma_m^2)$. In sum, all the parameters to calibrate include: 1) model parameters $\boldsymbol{\theta}_m$; 2) hyper-parameters $\boldsymbol{\theta}_\delta$ of the model error $\delta(\boldsymbol{x})$; and 4) standard deviation σ_m of ϵ_m.

Eq. (2.8) has an underlying Bayesian network, as shown in Figure 2.6. Here the state variables are $\{\sigma_m, \boldsymbol{\theta}_m, \boldsymbol{\theta}_\delta, S, \delta\}$ and the observation variables are $\{\boldsymbol{X}, \boldsymbol{Z}\}$. For the input-output pairwise data from a single experiment, the likelihood function can be constructed by Eq. (2.5), and the entire likelihood function is the product the likelihood function for each experiment. After assigning prior distributions to all the root nodes and implementing Bayesian inference algorithms, we can obtain the posterior distributions of all the state variables, but usually we are mainly interested in the posterior of $\{\sigma_m, \boldsymbol{\theta}_m, \boldsymbol{\theta}_\delta\}$ for future system response prediction.

Note that Figure 2.6 is a generic expression of the model calibration, but its topology may vary in a specific problem. For example, the numerical example in Section 4.6 assume that δ is an unknown constant to be calibrated, thus 1) the node $\delta(\boldsymbol{X})$ reduces to δ; and 2) $\boldsymbol{\theta}_\delta$ and the edge from $\boldsymbol{X}$ to $\delta(\boldsymbol{X})$ will be removed from the BN.

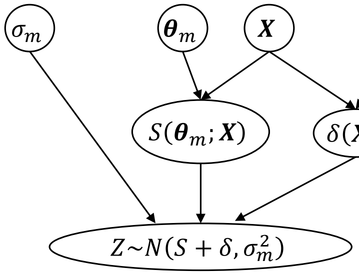

σ_m
$\boldsymbol{\theta}_m$
$\boldsymbol{X}$
$\boldsymbol{\theta}_\delta$
$S(\boldsymbol{\theta}_m;\boldsymbol{X})$
$\delta(\boldsymbol{X})$
$Z \sim N(S+\delta, \sigma_m^2)$

Figure 2.6 Bayesian network for calibration

Note that Eq. (2.8) and Figure 2.6 imply that the model input $\boldsymbol{X}$ is a vector random variables, which is NOT time dependent. If $\boldsymbol{X}$ represents time series input, then the output will be accumulative effects of the input across a period time. In this case, a time series model such as an autoregressive moving average (ARMA) model is needed to simulate the input, and the parameters of this time series model also need to be calibrated. Details for this case can be found in Chapter 3.

2.5.2 Model Validation

The term "model validation" has had different interpretations in different studies, and this research follows the AIAA definition [42], i.e., model validation is the process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model. Generally, model validation is realized by comparing the model prediction against experimental data. Both model calibration and model validation are conducted in this research, but they use different sets of experimental data (no calibration data is used in model validation). Comprehensive reviews on model validation can be found in [42–45]. A methodology for integrating model validation results from multiple experiments, each of which tests one part of the physics in the target application, can be found in Ref. [46].

Model calibration and model validation are distinct activities. Theoretically, for a computation model $F(\boldsymbol{\theta}_m;\boldsymbol{X})$ where $\boldsymbol{X}$ is a set of model inputs and $\boldsymbol{\theta}_m$ is a set of model parameters, model validation can be conducted exclusive of any model calibration [42] if the model parameters are assumed to be known. However, the model parameters $\boldsymbol{\theta}_m$ are often unknown. Therefore, prior to model validation, model calibration can be conducted to quantify the values of $\boldsymbol{\theta}_m$ or reduce the uncertainty about their values. Model calibration used in this research not only reduces the analyst's uncertainty about $\boldsymbol{\theta}_m$ by Bayesian inference, but also quantifies the model error $\delta(\boldsymbol{X})$ which is defined as the difference between model prediction and reality. For a new test input $\boldsymbol{X} = \boldsymbol{x}$, the corrected prediction model is $F(\boldsymbol{\theta}_m; \boldsymbol{x}) + \delta(\boldsymbol{x})$. Note that the prediction can be stochastic at fixed model inputs $\boldsymbol{X} = \boldsymbol{x}$ if the model parameters $\boldsymbol{\theta}_m$ are still uncertainty. In addition, uncertain model errors, surrogate model uncertainty are other reasons that the prediction can be stochastic at fixed model inputs. Compared to the original computational model, the new model is different in two aspects: 1) reduced uncertainty in $\boldsymbol{\theta}_m$; and 2) introduction of model error $\delta(\boldsymbol{X})$. In this research, the model to be assessed in model validation is this "corrected" model. Thus validation is a subsequent and distinct activity after calibration in this research. In other words, we consider model calibration and model validation as two distinct activities, and use two different sets of experimental data for these two activities, as suggested in Refs. [47,48]. Thus the calibration results of $\delta(\boldsymbol{X})$ and $\boldsymbol{\theta}_m$ do not change as a result of model validation in our approach.

Model validation is about comparing the model prediction against experimental data, and a model validation metric is needed to quantify this comparison. Among the validation metrics in the literature, classical hypothesis testing gives an acceptance/rejection decision. Confidence intervals have also been calculated for the difference between model prediction and observed data [42]. Validation metrics resulting in a single quantitative value indicating the degree of model validity have also been developed. In Bayesian hypothesis testing [47,49], the posterior distribution obtained by model calibration is used as the null hypothesis and an alternative distribution is selected for the alternative hypothesis. The result of Bayesian hypothesis testing is a Bayes factor (the likelihood ratio between the null and alternate hypotheses), measuring the support from validation data to the null and alternate hypotheses. This is a relative measure significantly depending on the choice of distribution of the alternate hypothesis. In contrast, Ferson et al. [50,51] proposed an area metric, which is the difference between CDFs and has the same unit as the prediction/data. For the case that the model output is stochastic at the fixed model input, this metric measures the area between the CDF of model output and the EDF (empirical distribution function) of experimental data at a fixed model input. If data are from experiments with different inputs, this metric is still applicable by building a single EDF for all the data with u-pooling method [50].

The model validation metric used in this research is the model reliability metric proposed by Rebba and Mahadevan [52] and further developed by Sankararaman and Mahadevan [53]. This metric measures the model validity by “model reliability”, which is defined as the probability that the difference between model prediction and observed data is less than a pre-defined tolerance. Details of this metric and its extensions will be illustrated in Sections 4.1 and 4.3.1.

2.6 Global Sensitivity Analysis: Sobol’ Index

Uncertainty propagation problems generally involve a deterministic function in the form of $Y = F(\boldsymbol{X})$ where $\boldsymbol{X} = \{X_1, \ldots, X_k\}$ is the vector of stochastic model inputs. Here the function is deterministic function if a give value of $\boldsymbol{X}$ results in a single value of Y. The computation model $F(\boldsymbol{\theta}_m; \boldsymbol{X})$ in Section 2.4 is also a deterministic function suitable for GSA.

Global sensitivity analysis (GSA) studies how the uncertainty in the output can be apportioned to the uncertainty in the stochastic model inputs. For the computational model $Y = F(\boldsymbol{\theta}_m; \boldsymbol{X})$ in Section 2.4, GSA is to quantify the contribution of each random variable in the model parameters

$\boldsymbol{\theta}_m$ and model inputs $\boldsymbol{X}$. In fact, GSA treats all the random variables in the same way, no matter this variable belongs to model inputs or model parameter. For the sake of notation convenience, this section does not distinguish model inputs and model parameters, but denotes $Y = F(\boldsymbol{X})$ as the generic function format for GSA where $\boldsymbol{X} = \{X_1, \dots, X_k\}$ is the vector of all the stochastic model inputs (also include stochastic model parameters).

GSA quantifies the contributions of the stochastic model inputs to the output variance so that their importance can be ranked. Based on the result of GSA, inputs with negligible contribution can be fixed at their mean values thus reducing the number of stochastic variables. Reviews on various GSA methods can be found in Refs. [54,55]. The Sobol' sensitivity indices method based on variance decomposition is a prominent one among these methods. Usage of the Sobol' indices in different engineering problems can be found in Refs. [56–60].

A brief introduction to the Sobol' index is given here. Assuming that $Y = F(\boldsymbol{X})$ is a real integrable function and all the model inputs $\boldsymbol{X} = \{X_1, \dots, X_k\}$ are mutually independent, Sobol' [61] proved the following formula to decompose the variance of Y:

$$V(Y) = \sum_{i}^{k} V_i + \sum\nolimits_{i_1=1}^{k} \sum\nolimits_{i_2=i_1+1}^{k} V_{i_1 i_2} + \sum\nolimits_{i_1=1}^{k} \sum\nolimits_{i_2=i_1+1}^{k} \sum\nolimits_{i_3=i_2+1}^{k} V_{i_1 i_2 i_3} + \cdots + V_{12\dots k} \qquad (2.9)$$

where V_i is the variance of Y caused by X_i individually, and $V_{i_1 \dots i_s} (s \geq 2)$ is the variance of y caused by the interaction of $\{X_{i_1}, \dots, X_{i_s}\}$.

Dividing $V(Y)$ at both sides of Eq. (2.9) for normalization, the Sobol' index is defined as:

$$1 = \sum_{i}^{k} S_i + \sum\nolimits_{i_1=1}^{k} \sum\nolimits_{i_2=i_1+1}^{k} S_{i_1 i_2} + \sum\nolimits_{i_1=1}^{k} \sum\nolimits_{i_2=i_1+1}^{k} \sum\nolimits_{i_3=i_2+1}^{k} S_{i_1 i_2 i_3} + \cdots + S_{12\dots k} \qquad (2.10)$$

where the index S_i measures the contribution of X_i alone to the variance of Y, without interacting with any other inputs. S_i is called first-order index or main effects index. Other indices $S_{i_1 \dots i_s} (s \geq 2)$ in Eq. (2.10) are higher-order indices, measuring the contribution of the interaction of $\{X_{i_1}, \dots, X_{i_s}\}$.

The calculation of S_i is based on the following formula:

$$S_i = \frac{V_i}{V(Y)} = \frac{V_{X_i}\left(E_{\boldsymbol{X}_{-i}}(Y|X_i)\right)}{V(Y)} \tag{2.11}$$

where $\boldsymbol{X}_{-i}$ means all the model inputs other than X_i.

Another index is the total effects index S_i^T, which overall contribution of X_i by itself plus interactions with other inputs. This total effects index is defined as the sum of all the indices in Eq. (2.10) related X_i. For example, if $k = 3$ so that Eq. (2.10) reduces to $1 = S_1 + S_2 + S_3 + S_{12} + S_{13} + S_{23} + S_{123}$, the total effects index of X_1 will be:

$$S_1^T = S_1 + S_{12} + S_{13} + S_{123} \tag{2.12}$$

Eq. (2.12) implies that we need to calculate multiple indices to obtain the total effects index, but it is not necessary. Similar to Eq. (2.11), the calculation of S_i^T is based on the following formula:

$$S_i^T = 1 - \frac{V\big(E(Y|\boldsymbol{X}_{-i})\big)}{V(Y)} \tag{2.13}$$

Eq. (2.11) and Eq. (2.13) can be extended to assess the contribution of a model input subset $\boldsymbol{X_p}$ which contains more than one input [62,63]. The main Sobol' index of $\boldsymbol{X_p}$ is defined by extending Eq. (2.11) as

$$S_{\boldsymbol{X_P}} = \frac{V\left(E\left(Y\middle|\boldsymbol{X_p}\right)\right)}{V(Y)} \tag{2.14}$$

$S_{\boldsymbol{X_p}}$ is a combined measure of the individual contributions of the components of $\boldsymbol{X_p}$ and of the interactions among them.

And the total effects Sobol' index of $\boldsymbol{X_p}$ is defined by extending Eq. (2.13) as

$$S^T_{\boldsymbol{X_P}} = 1 - \frac{V\left(E\left(Y\middle|\boldsymbol{X_{-p}}\right)\right)}{V(Y)} \tag{2.15}$$

where $\boldsymbol{X_{-p}}$ is the complementary subset of $\boldsymbol{X_p}$. $S^T_{\boldsymbol{X_P}}$ is a combined measure of the individual contributions of the components of $\boldsymbol{X_p}$, the interactions among them, and the interactions between $\boldsymbol{X_p}$ and $\boldsymbol{X_{-p}}$.

A key assumption of the Sobol' index is the mutual independence of model inputs. With correlated model inputs, Eqs. (2.9) and (2.10) are no longer valid. However, Saltelli [64] pointed out that the first-order index S_i is still an informed choice to rank the importance of correlated model inputs, since S_i can be defined in another way where independent model inputs are not assumed:

1. The importance of X_i at a particular location $\tilde{X}_i$ can be measured by $V_{\boldsymbol{X}_{-i}}\left(Y\middle|X_i = \tilde{X}_i\right)$, i.e., smaller $V_{\boldsymbol{X}_{-i}}\left(Y\middle|X_i = \tilde{X}_i\right)$ indicates greater importance of X_i;
2. The dependence of this measurement on the location of X_i is removed by taking the average of $V_{\boldsymbol{X}_{-i}}\left(Y\middle|X_i = \tilde{X}_i\right)$, i.e. $E_{X_i}\left(V_{\boldsymbol{X}_{-i}}(Y|X_i)\right)$;
3. By the law of total variance $V(Y) = E_{X_i}(V_{\boldsymbol{X}_{-i}}(Y|X_i)) + V_{X_i}(E_{\boldsymbol{X}_{-i}}(Y|x_i))$, a larger $V_{X_i}(E_{\boldsymbol{X}_{-i}}(Y|X_i))$ equally indicates a greater importance of X_i;

4. The first-order index is redefined by normalization, thus $S_i = V_{X_i}(E_{\boldsymbol{X}_{-i}}(Y|X_i))/V(Y)$.

In sum, we can use the first-order Sobol' index S_i whether the model inputs are correlated or not. In comparison, other higher order indices in Eq. (2.10) and the total effects index S_i^T are meaningless if the model inputs are correlated, since their derivations requires uncorrelated model inputs. In this research, S_i^T is utilized if the model inputs are uncorrelated, since it is a more comprehensive index considering the interaction between different inputs; and S_i is used if the model inputs are correlated.

In addition, it should be kept in mind that the Sobol' index requires the function $Y = F(\boldsymbol{X})$ to be a deterministic function, which means that a single realization of $\boldsymbol{X}$ gives a corresponding single realization of Y. This research emphasize the term "deterministic function" to contrast from "stochastic" functions such as Eq. (2.7) , where the function output is uncertain (i.e., it has many possible realizations) even if all the inputs are fixed. One objective of this research is to extend the usage of Sobol' index to stochastic functions, and the auxiliary variable method is required for this purpose. A brief introduction of the auxiliary variable method will be given in Section 2.7; and the proposed method of GSA for stochastic function of aleatory and epistemic uncertainty, considering both random variable input and time series input, can be found in Chapter 3.

Another key question in computing the Sobol' index is the computational cost. Direct calculation of S_i and S_T^i based on Eqs. (2.11) and (2.13) is quite expensive since a double-loop Monte Carlo simulation (MSC). For S_i in Eq. (2.11), the inner loop $E_{\boldsymbol{X}_{-i}}(Y|X_i)$ computes the mean value of Y using n_1 random samples of $\boldsymbol{X}_{-i}$; and the outer loop computes $V_{X_i}(E_{\boldsymbol{X}_{-i}}(Y|X_i))$ by iterating the inner loop n_2 times at different values of X_i. In addition, another n_3 MCS iterations are required to compute $V(Y)$ in Eq. (2.11). The cost of double-loop MCS, defined as the total number of model evaluations to compute all S_i ($i = 1$ to k), is

$$\text{Cost} = kn_{dl}^2 + n_{dl} \tag{2.16}$$

where we assume $n_1 = n_2 = n_3 = n_{dl}$.This cost increases with n_{dl} and k, and is unaffordable if a single model evaluation is time-consuming or economically expensive, since n_{dl} is often of the order greater than 1000 in many practical applications. The double-loop simulation for S_i^T is similar and also expensive.

Various algorithms have been proposed to reduce the computational cost, and one objective of this research is to propose a new efficient algorithm. A literature review on existing algorithms will be given in Section 7.2, and the proposed algorithm to compute S_i can be found in Section 7.3.

2.7 Auxiliary Variable Method

The auxiliary variable method was developed by Sankararaman and Mahadevan [65] to distinguish the contributions of aleatory natural variability and epistemic distribution parameter uncertainty in a random variable X. The distribution of X is conditioned on the value of its distribution parameter $\boldsymbol{\theta}_X$, which has uncertainty represented by a probability density $p(\boldsymbol{\theta}_X)$. This parameters distribution $p(\boldsymbol{\theta}_X)$ is also referred as second-order probability. The conditional distribution of X is denoted as $p(X|\boldsymbol{\theta}_X)$. This conditional distribution $p(X|\boldsymbol{\theta}_X)$ and the second-order probability $p(\boldsymbol{\theta}_X)$ actually constitutes a hierarchical Bayesian model. With different realizations of $\boldsymbol{\theta}_X$, $p(X|\boldsymbol{\theta}_X)$ constitutes a family of distributions, as shown in Figure 2.7. Each single distribution represents the natural variability of X at a particular realization of $\boldsymbol{\theta}_X$, and the spread of the distributions indicates the contribution of uncertainty in $\boldsymbol{\theta}_X$. This family of distribution only gives a qualitative representation of aleatory and epistemic uncertainties; a method of quantitative contribution assessment is still required.

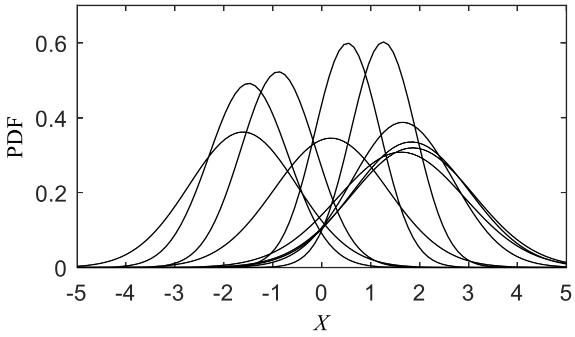


Figure 2.7 Family of PDFs

Based on the probability integral transform theorem [66], random sampling from the conditional distribution $p(X|\boldsymbol{\theta}_X)$ is realized in two steps: 1) define a variable U_X of standard uniform distribution $U(0,1)$ and generate its sample u_X, which is taken as the CDF value of X, and 2) obtain a sample x of X by the inverse conditional CDF $\mathcal{P}^{-1}(U_X|\boldsymbol{\theta}_X)$, i.e.,

$$x = \mathcal{P}^{-1}(U_X = u_X|\boldsymbol{\theta}_X) \tag{2.17}$$

The same procedure is repeated for other realizations of $\boldsymbol{\theta}_X$. Note that the distribution of U_X is independent of the realization of $\boldsymbol{\theta}_X$. At a given value of $\boldsymbol{\theta}_X$, the sample of U_X and the sample of X have a one-to-one mapping, i.e., a single value of X is determined once the value of U_X is decided. Thus the natural variability in X is represented by U_X.

This standard uniform random variable U_X, which is the CDF value of $p(X|\boldsymbol{\theta}_X)$, is named as the auxiliary variable. With U_X, Eq. (2.17) helps to build a deterministic input-output function $X = F(U_X, \boldsymbol{\theta}_X)$ for computing Sobol' indices, since a sample of $\boldsymbol{\theta}_X$ and a sample of U_X lead to a deterministic value of X based on Eq. (2.17). Then the resultant Sobol' index of $\boldsymbol{\theta}_X$ assesses the contribution of epistemic distribution parameter uncertainty, and the Sobol' index of U_X assesses the contribution of the natural variability of X.

Although the auxiliary variable approach is a standard procedure in sampling random variables, generally it is used implicitly and only the resultant samples of the random variables are recorded and utilized. However, as explained above, Ref. [65] found that if we use this auxiliary variable explicitly, it brings the benefit of separating the aleatory and epistemic uncertainty in a single random variable X and quantifying their contributions to the overall uncertainty in X. Ref. [65] only considered the aleatory and epistemic in the random variable model input, whereas this research extends the usage of the auxiliary variable to several topics: 1) assess the relative contributions of aleatory and epistemic uncertainty sources in time series prediction, as illustrated in Chapter 3; 2) global sensitivity analysis for Bayesian network, as illustrated in Chapter 5; and 3) development of an efficient Bayesian inference algorithm, as illustrated in Chapter 7.

CHAPTER 3

GLOBAL SENSITIVITY ANALYSIS UNDER ALEATORY AND EPISTEMIC UNCERTAINTY

3.1 Background

In many practical engineering systems, direct measurement of the system response under actual usage conditions is often not available; instead, a model is used to predict the response, in order to facilitate decisions related to design, risk management etc. In this case, the uncertainty in system response prediction is affected by various uncertainty sources. The importance of each uncertainty source can be measured by its contribution to the uncertainty in the system response prediction. Such information is useful in several ways, especially in problem dimension reduction (by ignoring the insignificant uncertainty sources) and in resource allocation for uncertainty reduction (by focusing additional data collection or model refinement efforts on significant uncertainty sources).

As introduced in Section 2.6, global sensitivity analysis (GSA) [54] provides a quantitative assessment of the relative contribution of model inputs towards the uncertainty in the model output. GSA methods can be either data-driven (e.g., based on analysis of variance ANOVA), or model-based, such as the computation of Sobol' indices [61] . In model-based prediction as shown in Figure 3.1, the computation of Sobol' indices is well-established for aleatory inputs [62,67,68], but their computation considering both aleatory and epistemic uncertainty sources (in model inputs and in model prediction) is not well-established [1–3]. Thus this chapter focuses on developing a framework for computing the Sobol' indices considering both aleatory and epistemic uncertainty sources when considering uncertainty propagation through a computational model.

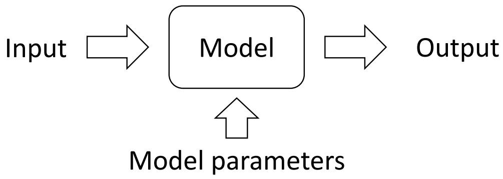


Figure 3.1 Model-Based prediction

Related to Figure 3.1, there is uncertainty in the model inputs, and in the model output even for a fixed input. A model input may be deterministic or random, and epistemic uncertainty can be present in both, due to inadequate data. In case of a deterministic input, its value may be unknown; in the case of a random input, its distribution type and/or distribution parameters may be unknown. The latter case is a mixture of aleatory and epistemic uncertainty. When the input is propagated through the computational model to compute the output, epistemic uncertainty sources in the model (uncertain model parameters, numerical approximations in the model, and model form assumptions) contribute to additional uncertainty in the model prediction. The objective of this section to quantify the contributions of various aleatory and epistemic uncertainty sources in the input and the model to the uncertainty in the model output.

The proposed framework for realizing this objective is shown in Figure 3.2. Due to inadequate data, the aleatory model inputs (either random variables or random processes) are mixed with epistemic uncertainty. Due to model uncertainty sources, the model output is uncertain even for a fixed input. When using an input-output model to compute the Sobol' indices, a deterministic input-output relationship, i.e., a one-to-one mapping, is needed. Therefore, a methodology is proposed in this research by introducing auxiliary variables based on the probability integral transform (explained in Section 2.7) to establish such a deterministic input-output relationship, and to separate the aleatory and epistemic uncertainty sources when the two are mixed. This strategy helps to calculate the Sobol' indices separately for both aleatory and epistemic uncertainty sources.

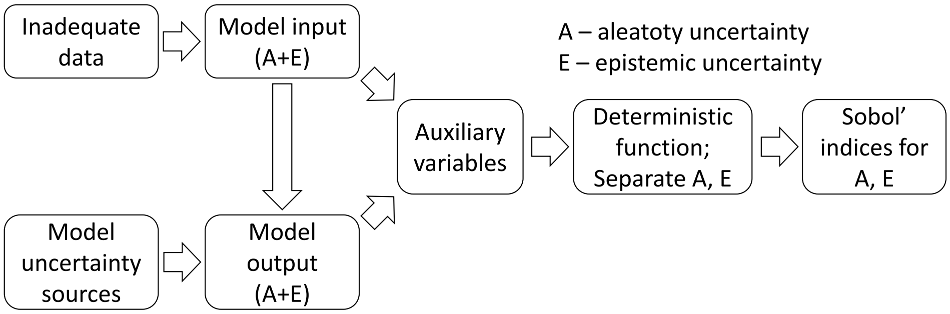


Figure 3.2 Proposed framework for Sobol' indices computation under aleatory and epistemic uncertainty

A particular problem of interest in this research is when the input to the computational model is a time series, such as the loading history on a mechanical component causing fatigue damage. Several options are available for modeling the time series input; this research uses the Autoregressive Moving Average (ARMA) approach, which is able to explicitly quantify the aleatory and epistemic uncertainty components in the time series input through the use of Bayesian calibration. The ARMA model and Bayesian calibration are described in Section 3.2. Sensitivity computation in the presence of time series input brings a significant challenge regarding computational effort, especially due to the introduction of a large number of noise terms (one in each time step). Therefore this research proposes a novel technique, based on the concept of pseudo-random number generation, to significantly improve the computational efficiency in calculating the Sobol' indices in the presence of time series input that has both aleatory and epistemic uncertainty.

In summary, this section makes three new important contributions to model-based sensitivity analysis: 1) computation of Sobol' indices in the presence of both input uncertainty (aleatory and epistemic) and model uncertainty (epistemic); 2) a novel technique to separate the aleatory and epistemic uncertainty sources in time series input; and 3) a novel computational technique (based on pseudo-random number generation) for efficient computation of Sobol' indices in the presence of time series input.

3.2 Autoregressive Moving Average (ARMA) Model

This section focuses on model-based GSA with time series input. The ARMA model is selected to model the time series input due to its ability to capture both natural variability and epistemic uncertainty in the time series input. An ARMA(p, q) model assumes that the input at time step t is a linear combination of 1) earlier input values from step $t-p$ to step $t-1$; 2) earlier values of noise from step $t-q$ to step $t-1$; and 3) the current value of noise at step t, i.e.,

$$X^t = \bar{X} + \sum_{i_a=1}^{p} \phi_{i_a} X^{t-i_a} + \epsilon^t + \sum_{i_m=1}^{q} \theta_{i_m} \epsilon^{t-i_m} \tag{3.1}$$

where X^t and X^{t-i_a} are the inputs at time step t and time step $t-i_a$; $\boldsymbol{\phi} = \{\phi_1, \dots, \phi_p\}$ are the coefficients of the AR model; $\boldsymbol{\theta} = \{\theta_1, \dots, \theta_q\}$ are the coefficients of the MA model; $\bar{X}$ is a constant; and ϵ^t and ϵ^{t-i_m} are the random noise terms at time step t and time step $t-i_m$. All the random noise terms are generally assumed to be independent and identically distributed Gaussian variables $N(0, \sigma_\epsilon^2)$, i.e., Gaussian white noise [69]. And these noise terms represent the natural variability of the time series input.

To build an ARMA model, the values of its orders p and q are first identified by matching the theoretical autocorrelation function to the sample autocorrelation function computed from the observed time series data. The Ljung-Box Q statistic [70] can be used to measure the adequacy of the matching.

The values of the ARMA parameters $\{\bar{X}, \boldsymbol{\phi}, \boldsymbol{\theta}, \sigma_\epsilon\}$ have epistemic uncertainty due to limited history data. The ARMA model can capture this epistemic uncertainty by assigning probability distributions to the ARMA parameters $\{\bar{X}, \boldsymbol{\phi}, \boldsymbol{\theta}, \sigma_\epsilon\}$. Bayesian inference may be used to calibrate the distributions of the ARMA parameters using the observed data [69]. In contrast, the counting matrix in the cycle counting methods is deterministic so that the epistemic uncertainty due to limited time series data is difficult to quantify.

3.3 GSA under Both Aleatory and Epistemic Uncertainty

The uncertainty sources in system response prediction have been listed in Table 2.1. Theoretically, GSA based on Sobol' indices can be used to assess the contribution of any uncertainty source, no matter whether it is aleatory or epistemic. However, the existence of both aleatory and epistemic uncertainties in Table 2.1 brings two challenges to computing the Sobol' indices using an input-output prediction model. First, the model prediction $Y = S(\boldsymbol{\theta}_m; \boldsymbol{X}) + \epsilon_h(\boldsymbol{X}) + \delta(\boldsymbol{X})$ is not deterministic, i.e., Y does not have a single deterministic value even if $\boldsymbol{\theta}_m$ and $\boldsymbol{X}$ are fixed. The reason is that $S(\boldsymbol{\theta}_m; \boldsymbol{X}), \epsilon_h(\boldsymbol{X})$ and $\delta(\boldsymbol{X})$ can each be uncertain even for fixed values of $\boldsymbol{X}$ and $\boldsymbol{\theta}_m$. In this research, since the GP surrogate model is used, the surrogate model prediction $S(\boldsymbol{\theta}_m; \boldsymbol{X})$ is a Gaussian random variable for fixed values of $\boldsymbol{X}$ and $\boldsymbol{\theta}_m$; the discretization error $\epsilon_h(\boldsymbol{X})$ and model form error $\delta(\boldsymbol{X})$ are also estimated by GP models, thus they are both Gaussian random variables for a fixed value of $\boldsymbol{X}$. Therefore Y is the sum of three Gaussian random variables.

Second, each uncertainty source in Table 2.1 should be represented by a single random variable of known or fixed probabilistic distribution if we want to compute the Sobol' indices. However, this required single random variable is not available for some uncertainty sources. The main reason is that one uncertainty source may depend on another one. For example, the uncertainty in the discretization error $\epsilon_h(\boldsymbol{X})$ depends on the value of $\boldsymbol{X}$. In this case, the distribution of $\epsilon_h(\boldsymbol{X})$ is not fixed but changes with the value of $\boldsymbol{X}$. The first contribution of this research is to use the auxiliary variable to decouple the dependent uncertainty sources, so that the uncertainty term that depends on other uncertainty sources can be separately represented by a single auxiliary variable of fixed uniform distribution $U(0,1)$, and the deterministic function required for the Sobol' indices computation can be established.

Identifying the single variable to represent the natural variability in the ARMA model is even more difficult. At given values of ARMA parameters, if we run the ARMA model N times, N different time series histories can be obtained. The variation among these histories represents the natural variability in the ARMA model (last row in Table 2.1), which is caused by the noise terms $\{\epsilon^1, \epsilon^2, \dots, \epsilon^N\}$ in the ARMA model at each time step. Although we can consider all the noise terms in the GSA, this will make the GSA extremely high-dimensional. Thus the second contribution of this research is a new method defining a single auxiliary variable that captures all the noise terms, i.e., the natural variability in the ARMA model; this method is described in Section 3.3.2, and referred to as uncontrolled-seed method.

Although the proposed uncontrolled-seed method reduces the dimension of the GSA, its computational efficiency is still not satisfying. Therefore the third contribution of this research is a new controlled-seed method proposed in Section 3.3.3, which uses the seed as a single random variable capturing the natural variability in the ARMA model. This method obtains the same result as the uncontrolled-seed method and reduces computational cost significantly.

3.3.1 GSA for Random Variable Input

The auxiliary variable method introduced in Section 2.7 can be extended to any variable whose distribution is conditioned on other variables. Assume that the distribution of a random variable A depends on the value of another random value B by a conditional distribution $p(A|B)$. Then the uncertainty in $p(A|B)$ can be captured by a single auxiliary variable U_A, which is the CDF value of $p(A|B)$. In other words, the auxiliary variable can be used to represent any uncertainty term whose distribution depends on other uncertainty sources. The represented uncertainty term can be either aleatory or epistemic.

Assume that the model inputs $\boldsymbol{X} = \{X_1, \dots, X_k\}$ are random variables. For the random variables $S(\boldsymbol{\theta}_m; \boldsymbol{X}), \epsilon_h(\boldsymbol{X})$ and $\delta(\boldsymbol{X})$ in Table 2.1 whose distribution is conditioned on the value of $\boldsymbol{X}$ and $\boldsymbol{\theta}_m$, auxiliary variables U_S, U_{ε_h} and U_δ can be introduced to represent the uncertainties due to surrogate model, discretization error, and model discrepancy respectively at fixed values of $\boldsymbol{X}$ and $\boldsymbol{\theta}_m$. In addition, auxiliary variables $\boldsymbol{U_X} = \left\{U_{X_1}, U_{X_2}, \dots, U_{X_k}\right\}$ are also introduced for each model input $X_j (j = 1 \text{ to } k)$ that has both aleatory and epistemic uncertainty. Then a deterministic function suitable for Sobol' indices computation can be built as:

$$Y = F\left(\boldsymbol{\theta}_m, \boldsymbol{\theta}_X, \boldsymbol{U_X}, U_S, U_{\varepsilon_h}, U_\delta\right) \tag{3.2}$$

Note that no auxiliary variable is needed for $\boldsymbol{\theta}_X$ or $\boldsymbol{\theta}_m$ since their distributions are not conditioned on any other variables. Another observation is that either aleatory or epistemic uncertainty can be represented by the auxiliary variables depending on the situation. For example, $\boldsymbol{U_X}$ represents the aleatory uncertainty in model inputs; whereas U_S, U_{ε_h}, and U_δ represent the epistemic uncertainties caused by surrogate model uncertainty, discretization error, and model form error respectively.

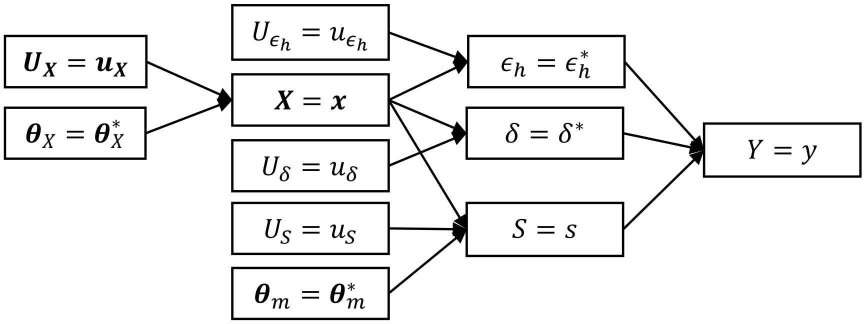

$U_{\epsilon_h} = u_{\epsilon_h}$
$\boldsymbol{U_X} = \boldsymbol{u_X}$
$\boldsymbol{X} = \boldsymbol{x}$
$\boldsymbol{\theta}_X = \boldsymbol{\theta}_X^*$
$U_\delta = u_\delta$
$U_S = u_S$
$\boldsymbol{\theta}_m = \boldsymbol{\theta}_m^*$
$\epsilon_h = \epsilon_h^*$
$\delta = \delta^*$
$S = s$
$Y = y$

Figure 3.3 Deterministic function for random variable input

The flowchart in Figure 3.3 illustrates the application of Eq. (3.2). A sample of the distribution parameters $\boldsymbol{\theta}_X$ gives the marginal distribution for each model input X, and auxiliary variables $\boldsymbol{U}_{\boldsymbol{X}} = \left\{U_{X_1}, U_{X_2}, \dots, U_{X_k}\right\}$ helps to generate a deterministic sample of $\boldsymbol{X}$ by CDF inversion on the joint distribution of model inputs $\boldsymbol{X}$. Note that the model inputs $\boldsymbol{X}$ discussed in this section is a set of scalar random variables. The case that $\boldsymbol{X}$ represents a time series input will be discussed in Section 3.3.2 and Section 3.3.3.

The sample of $\boldsymbol{X}$ decides the distribution of input-dependent discretization error ϵ_h and model form error δ, and the corresponding auxiliary variables U_{ϵ_h} and U_δ generate deterministic values of ϵ_h and δ respectively by inverting the corresponding CDFs. Similarly, the value of S is determined by the value of $\boldsymbol{X}$, $\boldsymbol{\theta}_m$, and auxiliary variable U_S. Finally a deterministic prediction is computed as $y = s + \epsilon_h + \delta$. The deterministic function in Eq. (3.2) is now ready for Sobol' indices computation. The resultant sensitivity indices of $\boldsymbol{\theta}_X$ assess the contributions of input distribution parameter uncertainty towards the uncertainty in model prediction Y; the indices of $\boldsymbol{\theta}_m$ assess the contributions of model parameter uncertainty; and the indices of auxiliary variables assess the contributions of the corresponding uncertainty sources, as shown in Table 2.1.

Note that Eq. (3.2) proposes a framework to assess the contribution of each uncertainty source with random variable inputs. If any uncertainty source is ignored in practice, this framework is still applicable by removing the corresponding variable in Eq. (3.2). For instance, if Richardson extrapolation is used to compute a deterministic discretization error and ignore the uncertainty in it, the auxiliary variable U_{ϵ_h} is not needed in Eq. (3.2). Similarly, if an input random variable X_j has only aleatory uncertainty and no epistemic uncertainty (i.e., its probability distribution is precisely known), then the corresponding auxiliary variable U_{X_j} is not needed; in this case, the probability density $p(X_j)$ represents the uncertainty (variability) in X_j.

3.3.2 GSA for Time Series Input

As discussed earlier, the epistemic uncertainty in the ARMA model of the times series input can be represented by assigning probability distributions to its parameters $\{\bar{X}, \boldsymbol{\phi}, \boldsymbol{\theta}, \sigma_\epsilon\}$ and updating these distributions using Bayesian inference. Like other random process representations, the ARMA model takes the input at each time step as a random variable X^t and the observed value at this time step is a realization of this random variable. Theoretically, this time series can be considered as a N_t-dimensional vector of random variables $\boldsymbol{X} = \{X^1, \dots, X^{N_t}\}$ where N_t is the number of time steps, so the flowchart in Figure 3.3 is still applicable. However, since N_t is usually very large, several studies have tried to reduce this N_t-dimensional time series input to a low-dimensional representation.

Ben-Haim [71] employed a deterministic convex model of Fourier series rather than probabilistic models to represent the uncertainty in a load history. However, this deterministic model ignores the aleatory uncertainty in the time series input, even if the epistemic uncertainty can be introduced into this model by allowing the Fourier coefficients to vary. Echard et al. [72] used nine displacement histories to represent the uncertainty of in-service loads. This method needs adequate observations of time series input, which may be impossible.

Another option to reduce the dimension of a random process is the Karhunen-Loeve expansion [73,74], which represents a random process by the eigenvalues and eigenfunctions of the covariance function. The first l largest eigenvalues and the corresponding engenfunctions are retained if the explained variance of the random process reaches a threshold such as 95% or 99%.

The explained variance is given by $\sum_{i_e=1}^{l} \lambda_{i_e} / \sum_{i_e=1}^{\infty} \lambda_{i_e}$, where λ_{i_e} is the i_e-th largest eigenvalue [74]. However, the value of l highly depends on the autocorrelation function of the random process: more eigenvalues and eigenfunctions are needed to explain the same variance if the autocorrelation function decays faster. Consider a random process represented by an ARMA(1, 1) model $X^t = -2 + 0.2X^{t-1} + \epsilon^t + 0.2\epsilon^{t-1}$ where the noise terms have a Gaussian distribution $N(0, 0.1^2)$. Figure 3.4 shows the autocorrelation function and the first 50 eigenvalues. The autocorrelation function decays to almost zero after 3 lags. No dominant eigenvalue is observed, therefore most eigenvalues should be retained to explain the variance. Thus the dimension of the random process cannot be significantly reduced in some cases.

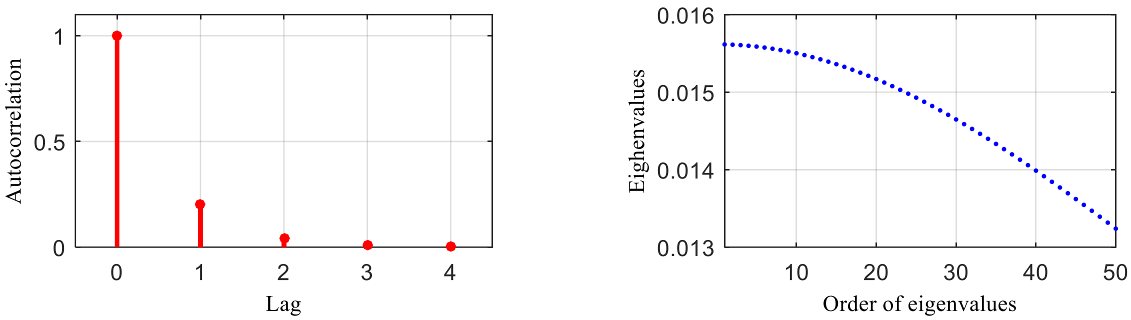


Figure 3.4 Autocorrelation and eigenvalues for ARMA model

The objective of this research is not only to make the Sobol' indices computation affordable but also to distinguish the contributions of aleatory and epistemic uncertainties towards the uncertainty in the prediction. Here the auxiliary variable method is extended to assess the individual contribution of each uncertainty source. The deterministic function required for the Sobol' indices computation is:

$$Y = F\left(\bar{X}, \boldsymbol{\phi}, \boldsymbol{\theta}, \sigma_\epsilon, \boldsymbol{\theta}_m, U_S, U_{\epsilon_h}, U_\delta, U_\epsilon\right) \tag{3.3}$$

An evaluation of Eq. (3.3) is shown in Figure 3.5, which can be realized in 7 steps:

1. Generate a sample of the ARMA model parameters $\bar{X}, \boldsymbol{\phi}, \boldsymbol{\theta}, \sigma_\epsilon$ from their joint distribution. This joint distribution represents the epistemic uncertainty regarding the ARMA model parameters, and can be obtained by Bayesian inference using observed time series data.
2. Generate a sample $\boldsymbol{\theta}_m^*$ of the physics model parameters $\boldsymbol{\theta}_m$.
3. Generate N time histories $\{\boldsymbol{\chi}_1, \dots, \boldsymbol{\chi}_N\}$ based on the samples of $\bar{X}, \boldsymbol{\phi}, \boldsymbol{\theta}, \sigma_\epsilon$ from Step 1. Here the model input $\boldsymbol{X} = \{X^1, \dots, X^{N_t}\}$ is a time series input of N_t time steps. A generated history $\boldsymbol{\chi}_i (i = 1, \dots, N)$ is a realization of $\boldsymbol{X}$, thus $\boldsymbol{\chi}_i$ is a vector of N_t elements. The difference between these time histories represents the natural variability in the ARMA model caused by the noise terms. By propagating each time history with the sample of $\boldsymbol{\theta}_m^*$ through the stochastic surrogate model $S(\boldsymbol{\theta}_m; \boldsymbol{X})$, a family of N distributions can be constructed. Each distribution $S_i(\boldsymbol{\theta}_m^*, \boldsymbol{\chi}_i)$ $(i = 1, \dots, N)$ represents the effect of epistemic surrogate model uncertainty at a given time history, thus an auxiliary variable U_s is introduced to represent it.
4. Generate a sample of U_s to conduct CDF inversion of each distribution $S_i(\boldsymbol{\theta}_m^*, \boldsymbol{\chi}_i)$ $(i = 1, \dots, N)$in Step 3. The resultant N samples $\{s_1, \dots, s_N\}$ from the N distributions constitute a new random variable S whose uncertainty is caused by the ARMA model natural variability.
5. If the discretization error $\epsilon_h(\boldsymbol{X})$ is stochastic (e.g., due to the use of a GP model) at a given time series input, each time history from Step 3 gives a distribution of discretization error, thus a family of N distributions $\epsilon_{h_i}(\boldsymbol{\chi}_i)$ $(i = 1 \text{ to } N)$ can be constructed. An auxiliary variable U_{ϵ_h} representing the discretization error uncertainty is introduced to obtain a sample from each distribution, and the resultant N samples construct a random variable ϵ_h whose uncertainty is caused by the ARMA model natural variability.

6. Use the same procedure as Step 5 for $\delta(\boldsymbol{X})$: each time history from Step 3 gives a distribution of model discrepancy, thus a family of N distributions $\delta_i(\boldsymbol{\chi}_i)$ ($i = 1$ to N) can be constructed. An auxiliary variable U_δ is introduced to obtain a sample from each distribution, and the resultant N samples construct a random variable δ whose uncertainty is caused by the ARMA model natural variability.
7. Define a new variable Y_ϵ as the sum of S, δ, and ϵ_h from steps 3 to 6. The uncertainty in S, δ and ϵ_h is caused by the natural variability in the ARMA model, thus the uncertainty in Y_ϵ is also caused by natural variability in the ARMA model. Another auxiliary variable U_ϵ is introduced to represent the uncertainty in Y_ϵ. (Note that S, δ, and ϵ_h can be correlated, and the calculation in Figure 3.5 correctly accounts for this correlation by generating correlated samples of S, δ, and ϵ_h). With N samples of S from step 4, N samples of δ from step 5, and N samples of ϵ_h from step 6, we can obtain N samples of Y_ϵ to represent its distribution. A sample of U_ϵ is generated to conduct CDF inversion on Y_ϵ to obtain a deterministic value y so that a deterministic function can be established.

Note that Eq. (3.3) is as flexible as Eq. (3.2). The corresponding variable in Eq. (3.3) can be removed if any uncertainty source is ignored.

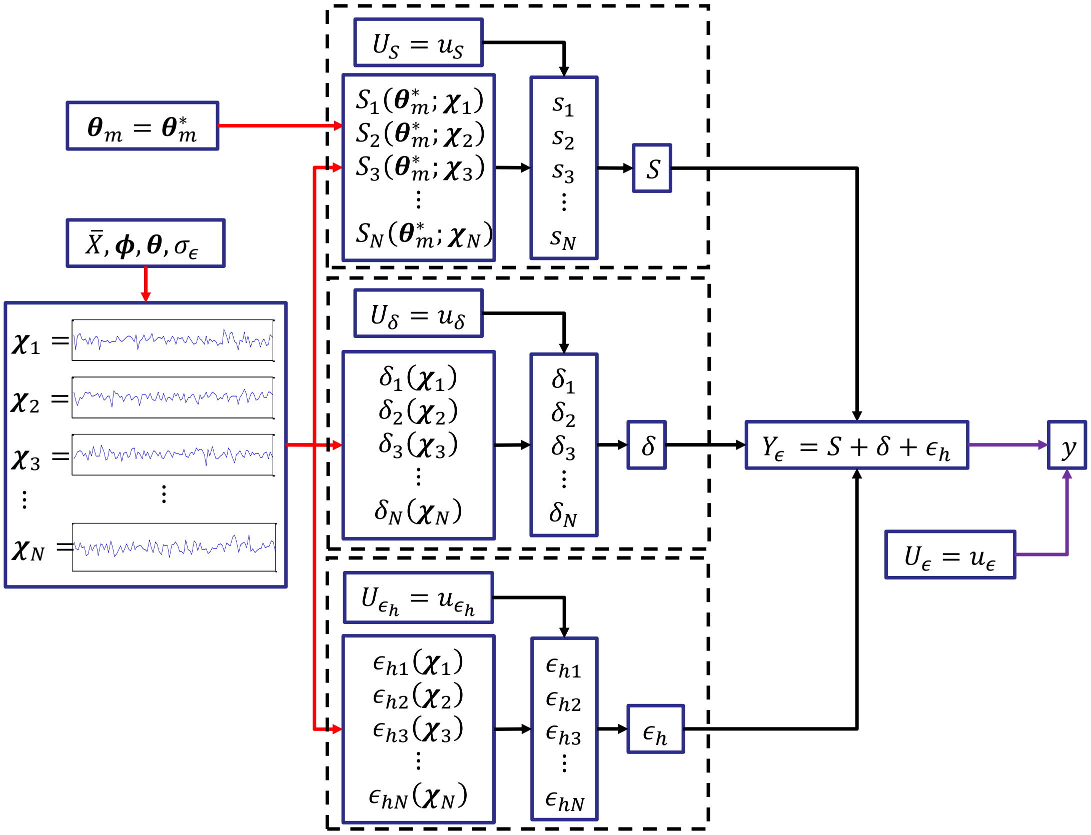

$U_S = u_S$
$\boldsymbol{\theta}_m = \boldsymbol{\theta}_m^*$
$S_1(\boldsymbol{\theta}_m^*; \boldsymbol{\chi}_1)$
$S_2(\boldsymbol{\theta}_m^*; \boldsymbol{\chi}_2)$
$S_3(\boldsymbol{\theta}_m^*; \boldsymbol{\chi}_3)$
$S_N(\boldsymbol{\theta}_m^*; \boldsymbol{\chi}_N)$
s_1
s_2
s_3
s_N
S
$\bar{X}, \boldsymbol{\phi}, \boldsymbol{\theta}, \sigma_\epsilon$
$\boldsymbol{\chi}_1 =$
$\boldsymbol{\chi}_2 =$
$\boldsymbol{\chi}_3 =$
$\boldsymbol{\chi}_N =$
$U_\delta = u_\delta$
$\delta_1(\boldsymbol{\chi}_1)$
$\delta_2(\boldsymbol{\chi}_2)$
$\delta_3(\boldsymbol{\chi}_3)$
$\delta_N(\boldsymbol{\chi}_N)$
δ_1
δ_2
δ_3
δ_N
δ
$Y_\epsilon = S + \delta + \epsilon_h$
y
$U_\epsilon = u_\epsilon$
$U_{\epsilon_h} = u_{\epsilon_h}$
$\epsilon_{h1}(\boldsymbol{\chi}_1)$
$\epsilon_{h2}(\boldsymbol{\chi}_2)$
$\epsilon_{h3}(\boldsymbol{\chi}_3)$
$\epsilon_{hN}(\boldsymbol{\chi}_N)$
ϵ_{h1}
ϵ_{h2}
ϵ_{h3}
ϵ_{hN}
ϵ_h

Figure 3.5 An evaluation of Eq. (3.3)

3.3.3 Controlled-Seed Method for GSA with Time Series Input

An important challenge in the application of Eq. (3.3) is the computational cost. Here we define "one evaluation of the deterministic function such as Eq. (3.3)" as a function evaluation. Computation of the Sobol' indices based on Eqs. (2.11) and (2.13) is computationally intensive since it requires repeated function evaluations at different values of the inputs. If the double-loop method introduced in Section 2.6 is used, the cost to compute all the first-order indices is $N_f = kn^2 + n$, as shown in Eq. (2.16). The number of function evaluations for the total effects indices is the same as the first order indices. If $X_i (i = 1 \text{ to } k)$ are uncorrelated with each other, a single loop method [67] has been developed to reduce the cost in Eq. (2.11) to $kn + n$. But when the model inputs are correlated, there is no alternative to the double loop method [67]. This section only applies the double loop method, considering the general case of correlated inputs.

Regarding Eq. (3.3) for GSA with time series input, one function evaluation shown in Figure 3.5 requires computations over N time histories. If the time cost for one time history is t_0, the time cost for one function evaluation of Eq. (3.3) is Nt_0. Thus the overall time cost for the first-order indices is:

$$T_1 = Nt_0 \times N_f = Nt_0(kn^2 + n) \tag{3.4}$$

The time cost given by Eq. (3.4) is sometimes unaffordable. Consider a simple example where 1) the time series input is generated by an ARMA(1, 1) model of four model parameters, i.e., $\bar{X}, \phi_1, \theta_1$ and σ_ϵ; 2) the discretization error and surrogate model uncertainty are ignored; and 3) the values of the model parameters $\boldsymbol{\theta}_m$ are precisely known. Then the deterministic function of Eq. (3.3) reduces to $Y = F\big(\bar{X}, \phi_1, \theta_1, \sigma_\epsilon, U_\delta, U_\epsilon\big)$, which requires a six-dimensional GSA ($k = 6$). Assume $t_0 = 0.01$s, which is quite fast and implies the use of a surrogate or a simplified reduced-order model for a realistic structure. Suppose $N = 100$ and $n = 500$, the overall time cost by Eq. (3.4) is about 417 hours, which is rarely affordable. Of course, parallel computing can be used to reduce this time cost, but that requires more computational resources.

The reason for the unaffordable time cost by Eq. (3.4) is as follows: in Eq. (3.4) the natural variability of time series input is represented by N sampled time histories (Figure 3.5), so the auxiliary variable U_ϵ can be introduced only after computing all the sampled time histories to predict the system response. In other words, one function evaluation of Eq. (3.4) requires computing N time histories. In contrast, in Eq. (3.2) for random variable inputs, the natural variability in random variable input X is represented by a single PDF (a PDF in Figure 2.7), and the auxiliary variable U_X generates a deterministic value of X from this distribution. Thus in a function evaluation of Eq. (3.2), only a single value of X is propagated into the model of $S(\boldsymbol{\theta}_m; \boldsymbol{X})$,

$\epsilon_h(\boldsymbol{X})$ and $\delta(\boldsymbol{X})$ to predict the system response. Therefore, the function evaluation of Eq. (3.4) can be accelerated significantly if the natural variability in time series input can be captured by a single PDF before propagating the time histories through the prediction model. The next subsection proposes a controlled-seed method to achieve this outcome.

As explained earlier, the natural variability of time series input is represented by generating multiple time histories, which is basically a process of generating random numbers. Random numbers in computers are always generated by deterministic algorithms such as Mersenne Twister generator [75], Combined Multiplicative Recursive generator [76] or Wichmann-Hill generator [77]. These pseudo-random number generators use a positive integer known as a seed to generate a random number of various distribution types, and a new seed is deterministically computed before generating the next random number. A fixed initial seed value will give a fixed set of random numbers. Nevertheless, the deterministic generators are sufficiently complicated so that the generated pseudo-random samples can pass various statistical tests of randomness.

Therefore, if a code is used to generate time series input using a mathematical model such as the ARMA model, the sample at each time step is determined once the initial seed value for sampling the first time step is given. For example, Figure 3.6 shows that the same initial seed K leads to the same load history at different runs of the ARMA model in MATLAB.

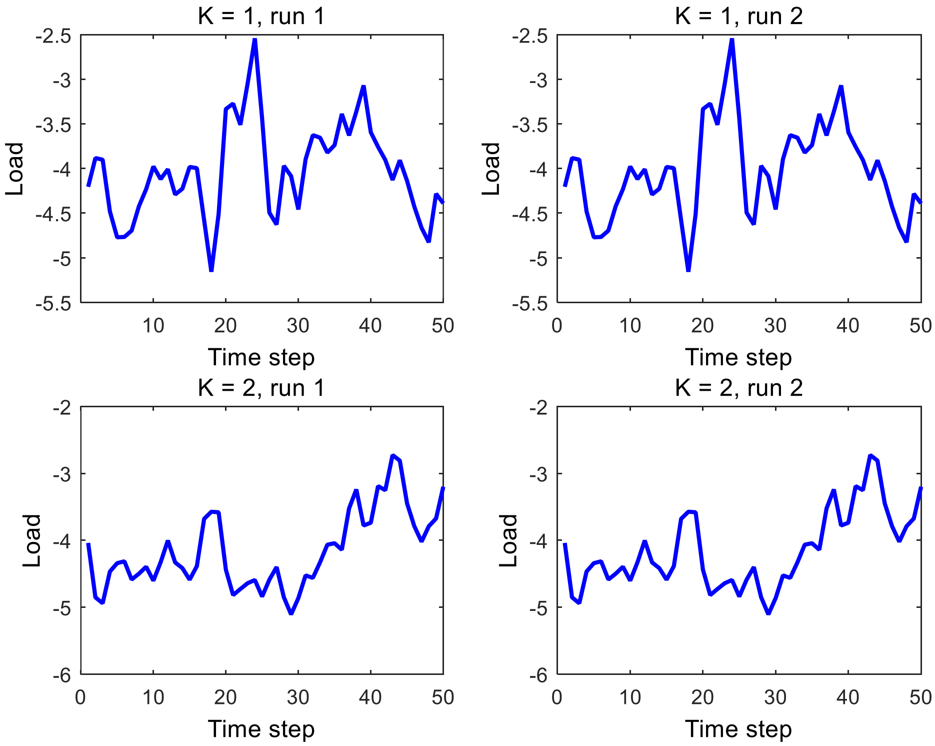


Figure 3.6 Seed and ARMA model simulation in MATLAB

This initial seed K is considered as a random variable controlling the generation of the time series input. For the random variable input, the auxiliary variable U_X captures the natural variability in the random variable input X due to the one-to-one mapping between each value of U_X and the value of X; similarly, the initial seed K captures the natural variability in the time series input due to one-to-one mapping between the value of K and the realization of the time series input. It is equally possible for any positive integer to be used as a seed, so theoretically K has a discrete uniform distribution $U_d(1, n_c)$ where the upper bound n_c is a very large positive integer decided by the specific programing language and computer. But in practice we can define the bounds of this discrete uniform distribution, depending on how many different possible histories are adequate to represent the natural variability in time series input. The numerical example in Section 3.4 assigns a discrete uniform distribution $U_d(1, 100)$ to the initial seed K by implying that 100 possible histories are adequate to represent the natural variability in the ARMA model.

An additional step is needed to apply the initial seed K to global sensitivity analysis. Although the initial seed K captures the natural variability in the time series input, its distribution is discrete but Sobol' indices requires continuous random variables. Therefore another auxiliary variable U_K, which is the CDF value of K, is introduced to represent K. The mapping between the value of U_K and the value of K is:

$$K = a + \lfloor U_K(b - a + 1) \rfloor \tag{3.5}$$

where a and b are the positive integers of lower and upper bounds respectively, and $\lfloor \cdot \rfloor$ is the floor function. The first constraint for a and b is that $a < b \leq n_c$. In addition, the difference between a and b should be large enough to guarantee the diversity of resultant seed values, so that adequate different time histories can be generated to represent the natural variability in the time series input.

In Figure 3.5, the auxiliary variable U_ϵ is to pick one sample of Y from N samples. In other words, U_ϵ actually picks one time series time history. Now the auxiliary variable U_K reaches the same objective, thus it equivalently captures the natural variability in the time series input via K. Then a new deterministic function for GSA is proposed as:

$$Y = F\left(\bar{X}, \boldsymbol{\phi}, \boldsymbol{\theta}, \sigma_\epsilon, \boldsymbol{\theta}_m, U_S, U_{\epsilon_h}, U_\delta, U_K\right) \tag{3.6}$$

where U_K plays the same role as U_ϵ in Eq. (3.3).

Similar to Eq. (3.3), an evaluation of Eq. (3.6) is shown in Figure 3.7, which can be realized in five steps:

1. Generate a sample of $\bar{X}, \boldsymbol{\phi}, \boldsymbol{\theta}, \sigma_\epsilon$ from their joint distribution;
2. Sample U_K and compute the corresponding value of K using Eq. (3.5). Then generate a time history $\boldsymbol{\chi}$ by taking the value of K as the initial seed;

3. Generate a sample $\boldsymbol{\theta}_m^*$ of model parameters $\boldsymbol{\theta}_m$;

4. Compute $S(\boldsymbol{\theta}_m^*;\boldsymbol{\chi})$, $\epsilon_h(\boldsymbol{\chi})$ and $\delta(\boldsymbol{\chi})$, where each one is a distribution;

5. Sample the auxiliary variables U_S, U_ϵ and U_δ to obtain deterministic values of s, ϵ_h and δ by CDF inversion on the distributions in Step 4, respectively. Then the deterministic value of response prediction is $y = s + \epsilon_h + \delta$.

Note that Eq. (3.6) is as flexible as Eqs. (3.2) and (3.3). The corresponding variable can be removed if any uncertainty source is ignored.

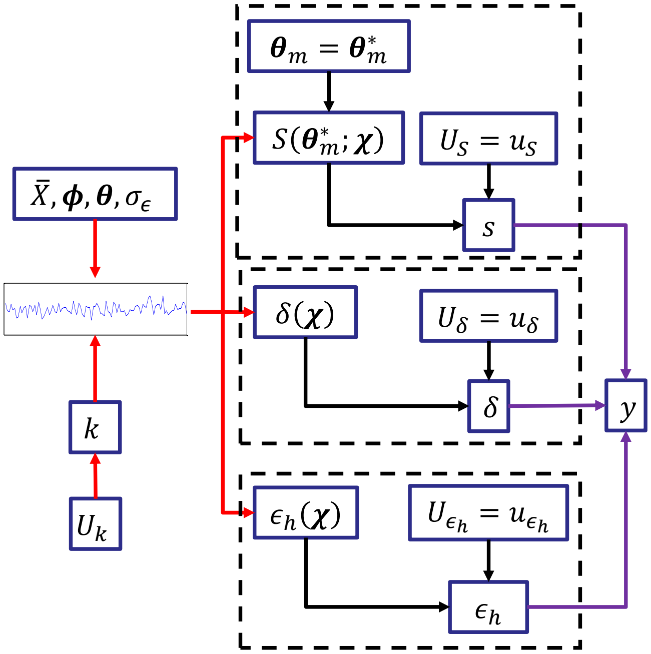

$\boldsymbol{\theta}_m = \boldsymbol{\theta}_m^*$
$S(\boldsymbol{\theta}_m^*;\boldsymbol{\chi})$
$U_S = u_S$
s
$\bar{X}, \boldsymbol{\phi}, \boldsymbol{\theta}, \sigma_\epsilon$
$\delta(\boldsymbol{\chi})$
$U_\delta = u_\delta$
δ
y
k
U_k
$\epsilon_h(\boldsymbol{\chi})$
$U_{\epsilon_h} = u_{\epsilon_h}$
ϵ_h

Figure 3.7 An evaluation of Eq. (3.6)

As the initial seed K is introduced, the proposed method by Eq. (3.6) is named as "controlled-seed method"; in contrast, the method by Eq. (3.2) is named as "uncontrolled-seed method". By developing the controlled-seed method, the natural variability in time series input is captured by U_K. As shown in Figure 3.7, only one time history requires computation in each function evaluation of Eq. (3.6). Thus the overall time cost for the first-order indices is:

$$T_2 = t_0(kn^2 + n) \tag{3.7}$$

Compared with Eq. (3.3) , the computational effort is significantly reduced by the factor N. For the earlier example in Section 3.4 where $t_0 = 0.01\text{s}, n = 500$, and $N = 100$, the time cost reduces to 4.17 hours, instead of 417 hours.

3.4 Numerical Example

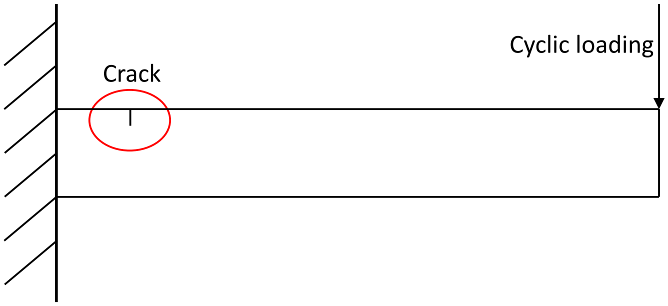


Figure 3.8 Cantilever beam

Consider a single cantilever beam shown in Figure 3.8. An edge crack is assumed to have initiated at the top surface, and this crack grows under the time series loading $\boldsymbol{X}$ of N_t cycles imposed at the other end of the beam. The initial crack size a_0 is assumed to have a normal distribution $N(0.03{,}0.0015^2)$, representing the uncertainty in measuring a_0. The objective of this example is to assess the contribution of each uncertainty source to the uncertainty in the final crack length prediction. The uncertainty sources include structure properties (structure geometry, initial crack size, material properties, and crack growth parameters), loading history, and various model errors (surrogate model error, discretization error, and model form error). In this example, for the sake of illustration, we only consider the uncertainty in initial crack size, loading history, and model errors. Properties of the structure are assumed to be fixed and known. However, the proposed methodology can easily include these additional uncertainty sources.

Section 3.4.1 illustrates the prediction model to compute the final crack length, i.e., how to compute the crack growth at a given time series history generated by ARMA. Section 3.4.2 develops the deterministic functions required for global sensitivity analysis and provides two scenarios: 1) assumes known ARMA model parameter distributions, and compares the efficiency of the uncontrolled-seed method and the controlled-seed method, and 2) calibrates ARMA model parameters by Bayesian inference, and the effect of correlation between ARMA parameters is investigated.

3.4.1 Computational Models

As shown in Figure 3.9, two finite element (FE) models are established by the commercial software ANSYS to compute the stress intensity factor ΔK_s under load X and crack length A. The first FE model has coarse mesh around the crack tip, while the second FE model has fine mesh around the crack tip.

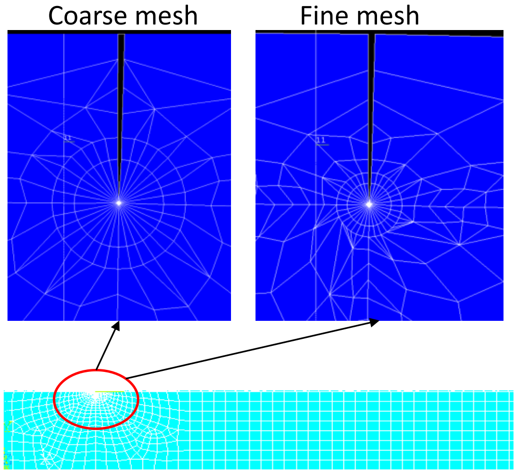


Figure 3.9 FEA model

At given stress intensity factor ΔK_s, an empirical curve of crack growth rate vs. stress intensity factor obtained in material experiment can be used to compute crack growth ΔA in each cycle, as shown in Figure 3.10.

Alternatively, the Paris' law can be also used as the crack growth model to compute ΔA:

$$\mathrm{d}A/\mathrm{d}N = C\Delta K_s^m \tag{3.8}$$

In Eq. (3.8), C and m are Paris' law parameters; $\mathrm{d}A/\mathrm{d}N$ is the crack growth rate, and its magnitude is equal to the predicted crack growth ΔA in one cycle. Since ΔK_s depends on load X and the crack length A, ΔA is a also function of X and A, i.e., $\Delta A(X, A)$.

Paris' law fits the linear behavior part of the empirical curve well, but diverges from the empirical curve in the non-linear behavior parts and brings errors. Using the linear behavior data of the empirical curve, the Paris's law parameters C and m are obtained by a linear regression model of $\log(\mathrm{d}A/\mathrm{d}N) = \log C + m \log \Delta K_s$. The values of C and m by the linear regression are $C = 3.2379 \times 10^{-8}$ and $m = 2.1577$. Since this linear regression gives a high R-squared value of 0.997, this research fixes C and m as constants.

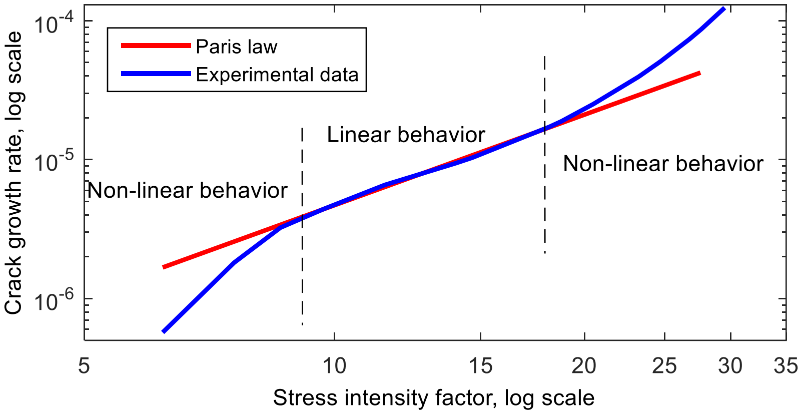


Figure 3.10 Paris law vs. Empirical crack growth curve

Depending on different mesh resolutions and crack growth models, three models with different levels of fidelity are established, as shown in Table 3.1. The crack growth predicted by each of these three models in the t-th cycle are denoted as ΔA_t^l, ΔA_t^m, and ΔA_t^h, respectively. Note the time t is put in subscripts in this example, since the superscripts are used for other purpose.

Table 3.1. Models of different fidelities

Models	Predicted crack growth in each cycle	Mesh type	Crack growth model
Low fidelity model	ΔA_t^l	Coarse mesh	Paris's law
Mid-fidelity model	ΔA_t^m	Fine mesh	Paris' law
High fidelity model	ΔA_t^h	Fine mesh	Empirical curve

For the sake of illustration, the crack growth prediction by the high fidelity model is assumed to be the true value, and the low fidelity model is assumed to be the computational model. As illustrated earlier, the computational model needs two corrections to approximate the true value. At the t-th cycle, the low fidelity model prediction ΔA_t^l is corrected as:

$$\begin{aligned}\Delta A_t^c(X_t, A_{i-1}) &= \Delta A_t^l + \left(\Delta A_t^m - \Delta A_t^l\right) + \left(\Delta A_t^h - \Delta A_t^m\right) \\ &= \Delta A_t^l(X_t, A_{t-1}) + \epsilon_h(X_t, A_{t-1}) + \delta(X_t, A_{t-1})\end{aligned} \tag{3.9}$$

where X_t is the load at the t-th cycle, and A_{t-1} is the crack length after the $(t-1)$-th cycle. The difference between ΔA_t^m and ΔA_t^l is caused by mesh resolutions (indicating discretization error) and denoted as $\epsilon_h(X_t, A_{t-1})$; and the difference between ΔA_t^h and ΔA_t^m is caused by different crack growth models (model form error or model discrepancy) and denoted as $\delta(X_t, A_{t-1})$.

25 values of $\epsilon_h(X_t, A_{t-1})$ and $\delta(X_t, A_{t-1})$ at different load X and crack length A are computed to train Gaussian process (GP) models for $\epsilon_h(X_t, A_{t-1})$ and $\delta(X_t, A_{t-1})$, which are used to compute the error terms at desired values of load and crack length. The GP model output for $\epsilon_h(X_t, A_{t-1})$ is denoted as $gp_{\epsilon_h}(X_t, A_{t-1})$, and the GP model output for $\delta(X_t, A_{t-1})$ is denoted as $gp_\delta(X_t, A_{t-1})$. In addition, for the sake of computational efficiency during uncertainty propagation (since many Monte Carlo samples will be used), a third GP model denoted as $gp_s(X_t, A_{t-1})$ is built to replace the low fidelity model in Table 3.1 and used in the prediction. Therefore the crack growth prediction at the t-th cycle is:

$$\Delta A_t^c(X_t, A_{t-1}) = gp_s(X_t, A_{t-1}) + gp_{\epsilon_h}(X_t, A_{t-1}) + gp_\delta(X_t, A_{t-1}) \tag{3.10}$$

Note that all three terms at the right-hand side of Eq. (3.10) are GP models, thus their outputs are Gaussian random variables with given values of X_t and A_{t-1}, so that the crack growth ΔA_t^c is also a Gaussian variable. In our computation, the standard deviation of this Gaussian variable is less than 1% of its mean value, so that the probability that Eq. (3.10) gives a negative crack growth is almost zero.

Eq. (3.10) is for one cycle. The final crack length is predicted by applying Eq. (3.10) at all cycles sequentially and using $A_t = A_{t-1} + \Delta A_t^c$. Since the crack growth $\Delta A_t^c(X_t, A_{t-1})$ is the sum of three Gaussian distributions, the crack growth in each cycle is stochastic so the starting crack length for each cycle is also stochastic. But this stochastic starting crack length will make the application of Eq. (3.10) tedious due to the nesting of Monte Carlo sampling loops from one cycle to another. Since this numerical example is mainly used to illustrate the proposed framework of contribution assessment, we simply use the mean value of A_{t-1} to compute the crack growth; the uncertainty in A_t is the accumulated uncertainty from the three GP models. Specifically, each of the three Gaussian distributions on the right hand side of Eq. (3.10) are separated into the sum of its mean value and a zero mean Gaussian distribution:

$$\begin{gathered} gp_s(X_t, A_{t-1}) = \mu_s(X_t, A_{t-1}) + gp_s^0(X_t, A_{t-1}) \\ gp_{\epsilon_h}(X_t, A_{t-1}) = \mu_{\epsilon_h}(X_t, A_{t-1}) + gp_{\epsilon_h}^0(X_t, A_{t-1}) \\ gp_\delta(X_t, A_{t-1}) = \mu_\delta(X_t, A_{t-1}) + gp_\delta^0(X_t, A_{t-1}) \end{gathered} \tag{3.11}$$

The crack length prediction A_t after the t-th cycle is assumed to be the sum of a mean value μ_{A_t} and three zero mean Gaussian distributions:

$$A_t = \mu_{A_t} + \sum\nolimits_1^t gp_s^0(X_t, \mu_{A_{t-1}}) + \sum\nolimits_1^t gp_{\epsilon_h}^0(X_t, \mu_{A_{t-1}}) + \sum\nolimits_1^t gp_\delta^0(X_t, \mu_{A_{t-1}}) \tag{3.12}$$

where:

$$\begin{aligned} \mu_{A_t} &= \mu_{A_{t-1}} + \mu_s\left(X_t, \mu_{A_{t-1}}\right) + \mu_{\epsilon_h}\left(X_t, \mu_{A_{t-1}}\right) + \mu_\delta\left(X_t, \mu_{A_{t-1}}\right) \text{ for } t \geq 2 \\ \mu_{A_t} &= a_0 + \mu_s(X_t, a_0) + \mu_{\epsilon_h}(X_t, a_0) + \mu_\delta(X_t, a_0) \text{ for } t = 1 \end{aligned} \tag{3.13}$$

Eq. (3.12) is the prediction model used in this numerical example. In Eq. (3.12), $\sum_1^i gp_s^0(X_t, A_{t-1})$ is the variable of accumulated surrogate model uncertainty, denoted as S_t^a; $\sum_1^t gp_{\epsilon_h}^0(X_t, A_{t-1})$ is the variable of accumulated discretization error uncertainty, denoted as ${\epsilon_h}_t^a$; $\sum_1^i gp_\delta^0(X_t, A_{t-1})$ is the variable of accumulated mode discrepancy uncertainty, denoted as δ_t^a. Auxiliary variables will be introduced to assess the contribution of S_t^a, ${\epsilon_h}_t^a$ and δ_t^a to the uncertainty of A_t.

3.4.2 Contribution Assessment of Each Uncertainty Source

First, the uncertainty in the final crack length A_{N_t} is from the time series input represented by an ARMA model, including the natural variability in ARMA model and the epistemic uncertainty in the ARMA parameters. Second, for a given time series input, the uncertainty in A_{N_t} is from the three accumulative error terms in Eq. (3.12). Based on Eqs. (3.3) and (3.6), the deterministic functions required in global sensitivity analysis are:

$$A_{N_t} = F\left(a_0, \bar{X}, \boldsymbol{\phi}, \boldsymbol{\theta}, \sigma_\epsilon, U_S, U_{\epsilon_h}, U_\delta, U_\epsilon\right) \quad \text{for uncontrolled-seed method} \tag{3.14}$$

$$A_{N_t} = F\left(a_0, \bar{X}, \boldsymbol{\phi}, \boldsymbol{\theta}, \sigma_\epsilon, U_S, U_{\epsilon_h}, U_\delta, U_K\right) \quad \text{for controlled-seed method} \tag{3.15}$$

The evaluations of Eqs. (3.14) and (3.15) follow the steps in Section 3.3.2 and Section 3.3.3, respectively.

Case 1: Uncontrolled-seed method vs. controlled-seed method with known ARMA model

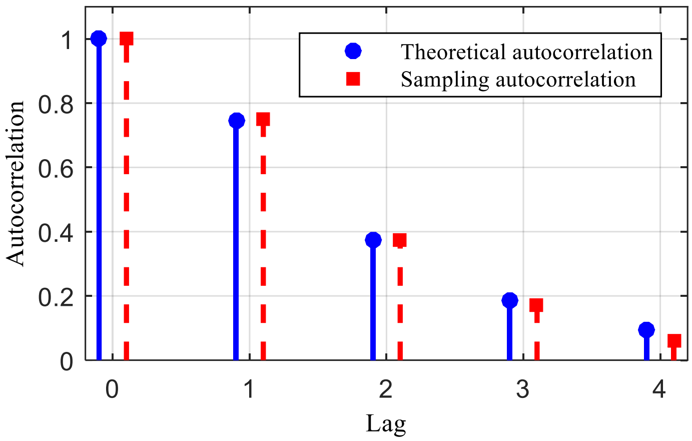


Figure 3.11 Theoretical and sampling autocorrelation function for ARMA(1, 1) model

This section assumes that the time series input is an ARMA(1, 1) model. The distributions of ARMA parameters are assumed as $\bar{X} \sim N(-2, 0.2^2)$, $\phi_1 \sim N(0.5, 0.1^2)$, $\theta_1 \sim N(0.75, 0.1^2)$, $\sigma_\epsilon \sim U(0.1, 0.5)$ and they are uncorrelated. To make the uncontrolled-seed method computationally affordable, we assume that the time series input only has 10 time steps. Assuming that 100 time series histories are adequate to represent the ARMA model with given parameters, Eq. (3.13) generates $N = 100$ time series in each function evaluation and Eq. (3.14) sets the distribution of seed as $K \sim U_d(1, 100)$. This assumption can be verified by checking the consistency of the autocorrelation function $R_s(\tau)$ based on 100 sample histories and the theoretical autocorrelation function $R(\tau)$ of ARMA(1, 1) model with the given ARMA parameters. The comparison of $R_s(\tau)$ and $R(\tau)$ is shown in Figure 3.11 for a ARMA model with $\bar{X} = -2, \phi_1 = 0.5, \theta_1 = 0.75$ and $\sigma_\epsilon = 0.1$, where $R_s(\tau)$ are computed based on 100 sample histories with initial seed values ranging from 1 to 100. In Figure 3.11, $R_s(\tau)$ and $R(\tau)$ is consistent, so our assumption of 100 sample histories is reasonable.

Table 3.2 Global sensitivity analysis for case 1

		First-order effects		Total effects	
	Methods	Uncontrolled-seed	Controlled-seed	Uncontrolled-seed	Controlled-seed
	n	120	500	120	500
	Time (hrs.)	17.4	3.0	17.7	3.1
Indices	a_0	0.073	0.075	0.086	0.084
	$\bar{X}$	0.107	0.100	0.297	0.290
	ϕ_1	0.497	0.484	0.735	0.737
	θ_1	0.000	0.000	0.001	0.001
	σ_ϵ	0.004	0.006	0.032	0.039
	U_S	0.000	0.000	0.000	0.000
	U_{ϵ_h}	0.000	0.000	0.000	0.000
	U_δ	0.000	0.000	0.000	0.000
	U_ϵ/U_K	0.051	0.055	0.068	0.072

The result of GSA using both the uncontrolled-seed and controlled-seed method are reported in Table 3.2. Both the first order and total effects indices can be reported since all the variables in this example are uncorrelated.

Table 3.2 shows consistent results between the two methods: 1) the indices for the same uncertainty source using different methods are very close; 2) both U_ϵ in the uncontrolled-seed method and U_K in the seed method equivalently capture the natural variability in time series input; 3) ϕ_1 is the most dominant variable in the prediction uncertainty. The slight difference between the indices for the same uncertainty source is mainly caused by the limited number of samples in the uncontrolled-seed method (n =120). But if we also apply 500 samples for the uncontrolled-seed method, its time cost will be over unaffordable 300 hours. This also proves the efficiency of the controlled-seed method.

Case 2: Correlation vs. non-correlation with calibrated ARMA model

Figure 3.12 shows a synthetic time history generated as observed data. The loading at each cycle includes a maximum value and a minimum value. Figure 3.12 only shows the minimum value at each cycle since the maximum values are assumed to be zero.

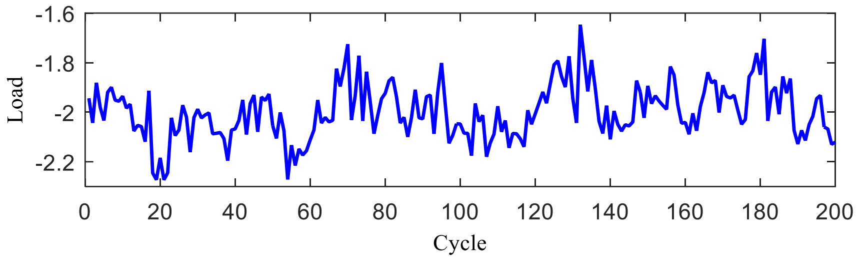


Figure 3.12 Synthetic time series data

An ARMA(2, 2) model is selected to model this time series input. The parameters of the ARMA(2, 2) model are $\bar{X}$, $\boldsymbol{\phi} = \{\phi_1, \phi_2\}$, $\boldsymbol{\theta} = \{\theta_1, \theta_2\}$ and σ_ϵ. Prior distributions are assumed for the ARMA model parameters, and posterior distributions are obtained from Bayesian calibration using Markov Chain Monte Carlo (MCMC) sampling [78].

The marginal PDFs of the priors and the posterior distributions are shown in Figure 3.13. The posteriors of some ARMA parameters are highly correlated, as shown in bold in the correlation matrix of

Table 3.3. For example, the correlation between ϕ_1 and ϕ_2 is -0.8. This can also be explained physically: one criterion to guarantee the stationarity of the ARMA(2, 2) model is $\phi_1 + \phi_2 < 1$ [40], i.e., a larger ϕ_1 requires a smaller ϕ_2 thus indicates a negative correlation between them. The correlation of ARMA parameters has a significant influence on assessing the contribution of each uncertainty source, as shown later.

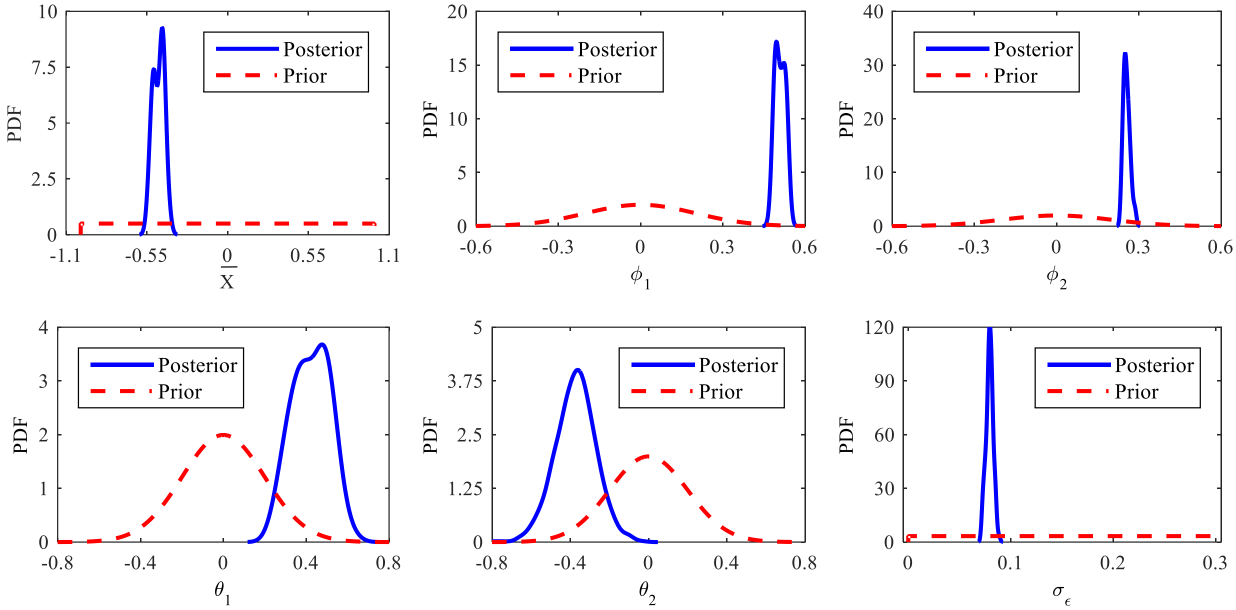


Figure 3.13 Prior and posterior distributions of ARMA parameters

Table 3.3 Correlation matrix of ARMA parameters

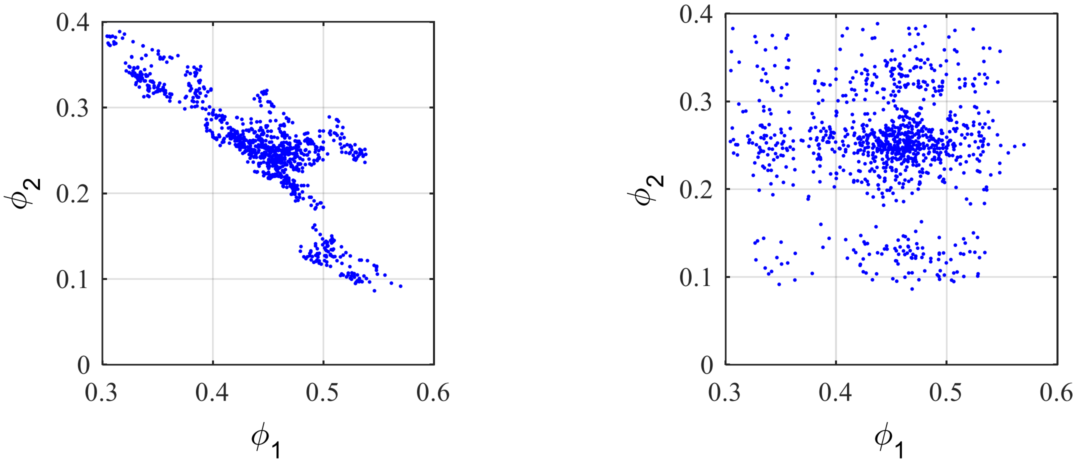
	$\bar{X}$	ϕ_1	ϕ_2	θ_1	θ_2	σ_ϵ
$\bar{X}$	1.000	0.174	0.451	0.082	-0.134	0.004
ϕ_1	0.174	1.000	-0.800	0.059	-0.125	-0.082
ϕ_2	0.451	-0.800	1.000	-0.004	0.033	0.080
θ_1	0.082	0.059	-0.004	1.000	-0.105	-0.400
θ_2	-0.134	-0.125	0.033	-0.105	1.000	0.279
σ_ϵ	0.004	-0.082	0.080	-0.400	0.279	1.000

By assuming that 100 samples of the time series are adequate to represent the ARMA model with given parameters, Eq. (3.13) generates 100 time series in each evaluation and Eq. (3.14) sets the distribution of seed as $K \sim U_d(1, 100)$.

Table 3.4. Global sensitivity analysis: First-order indices for example 2

	a_0	$\bar{X}$	ϕ_1	ϕ_2	θ_1	θ_2	σ_ϵ	U_S	U_{ϵ_h}	U_δ	U_K
Corr. ignored	0.001	0.055	0.249	0.314	0.000	0.000	0.000	0.000	0.000	0.000	0.001
Corr. considered	0.291	0.002	0.001	0.000	0.003	0.002	0.002	0.001	0.001	0.005	0.525

In addition to the longer time series input (200 verses 10), this example is different from the previous one regarding the correlation of ARMA parameters. To show the impact of this correlation, two results of the global sensitivity analysis are shown in Table 3.4. One result intentionally uses the marginal distributions in Figure 3.13 and ignores the correlation in ARMA parameters and another correctly considers the correlation. Only the first-order indices are reported since the total effects indices are not applicable for correlated variables. This example only uses the controlled-seed method since the uncontrolled-seed method is not affordable for a time series with 200 cycles, given the computational resources available.



(a) Correlation = -0.8

(b) Correlation ignored

Figure 3.14 Scatter plot of ϕ_1 and ϕ_2

The indices in Table 3.4 indicate the impact of ARMA parameter correlation in assessing the contribution of each uncertainty source. The result ignoring correlation misleads us to take ϕ_1 and ϕ_2 as the dominant factors, while actually their contribution reduces when the correlation is considered. The reason for this overestimation can be revealed by the scatter plots in Figure 3.14; the scatter width of ϕ_1 and ϕ_2 is much narrower due to the correlation of -0.8 between them, so the uncertainty caused by them in the prediction is reduced significantly.

In the result considering correlation, the important uncertainty sources are initial crack size a_0 and time series input natural variability U_K. The indices for ARMA model parameters are all small, indicating that collecting more time series data cannot help us reduce uncertainty in the final crack length prediction.

In this example, the sensitivity indices for surrogate model error (captured by U_S), discretization error (captured by U_{ϵ_h}) and model form error (captured by U_δ) are very small because our GP models are quite accurate and the variance of the GP model prediction is very small. Here the GP models reach high accuracy because 1) they have only two inputs (load and crack length); and 2) the crack growth is a smooth function with weak non-linearity. Thus 25 training points were enough to achieve very low prediction variance.

3.5 Summary

Various uncertainty sources arise at different steps in the computational prediction of the system response, including surrogate model uncertainty, model discrepancy, model input uncertainty, etc. Some uncertainty sources are aleatory and some are epistemic. In this research, global sensitivity analysis (GSA) based on Sobol' indices is used to quantify the contribution of each uncertainty source. One challenge is that under aleatory and epistemic uncertainty the prediction model is stochastic whereas the Sobol' indices computation requires a deterministic model. Another challenge is that with time series input the GSA will be extremely high-dimensional since each time step introduces a random noise term in the ARMA model.

To solve the first challenge, this research uses the auxiliary variable to represent each uncertainty source explicitly and establish the required deterministic function such that the Sobol' indices can be computed. Based on the auxiliary variable, this research proposes an uncontrolled-seed method to solve the second challenge, by defining a single variable to represent the natural variability in the time series input, thus reducing the problem dimension. Furthermore, a novel controlled-seed method is proposed based on the concept of pseudo-random number generation.

This method requires computing only one history in each function evaluation thus the computation of the Sobol' indices is significantly accelerated. These contributions help to assess the contributions of each aleatory and epistemic uncertainty source to the uncertainty in the time series prediction, such as fatigue crack growth.

CHAPTER 4

UNCERTAINTY INTEGRATION AND RESPONSE PREDICTION IN MULTI-LEVEL PROBLEMS

4.1 Background

Parameters of computational models are often calibrated using experimental data. For a complicated system, it may be difficult to conduct full-scale experiments, but it may be possible to obtain data at lower levels of complexity (e.g., isolated physics or simpler configurations). Figure 4.1 shows such a multi-level problem with two lower levels (G_1, G_2) and a system level (H). The lower levels and the system level constitute a hierarchy, and different levels have the same set of model parameters ($\boldsymbol{\theta}_m$) that need to be calibrated.

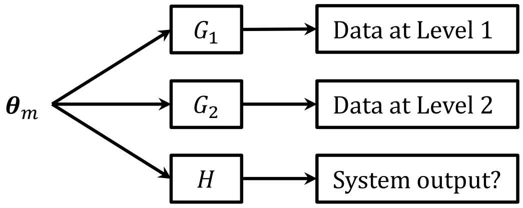


Figure 4.1 Multi-level parameter estimation problem

In order to predict the system level output when data are only available at lower levels, a reasonable route is to quantify the model parameters using lower level data and propagate the results through the computational model at the system level. Several issues need to be addressed in realizing such a multi-level parameter estimation problem. First, even if model input and output are measured in the lower level tests, thereby forming pairwise input-output data, the calibration result can still be uncertain due to several sources, including 1) model errors in the lower level computational models; 2) measurement errors in the experiments; and 3) sparse experimental data.

Second, the existence of multiple lower levels provides multiple possibilities to conduct model calibration and leads to multiple calibration results. In a multi-level problem, model calibration can be conducted using the data from a single level or multiple levels. For the problem in Figure 4.1 with two lower levels, 3 calibration options are possible: 1) calibration using the data and model from Level 1 alone; 2) calibration using the data and model from Level 2 alone, and 3) calibration using the data and models from both Level 1 and Level 2. Generally, if data are available at n different levels, $2^n - 1$ model calibration options are possible to quantify the uncertainty of model parameters [79].

As introduced in Section 2.5, this research uses Bayesian inference for model calibration, thus the result of model calibration is a joint posterior distribution of model parameters. As Kennedy and O'Hagan [32] pointed out, the posterior distribution is the "best-fitting" results in the sense of representing the calibration data faithfully, not necessarily representing the true physical values. The main objective of this research is to determine the appropriate distribution for model parameters $\boldsymbol{\theta}_m$ to be used in system level prediction. One possibility is to use all the lower level data in model calibration and propagate the resultant posterior distribution to predict the system level output. However, this result is conditioned on the event that both the models at Level 1 and Level 2 are valid, which may or may not be true [80]. This research answers this question by assigning a "confidence" measure to each posterior distribution. Note that this research is not using the term "confidence" in the same sense as is used in statistics (as in confidence interval). This "confidence" measure constitutes of two components: 1) the model validity at the corresponding lower level (one can think of this as local confidence regarding each lower level); 2) the relationship between the lower level and the system level, i.e., the relevance of the posterior distribution obtained at the lower level to the system level prediction problem (one can think of this as inter-level confidence). The relationship between two lower levels can be also important. However, this relationship is not considered here since in this research the obtained information in a lower level is extrapolated to the system level, but not to another lower level.

Before quantifying the local confidence, the relationship between model calibration and model validation should be clarified. This topic has been covered in Section 2.5, and can be summarized as:

1. The purpose of model calibration is to adjust a set of parameters associated with a computational model so that the agreement between model prediction and experimental observation is maximized [41].
2. Model validation is the process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model. Generally, model validation is realized by comparing the model prediction against experimental data.
3. Model calibration and model validation are distinct activities. But usually, before model validation, model calibration can be conducted to quantify the values of $\boldsymbol{\theta}_m$ or reduce the uncertainty about their values.

With the calibration and validation perspectives to be used in this research defined as above, the reason to use model validation to quantify the local confidence is explained next. In model validation, the assessed model validity of the corrected prediction model $F(\boldsymbol{\theta}_m; \boldsymbol{x}) + \delta(\boldsymbol{x})$ at a lower level is a combined effect of three components: 1) $F(\boldsymbol{\theta}_m; \boldsymbol{x})$; 2) $\delta(\boldsymbol{x})$; and 3) the posterior distribution of $\boldsymbol{\theta}_m$. The third aspect corresponds to the "local confidence" (not to be confused with confidence intervals used in statistics), thus this research takes the model validity as one factor affecting our confidence in extrapolating the posterior distribution of the model parameter from the lower level to the system level. This is reasonable since the model parameter has been calibrated with a model corresponding to the lower level experiment, and it is important to know whether the model was calibrated accurately; the calibration result is obviously affected by how accurately the lower level model represents the physics in the lower level experiment.

Model validation is about comparing the model prediction against experimental data, and a model validation metric is needed to quantify this comparison.

The model validation metric used in this research is the model reliability metric proposed by Rebba and Mahadevan [52] and further developed by Sankararaman and Mahadevan [53]. This metric measures the model validity by “model reliability”, which is defined as the probability that the difference between model prediction and observed data is less than a pre-defined tolerance. Here the model prediction is stochastic, whose uncertainty is caused by the uncertainty in the posterior distribution of model parameters as well as the uncertainty regarding the model error. In other words, the model reliability metric considers the combined effect of these two sources of uncertainty. The value of model reliability is between 0 and 1, thus it can be conveniently used as a weighting term in subsequent uncertainty integration across multiple levels.

For a given validation data point, the model reliability is a deterministic value. However, its value is different for different data points. To capture this variability in model reliability, this research proposes a stochastic model reliability metric where the model reliability is treated as a random variable instead of a deterministic value. In addition, this research extends the model reliability metric to handle the multivariate output.

As mentioned earlier, the inter-level confidence to extrapolate a lower level posterior distribution to the system level is about the relationship between the lower level and the system level. In this research, the relationship between the lower level and the system level is quantified by a proposed relevance analysis. The necessity of relevance analysis is explained here. An inherent assumption in the proposed relevance analysis is that if the physical configuration and inputs of a lower level experiment (say Level 2 in Figure 4.1) is more similar to the system level than another lower level experiment (say Level 1 in Figure 4.1), it is reasonable to assign higher confidence to the calibration result at this level (i.e., Level 2). Thus the relevance of the lower level to the system level is the degree to which the experimental configuration and inputs at a lower level reflect the physical characteristics of the system so that the calibration results can be reliably used in the system level prediction. The relevance decides the inter-level confidence on the calibration at lower levels and influences the uncertainty integration. This research proposes a method to quantify the relevance using Sobol' indices and the cosine similarity of sensitivity vectors.

With the local confidence and inter-level confidence quantified, uncertainty integration is needed to aggregate all the available information from model calibration, model validation (for local confidence) and relevance analysis (for inter-level confidence). A roll-up methodology for uncertainty integration was proposed in Ref. [80], which results in the integrated distribution of model parameters as a weighted average of the posterior distributions, and the weight terms are the model reliability at lower levels. A brief introduction to this methodology is given in Section 4.5, and this research extends it to incorporate more information from the lower levels, including 1) the stochastic model reliability; and 2) the relevance between any lower level and the system level.

In summary, the motivation of this research is to quantify the distributions of model parameters to be used in system level prediction, by using the available information at multiple levels from model calibration, model validation, relevance analysis, and uncertainty integration. The posterior distributions of model parameters are computed by Bayesian inference. The integration of multiple posterior distributions for each model parameter is assisted by model validation and relevance analysis and realized in a proposed new roll-up method. This research develops a methodology to compute the relevance using Sobol' indices and cosine similarity of vectors. In model validation, the model reliability metric is extended to capture the variability in model reliability among different validation points and to consider multivariate output. Finally, the integrated distributions of model parameters are propagated through the computational model at the system level to predict the system output and quantify its uncertainty.

4.2 Model Calibration

Model calibration has been covered in Section 2.5.1. However, since calibration is the first step of the proposed methodology in this section, a brief summary is given here for the sake of completeness.

Suppose the physical input-output relationship at a single level is described by a computational model $Y_c = F(\boldsymbol{\theta}_m; \boldsymbol{X})$, where Y_c is the computational model output, and $\boldsymbol{\theta}_m$ is a set of unknown model parameters, and $\boldsymbol{X}$ is the model input. Kennedy and O'Hagan (KOH) [32] expressed the relationship between the experimental observation z and the computational model as:

$$Z = F(\boldsymbol{\theta}_m; \boldsymbol{X}) + \delta(\boldsymbol{X}) + \epsilon_m \tag{4.1}$$

where $\delta(\boldsymbol{X})$ is the model error (input-dependent); ϵ_m is the measurement error which is usually assumed to be Gaussian distribution $N(0, \sigma_m^2)$. The model error $\delta(\boldsymbol{X})$ can be modeled using different formulations [31], which introduces more parameters. In addition, to reduce the computational effort, the computational model $F(\boldsymbol{\theta}_m; \boldsymbol{X})$ may be replaced by a surrogate model, and this research uses the GP model. The parameters of $\delta(\boldsymbol{X})$ and the surrogate model for

$F(\boldsymbol{\theta}_m; \boldsymbol{X})$ are also uncertain and need to be estimated. These parameters are also called hyper-parameters to distinguish them from model parameters $\boldsymbol{\theta}_m$. In sum, all the parameters to calibrate include: 1) model parameters $\boldsymbol{\theta}_m$; 2) hyper-parameters in the surrogate model for $F(\boldsymbol{\theta}_m; \boldsymbol{X})$; 3) hyper-parameters $\boldsymbol{\theta}_\delta$ of the model error $\delta(\boldsymbol{X})$; and 4) standard deviation σ_m of ϵ_m. The presence of so many calibration parameters is challenging if calibration data are sparse.

This research ignores the hyper-parameter uncertainty in the GP model of $F(\boldsymbol{\theta}_m; \boldsymbol{X})$ for three reasons: 1) enough training points are used to build an accurate GP model with small variance in the GP prediction, thus the hyper-parameter uncertainty is expected to be small; 2) considering this hyper-parameter uncertainty will bring enormous computational effort [81] in model calibration and validation, whereas this hyper-parameter uncertainty is not the focus of this research; and 3) the uncertainty in the hyper-parameters is typically negligible compared to actual model parameters [82]. Thus we first estimate the hyper-parameters of the GP model and then fix them as deterministic values in the subsequent calibration of model parameters $\boldsymbol{\theta}_m$. In addition, if the model input is fixed, then the input dependent model discrepancy $\delta(\boldsymbol{X})$ will become a single parameter δ. In the numerical example of this section, for each lower level calibration test, the model/experimental input is fixed and so the vector of calibration parameters $\boldsymbol{\theta}$ includes: 1) model parameters $\boldsymbol{\theta}_m$; 2) model error δ; and 3) the standard deviation σ_m of measurement error ϵ_m.

In a multi-level problem, each lower level may provide data for multivariate output quantities, and each output quantity at any level has a corresponding model error $\delta(\boldsymbol{X})$ and measurement error standard deviation σ_m to be calibrated. In Figure 4.1, if calibration data consist of two output quantities at Level 1, model calibration includes two model error terms and two measurement error terms; and if two output quantities at Level 2 are also included for calibration, model calibration includes four model error terms and four measurement error terms.

For the model error $\delta(\boldsymbol{X})$, we need to select the prior distribution for each hyper-parameter in the above formulation. But if model input $\boldsymbol{X}$ is fixed and the hyper-parameters are fixed, we only need to select a prior distribution for δ. In the numerical example in Section 4.6, since there is no information available on δ, a uniform prior distribution is assumed as $\delta \sim U(a, b)$ where a and b are the lower and upper bounds of the uniform distribution. The prior distribution of σ_m is chosen as the non-informative Jeffrey's prior $p'(\sigma_m) \propto 1/\sigma_m$, which is invariant under re-parameterization [83]. In addition, the prior distributions for $\boldsymbol{\theta}_m$ are constructed based on expert opinion.

With prior distributions for $\boldsymbol{\theta} = \{\boldsymbol{\theta}_m, \boldsymbol{\theta}_\delta, \sigma_m\}$ defined and experimental data at lower levels obtained, the Bayesian inference expresses the posterior distribution of $\boldsymbol{\theta}$ as:

$$p''(\boldsymbol{\theta}) = \frac{L(\boldsymbol{\theta})p'(\boldsymbol{\theta})}{\int L(\boldsymbol{\theta})p'(\boldsymbol{\theta})\,\mathrm{d}\boldsymbol{\theta}} \tag{4.2}$$

where $L(\boldsymbol{\theta})$ is the likelihood function of $\boldsymbol{\theta}$ and $p'(\boldsymbol{\theta})$ is the joint prior PDF of $\boldsymbol{\theta}$. The samples of $p''(\boldsymbol{\theta})$ are often generated numerically by Markov Chain Monte Carlo (MCMC) methods [78]. Note that if the computational model $F(\boldsymbol{\theta}_m; \boldsymbol{X})$ is replaced by a GP model $GP(\boldsymbol{\theta}_m; \boldsymbol{X}) \sim N\big(\mu_s(\boldsymbol{\theta}_m; \boldsymbol{X}), \sigma_s^2(\boldsymbol{\theta}_m; \boldsymbol{X})\big)$, this research not only considers its mean prediction $\mu_s(\boldsymbol{\theta}_m; \boldsymbol{X})$ but also its variance $\sigma_s^2(\boldsymbol{\theta}_m; \boldsymbol{X})$. Therefore Eq. (4.1) will change to $Z = N\big(\mu_s(\boldsymbol{\theta}_m; \boldsymbol{X}), \sigma_s^2(\boldsymbol{\theta}_m; \boldsymbol{X})\big) + \delta(\boldsymbol{X}) + N(0, \sigma_m^2)$, and the likelihood function $L(\boldsymbol{\theta})$ is established based on this modified equation so that the surrogate model uncertainty is also incorporated in model calibration.

4.3 Model Validation

As mentioned in Section 4.1, a multi-level problem with n lower levels can provide $2^n - 1$ alternative model calibration results, but model calibration cannot answer the question regarding how to integrate them. Thus model validation is necessary to assess the validity of the model calibration before using the calibrated model parameters for system output prediction.

In this research, the basic concept in uncertainty integration is to combine all the information from lower levels and results in an integrated distribution of model parameter θ as the weighted average of multiple posterior distributions. To make the integrated distribution as a valid PDF, the sum of the weight terms computed in model validation should be unity. The model reliability metric directly satisfies this requirement and is selected in this research.

Section 4.3.1 introduces the model reliability metric; Section 4.3.2 extends it to consider the model reliability as a stochastic variable to aggregate the validation results at different validation points, and Section 4.3.3 extends the model reliability metric to deal with multivariate output

4.3.1 Model Reliability Metric

In model reliability metric, for a specific application, the model is defined to be valid if the difference between the model prediction y and the corresponding validation measurement is less than a predefined tolerance λ. Due to the measurement error ($\epsilon_m \sim N(0, \sigma_m^2)$), the measurement is actually a random variable. For a single observed value d, this random variable is denoted by D with mean value d and standard deviation σ_m, i.e. $D \sim N(d, \sigma_m^2)$. Let G denote the event that the model is valid, then the model reliability is defined as the probability of event G:

$$P(G|d) = P(|Y - d| < \lambda) \tag{4.3}$$

The probability in Eq. is used as a metric to measure model validity, thus this metric is named as “model reliability metric”. If Y and σ_m are deterministic, Eq. (4.3) computes the model reliability where ε is a dummy variable for integration:

$$p(G|d) = \int_{-\lambda}^{\lambda} \frac{1}{\sigma_m\sqrt{2\pi}} \exp\left[-\frac{(\varepsilon-(Y-d))^2}{2\sigma_m^2}\right] \mathrm{d}\varepsilon \tag{4.4}$$

In this research, the model prediction y refers to the computational model output corrected by the model error, i.e., $Y = F(\boldsymbol{\theta}_m;\boldsymbol{X}) + \delta(\boldsymbol{X})$. Although model input $\boldsymbol{x}$ is known, the model prediction y is still stochastic due to the uncertainty of $\delta(\boldsymbol{X})$ and $\boldsymbol{\theta}_m$. Furthermore, another calibration parameter σ_m can be also uncertain. In this case, the model reliability is:

$$P(G|d) = \int P(G|\boldsymbol{\theta},d)p''(\boldsymbol{\theta})\,\mathrm{d}\boldsymbol{\theta} \tag{4.5}$$

where $P(G|\boldsymbol{\theta},d)$ is given by the right side of Eq. (4.4), and $p''(\boldsymbol{\theta})$ is the joint posterior distribution of $\boldsymbol{\theta} = \{\boldsymbol{\theta}_m, \boldsymbol{\theta}_\delta, \sigma_m\}$. Note that if the computational model $F(\boldsymbol{\theta}_m;\boldsymbol{X})$ is replaced by a GP model $GP(\boldsymbol{\theta}_m;\boldsymbol{X}) \sim N\big(\mu_s(\boldsymbol{\theta}_m;\boldsymbol{X}), \sigma_s^2(\boldsymbol{\theta}_m;\boldsymbol{X})\big)$, this research not only considers its mean prediction $\mu_s(\boldsymbol{\theta}_m;\boldsymbol{X})$ but also its variance $\sigma_s^2(\boldsymbol{\theta}_m;\boldsymbol{X})$, thus the model prediction will be $Y = N\big(\mu_s(\boldsymbol{\theta}_m;\boldsymbol{X}), \sigma_s^2(\boldsymbol{\theta}_m;\boldsymbol{X})\big) + \delta(\boldsymbol{X})$. Then the model reliability in Eq. (4.4) is computed based on this formula so that the surrogate model uncertainty is also incorporated in model validation.

Eqs. (4.4) and (4.5) are only suitable for a single observed value d from an output quantity. If multiple data points are observed for an output quantity (i.e., multiple validation experiments), then Eqs. (4.4) and (4.5) are not correct. Model validation is further complicated if experimental data are observed for a multivariate output and multiple validation data points are available. Therefore the concept of the model reliability metric needs to be extended to deal with multiple data points and multivariate output. The first issue will be addressed in Section 4.3.2 by proposing a stochastic model reliability metric, while the second issue will be addressed in Section 4.3.3.

4.3.2 Stochastic Model Reliability Metric

As shown in Eqs. (4.4) and (4.5), the value of model reliability $P(G)$ is deterministic at a single data point D, but changes over different data points. If model inputs $\boldsymbol{X}$ of these data points are known, a mathematical function $P(G|\boldsymbol{X}) = S(\boldsymbol{X})$ can be established where $P(G|\boldsymbol{X})$ is the model reliability at model input $\boldsymbol{X}$. However, this function may be not accurate due to validation data sparseness (only five validation points are available in the numerical example in Section 4.6). Thus constructing a mathematical function for model reliability (as a function of $\boldsymbol{X}$) is not considered in this research. Instead, this research uses a probability distribution to represent the variability in $P(G)$, and this distribution is constructed using the model reliability values at different validation data points. (The first option could be considered if a large number of validation experiments are conducted).

In this research, model reliability $P(G)$ is assumed to have a beta distribution since $P(G) \in [0,1]$ and the sample space of beta distribution is also the interval $[0,1]$. If a data set $\boldsymbol{d} = \{d_1, d_2, \cdots, d_n\}$ of one output quantity is observed for model validation from n experiments with different inputs, the corresponding model reliability values computed by Eq. (4.5) at each experiment are $\boldsymbol{d}_R = \{d_{R_1}, d_{R_2}, \cdots, d_{R_n}\}$. Using $\boldsymbol{d}_R$, several methods can be used to construct the PDF of model reliability, such as the method of maximum likelihood, method of moments, or Bayesian inference. This research uses the method of moments to construct the PDF of $P(G)$. In summary, this approach gives a stochastic representation of model reliability, i.e., $P(G)$ is not a single value but represented by a probabilistic distribution. The next section extends the model reliability metric to deal with multivariate output.

4.3.3 Extension to Multivariate Output

If K output quantities are observed in a validation experiment, we have a set of K models sharing the same model input and model parameters:

$$\boldsymbol{Y} = \boldsymbol{F}(\boldsymbol{\theta}_m; \boldsymbol{X}) + \boldsymbol{\delta}(\boldsymbol{X}) \leftrightarrow \begin{cases} Y_1 = F_1(\boldsymbol{\theta}_m; \boldsymbol{X}) + \delta_1(\boldsymbol{X}) \\ Y_2 = F_2(\boldsymbol{\theta}_m; \boldsymbol{X}) + \delta_2(\boldsymbol{X}) \\ \cdots \\ Y_K = F_K(\boldsymbol{\theta}_m; \boldsymbol{X}) + \delta_K(\boldsymbol{X}) \end{cases} \tag{4.6}$$

where $F_j(\boldsymbol{\theta}_m; \boldsymbol{X})$ and $\delta_j(\boldsymbol{X})$ $(j = 1 \text{ to } K)$ are the computational model and model error of the j^{th}quantity. Each quantity also has a measurement error $\epsilon_{m_j} \sim N(0, \sigma_{m_j}^2)$ and the corresponding variable $Z_j = Y_j + N(0, \sigma_{m_j}^2)$ representing the measurement. We denote $\boldsymbol{Z} = \{Z_1, \ldots, Z_j, \ldots Z_K\}^T$. Assume that n experiments are conducted. In the i^{th} experiment ($i = 1 \text{ to } n$), data points for K quantities form a data set $\boldsymbol{d}_i = \{d_{i1}, \ldots, d_{ij}, \ldots, d_{iK}\}^T$. In addition, the pre-defined tolerance for each quantity is included in a vector $\boldsymbol{\lambda} = \{\lambda_1, \ldots, \lambda_j, \ldots, \lambda_K\}^T$.

The distance between $\boldsymbol{Z}$ and $\boldsymbol{d}_i$ can be measured by multiple distance functions such as the Euclidean distance, Chebyshev distance, Manhattan distance, and Minkowski distance [84]. This research uses the Mahalanobis distance [85]. The Mahalanobis distance between $\boldsymbol{Z}$ and $\boldsymbol{d}_i$ is defined as $M = \sqrt{(\boldsymbol{Z} - \boldsymbol{d}_i)^T \boldsymbol{\Sigma}_{\boldsymbol{Z}}^{-1} (\boldsymbol{Z} - \boldsymbol{d}_i)}$ where $\boldsymbol{\Sigma}_{\boldsymbol{Z}}$ is the covariance matrix of $\boldsymbol{Z}$. The Mahalanobis distance transfers $\boldsymbol{Z}$ and $\boldsymbol{d}_i$ into the normalized principal component (PC) space [85] by using $\boldsymbol{\Sigma}_{\boldsymbol{Z}}^{-1}$. Compared to other distance functions, the Mahalanobis distance brings two advantages: 1) the correlations between output quantities are considered; and 2) the output quantities are normalized to the same scale to prevent any quantity from dominating the metric simply due to large numerical values. Using the Mahalanobis distance, the model reliability for multivariate output is defined as:

$$P(G|\boldsymbol{d}_i) = P(M < \lambda_M) = P\left(\sqrt{(\boldsymbol{Z}-\boldsymbol{d}_i)^T\boldsymbol{\Sigma}_{\boldsymbol{Z}}^{-1}(\boldsymbol{Z}-\boldsymbol{d}_i)} < \sqrt{\boldsymbol{\lambda}^T\boldsymbol{\Sigma}_{\boldsymbol{Z}}^{-1}\boldsymbol{\lambda}}\right) \tag{4.7}$$

where $\lambda_M = \sqrt{\boldsymbol{\lambda}^T\boldsymbol{\Sigma}_{\boldsymbol{Z}}^{-1}\boldsymbol{\lambda}}$ is the normalized tolerance.

Generally, the posterior distributions obtained in model calibration are numerical samples generated by MCMC, so the subsequent model reliability in Eqs. (4.4) and (4.5) is also computed numerically. Numerical computation also facilitates the realization of the extended model reliability in Eq. (4.7). Here the model reliability is expressed as:

$$\begin{aligned} P(G|\boldsymbol{d}_i) &= P(M < \lambda_M|\boldsymbol{d}_i) = \int_0^{\lambda_M} p(M|\boldsymbol{d}_i)\mathrm{d}M \\ &= \int_0^{\lambda_M}\left(\int p(M|\boldsymbol{d}_i,\boldsymbol{\theta})p''(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}\right)\mathrm{d}M \end{aligned} \tag{4.8}$$

Eq. (4.8) indicates a numerical algorithm to compute the model reliability:

1. Generate a random sample of $\boldsymbol{\theta}$ from its posterior distribution $p''(\boldsymbol{\theta})$;
2. Generate a sample of M conditioned on $\boldsymbol{\theta}$ by generating a sample of $\boldsymbol{Z}$ and computing its Mahalanobis distance from $\boldsymbol{d}_i$;
3. Repeat steps 1 and 2 to obtain N samples of M; these samples can be used to construct the distribution $p(M|\boldsymbol{d}_i)$, which is not conditioned on $\boldsymbol{\theta}$;
4. If N' out of N samples in step 3 satisfy $M < \lambda_M$, the model reliability is $P(G|\boldsymbol{d}_i) = N'/N$.

The model reliability $P(G|\boldsymbol{d}_i)$ by Eq. (4.8) is regarding a single experiment and $P(G|\boldsymbol{d}_i)$ is a deterministic value. Thus n experiments will give n different model reliability values $\{P(G|\boldsymbol{d}_1),\dots,P(G|\boldsymbol{d}_n)\}$. As proposed in Section 4.3.2, these values can be used to build a probability distribution for the model reliability $P(G)$, by treating $P(G)$ as a random variable instead of a deterministic value.

4.4 Relevance Analysis

Section 4.1 explains the necessity to assign a larger weight to the level physically “closer” or more relevant to the system level than the other. For instance, to predict the battery temperature of a spacecraft on the way to Mars, the data of the same quantity collected from its journey to the Moon will be more valuable than the data collected in any laboratory experiment on earth, since the former ones come from a physical environment more similar to the system of interest. Hence this section develops a method for relevance analysis, which measures the degree to which the experimental configuration and inputs at a lower level reflect the physics captured in the system-level model. Currently, such measure is only intuitive and qualitative; an objective quantitative measure of relevance is needed for uncertainty integration.

The methodology to measure relevance should have two desired features. First, the defined methodology needs no mathematical details of the model in each level, since the model in each level could be a black box. Second, the resultant relevance measure can be used conveniently as a weighting term in uncertainty integration. To fulfill these two criteria, a relevance analysis using Sobol’ indices is proposed in this section.

Consider a model $Y = F(\boldsymbol{X})$ where $\boldsymbol{X} = \{X_1, \dots, X_N\}$ is a vector containing all the inputs. Sensitivity analysis measures the contribution of each input to the uncertainty of $\boldsymbol{Y}$ [54]. Compared to local sensitivity analysis, global sensitivity analysis (GSA) considers the entire probability distribution of the input, not just the contribution at a local point. The Sobol’ indices for GSA have been developed in the literature based on the variance decomposition theorem [61], including first-order index and total effects index. For a particular input X_i, its first-order index is $S_1^i = V(E(Y|X_i))/V(Y)$; and its total effects index is $S_T^i = 1 - V(E(Y|\boldsymbol{X}_{-i}))/V(Y)$ where $\boldsymbol{X}_{-i}$ means all the inputs other than X_i. The first-order index S_1^i measures the contribution of X_i by itself, and the sum of first-order indices of all inputs is always less than or equal to unity. The difference between this sum and unity is the contribution of the interaction among inputs. In contrast, the total effects index S_T^i contains not only the contribution of X_i, but also the interaction effect of X_i with other inputs. The interaction between variables will be ignored if the first-order index is used, thus this research uses the total effects index to develop a method to quantify the relevance. In the following discussion the term sensitivity index indicates the total effects index.

Without loss of generality, this research takes the multi-level problem in Figure 4.1 for the illustration of relevance analysis. To predict the system output Y_s (such as the maximum acceleration at the top mass in the numerical example in Section 4.6), the same quantity is also measured at lower levels (in the numerical example the maximum acceleration at the top mass is also measured at Level 1 and Level 2). The three prediction models for this quantity at different levels are $Y_{L_1} = GP_{L_1}(\boldsymbol{\theta}_m, \boldsymbol{X}_{L_1}) + \delta_{L_1}(\boldsymbol{X}_{L_1})$, $Y_{L_2} = GP_{L_2}(\boldsymbol{\theta}_m, \boldsymbol{X}_{L_2}) + \delta_{L_2}(\boldsymbol{X}_{L_2})$, $Y_{L_1} = GP_{L_s}(\boldsymbol{\theta}_m, \boldsymbol{X}_{L_s})$ where $\boldsymbol{\theta}_m$ are model parameters and $\boldsymbol{X}_{L_1}, \boldsymbol{X}_{L_2}, \boldsymbol{X}_s$ are the model inputs at each level. Note that 1) the computational models are replaced by the GP models to improve computational efficiency; 2) model errors are considered in Level 1 and Level 2; and 3) model error at the system level is not considered since no information on it is available. These prediction models are stochastic, i.e., the output is stochastic even at fixed values of model inputs and model parameters. However, the Sobol' indices computation requires a deterministic model, i.e., deterministic output at given values of model inputs and model parameters. This research applies the auxiliary variable methodology based on the probability integral transform, as developed in Refs [86][59], to obtain a deterministic value of the output for a given realization of inputs and model parameters; thus the Sobol' indices can be computed.

Assume model parameters, model inputs, auxiliary variables constitute N_{L_1} elements in total at Level 1; since each element has a corresponding sensitivity index, a N_{L_1}-dimensional vector V_{L_1} of sensitivity indices will be obtained at Level 1. Similarly, a N_{L_2}-dimensional sensitivity vector V_{L_2} will be obtained at Level 2 and a N_s-dimensional sensitivity vector V_s will be obtained at the system level.

Rigorously, measuring the relevance requires comparing the mathematical model of the lower level and the mathematical model of the system level. However, this comparison is not easy if the models at different levels have distinct formats and are addressing different physical configurations (3-mass-spring vs. 3-mass-spring-on-beam in the numerical example) and are under different inputs (sinusoidal inputs vs. random process inputs in the numerical example). Further, the model sometimes may be a black box; thus we cannot access its mathematical details and a direct comparison would be difficult. The obtained sensitivity vectors quantify the contribution of each model input/parameter towards the uncertainty in the model output. In other words, the sensitivity vector indicates which model input/parameter is more important in affecting the model output uncertainty. Actually, whether the model input/parameter is important is determined by the physics of the model, thus the sensitivity vector is a representative of the physics, to the extent that the model represents the physics accurately. Therefore, this research considers the sensitivity vector as an indicator of the physics captured in the model. (Of course, how well the physics is captured in the model is already indicated by the model reliability metric); thus the comparison of the vectors from two different levels is used to quantify the relevance between these two levels.

One issue in the comparison of $V_{L_i}(i = 1,2)$ and V_s is that they may have different sizes (N_{L_1}, N_{L_2}, N_s may not be equal to each other) and some elements in one vector may not be present in the other vector. The shared dimensions of V_{L_i} and V_s are model parameters $\boldsymbol{\theta}_m$; and the unshared dimensions are the different model inputs and auxiliary variables at each level. To solve this problem we add the unshared dimension in V_{L_i} or V_s to the other vectors but set the corresponding sensitivity indices as zero since the added dimensions have no effect in the computation of the original sensitivity vector. Thus all the vectors V_{L_i} or V_s are brought to the same size.

Several methods are available to compare two vectors, such as Euclidean distance [84], Manhattan distance [84], Chebyshev distance [84], and cosine similarity [84,87]. To include the relevance in the subsequent uncertainty integration conveniently, we define the relevance index R as the square of cosine similarity of the sensitivity vectors, where the cosine similarity is the normalized dot product of two vectors:

$$R = \left(\frac{V_{L_i} \cdot V_s}{\left\| V_{L_i} \right\| \| V_s \|} \right)^2 \tag{4.9}$$

In other words, the above relevance index is the square of the cosine value of the angle between two sensitivity vectors, the elements in which are all positive. If the angle is zero, the relevance between these two levels is 1; if the two vectors are perpendicular, the relevance is 0.

In addition, this definition of relevance generates a value on the interval [0, 1]; and its complement, the square of the sine value, indicates physical non-relevance; hence the sum of “relevance” and “non-relevance” is the unity. Here the relevance index is a plausibility model for the proposition “The lower level model reflects the physical characteristics of the system level model”, and the plausibility of this proposition is the relevance index. Based on Cox’s theorem [88], this plausibility model is isomorphic to probability, since 1) the relevance index is a real value depending on the information of sensitivity vectors we obtained, and 2) the relevance index changes sensibly as the sensitivity vectors change. Thus the relevance index can be converted to probability by scaling, which has been done since the relevance index defined in Eq. (4.9) is already on the interval [0, 1]. Therefore in the roll-up methodology proposed in Section 4.5, we treat the relevance index as a probability and conveniently include it as a weighting term in the uncertainty integration.

However, the relevance index is only calculated based on the prediction models at each level, and data at lower levels; but no system-level observation data is assumed to be available. Therefore, if the system-level model does not capture the system-level physics very well, the relevance index cannot capture the effect of this discrepancy. Thus the proposed relevance index approach is not a fully physics-based approach and does not provide a comprehensive comparison of the actual physics at different levels. However, the sensitivity vector does provide an indication of the physics captured in the models through variance decomposition, and we seek to include this information in the distributions of those system level model parameters that are inferred using lower level tests and models.

When the system-level model has additional physics, there may be additional parameters in the system-level model to reflect this. The sensitivity vector of the system level model will quantify the contribution of these additional parameters, as well as the contribution of the parameters shared with the lower level models. The relevance index is based on the dot product of sensitivity vectors for the models at two different levels. Therefore, if the additional physics parameters in the system-level model have a significant contribution, then the physics in the Level 1 model may not be closely related to the physics in the system-level model. In that case, the two corresponding sensitivity vectors will diverge, and the relevance index of Level 1 will be small. Similarly, if the physics in the Level 2 model is not closely related to the physics in the system-level model, the relevance index of Level 2 will be small.

A further question arises in the computation of relevance index. Sobol' indices consider the entire distribution of the influencing variable, but the posterior distribution of $\boldsymbol{\theta}_m$ (to be used in system level prediction) is unknown before the uncertainty integration. In order to solve this problem, a straightforward iterative algorithm to compute the relevance index R is proposed below:

1. Set an initial value of R.
2. Obtain the integrated distribution of each model parameter using the current relevance and the proposed roll-up method in Section 4.5 below.
3. Use the integrated distributions from step 2 to compute the sensitivity indices, and re-compute the updated relevance index R.
4. Repeat steps 2 and 3 until the relevance index R converges.

Thus, the results of calibration and validation at each lower level and relevance indices between the lower levels and the system level have been obtained. The next task is to construct the integrated distribution of the system level model parameters and predict the system output.

4.5 Uncertainty Integration and Prediction

For a multi-level problem, the purpose of uncertainty integration is to combine all the available information (from calibration, validation and relevance analysis) from the lower levels and predict the response at the system level. In this research the information from the lower level includes: 1) the posterior distributions from model calibration by considering data at each individual lower level, as well as data from multiple lower levels; 2) the model reliability distributions from model validation at each lower level; and 3) the relevance indices between each lower level and the system level. A roll-up methodology has been proposed in Ref. [80] for uncertainty integration. For the multi-level problem in Figure 4.1, this methodology results in an integrated distribution [89] for a model parameter $\theta \in \boldsymbol{\theta}_m$:

$$
\begin{aligned}
p\left(\theta\middle|D_1^{C,V}, D_2^{C,V}\right) = {} & P(G_1)P(G_2)p(\theta|D_1^C, D_2^C) + P(G_1')P(G_2)p(\theta|D_2^C) \\
& + P(G_1)P(G_2')p(\theta|D_1^C) + P(G_1')P(G_2')p(\theta)
\end{aligned}
\tag{4.10}
$$

In Eq. (4.10) the integrated distribution $p\left(\theta\middle|D_1^{C,V}, D_2^{C,V}\right)$ is a weighted average of multiple posterior distributions and contains four terms: in the first term the posterior distribution $p(\theta|D_1^C, D_2^C)$ uses the calibration data of both Level 1 and Level 2 and its weight $P(G_1)P(G_2)$ is the probability that both of the models are valid; in the second and third terms the posterior distribution $p(\theta|D_i^C)$ uses the calibration data at Level i alone and its weight is the probability that the model at Level i is valid but the model at another level is invalid; in the last term the weight $P(G_1')P(G_2')$ of the prior distribution $p(\theta)$ is the probability that both of the models are invalid. After obtaining the integrated distributions for all the parameters in $\boldsymbol{\theta}_m$, the system response can be predicted by propagating all these integrated distributions through the computational model of the system level.

Obviously, the weight of each PDF on the right-hand side of Eq. (4.10) is purely decided by model validation. This research proposes an extension of Eq. (4.10) to include two additional concepts:

1. Stochastic model reliability: The model reliability $P(G_i)$ in Eq. (4.10) is a deterministic value, where G_i is the event that the model at Level i is valid; and this research proposes the stochastic model reliability metric, where $P(G_i)$ is a random variable with PDF $p(P(G_i))$ as explained in Section 4.3.2;
2. Relevance index: This has been defined in Section 4.4 as the square of the cosine value of the angle between the sensitivity vectors at a lower level and system level. We treat the relevance index similar to probability in the roll-up methodology, based on Cox's theorem.

> If S_i denotes the event that Level i is relevant to the system level, then the probability $P(S_i|G_i)$ is equal to the value of the relevance index R; this probability is conditioned on G_i since the computation of the relevance index uses the model at Level i; in contrast $P(S_i'|G_i)$ denotes the probability of non-relevance, and is equal to $1-R$.

The roll-up formula in Eq. (4.10) can be extended to consider stochastic model reliability by rewriting the left-hand side as $p(\theta|D_1^{C,V}, D_2^{C,V}, P(G_1), P(G_2))$ and averaging it over $p(P(G_1))$ and $p(P(G_2))$. But a new formula is required to include the relevance index. Take the multi-level problem in Figure 4.1 as an example. The integrated distribution of a model parameter θ conditioned on the calibration and validation data and model reliability $P(G_i)(i=1,2)$ is redefined as:

$$\begin{aligned}
&p\left(\theta\middle|D_1^{C,V}, D_2^{C,V}, P(G_1), P(G_2)\right)\\
&= P(G_1G_2S_1S_2)p(\theta|D_1^C, D_2^C) + P\big(G_1S_1 \cap (G_2' \cup S_2')\big)p(\theta|D_1^C) \qquad (4.11)\\
&+P\big(G_2S_2 \cap (G_1' \cup S_1')\big)p(\theta|D_2^C) + P((G_1' \cup S_1') \cap (G_2' \cup S_2'))p(\theta)
\end{aligned}$$

From the view of generating samples, Eq. (4.11) indicates two criteria: 1) whether a level is relevant to the system level; 2) whether a level has a valid model. A sample of θ is generated from $p(\theta|D_1^C, D_2^C)$ only when both levels satisfy both criteria; a sample of θ is generated from $p\big(\theta\big|D_i^C\big)$ if level i satisfies both criteria but the other level does not; and a sample of θ is generated from the prior distribution $p(\theta)$ if neither level satisfies both criteria. By assuming independence of model validity and relevance between different lower levels, the weight terms in Eq. (4.11) are computed by using the values of $P(G_i), P(S_i|G_i)$ and two fundamental probability relationships: $P(G_iS_i) = P(G_i)P(S_i|G_i), P(G_i' \cup S_i') = 1 - P(G_iS_i)$. Eq. (4.11) also implies the option of “using only data from one level”. If both the model validity and relevance are 1 for Level 1, and either model validity or relevance is 0 for Level 2, Eq. (4.11) reduces to $p\left(\theta\middle|D_1^{C,V},D_2^{C,V}\right)=p(\theta|D_1^C)$, i.e., only Level 1 data is used.

The integrated distribution of θ, which is conditioned on both calibration and validation data, can now be computed as:

$$p\left(\theta\middle|D_1^{C,V},D_2^{C,V}\right)=\iint p\left(\theta\middle|D_1^{C,V},D_2^{C,V},P(G_1),P(G_2)\right)p\big(P(G_1)\big)p\big(P(G_2)\big)\mathrm{d}P(G_1)\mathrm{d}P(G_2) \quad (4.12)$$

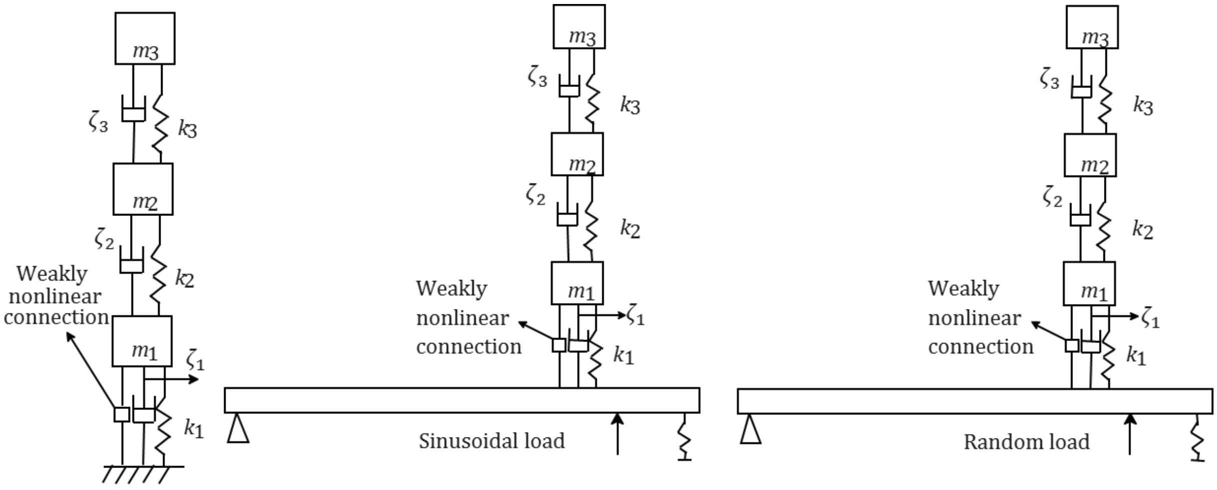
Eqs. (4.11) and (4.12) express the proposed approach of integrating calibration, validation and relevance results at lower levels. Note that Eq. (4.12) accounts for stochastic model reliability. The analytical expression of $p\left(\theta\middle|D_1^{C,V},D_2^{C,V}\right)$ is difficult to derive since the results we collect in model calibration and validation are all numerical. A single loop sampling approach is proposed to construct $p\left(\theta\middle|D_1^{C,V},D_2^{C,V}\right)$ numerically, as follows:

1. Generate a sample of $P(G_1)$ and $P(G_2)$ from their distributions.
2. Compute the weight terms in Eq. (4.11). Divide the interval [0, 1] into four ranges; the length of the k^{th} range is equal to the value of the k^{th} weight in Eq. (4.11).
3. Generate a random number from the uniform distribution $U(0,1)$.
4. Generate a sample of θ using stratified sampling, i.e., from $p(\theta|D_1^C,D_2^C)$ if the random number in step 3 is located in the first range; from $p(\theta|D_1^C)$ if located in the second range; from $p(\theta|D_2^C)$ if located in the third domain; from $p(\theta)$ if located in the fourth domain.
5. Repeat steps 1 to 4 to obtain multiple samples of θ; then construct the PDF $p\left(\theta\middle|D_1^{C,V},D_2^{C,V}\right)$ by any method such as kernel density estimation [90].

After obtaining the integrated distributions of all the model parameters, the final step is to propagate the integrated distributions through the computational model of the system of interest to predict the system level output. This can be done by Monte Carlo sampling or other preferred stochastic analysis methods. Due to the uncertainty in the model parameters, the predicted system output will also be stochastic, and its distribution can be constructed by kernel density estimation. The distribution of the system output now systematically includes the contributions from calibration and validation activities at lower levels, and also accounts for the relevance of the lower levels to the actual system.

4.6 Numerical Example

4.6.1 Problem Description



(a) Level 1 (b) Level 2 (c) Level 3

Figure 4.2 Structural dynamics challenge problem

A multi-level structural dynamics challenge problem provided by Sandia National Laboratories [91] is used to illustrate the methodology developed in Sections 4.2 to 4.5. As shown in Figure 4.2, Level 1 contains three mass-spring-damper dynamic components in series, and a sinusoidal force input $P_s = 300\sin(500t)$ is applied to m_1. At Level 2, the dynamic system is mounted on a beam supported by a hinge at one end and a spring at the other end; a sinusoidal force input $P_s = 3000\sin(350t)$ is applied on the beam. The configuration of the system level is the same as Level

2, but the input is a random process loading (indicating difference in usage condition). Here Level 1 and Level 2 are defined as lower levels, and experimental data are assumed to be available only at the lower levels. All levels share six model parameters: three spring stiffnesses $k_i (i = 1,2,3)$ and three damping ratios $\zeta_i (i = 1,2,3)$; and they are assumed to be deterministic but unknown parameters, which are to be calibrated. The units of all quantities are non-dimensional.

Suppose ten experiments are conducted at each of Level 1 and Level 2; and the displacement, velocity and acceleration history at each degree of freedom are recorded. Six quantities at each lower level are extracted from these records as the synthetic experimental data in model calibration and validation: 1) $A_i (i = 1,2,3)$: the maximum acceleration in the i^{th} mass; 2) $D_i (i = 1,2,3)$: the energy dissipated by the i^{th} damper in 1000 time units.

Table 4.1 Synthetic experimental data at Level 1

	Calibration Data					Validation Data				
A_1	10749	8146	9195	9500	10185	9940	10233	9887	9837	10409
A_2	6362	6827	6780	5759	6319	6579	6346	6730	6160	6126
A_3	1509	1465	1431	1556	1512	1416	1288	1293	1548	1360
D_1	93230	93059	84033	86102	92717	84258	89758	95249	85275	90709
D_2	8110	7283	8377	8590	8736	7490	8407	8127	8710	8477
D_3	33948	30740	30693	34290	24536	34579	31193	29959	33172	33723

Table 4.2 Synthetic experimental data at Level 2

	Calibration Data					Validation Data				
A_1	3876	4110	4372	4187	4443	4486	3912	4237	4394	4807
A_2	4316	4051	4488	3947	4596	4347	5008	4930	4455	4809
A_3	3648	4133	4311	4558	4126	4410	4037	4380	4523	4277
D_1	8593	9009	8966	8910	9746	8606	8644	8757	9050	8458
D_2	1566	1563	1749	1616	1602	1718	1577	1597	1614	1451
D_3	2490	2975	2679	2891	3017	2654	2834	3021	2983	3121

The synthetic experimental data are listed in Table 4.1 and

Table 4.2. The data points for each quantity from the first five tests are selected as calibration data and the rest as validation data.

Computational models for the three levels have been established. The method to solve the dynamic problem at Level 1 can be found in structural dynamics textbooks [92]; and the computational models using the finite element method for Level 2 and the system level are provided by Sandia National Laboratories [32].

Since the model input at each level is fixed, the input-dependent model error is an unknown deterministic value. Thus the parameters to be calibrated in this example are: the spring stiffnesses $k_i (i = 1,2,3)$, the damping ratios $\zeta_i (i = 1,2,3)$, model error δ and the output measurement error standard deviation σ_m if the data of the corresponding quantity are used in model calibration. Based on expert opinion, suppose the prior distribution of each k_i and ζ_i is assumed to be lognormal with a coefficient of variation of 10% and mean values of $\mu_{k_1} = 5000$, $\mu_{k_2} = 9000$, $\mu_{k_3} = 8000$, $\mu_{\zeta_i} = 0.025$ $(i = 1, 2, 3)$. The prior distribution of model error is assumed to be uniform, i.e., $\delta \sim U(a, b)$ and the prior of σ_m is Jeffrey's prior $p'(\sigma_m) \propto 1/\sigma_m$.

The objective in this numerical example is to quantify the uncertainty in the prediction of maximum acceleration at m_3 in the system level, by using available models and experimental data.

Since as many as six quantities are measured, we can choose any combination of these six quantities in the analysis. Measurement data on more output quantities reduce the uncertainty in the system output prediction, but the computational effort will also increase and each quantity will bring two more related terms (δ and σ_m) for calibration. For the sake of brevity, only the calibration and validation results using the test data for all six quantities are provided below. But a plot showing the reduction in the uncertainty of system output prediction with the increase of output quantity measurements is also provided at the end.

4.6.2 Results and Analysis

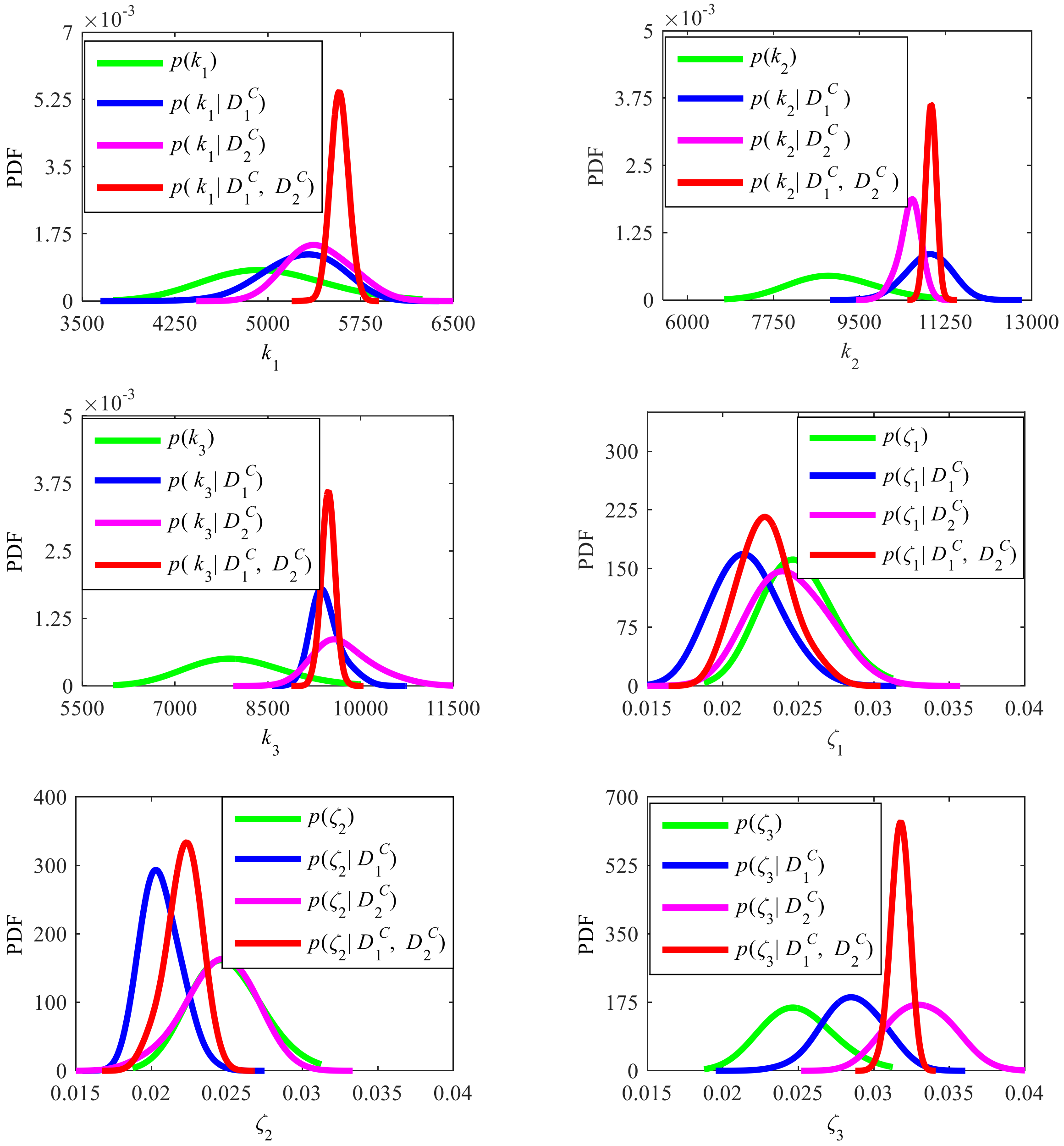


Figure 4.3 Posterior distributions of model parameters

In order to reduce the computational effort, Gaussian process (GP) surrogate models are established to replace the computational models for all the output quantities. The surrogate model uncertainty introduced by the GP models is incorporated in model calibration and validation. The calibration results of k_i and ζ_i using the calibration data of the six output quantities at different levels are shown in Figure 4.3, including all the PDFs needed in Eq. (4.11). As more data are used in the calibration, the uncertainty of the model parameters will decline. Thus Figure 4.3 shows that the posterior distributions using the data at both levels always have less uncertainty than those using data at a single level. The difference between the posterior distributions within each sub-figure also indicates that the posterior distribution is a best-fitting result in the sense of representing that particular data-set, but we do not yet know how to combine these alternatives in the subsequent prediction. This is answered by model validation and relevance analysis.

Next model validation is performed using the stochastic model reliability metric with the multivariate output. The tolerance for each quantity is chosen to be 15% of the validation data. Level 2 is expected to have lower model reliability value for two main factors:

1. The discretization error at Level 2 due to a limited number of finite elements for the beam (41 in this example). But this factor is not effective here since the data at Level 2 are synthetic data generated using the computational model, meaning that the difference between the computational model and the physics model is ignored. This factor will come into play if experimental data instead of synthetic data are used.
2. The coupling between the beam and the damped mass-spring system brings stronger nonlinearity at Level 2. Under the same number of training points, the GP surrogate model at Level 2 has more surrogate uncertainty (larger GP model prediction variance) than the GP surrogate model at Level 1. This factor is included in the numerical example.

The model reliability values given by the validation data from each validation test are listed in Table 4.3, which indicate lower model reliability at Level 2. In Figure 4.4, these values are used to construct the distributions of model reliability at Level 1 and Level 2 using the method of moments.

However, even though the model at Level 1 has higher model reliability than the model at Level 2, Level 2 is closer to the system level of interest since they have the same configuration. Therefore relevance analysis also needs to be considered.

Table 4.3 Model reliability values

Validation Test	1	2	3	4	5
Model reliability at Level 1	0.9702	0.9580	0.9398	0.9828	0.9800
Model reliability at Level 2	0.9616	0.8564	0.9208	0.9796	0.7904

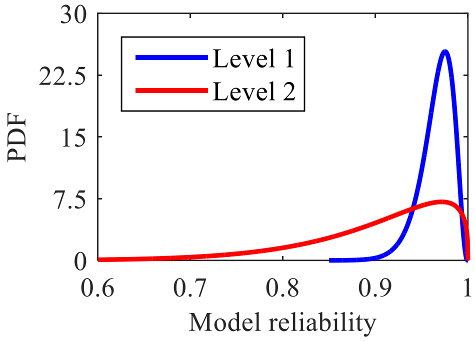


Figure 4.4 Distribution of model reliability

The relevance index of each lower level to the system level is computed using the iterative algorithm in Section 4.4. The initial values of relevance indices for both lower levels are set as 1. The algorithm converges after three iterations for Level 1, and after five iterations for Level 2. The results are: $P(S_1) = 0.5785, P(S_2) = 0.8971$. This result means that Level 2 is more relevant to the system level, which is consistent with our intuition since Level 2 has the same structural configuration as the system and differs only in the load input (sinusoidal vs. random process). Compared with the result of model validation, Level 2 has a lower value of model reliability but higher relevance index.

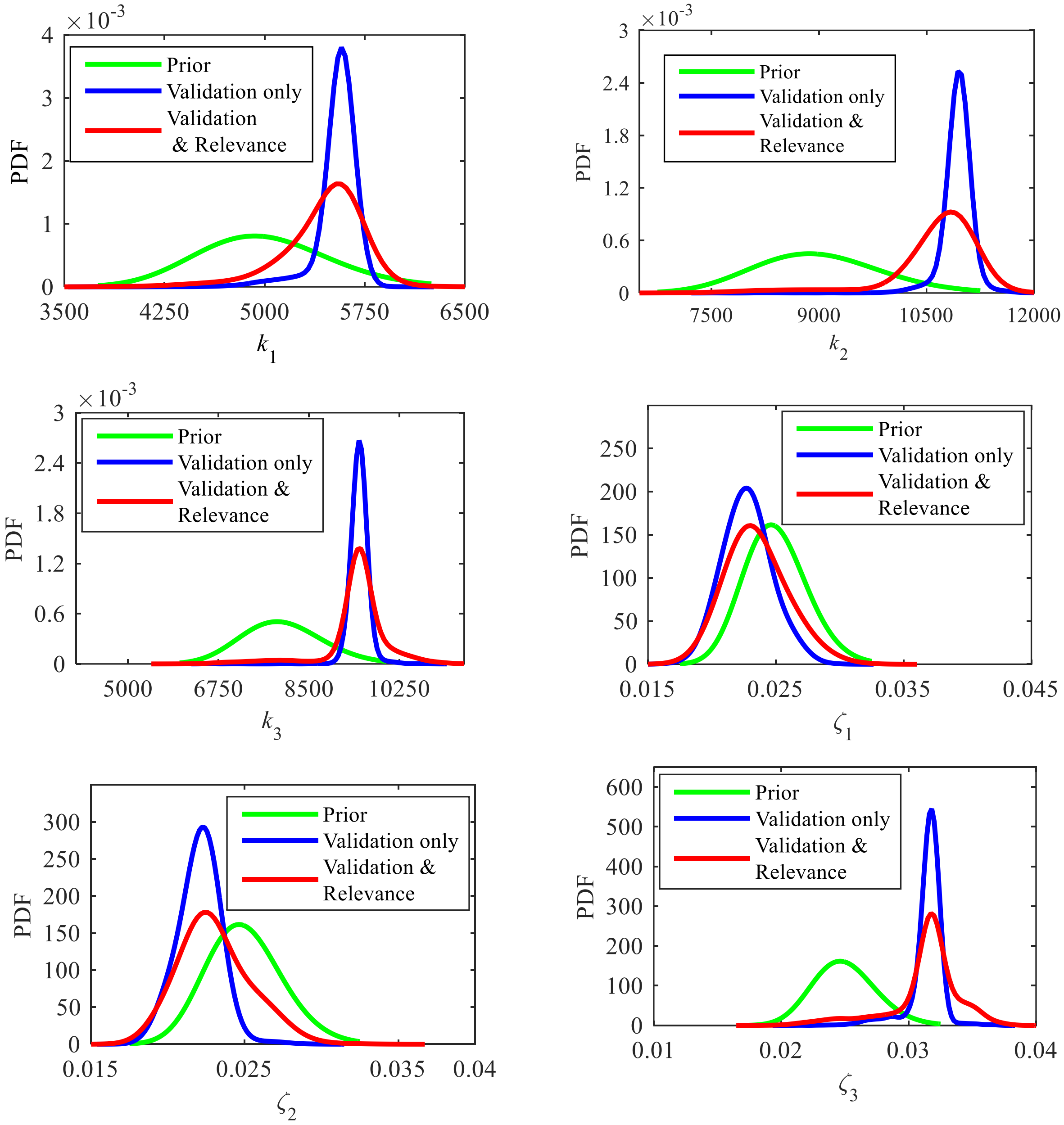


Figure 4.5 Integrated distributions of model parameters

Based on all the information from calibration, validation and relevance analyses, the integrated distributions of all six model parameters are constructed in Figure 4.5 using Eqs. (4.11) and (4.12). Figure 4.5 also shows the result by considering validation only (no relevance) using the previous rollup method in Eq. (4.10) but extended for stochastic model reliability metric. It is shown that the proposed roll-up method is more conservative than the previous one since we add one more criterion of relevance during the generation of samples from the posterior distribution.

The system output is predicted by propagating the integrated distribution of model parameters through the computational model at the system level. Figure 4.6 gives not only the prediction using the data of all six quantities but also the prediction by other combinations of quantities whose names are shown in the legend. The mean values and variances of the predictions are shown in Table 4.4. As more quantities are employed, the mean value of prediction decreases from 712 to 656; and the variance shows an overall decreasing tendency, but not monotonic (the variance increases slightly when the number of outputs considered rises from 2 to 3, and from 4 to 5).

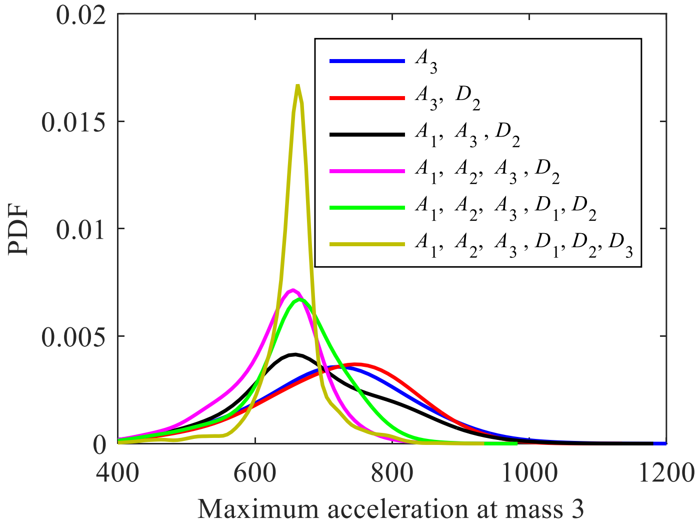


Figure 4.6 System output prediction

Table 4.4. Mean values and variances of predictions

Number of quantities	1	2	3	4	5	6
Mean values	710	713	690	632	655	656
Variance	12202	10499	10959	4868	5432	2301

4.7 Summary

This research developed a methodology to quantify the uncertainty in the system level output in a multi-level problem if experimental data are available only at lower levels and no data is available at the system level. The particular focus of this research was to determine the appropriate distribution for model parameters $\boldsymbol{\theta}_m$ to be used in system level prediction, using calibration, validation, and sensitivity analyses at lower levels.

Note that the focus is not on improving the precision of calibration, but on including as much information as possible. The lower level models have different physical configurations and/or excitation compared to the system level prediction model (e.g., 3-mass-spring vs. 3-mass-spring-on-beam and sinusoidal inputs vs. random process inputs), and no calibration data is available corresponding to the system level configuration. Thus the proposed approach results in increasing the uncertainty of the posteriors because the lower-level models do not have 100% reliability or 100% relevance to the system level.

The quantification of relevance is an important contribution to uncertainty integration. The relevance index quantifies the extent to which the lower level model reflects the physics captured in the system level model, and contributes to the weight of each posterior distribution in the uncertainty integration. In the proposed method, the relevance index is computed using the Sobol' indices, and defined as the square of the cosine of the angle between two sensitivity vectors. As mentioned in Section 4.4, this approach does not provide a comprehensive comparison of the actual physics at different levels but seeks to include the indication of physics given by variance-based sensitivity analysis, based on the prediction models at different levels.

For model validation, the proposed stochastic model reliability metric solves the problem of properly integrating results from multiple validation experiments. This research also extends the model reliability metric to deal with multivariate data, i.e., measurements of multiple output quantities.

The third contribution of this research is the development of the roll-up formula (Eqs. (4.11) and (4.12)) to integrate the information from three sources: 1) posterior distribution of model parameters by model calibration; 2) stochastic model reliability in model validation; 3) and relevance index of each lower level to the system level. The steps to realize this integration numerically are also developed.

In conclusion, model calibration obtains posterior distributions of each parameter within and across different lower levels; model validation evaluates the model reliability at each lower level separately, and the relevance analysis reveals the relationship between each lower level and the system level. All the above activities provide information to obtain the integrated distribution of model parameters. Using all this information, the system level output is predicted by propagating the integrated distributions of model parameters through the computational model at the system level.

CHAPTER 5

USE OF GLOBAL SENSITIVITY ANALYSIS IN TEST RESOURCE ALLOCATION FOR ROBUST PREDICTIONS

5.1 Background

In engineering applications, it is often required to estimate the system response under untested conditions using available computational models and test data at different conditions. The computational model aims to describe the physics of the system and can be denoted as $Y = F(\boldsymbol{X}; \boldsymbol{\theta})$, where Y is the system response, and $\boldsymbol{X}$ is the set of model inputs, and $\boldsymbol{\theta}$ is the set of model parameters. Usually the inputs $\boldsymbol{X}$ for a test are measurable, and their natural variability across different tests is represented by a probability distribution $p(\boldsymbol{X})$. Note that this natural variability is irreducible (aleatory uncertainty). The model parameters $\boldsymbol{\theta}$ have fixed but unknown values in all tests on the same specimen. The uncertainty regarding the values of $\boldsymbol{\theta}$ is epistemic uncertainty due to lack of information, which can be reduced using test data. (In some problems, the model parameters could be input-dependent; this research does not consider such cases).

Two important questions in system model development are: 1) how to quantify and reduce the uncertainty in $\boldsymbol{\theta}$; and 2) how to validate the agreement of the computational model to the true physics or quantify their difference. Activities that answer these two questions respectively are model calibration and model validation. Various approaches to model calibration and validation have been studied in the literature. Consider for example model calibration using Bayesian inference. While some researchers directly use the computational model $Y = F(\boldsymbol{X}; \boldsymbol{\theta})$ and calibrate $\boldsymbol{\theta}$, others [32] use a model discrepancy term $\delta(\boldsymbol{X})$ to correct the computational model and calibrate both $\boldsymbol{\theta}$ and $\delta(\boldsymbol{X})$. Consider another example regarding the use of test data. Some researches treat all the data as calibration data and use the calibrated model parameters in predicting the system response [93,94]; others integrate the results of model calibration and model validation (each done with different sets of data) in predicting the system response [95–97].

Both model calibration and validation require test data. Due to the variability in test outcomes, two sets of test data of the same size may lead to two distinct system response predictions (after calibration and/or validation) even if the same computational model and the same framework of model calibration/validation are used. This raises the question as to how many tests of each type are necessary to "optimize" the resultant system response prediction under limited test budget. The focus of this research is to develop an optimization approach to answer this question, assuming the computational model and the framework of model calibration/validation are given. The design variables of this optimization are the numbers of each type of test, denoted as $\boldsymbol{N} \in \mathbb{N}^q$ if q types of tests are available; the objective function and constraints will be discussed later. Note that 1) this optimization needs to be solved before any actual test is conducted [95]; and 2) this optimization needs to consider test outcome uncertainty due to which the subsequent system response prediction is also uncertain.

The actual physical test data from a certain type of test are obtained by 1) selecting the values of inputs $\boldsymbol{X}$; 2) propagating $\boldsymbol{X}$ through the physical test configuration where the model parameters $\boldsymbol{\theta}$ are at their true but unknown values; and 3) recording the input-output data, where both the input and output may have measurement errors. In actual tests where the values of $\boldsymbol{X}$ have been decided, the test outcome uncertainty arises from the measurement errors. However, the data considered in test resource allocation analysis always has to be synthetic since it is done before any actual test. The generation of synthetic data is a simulation of the three steps above, with the physical test configuration replaced by a computational model and the model parameters being unknown. Thus two additional uncertainty sources are introduced in the synthetic data: 1) uncertainty regarding the value of $\boldsymbol{\theta}$; and 2) model discrepancy, i.e., the difference between the computational model and the actual physics. In a Bayesian framework, the first one can be represented by the prior distribution of $\boldsymbol{\theta}$ based on available knowledge. No information on model discrepancy is available before any actual test.

Starting from the synthetic data generation explained above, several approaches for test resource allocation have been studied in the literature [95,98–102], and the main difference between these approaches is the choice of the objective function. Note that model calibration aims to reduce the uncertainty in model parameters, and thus reduce the uncertainty in the subsequent prediction. Thus in the case that only model calibration is considered in system response prediction, generally the objective of test resource allocation optimization is to minimize the prediction uncertainty subject to limited budget. Several quantities have been used to represent prediction uncertainty, and the first one is variance. Sankararaman et al. [95] minimized $E(V(Y))$ where $V(Y)$ is the variance of the prediction at given numbers of each type of test, and $E(\cdot)$ denotes the average of $V(Y)$ over different synthetic data sets. Similarly, Vanlier et al. [99] defined the variance reduction via model calibration as $1 - E(\sigma_{new}^2/\sigma_{old}^2)$ and maximized it, where σ_{new}^2 is the variance of the prediction using the posterior distribution and σ_{old}^2 is the variance of the prediction using the prior distribution. Entropy measures have also been used to represent prediction uncertainty. In [100], the authors maximized the relative entropy (Kullback–Leibler divergence) from the prediction $p'(y)$ using the prior distribution and the prediction $p''(y)$ using the posterior distribution; while in [101,102], the authors maximized the mutual information, i.e., the change of entropy from $p'(y)$ to $p''(y)$.

The above approaches that directly minimize the uncertainty in the prediction are not applicable when model validation is also incorporated in the system response prediction. The reason is that model validation may indicate that the calibrated model is not exactly valid; accounting for this result increases the uncertainty in the prediction. Thus the earlier approaches tend to conclude that model validation is not necessary. Mullins et al. [48] proposed a method considering both model calibration and model validation, in which model calibration is via Bayesian inference, and model validation is via a stochastic model reliability metric describing model validity through a probability distribution. In this method, the objective regarding model validation tests is to minimize the spread in the family of predictions that results from the uncertainty in model validity, denoted as $E\{V[E(Y)]\}$ where the inner $E(Y)$ is the prediction mean at given synthetic data set and given value of model validity, and $V[\cdot]$ is the average over the distribution of model validity, and the outer $E\{\cdot\}$ is the average over the different data sets. The objective regarding model calibration tests is still to minimize the variance of the prediction, denoted as $E\{E[V(Y)]\}$ where $V(Y)$ is the prediction variance based on a given synthetic data set and given value of model validity; the inner $E[\cdot]$ is the average over the distribution of model validity, and the outer $E\{\cdot\}$ is the average over different synthetic data sets.

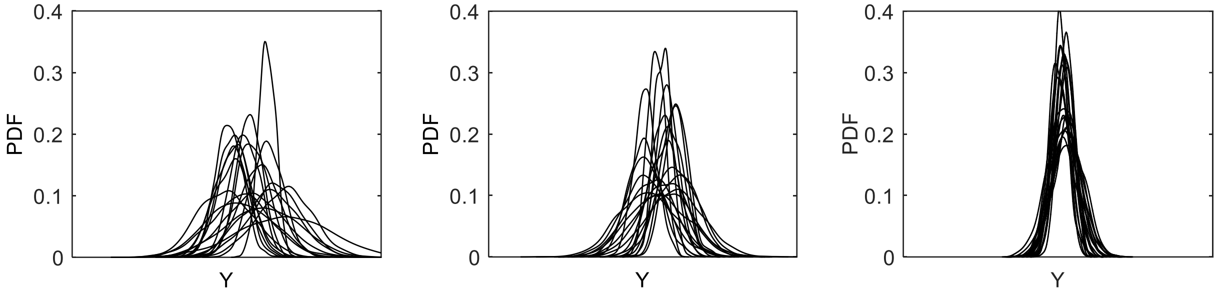
In this section, the proposed concept of “robust test resource allocation” means that the system response prediction is non-sensitive to the variability in test outcomes; so that at the optimal value of the design variables $\boldsymbol{N} \in \mathbb{N}^q$, different test outcomes result in consistent predictions. This concept and the required objective function will be explained in Section 5.2. The approach is suitable in different situations when only model calibration is considered or when both model calibration and model validation are considered, as shown in the numerical examples in Section 5.5.

The constraint in the optimization of test resource allocation is generally the budget. Note that the constraint and objective are interchangeable, i.e., the optimization may have two alternative formats: 1) subject to the budget constraint, optimize the design variable $\boldsymbol{N} \in \mathbb{N}^q$ (the number of each type of test) to reach the most robust prediction; or 2) subject to the robustness requirement in the prediction, find $\boldsymbol{N}$ to minimize the budget. The proposed approach can be used with either formulation.

In sum, the objectives of this research are to 1) find the optimal number of each type of test such that different data sets result in consistent system response predictions; 2) develop solutions for both formats of the optimization problem; and 3) adapt to different cases when only model calibration is considered or when both model calibration and model validation are considered. The rest of this research is organized as follows. Section 5.2 proposes the objective in the optimization of robust test allocation. Section 5.3 analyzes the uncertainty sources in the synthetic data and the use of Sobol' indices to assess their contributions towards the uncertainty in the prediction. Section 5.4 develops a flexible approach for test resource allocation optimization. Section 5.5 uses two numerical examples to illustrate the proposed approach.

5.2 Objective of Robust Test Resource Allocation

The objective of the proposed test resource allocation optimization can be visually represented as in Figure 5.1, which shows the families of the prediction PDFs at different values of the design variables $\boldsymbol{N}$. Within a sub-figure, the variation between the PDFs is caused by the test outcome variability among different data sets. From Figure 5.1(a) to Figure 5.1(c) this variation becomes smaller and the predictions reveal stronger consistency due to: 1) the decreased variability of mean values $E(Y)$ across the PDFs, meaning that the centroids of the family members are closer; and 2) the decreased variability of the variance $V(Y)$ across the PDFs, meaning that the ranges of values covered by the PDF are similar. In other words, at the value of optimal $\boldsymbol{N}$ in Figure 5.1(c), the effects of test outcome uncertainty on $E(Y)$ and $V(Y)$ are small so that consistent response predictions can be obtained with different sets of test data.



(a) **(b)** **(c)**

Figure 5.1 System response prediction: non-robust to robust

Therefore, this research defines the objective for robust test resource allocation as: minimize the contribution of test outcome uncertainty towards the variability (i.e., scatter) in the prediction mean value $E(Y)$ and the prediction variance $V(Y)$.

Global sensitivity analysis using Sobol' indices is a prominent approach to quantify the contributions of input uncertainty towards the uncertainty in the output. A brief introduction to Sobol' indices has been given in Section 2.6. The remaining challenge is to establish a deterministic function required by the Sobol' indices to map the test outcome uncertainty to the prediction uncertainty. This challenge will be analyzed and resolved in Section 5.3.

5.3 Uncertainty Sources in Test Outcomes

Recall that all the data considered in test resource allocation analysis has to be synthetic since the analysis is done before any actual test. The uncertainty in the synthetic data depends on specific test conditions, including 1) the possible values of inputs $\boldsymbol{X}$; 2) the number of test types; and 3) whether a single test specimen or multiple specimens are used for each type of test. Regarding the first condition, this research assumes that the testing personnel will provide the range of the possible values of the test inputs. In the absence of any other information, the range may be represented by a uniform distribution, thus for a single model input $X \in \boldsymbol{X}$ we have $X \sim U(L_X, U_X)$ where L_X is the lower bound and U_X is the upper bound. Other types of distributions can also be used to represent the possible values of model inputs if additional information is available.

This section will analyze the uncertainty sources in the synthetic data regarding the second and third conditions; the corresponding deterministic function required by the Sobol' indices also varies correspondingly. The rest of this section starts with the simplest case of one type of test and single specimen and subsequently extends it to multiple types of tests and multiple test specimens.

5.3.1 Single Type of Test and Single Test Specimen

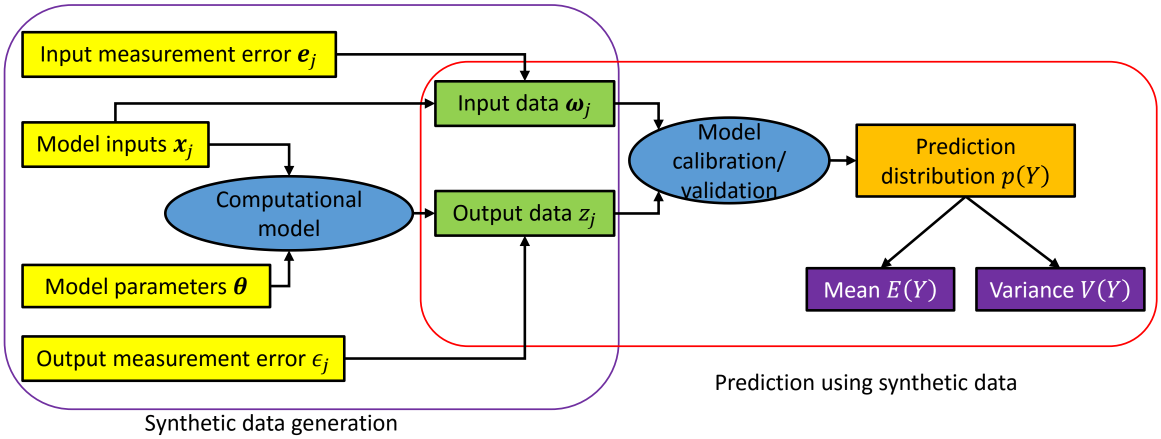

Input measurement error $\boldsymbol{e}_j$
Model inputs $\boldsymbol{x}_j$
Computational model
Model parameters $\boldsymbol{\theta}$
Output measurement error ϵ_j
Input data $\boldsymbol{\omega}_j$
Output data z_j
Model calibration/ validation
Prediction distribution $p(Y)$
Mean $E(Y)$
Variance $V(Y)$
Prediction using synthetic data
Synthetic data generation

Figure 5.2 Synthetic data: single type of test and single specimen

If only one type of test is available and all tests are conducted on a single specimen, the actual test data is a set of N data points obtained from the same specimen. Figure 5.2 shows the generation and usage of the synthetic data in this case. As shown in the left part of Figure 5.2, to generate a data set of N synthetic data points, four steps should be followed: 1) select and fix the values of $\boldsymbol{\theta} \in \mathbb{R}^{d_{\boldsymbol{\theta}}}$, where $d_{\boldsymbol{\theta}}$ is the dimension of model parameters; 2) generate N samples of model inputs $\boldsymbol{x}_j \in \mathbb{R}^{d_X}(j = 1 \text{ to } N)$ where $d_{\boldsymbol{X}}$ is the dimension of model inputs; and 3) propagate $\boldsymbol{x}_j(j =$

1 to N) and $\boldsymbol{\theta}$ through the computational model $F(\cdot)$; and 4) record the model input and output with measurement errors added. The resultant data set contains pairwise data points $\{\boldsymbol{\omega}_j, z_j\}(j = 1 \text{ to } N)$ as

$$\begin{gathered}\boldsymbol{\omega}_j = \boldsymbol{x}_j + \boldsymbol{e}_j \\ z_j = F(\boldsymbol{x}_j, \boldsymbol{\theta}) + \epsilon_j\end{gathered} \tag{5.1}$$

where $\boldsymbol{e}_j \in \mathbb{R}^{dx}$ is the model input measurement error and $\epsilon_j \in \mathbb{R}$ is the model output measurement error. If the model input measurement error is ignored, then $\boldsymbol{\omega}_j = \boldsymbol{x}_j$.

A crucial point in the generation of synthetic data is regarding the model parameters $\boldsymbol{\theta}$. For a single specimen, $\boldsymbol{\theta}$ have true but unknown values, meaning that the uncertainty in $\boldsymbol{\theta}$ is epistemic. Thus the uncertainty brought by $\boldsymbol{\theta}$ is the uncertainty in selecting the values of $\boldsymbol{\theta}$ before generating a synthetic data set; once selected, the values of $\boldsymbol{\theta}$ are fixed within the synthetic data set. This uncertainty in $\boldsymbol{\theta}$ only exist in the synthetic data; actual tests will fix the value of $\boldsymbol{\theta}$ at their true values.

The four steps above indicate three uncertainty sources in generating a pairwise synthetic data point $\{\boldsymbol{\omega}_j, z_j\}$, including:

1. Uncertainty regarding the values of model parameters $\boldsymbol{\theta}$, which can be represented by their prior distribution $p'(\boldsymbol{\theta})$ based on available knowledge before conducting any physical test. This uncertainty is epistemic since $\boldsymbol{\theta}$ have unknown but fixed true values.
2. Uncertainty regarding the possible values of inputs $\boldsymbol{x}_j$ to be used in the tests. As mentioned earlier, this uncertainty can be represented by uniform distribution $X \sim U(L_X, U_X)$ for $X \in \boldsymbol{X}$. This uncertainty is also epistemic since the values of $\boldsymbol{X}$ are unknown during test selection analysis, but will be decided by the test personnel in actual tests.

3. Uncertainty regarding input measurement errors $\boldsymbol{e}_j$ and output measurement error ϵ_j. Usually measurement error is assumed to have a zero mean Gaussian distribution thus $\boldsymbol{e}_j \sim N(\mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{X}})$ and $\epsilon_j \sim N(0, \sigma^2)$. The uncertainty in $\boldsymbol{e}_j$ and ϵ_j is aleatory if the values of $\boldsymbol{\Sigma}_{\boldsymbol{X}}$ and σ are known; but additional epistemic uncertainty regarding $\boldsymbol{\Sigma}_{\boldsymbol{X}}$ and σ will be introduced if their values are unknown.

In sum, Figure 5.2 shows that for a given number of tests, the synthetic data set $\{\boldsymbol{\omega}_j, z_j\}(j = 1 \text{ to } N)$ is uniquely determined once $\boldsymbol{\theta}$, $\boldsymbol{x}_j$, $\boldsymbol{e}_j$ and ϵ_j $(j = 1 \text{ to } N)$ are determined. Then for a given framework of model calibration/validation, the subsequent prediction distribution $\pi_Y(y)$ and its mean value $E(Y)$ and variance $V(Y)$ are also uniquely determined. Thus the deterministic functions suitable for computing Sobol' indices are

$$\begin{aligned} E(Y) &= E\big(G(\boldsymbol{\theta}, \boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_N)\big) \\ V(Y) &= V\big(G(\boldsymbol{\theta}, \boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_N)\big) \end{aligned} \tag{5.2}$$

where $\boldsymbol{\alpha}_j = \{\boldsymbol{x}_j, \boldsymbol{e}_j, \epsilon_j\} \in \mathbb{R}^{2dx+1}$ for $j = 1$ to N representing the uncertainty sources in generating a single pairwise data point $\{\boldsymbol{\omega}_j, z_j\}$, and N is the number of pairwise data points; $G(\cdot)$ represents the entire process shown in Figure 5.2, including both synthetic data generation and model calibration/validation analyses before predicting the system response. A model calibration/validation framework considering only model calibration is considered in Section 5.5.1; another framework incorporating both model calibration and model validation is considered in Section 5.5.2.

In Eq. (5.2), the uncertainty in $\{\boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_N\}$ represents the variability in the actual test outcomes; while the epistemic uncertainty in $\boldsymbol{\theta}$ only exist in the synthetic data, not in actual test data. To minimize the sensitivity of the prediction to the variability in the test outcomes, we need to minimize the sensitivity index of $\{\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_N\}$ in Eq. (5.2) so that $E(Y)$ and $V(Y)$ are non-sensitive to the variability in test outcomes and consistent prediction distributions can be achieved under different actual test outcomes. However, this minimization requires the sensitivity index closer to zero while numerical accuracy is always a challenge for small sensitivity indices.

Instead, this research chooses to maximize the sensitivity index of $\boldsymbol{\theta}$. If that is achieved, the epistemic uncertainty in $\boldsymbol{\theta}$ will be dominant towards the uncertainty in the prediction mean $E(Y)$ and the prediction variance $V(Y)$ (based on synthetic data). In the system response prediction using actual test data where $\boldsymbol{\theta}$ are fixed at their true values, the most dominant uncertainty contribution to $E(Y)$ and $V(Y)$ will be removed. Therefore the uncertainty in $E(Y)$ and $V(Y)$ caused by test outcome uncertainty will reduce significantly and consistent prediction distributions can be achieved under different actual test outcomes. In sum, the basic idea of the proposed approach is to maximize the contribution of epistemic uncertainty regarding model parameters in the synthetic data.

Note that the proposed approach guarantees consistent predictions regardless of what the true values of $\boldsymbol{\theta}$ are, since the Sobol' index is a global sensitivity analysis method and considers the entire distribution of $\boldsymbol{\theta}$.

5.3.2 Single Type of Test and Multiple Test Specimens

For a single type of test, multiple test specimens are required if the test is destructive so that each specimen can be used only once. Two examples of destructive tests are fatigue test and tensile strength test. The true value of a model parameter $\theta_l \in \boldsymbol{\theta}$ for $l = 1$ to $d_{\boldsymbol{\theta}}$ is fixed for a single specimen, but varies across different specimens. This variability of $\boldsymbol{\theta}$ may be represented by a probability distribution $p(\theta_l|\boldsymbol{P}_{\theta_l})$ where $\boldsymbol{P}_{\theta_l}$ are the distribution parameters of θ_l. For example, $\boldsymbol{P}_{\theta_l}$ are the mean and variance if θ_l has a Gaussian distribution. In addition, the entire set of distribution parameters for all components of $\boldsymbol{\theta}$ are denoted as $\boldsymbol{P_\theta}$ where $\boldsymbol{P}_{\theta_l} \in \boldsymbol{P_\theta}$ for $l =$ 1 to $d_{\boldsymbol{\theta}}$. In this case, $\boldsymbol{P_\theta}$ have unknown true values thus the uncertainty in $\boldsymbol{P_\theta}$ is epistemic; and this uncertainty can be represented by a prior distribution $p(\boldsymbol{P_\theta})$ based on available knowledge. Thus model calibration aims to quantify the uncertainty in $\boldsymbol{P_\theta}$, instead of $\boldsymbol{\theta}$. (Note that $\boldsymbol{\theta}$ have both aleatory and epistemic uncertainty, whereas the uncertainty in $\boldsymbol{P_\theta}$ is epistemic).

In the case of single type of test and multiple test specimens, the steps in generation and usage of the synthetic data set of N data points are similar to those in Figure 5.2, but the box "Model parameters $\boldsymbol{\theta}$" should be replaced by "$\boldsymbol{P_\theta} \rightarrow \boldsymbol{\theta}_j$", where $\boldsymbol{\theta}_j$ is the value of $\boldsymbol{\theta}$ generated for the j-th specimen (i.e., the j-th test). Compared to Figure 5.2, the values of $\boldsymbol{P_\theta}$ are now selected *before* generating a synthetic data set; once selected, the values of $\boldsymbol{P_\theta}$ are fixed within the synthetic data set. The values of model parameters $\boldsymbol{\theta}_j (j = 1 \text{ to } N)$for each of the N specimens are generated from the conditional distribution $p(\theta_l|\boldsymbol{P}_{\theta_l})$ for $l\ = 1$ to $d_{\boldsymbol{\theta}}$.

It seems natural to replace $\boldsymbol{\theta}$ in Eq. (5.2) with $\boldsymbol{P_\theta}$ and build new functions for the Sobol' indices computation. However, the new functions will not be deterministic functions as required by the Sobol' indices. A specific realization of $\boldsymbol{P_\theta}$ does not determine the values of $\boldsymbol{\theta}$ but only the distribution $p(\theta_l|\boldsymbol{P}_{\theta_l})$ for $l\ = 1$ to $d_{\boldsymbol{\theta}}$; thus $\boldsymbol{\theta}$ are still stochastic at given $\boldsymbol{P_\theta}$. Only deterministic values of $\boldsymbol{\theta}$ and $\alpha_i = \{\boldsymbol{x}_j, \boldsymbol{e}_j, \epsilon_j\}$ $(j = 1 \text{ to } N)$ can decide the subsequent prediction distribution $p(Y)$ and its mean value $E(Y)$ and variance $V(Y)$. In sum, an approach to establish a deterministic relationship from $\boldsymbol{P_\theta}$ to $\boldsymbol{\theta}$ is needed.

This required deterministic relationship can be provided by the auxiliary variable method developed in Ref. [59,65,86]. This method introduces an auxiliary variable U_{θ_l}, which is the CDF value of $p(\theta_l|\boldsymbol{P}_{\theta_l})$, and builds the needed deterministic relationship using the probability integral transform as:

$$\theta_l = \mathcal{P}^{-1}_{\theta_l|\boldsymbol{P}_{\theta_l}}\left(U_{\theta_l}\right) \tag{5.3}$$

where $\mathcal{P}^{-1}_{\theta_l|\boldsymbol{P}_{\theta_l}}(\cdot)$ is the inverse CDF (cumulative distribution function) of θ_l at given $\boldsymbol{P}_{\theta_l}$. Note that U_{θ_l} has the standard uniform distribution $U(0,1)$. Eq. (5.3) indicates three steps: 1) generate the values of $\boldsymbol{P}_{\theta_l}$ from their prior distribution to produce the conditional distribution $p(\theta_l|\boldsymbol{P}_{\theta_l})$; 2) generate the value of U_{θ_l} from $U(0,1)$; and 3) substitute U_{θ_l} into the inverse CDF $\mathcal{P}^{-1}_{\theta_l|\boldsymbol{P}_{\theta_l}}(\cdot)$ to obtain a unique value of θ_l.

The uncertainty in model parameter θ_l consists of two components: 1) the epistemic uncertainty in distribution parameters $\boldsymbol{P}_{\theta_l}$, represented by the prior distribution $p(\boldsymbol{P}_{\theta_l})$; and 2) the aleatory uncertainty in θ_l at given $\boldsymbol{P}_{\theta_l}$, represented by the conditional distribution $p(\theta_l|\boldsymbol{P}_{\theta_l})$. These two parts are coupled since $p(\theta_l|\boldsymbol{P}_{\theta_l})$ depends on the value of $\boldsymbol{P}_{\theta_l}$. The introduced auxiliary variable U_{θ_l} captures the aleatory uncertainty, and also helps to decouple the aleatory and epistemic uncertainties [65] since the distribution of $U_{\theta_l} \sim U(0,1)$ does not depend on $\boldsymbol{P}_{\theta_l}$.

With the introduction of the auxiliary variable, deterministic functions suitable for Sobol' indices computation can be established as

$$\begin{gathered} E(Y) = E\left(G(\boldsymbol{P}_{\boldsymbol{\theta}}, \boldsymbol{U}_{\boldsymbol{\theta}}, \boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_N)\right) \\ V(Y) = V\left(G(\boldsymbol{P}_{\boldsymbol{\theta}}, \boldsymbol{U}_{\boldsymbol{\theta}}, \boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_N)\right) \end{gathered} \tag{5.4}$$

where $\boldsymbol{\alpha}_j = \left\{\boldsymbol{x}_j, \boldsymbol{e}_j, \epsilon_j\right\}$ for $j = 1$ to N as in Eq. (5.2); $\boldsymbol{U}_{\boldsymbol{\theta}}$ contains all the auxiliary variables introduced for each θ_l, thus $U_{\theta_l} \in \boldsymbol{U}_{\boldsymbol{\theta}}$ for $l = 1$ to $d_{\boldsymbol{\theta}}$; $G(\cdot)$ represents the entire process of synthetic data generation and the framework of model calibration/validation (using the synthetic data) to predict the system response.

As explained earlier, the basic idea of the proposed approach is to maximize the contribution of the epistemic uncertainty of $\boldsymbol{\theta}$ in the synthetic data. Thus we need the contribution of $\boldsymbol{P_\theta}$ is dominant in the context of Eq. (5.4). If that is achieved, in the system response prediction using actual test data where $\boldsymbol{P_\theta}$ are fixed at their true values, the most dominant uncertainty contribution to $E(Y)$ and $V(Y)$ will be removed. Therefore the uncertainty in $E(Y)$ and $V(Y)$ caused by test outcome uncertainty will reduced significantly, and different actual test outcomes will lead to consistent predictions.

5.3.3 Multiple Types of Tests and Single Test Specimen

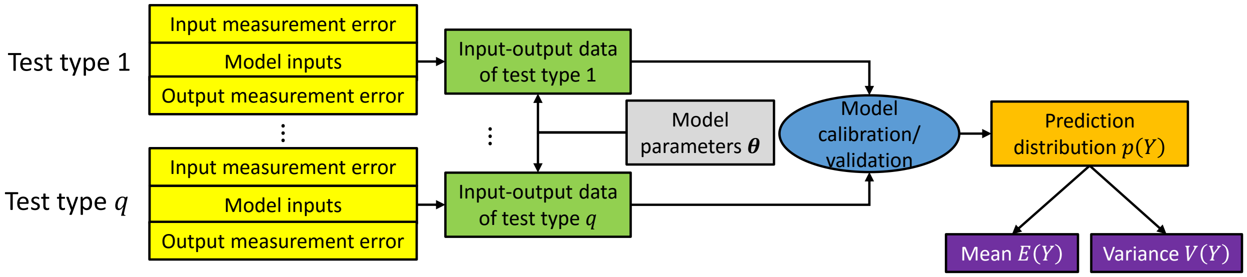


Figure 5.3 Synthetic data: q types of tests and single specimen for each type

In the case that q different types of tests are to be considered and each type utilizes only one specimen (non-destructive test), Figure 5.2 expands to Figure 5.3, and Eq. (5.2) expands to

$$\begin{aligned} E(Y) &= E\left(G(\boldsymbol{\theta}, \mathbf{A}_1, \dots, \mathbf{A}_q)\right) \\ V(Y) &= V\left(G(\boldsymbol{\theta}, \mathbf{A}_1, \dots, \mathbf{A}_q)\right) \end{aligned} \tag{5.5}$$

Eq. (5.5) gives the required deterministic functions for Sobol' indices computation. In Eq. (5.5), $\mathbf{A}_i = \{\boldsymbol{\alpha}_1^i, \dots, \boldsymbol{\alpha}_{N_i}^i\}$ for $i = 1$ to q represents the uncertainty regarding inputs and measurement errors in generating the synthetic data for the i-th type of test, where $\boldsymbol{\alpha}_j^i = \{\boldsymbol{x}_j^i, \boldsymbol{e}_j^i, \epsilon_j^i\}$ for $i = 1$ to q

and $j = 1$ to N_i; j represents the test number and N_i is the total number of the i-th type of test. Note that here $\boldsymbol{\theta}$ is the vector of the model parameters in all types of tests.

Similar to the earlier discussion, in the test resource allocation optimization regarding Eq. (5.5), we need the contribution of the epistemic uncertainty in $\boldsymbol{\theta}$ towards the uncertainty in $E(Y)$ and $V(Y)$ to be dominant.

5.3.4 Multiple Types of Tests and Multiple Test Specimens

The most complex case is that q different types of tests are to be considered and the i-th type of test utilizes N_i specimens corresponding to N_i tests. Similarly to Eq. (5.5), the epistemic uncertainty is regarding the unknown true values of distribution parameters $\boldsymbol{P_\theta}$; and an auxiliary variable is introduced for each model parameter in order to establish deterministic functions required by the Sobol' indices computation, as explained in Section 5.3.2. The resultant functions are:

$$\begin{aligned} E(Y) &= E\left(G\big(\boldsymbol{P_\theta}, \boldsymbol{U_\theta}, \mathbf{A}_1, \dots, \mathbf{A}_q\big)\right) \\ V(Y) &= V\left(G\big(\boldsymbol{P_\theta}, \boldsymbol{U_\theta}, \mathbf{A}_1, \dots, \mathbf{A}_q\big)\right) \end{aligned} \tag{5.6}$$

Similarly, in the test resource allocation optimization regarding Eq. (5.6), we need the contribution of the epistemic uncertainty in $\boldsymbol{P_\theta}$ towards the uncertainty in $E(Y)$ and $V(Y)$ to be dominant.

5.3.5 Selection of Sobol' Indices

So far deterministic functions for Sobol' indices computation in different test conditions have been established. Robust design of resource allocation can be achieved by maximizing the contribution of the epistemic uncertainty regarding either $\boldsymbol{\theta}$ (single specimen) or $\boldsymbol{P_\theta}$ (multiple specimen). This epistemic uncertainty is represented by a set of random variables ($\boldsymbol{\theta}$ in Eqs. (5.2) and (5.5); $\boldsymbol{P_{\theta}}$ in Eqs. (5.4) and (5.6)). The total effect sensitivity index considers the interactions between the subset of random variables and its complement; thus to be more comprehensive, the optimization in this research uses Eq. (2.15) to compute the total effect index for the subset of epistemic uncertainty (either $\boldsymbol{\theta}$ or $\boldsymbol{P_{\theta}}$). In the following sections, Sobol' index indicates the total effect index in Eq. (2.15). The computed Sobol' indices are denoted as $S_{\boldsymbol{m}}^{E(Y)}$ for $E(Y)$ and $S_{\boldsymbol{m}}^{V(Y)}$ for $V(Y)$. In the case of single specimen, $\boldsymbol{m} = \boldsymbol{\theta}$ so that $S_{\boldsymbol{m}}^{E(Y)}$ and $S_{\boldsymbol{m}}^{V(Y)}$ are the Sobol' indices of $\boldsymbol{\theta}$; in the case of multiple specimen, $\boldsymbol{m} = \boldsymbol{P_{\theta}}$ so that $S_{\boldsymbol{m}}^{E(Y)}$ and $S_{\boldsymbol{m}}^{V(Y)}$ are the Sobol' indices of $\boldsymbol{P_{\theta}}$.

5.4 Optimum Test Resource Allocation

5.4.1 Formulation

As discussed in Sections 5.1 and 5.2, the proposed robust test resource allocation means that the system response prediction is non-sensitive to the variability in test outcomes, so that consistent predictions of the system response under different sets of test data. This consistency can be obtained if the contribution of epistemic uncertainty in $\boldsymbol{\theta}$ or $\boldsymbol{P_{\theta}}$towards the uncertainty in $E(Y)$ and $V(Y)$ is dominant. That gives two objectives in the optimization: 1) maximize $S_{\boldsymbol{m}}^{E(Y)}$, the Sobol' index of $\boldsymbol{\theta}$ or $\boldsymbol{P_{\theta}}$ with respect to $E(Y)$; and 2) maximize $S_{\boldsymbol{m}}^{V(Y)}$, the Sobol' index of $\boldsymbol{\theta}$ or $\boldsymbol{P_{\theta}}$ with respect to $V(Y)$. Several methods are available to solve multi-objective problems. One simple method is to combine $S_{\boldsymbol{m}}^{E(Y)}$ and $S_{\boldsymbol{m}}^{V(Y)}$ through a weighted sum since they are both dimensionless and have the same scale [0,1]. This constitutes the first optimization formulation of robust test resource allocation:

$$\begin{aligned} &\text{Max} \quad p_1 S_{\boldsymbol{m}}^{E(Y)} + p_2 S_{\boldsymbol{m}}^{V(Y)} \\ &\text{s.t.} \quad \sum\nolimits_{i=1}^{q} C_i N_i \leq C_0 \text{ and } N_i \geq 0 \end{aligned} \tag{5.7}$$

where $C_i > 0$ is the unit cost of the i-th $(i = 1 \text{ to } q)$ type of test and N_i is the number of tests of the i-th type; and C_0 is the budget constraint; and p_1 and p_2 are use-defined positive constant weight coefficients.

Eq. (5.7) tries to reach the most optimal test design subject to the budget constraint. As explained in Section 5.1, another possible format of optimization is to minimize the budget subject to the sensitivity threshold. Thus the alternative optimization formulation for robust test resource allocation is

$$\begin{aligned} &\text{Min} \sum\nolimits_{i=1}^{q} C_i N_i \\ &\text{s.t.} \quad S_{\boldsymbol{m}}^{E(Y)} \geq \lambda^{E(Y)}, S_{\boldsymbol{m}}^{V(Y)} \geq \lambda^{V(Y)} \text{ and } N_i \geq 0 \end{aligned} \tag{5.8}$$

where $\lambda^{E(Y)}$ and $\lambda^{V(Y)}$ are the desired lower bounds of the Sobol' index for $E(Y)$ and $V(Y)$, respectively.

Eqs. (5.7) and (5.8) are both integer optimization problems since the design variables $N_i(i = 1 \text{ to } q)$ are integers. Sometimes integer optimization is solved using a relaxation approach [103], where the integer constraint is first relaxed, and the integers nearest to the resultant optimal solution are used as the solution of the original (unrelaxed) problem. Unfortunately, this approach is not applicable here because the synthetic data to be used in model calibration/validation can be generated only if $N_i(i = 1 \text{ to } q)$ are integers. It is not possible to generate test data for a non-integer number of tests.

5.4.2 Solution Algorithm

A simulated annealing algorithm [104] is used for the solution of Eqs. (5.7) and (5.8) because it can handle stochastic discrete optimization problems without requiring relaxation. For discrete optimization problems such as in Eqs. (5.7) and (5.8), this algorithm aims to minimize an objective function $f(\boldsymbol{s})$ where $\boldsymbol{s} = \{s_1, \ldots, s_L\}$ is a vector of integers and its feasible region is $\boldsymbol{\Omega}$. If the objective is to maximize $f(\boldsymbol{s})$ as shown in Eq. Eqs. (5.7), $-f(\boldsymbol{s})$ ought to be minimized.

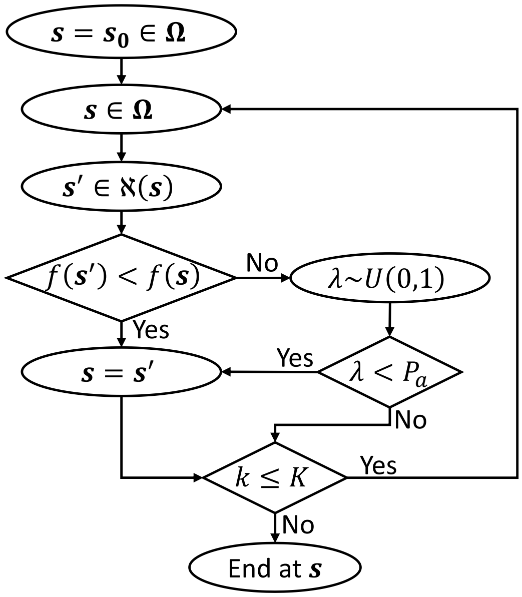

$\boldsymbol{s} = \boldsymbol{s_0} \in \boldsymbol{\Omega}$
$\boldsymbol{s} \in \boldsymbol{\Omega}$
$\boldsymbol{s}' \in \aleph(\boldsymbol{s})$
$f(\boldsymbol{s}') < f(\boldsymbol{s})$
No
$\lambda \sim U(0,1)$
Yes
Yes
$\lambda < P_a$
$\boldsymbol{s} = \boldsymbol{s}'$
No
$k \leq K$
Yes
No
End at $\boldsymbol{s}$

Figure 5.4 Simulated annealing algorithm

As shown in Figure 5.4, the simulated annealing algorithm starts from an initial value $\boldsymbol{s}_0 \in \boldsymbol{\Omega}$. If $\boldsymbol{s}$ is the optimal solution in an iteration, a new value $\boldsymbol{s}'$ will be randomly selected within the neighborhood of $\boldsymbol{s}$. This neighborhood, denoted as $\aleph(\boldsymbol{s})$, can be defined by different proposal density functions; and this research defines $\aleph(\boldsymbol{s}) = [s_1 \pm d_1, \ldots, s_l \pm d_L] \cap \boldsymbol{\Omega}$ where d_l is a user-defined positive integer for $l = 1$ to L. In one iteration, if $f(\boldsymbol{s}') < f(\boldsymbol{s})$ the new value $\boldsymbol{s}'$ is accepted as the new optimal solution; otherwise the probability to accept $\boldsymbol{s}'$ is

$$P_a = \exp\left(-\frac{f(\boldsymbol{s}') - f(\boldsymbol{s})}{T}\right) \tag{5.9}$$

where T is the parameter that governs how tight the acceptance criterion should be. Specifically, a random sample λ is generated from the standard uniform distribution $U(0,1)$, and $\boldsymbol{s}'$ is accepted if $\lambda < P_a$. The reason for accepting $\boldsymbol{s}'$ with a probability P_a even when it does not improve the objective function is to explore additional regions and reduce the opportunity to stop at a local minimum. As the algorithm proceeds, the threshold for acceptance becomes tighter, so only reductions and very small increases to the objective function can be accepted. This threshold tightening is governed by a reduction to the parameter T as

$$T = T_0\left(1 - \frac{k}{K}\right)^{\alpha} \tag{5.10}$$

where T_0 is the user-defined starting value of T, k is the current iteration number, K is the total number of iterations allowed, and α is a user-defined exponent that determines the rate of decrease of T. This iteration proceeds until the total allowed number of iterations K is expended.

5.5 Numerical Examples

This section uses two examples to illustrate the proposed method. The first example is a mathematical problem considering model calibration only, and the second example is a dynamics problem considering both model calibration and validation.

5.5.1 Mathematical Example

This sub-section presents a simple mathematical example to illustrate the proposed approach for robust resource allocation. In this example, the system output is the sum of two subsystem outputs, and each sub-system has separate model inputs and model parameters:

$$Y = W_1 + W_2,\, W_1 = X_1\theta_1,\, W_2 = X_2\theta_2 \tag{5.11}$$

The inputs X_1 and X_2 are assumed to be independent random variables; the uncertainty regarding their values in tests is represented by uniform distributions $X_1 \sim U(90{,}110), X_2 \sim U(40{,}60)$, based on ranges obtained from the test personnel.

Two types of tests are available. Test Type I measures W_1 with measurement error $\epsilon_1 \sim N(0{,}50^2)$; and test Type II measures W_2 with measurement error $\epsilon_2 \sim N(0{,}40^2)$. The resultant synthetic data are pairwise data $\{X_1, W_1\}$ and $\{X_2, W_2\}$, respectively. Assume that the unit cost of Type I test is 4 and the unit cost of Type II test is 1.

Two cases are considered in this example: single test specimen vs. multiple test specimens. In case 1 of single specimen, model parameter $\boldsymbol{\theta} = \{\theta_1, \theta_2\}$ have true but unknown values to be calibrated. In case 2 of multiple specimens, $\{\theta_1, \theta_2\}$ follow normal distributions $N\left(\mu_{\theta_1}, \sigma_{\theta_1}^2\right)$ and $N\left(\mu_{\theta_1}, \sigma_{\theta_1}^2\right)$ across specimens, and the parameters to be calibrated are $\boldsymbol{P_\theta} = \left\{\mu_{\theta_1}, \sigma_{\theta_1}, \mu_2, \sigma_{\theta_2}\right\}$.

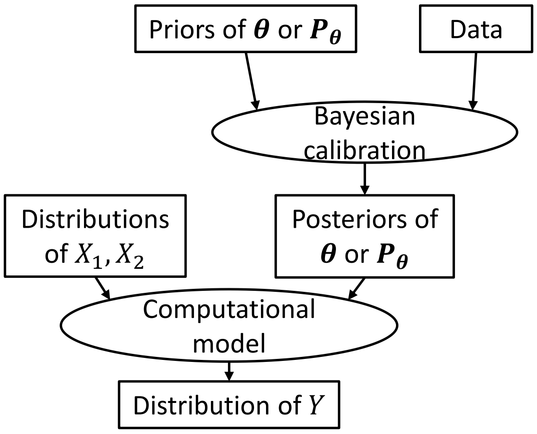

Priors of $\boldsymbol{\theta}$ or $\boldsymbol{P_\theta}$
Data
Bayesian calibration
Distributions of X_1, X_2
Posteriors of $\boldsymbol{\theta}$ or $\boldsymbol{P_\theta}$
Computational model
Distribution of Y

Figure 5.5 Prediction after model calibration with test data

The process to realize the system prediction Y, i.e., the framework of model calibration/validation with the synthetic data is shown in Figure 5.5, where the posterior distributions of calibration parameters together with the known distributions of X_1 and X_2 are propagated through the computational model in Eq. (5.11) to obtain the distribution of Y. Note that model validation is not considered in this example; only calibration is considered. The proposed test resource allocation approach can also handle model validation, as shown in the next numerical example.

Case 1: Single test specimen

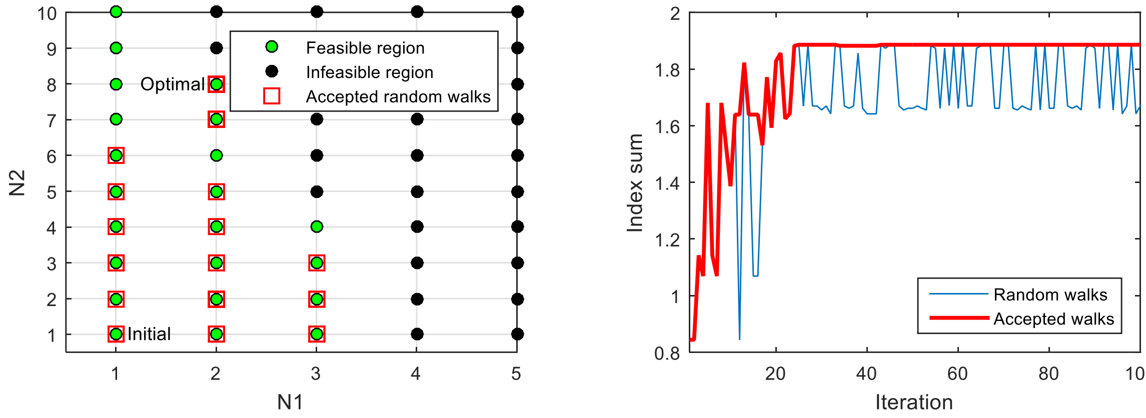
In this case, model parameters $\boldsymbol{\theta} = \{\theta_1, \theta_2\}$ have unknown deterministic values and prior distributions $\theta_1 \sim N(5,\ 0.5^2)$, $\theta_2 \sim N(10,\ 1^2)$ are assumed for them.

This case is applied to the two optimizations in Eqs. (5.7) and (5.8). For the optimization in Eq. (5.7), we set the total budget constraint at 16; thus Eq. (5.7) becomes (assuming equal weights $p_1 = p_2$)

$$\begin{gathered} \text{Max} \quad S_{\boldsymbol{\theta}}^{E(Y)} + S_{\boldsymbol{\theta}}^{V(Y)} \\ \text{s.t.} \quad 4N_1 + N_2 \leq 16 \text{ and } N_i \geq 0 \end{gathered} \tag{5.12}$$

where N_1 is the number of Type I tests and N_2 is the number of Type II tests. N_1 and N_2 are the decision variables, i.e., we need to decide the number of replications of each type of test.

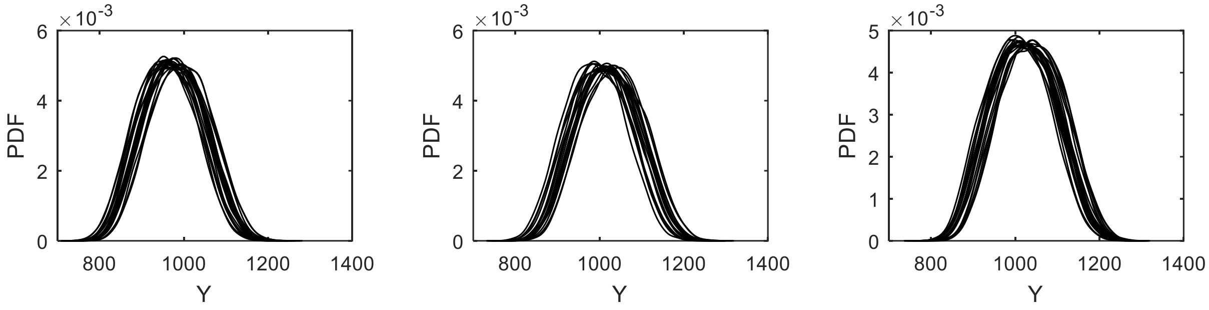
The simulated annealing algorithm is used to solve Eq. (5.12), and Figure 5.6 records the process of optimization. Figure 5.6(a) shows that the optimization starts at an initial design point $(N_1, N_2) = (1,1)$ and terminates at the optimal solution $(N_1, N_2) = (2,8)$. Figure 5.6(b) shows that only some of the random walks are accepted and the maximized Sobol' index sum $S_{\boldsymbol{\theta}}^{E(Y)} + S_{\boldsymbol{\theta}}^{V(Y)}$ is 1.89. The feasible region in Figure 5.6(a) covers the combinations of N_1 and N_2 such that $4N_1 + N_2 \leq 16$. Note that 1) this feasible region is obtained by extra computation; and 2) this feasible region is shown only to help in visualizing the result but is NOT needed in the optimization.



(a) History of accepted random walks **(b) History of the Sobol' indices sum $S_{\theta}^{E(Y)} + S_{\theta}^{V(Y)}$**

Figure 5.6 Optimization of the mathematical example based on Eq. (5.12)

As discussed in Section 5.3.1, since the robustness objective $S_{\boldsymbol{\theta}}^{E(Y)} + S_{\boldsymbol{\theta}}^{V(Y)}$ is maximized, the optimal solution $(N_1, N_2) = (2{,}8)$ for Eq. (5.12) should lead to consistent system response prediction regardless of the true values of $\boldsymbol{\theta}$. Three steps are pursued to verify it: 1) assume true values of $\boldsymbol{\theta}$; 2) generate multiple sets of synthetic data with the size of $(N_1, N_2) = (2{,}8)$; and 3) plot the family of prediction PDFs using the data sets in step 2 and observe whether they are consistent. Although the data are still synthetic, this is a simulation of the prediction using the actual test data since the model parameters $\boldsymbol{\theta}$ are fixed at the same value across different data sets; while in the synthetic data generation for test resource allocation shown in Figure 5.2, the model parameters are fixed within a single data set but vary across different data sets. The results of this verification are shown in Figure 5.7. Figure 5.7(a) indicates that $(N_1, N_2) = (2{,}8)$ leads to consistent predictions if the true values of model parameters are $\{\theta_1, \theta_2\} = \{4.9, 9.5\}$; similarly, Figure 5.7(b) and Figure 5.7(c) show that consistent predictions are also obtained if $\{\theta_1, \theta_2\} = \{5.4, 9.8\}$ or $\{\theta_1, \theta_2\} = \{5.0, 10.5\}$.



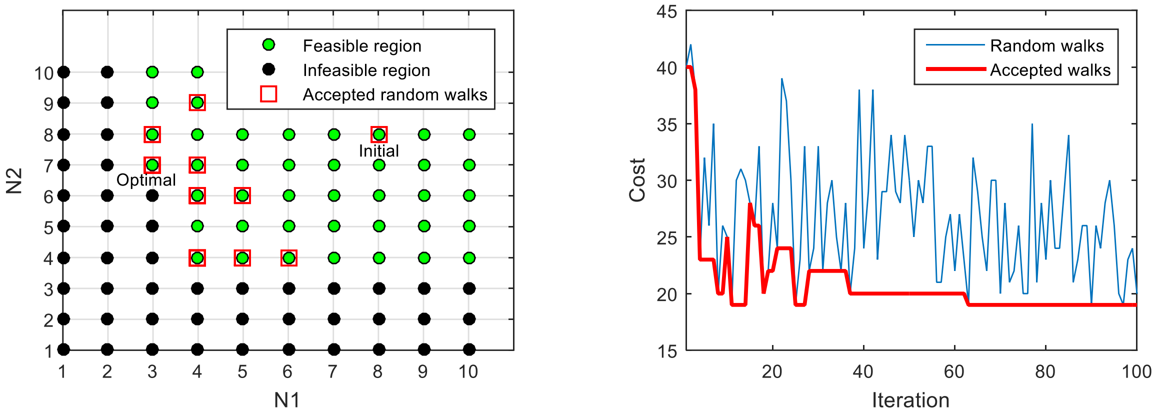
(a) $\theta_1 = 4.9, \theta_2 = 9.5$ **(b)** $\theta_1 = 5.4, \theta_2 = 9.8$ **(c)** $\theta_1 = 5.0, \theta_2 = 10.5$

Figure 5.7 Family of prediction PDFs at the solution of Eq. (5.12) of $(\boldsymbol{N_1}, \boldsymbol{N_2}) = (\mathbf{2}, \mathbf{8})$

For the optimization in Eq. (5.8), we set the Sobol' index lower bounds as $\lambda^{E(Y)} = \lambda^{V(Y)} = 0.95$; thus Eq. (5.8) becomes

$$\begin{gathered}\text{Min } 4N_1 + N_2 \\ \text{s.t. } \; S_{\boldsymbol{\theta}}^{E(Y)} \geq 0.95, S_{\boldsymbol{\theta}}^{V(Y)} \geq 0.95 \text{ and } N_i \geq 0\end{gathered} \tag{5.13}$$

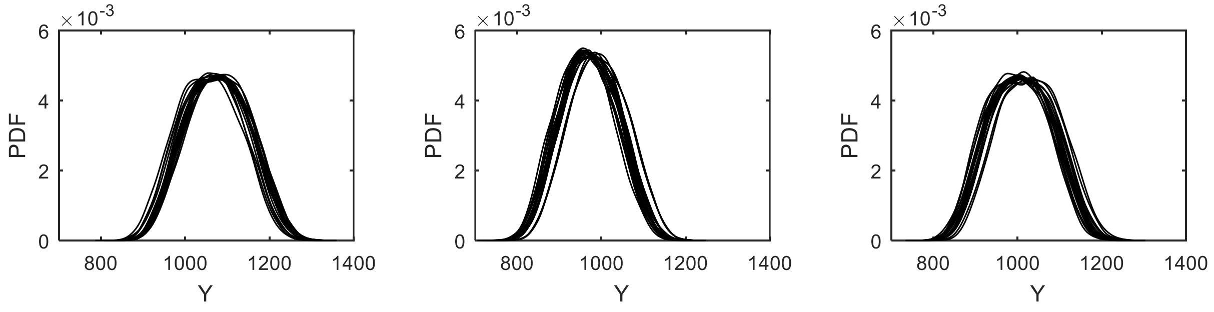
The simulated annealing algorithm is used to solve Eq. (5.13), and Figure 5.8 records the process of optimization. Figure 5.8(a) shows that the optimization starts at an initial design point $(N_1, N_2) = (8{,}8)$ and terminates at the optimal solution $(N_1, N_2) = (3{,}7)$. Figure 5.8(b) shows that only some of the random walks are accepted and the minimized cost is 19. The feasible region in Figure 5.8(a) covers the combinations of N_1 and N_2 such $S_{\boldsymbol{\theta}}^{E(Y)} \geq 0.95$ and $S_{\boldsymbol{\theta}}^{V(Y)} \geq 0.95$. Similar to Figure 5.6, note that 1) this feasible region is obtained by extra computation; and 2) this feasible region is shown only to help in visualizing the result but is NOT needed in the optimization.



(a) History of accepted random walks **(b) History of cost $4N_1 + N_2$**

Figure 5.8 Optimization of the mathematical example based on Eq. (5.13)

As discussed in Section 5.3.1, since the robustness constraints $S_{\boldsymbol{\theta}}^{E(Y)} \geq 0.95, S_{\boldsymbol{\theta}}^{V(Y)} \geq 0.95$ are satisfied, the optimal solution $(N_1, N_2) = (3{,}7)$ for Eq. (5.13) should lead to consistent system response prediction regardless of the true values of $\boldsymbol{\theta}$. The same three steps for Figure 5.7 are pursued to verify it. The results of this verification are shown in Figure 5.9. Figure 5.9(a) indicates that $(N_1, N_2) = (3{,}7)$ leads to consistent predictions if the true values of model parameters are $\{\theta_1, \theta_2\} = \{5.7, 10.5\}$; similarly, Figure 5.9(b) and Figure 5.9(c) show that consistent predictions are also obtained if $\{\theta_1, \theta_2\} = \{5.2, 9.1\}$ or $\{\theta_1, \theta_2\} = \{4.6, 10.8\}$.



(a) $\theta_1 = 5.7, \theta_2 = 10.5$ **(b)** $\theta_1 = 5.2, \theta_2 = 9.1$ **(c)** $\theta_1 = 4.6, \theta_2 = 10.8$

Figure 5.9 Family of prediction PDFs at the solution of Eq. (5.13) of $(\boldsymbol{N_1}, \boldsymbol{N_2}) = (\boldsymbol{3}, \boldsymbol{7})$

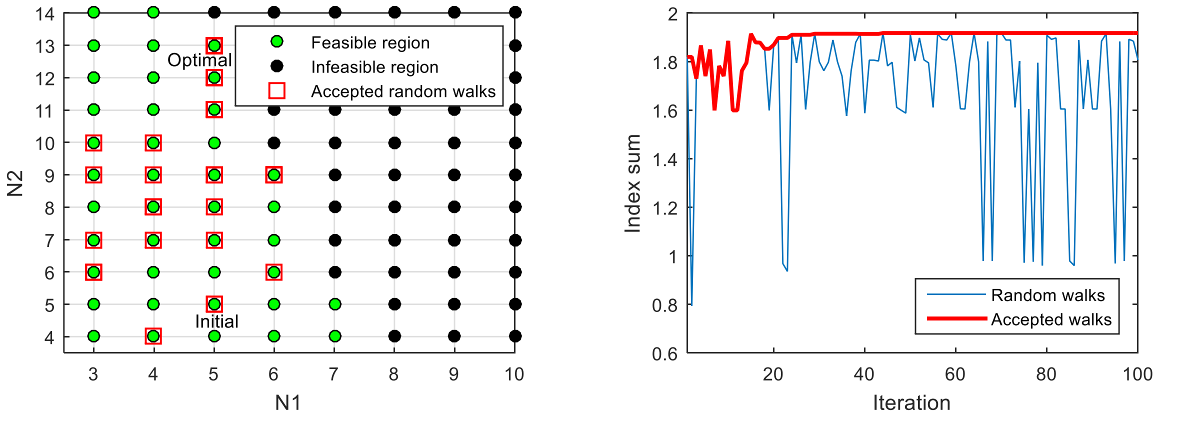
<u>Case 2: Multiple test specimens</u>

In this case, model parameters $\boldsymbol{P}_{\boldsymbol{\theta}} = \{\mu_{\theta_1}, \sigma_{\theta_1}, \mu_2, \sigma_{\theta_2}\}$ have unknown deterministic values and uniform prior distributions $\mu_{\theta_1} \sim U(4,6)$, $\sigma_{\theta_1} \sim U(0.2,1)$, $\mu_{\theta_2} \sim U(8,10)$, $\sigma_{\theta_2} \sim U(0.8,1.5)$ are assumed for them.

This case is also applied to the two optimizations in Eqs. (5.7) and (5.8). The unit cost of Type I test is 4 and the unit cost of Type II test is 1. For the optimization in Eq. (5.7), we set the total budget constraint at 33; thus Eq. (5.7) becomes (assuming equal weights $p_1 = p_2$)

$$\begin{gathered} \text{Max} \quad S_{\boldsymbol{P}_{\boldsymbol{\theta}}}^{E(Y)} + S_{\boldsymbol{P}_{\boldsymbol{\theta}}}^{V(Y)} \\ \text{s.t.} \quad 4N_1 + N_2 \leq 33 \text{ and } N_i \geq 0 \end{gathered} \tag{5.14}$$

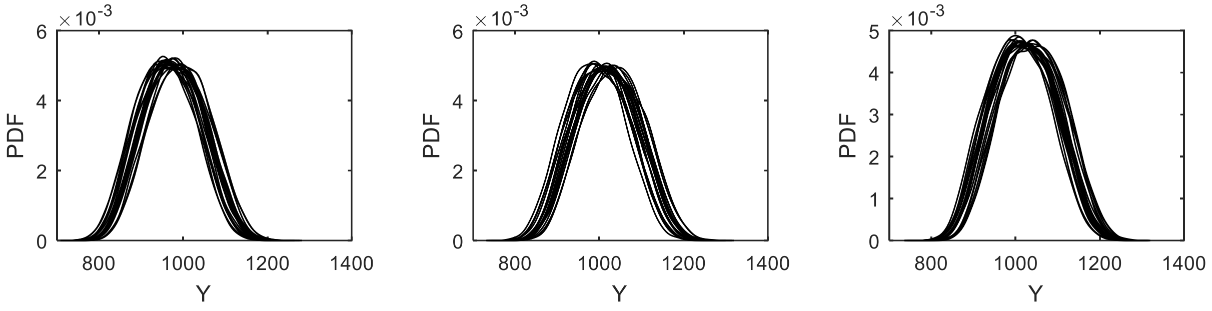
The simulated annealing algorithm is used to solve Eq. (5.14), and Figure 5.10 records the process of optimization. Figure 5.10(a) shows that the optimization starts at an initial design point $(N_1, N_2) = (5,5)$ and terminates at the optimal solution $(N_1, N_2) = (5,13)$. Figure 5.10(b) shows that only some of the random walks are accepted and the maximized Sobol' index sum $S_{\boldsymbol{P}_{\boldsymbol{\theta}}}^{E(Y)} + S_{\boldsymbol{P}_{\boldsymbol{\theta}}}^{V(Y)}$ is 1.92.



(a) History of accepted random walks **(b) History of the Sobol' indices sum $S_{\boldsymbol{\theta}}^{E(Y)} + S_{\boldsymbol{\theta}}^{V(Y)}$**

Figure 5.10 Optimization of the mathematical example based on Eq. (5.14)

As discussed in Section 5.3.1, since the robustness objective $S_{\boldsymbol{P_\theta}}^{E(Y)} + S_{\boldsymbol{P_\theta}}^{V(Y)}$ is maximized, the optimal solution $(N_1, N_2) = (5{,}13)$ for Eq. (5.14) should lead to consistent system response prediction regardless of the true values of $\boldsymbol{P_\theta}$. The results of this verification are shown in Figure 5.11.



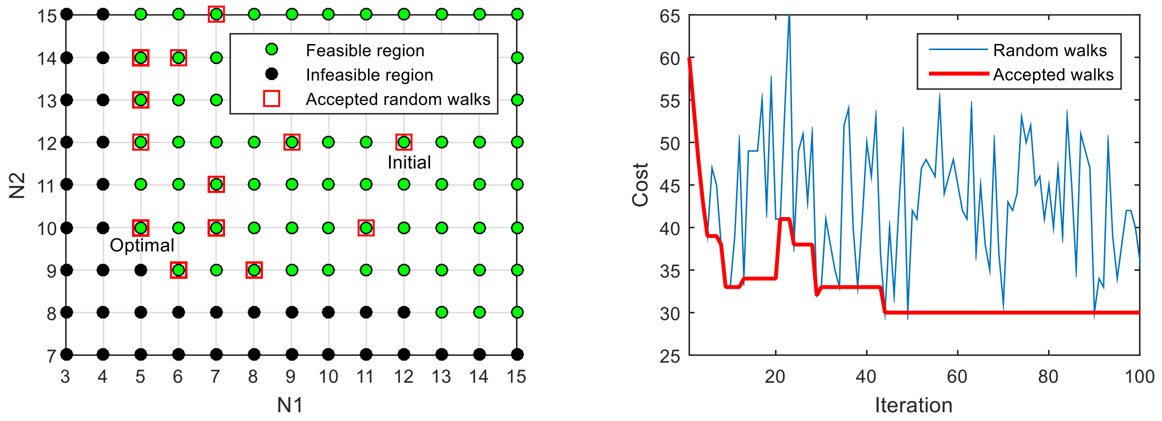
(a) $\boldsymbol{P_\theta} = \{\mathbf{4.2, 0.9, 8.3, 1.1}\}$ **(b)** $\boldsymbol{P_\theta} = \{\mathbf{5.8, 0.4, 9.1, 0.9}\}$ **(c)** $\boldsymbol{P_\theta} = \{\mathbf{4.7, 0.6, 9.6, 1.2}\}$

Figure 5.11 Family of prediction PDFs at the solution of Eq. (5.14) of $(\boldsymbol{N_1}, \boldsymbol{N_2}) = (\mathbf{5, 13})$

For the optimization in Eq. (5.8), we set the Sobol' index lower bounds as $\lambda^{E(Y)} = \lambda^{V(Y)} = 0.95$; thus Eq. (5.8) becomes

$$\begin{gathered} \text{Min } 4N_1 + N_2 \\ \text{s.t. } \; S_{\boldsymbol{P_\theta}}^{E(Y)} \geq 0.95, S_{\boldsymbol{P_\theta}}^{V(Y)} \geq 0.95 \text{ and } N_i \geq 0 \end{gathered} \tag{5.15}$$

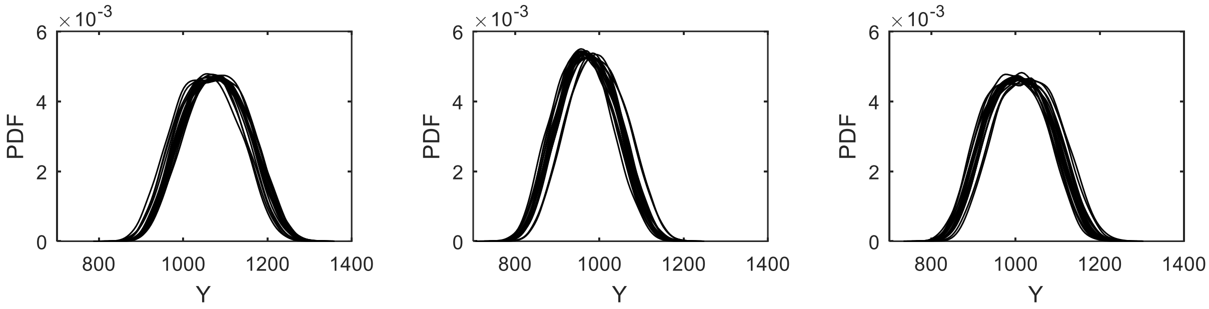
The simulated annealing algorithm is used to solve Eq. (5.15), and Figure 5.12 records the process of optimization. Figure 5.12(a) shows that the optimization starts at an initial design point $(N_1, \quad _2) = (12{,}12)$ and terminates at the optimal solution $(N_1, N_2) = (5{,}10)$. Figure 5.12(b) shows that only some of the random walks are accepted and the minimized cost is 30.



(a) History of accepted random walks **(b) History of cost $4N_1 + N_2$**

Figure 5.12 Optimization of the mathematical example based on Eq. (5.15)

As discussed in Section 5.3.1, since the robustness constraints $S_{\boldsymbol{P_\theta}}^{E(Y)} \geq 0.95, S_{\boldsymbol{P_\theta}}^{V(Y)} \geq 0.95$ are satisfied, the optimal solution $(N_1, N_2) = (5{,}10)$ for Eq. (5.15) should lead to consistent system response prediction regardless of the true values of $\boldsymbol{P_\theta}$. The results of this verification are shown in Figure 5.13.



(a) $\boldsymbol{P_\theta} = \{4.7, 0.3, 8.1, 1.0\}$ **(b) $\boldsymbol{P_\theta} = \{5.7, 0.9, 9.0, 0.4\}$** **(c) $\boldsymbol{P_\theta} = \{4.9, 0.5, 9.4, 1.2\}$**

Figure 5.13 Family of prediction PDFs at the solution of Eq. (5.15) of $(\boldsymbol{N_1}, \boldsymbol{N_2}) = (\boldsymbol{5}, \boldsymbol{10})$

5.5.2 Multi-level Problem

The second numerical example is a multi-level structural dynamics challenge problem from Section 4.6. In this example, we have four types of tests and a single specimen. As shown in Section 4.6 and Figure 4.2, this multi-level problem consists of three levels. Tests are available at Level 1 and Level 2, and it is required to predict the system response in Level 3. All three levels have the same model parameters, i.e., the three spring stiffnesses $\boldsymbol{k} = \{k_1, k_2, k_3\}$ (The damping ratios are assumed to known in this section). This example assumes the case of single test specimen thus $\boldsymbol{k}$ are the parameters to be calibrated. They are assumed to be deterministic but unknown, with independent prior distributions $k_1 \sim N(5000,\ 500^2)$, $k_2 \sim N(10000,\ 1000^2)$, and $k_3 \sim N(9000{,}900^2)$.

Four types of tests are available in this example:

1. Type I test measures $A_3^{L_1}$ and the resultant data set D_1^C is used in model calibration;
2. Type II test measures $A_3^{L_1}$ but the resultant data set D_1^V is used in model validation;
3. Type III test measures $A_3^{L_2}$ and the resultant data set D_2^C is used in model calibration;
4. Type IV test measures $A_3^{L_2}$ but the resultant data set D_2^V is used in model calibration.

The unit costs of these four types of tests are denoted as $C_i (i = 1 \text{ to } 4)$ respectively, and the number of each type of test is denoted as $N_i (i = 1 \text{ to } 4)$ respectively.

The key step to predict $A_3^{L_3}$ is to estimate the values of the model parameters $\boldsymbol{k} = \{k_1, k_2, k_3\}$. A reasonable route is to quantify the model parameters $\boldsymbol{k} = \{k_1, k_2, k_3\}$ using lower level calibration data of $A_3^{L_1}$ and $A_3^{L_2}$, and propagate the results through the computational model at the system level. However, either $A_3^{L_1}$ or $A_3^{L_2}$ can be used to calibrate the same model parameters, thus 3 calibration options are possible: 1) calibration using the data on $A_3^{L_1}$ alone; 2) calibration using the data on $A_3^{L_2}$ alone; and 3) calibration using the data on both $A_3^{L_1}$ and $A_3^{L_2}$. The challenge in such a multi-level problem is how to select from or combine these alternative calibration results. This research uses the roll-up method developed in Ref. [96] and [105] to solve this challenge. This roll-up method uses Bayesian model averaging of various calibration results and the weights for the averaging are obtained from model validation in each lower level. Thus the framework of model calibration/validation for prediction considers both model calibration and validation. A brief introduction of this framework is given here:

1. Model calibration by Bayesian inference to obtain the posterior distributions $p(\boldsymbol{k}|D_1^C)$, $p(\boldsymbol{k}|D_2^C)$, and $p(\boldsymbol{k}|D_1^C, D_2^C)$, respectively.

2. Model validation at lower levels using the model reliability metric in Refs. [53,96] and Section 4.3. The resultant model validity at Level 1 and Level 2 is denoted as $P(G_1)$ and $P(G_2)$, respectively.

3. Obtain the integrated distribution $p(\boldsymbol{k}|D_1^{C,V}, D_2^{C,V})$ by the roll-up formula [79,96,105] in Eq. (5.16):

$$\begin{aligned} p\left(\boldsymbol{k}\middle|D_1^{C,V}, D_2^{C,V}\right) = {} & P(G_1)P(G_2)p(\boldsymbol{k}|D_1^C, D_2^C) + P(G_1')P(G_2)p(\boldsymbol{k}|D_2^C) \\ & + P(G_1)P(G_2')p(\boldsymbol{k}|D_1^C) + P(G_1')P(G_2')p(\boldsymbol{k}) \end{aligned} \tag{5.16}$$

where $P(G_1') = 1 - P(G_1)$ and $P(G_2') = 1 - P(G_2)$ and $p(\boldsymbol{k})$ denotes the prior distribution of $\boldsymbol{k}$. In Eq. (5.16) the integrated distribution $p(\boldsymbol{k}|D_1^{C,V}, D_2^{C,V})$ is a weighted average of four terms: in the first term the posterior distribution $p(\boldsymbol{k}|D_1^C, D_2^C)$ uses the calibration data of both Level 1 and Level 2 and its weight $P(G_1)P(G_2)$ is the probability that both models are valid; in the second and third terms the posterior distribution $p(\boldsymbol{k}|D_i^C)$ uses the calibration data at Level i alone and its weight is the probability that the model at Level i is valid but the model at another level is invalid; in the last term the weight $P(G_1')P(G_2')$ of the prior distribution $p(\boldsymbol{k})$ is the probability that both of the models are invalid.

4. Propagate $p(\boldsymbol{k}|D_1^{C,V}, D_2^{C,V})$ through the computational model of $A_3^{L_3}$ to predict the distribution of $A_3^{L_3}$.

Since the computational models and measurement errors are known so that synthetic data of four types of test can be generated, and the framework of model calibration/validation is known, the proposed approach of test resource allocation is used to optimize the number of each type of test.

This example is applied to the two optimizations in Eqs. (5.7) and (5.8). Assume the unit cost of each type of test is $C_1 = C_2 = 1$, $C_3 = C_4 = 5$. For the optimization in Eq. (5.7), we set the total budget constraint at 60; thus Eq. (5.7) becomes (assuming equal weights $p_1 = p_2$)

$$\begin{gathered} \text{Max} \quad S_{\boldsymbol{\theta}}^{E(Y)} + S_{\boldsymbol{\theta}}^{V(Y)} \\ \text{s.t.} \quad N_1 + N_2 + 5N_3 + 5N_4 \leq 60 \text{ and } N_i \geq 0 \end{gathered} \tag{5.17}$$

The simulated annealing algorithm is used to solve Eq. (5.17). The initial value is $N_1 = N_2 = N_3 = N_4 = 3$. Among 500 iterations, the random walks of 226 iterations are accepted. Figure 5.14 shows the change of index sum over the iterations and the maximized index sum at the optimal solution is 1.88. The final optimal solution is $N_1 = 11, N_2 = 9, N_3 = 6, N_4 = 2$.

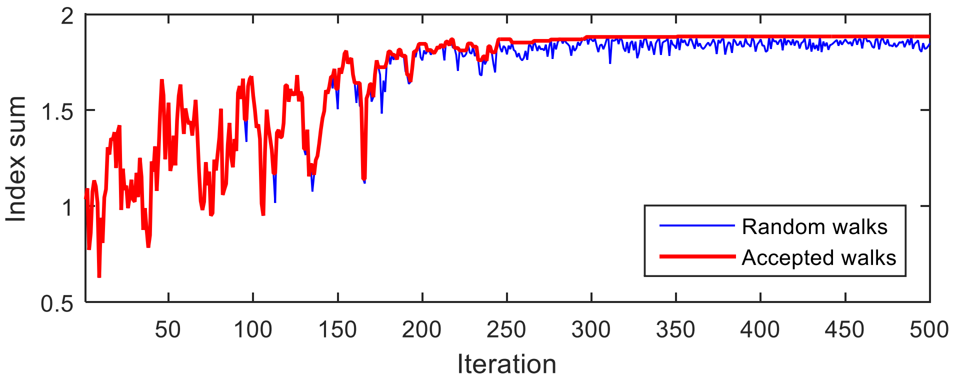
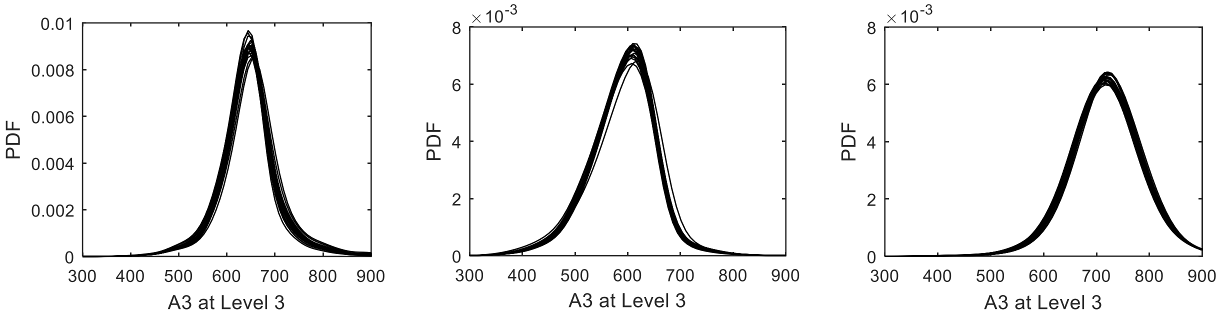


Figure 5.14 Optimization of the multi-level problem based on Eq. (5.17)

As discussed in Section 5.3.1, since the robustness objective $S_{\boldsymbol{\theta}}^{E(Y)} + S_{\boldsymbol{\theta}}^{V(Y)}$ is maximized, the optimal solution $(N_1, N_2, N_3, N_4) = (11,9,6,2)$ should lead to consistent system response prediction regardless of the true value of model parameters $\boldsymbol{k}$. Similar to the mathematical example in Section 5.5.1, verification of this multi-level test allocation result is shown in Figure 5.15. Figure 5.15 indicates that consistent predictions with three different assumed true values of model parameters.



(a) $\boldsymbol{k} = \{\mathbf{5600}, \mathbf{10433}, \mathbf{8638}\}$ **(b)** $\boldsymbol{k} = \{\mathbf{4483}, \mathbf{9112}, \mathbf{9987}\}$ **(c)** $\boldsymbol{k} = \{\mathbf{5776}, \mathbf{9812}, \mathbf{9393}\}$

Figure 5.15 Family of prediction PDFs at $(\boldsymbol{N_1}, \boldsymbol{N_2}, \boldsymbol{N_3}, \boldsymbol{N_4}) = (\mathbf{11}, \mathbf{9}, \mathbf{6}, \mathbf{2})$

This example is also applied to the optimization in Eq. (5.8). Assuming the unit cost of each type of test is $C_1 = C_2 = 1$, $C_3 = C_4 = 5$, and the threshold $\lambda^{E(Y)} = \lambda^{V(Y)} = 0.95$, Eq. (5.8) becomes

$$\begin{gathered} \text{Min } N_1 + N_2 + 5N_3 + 5N_4 \\ \text{s.t. } \; S_{\boldsymbol{\theta}}^{E(Y)} \geq 0.95, S_{\boldsymbol{\theta}}^{V(Y)} \geq 0.95 \text{ and } N_i \geq 0 \end{gathered} \tag{5.18}$$

The simulated annealing algorithm is used to solve Eq. (5.18). The initial value is $N_1 = N_2 = N_3 = N_4 = 15$. Among 500 iterations, the random walks of 164 iterations are accepted. Figure 5.16 shows the change of cost over the iterations and the minimized cost at the optimal solution is 66. The final optimal solution is $N_1 = 11, N_2 = 10, N_3 = 6, N_4 = 3$.

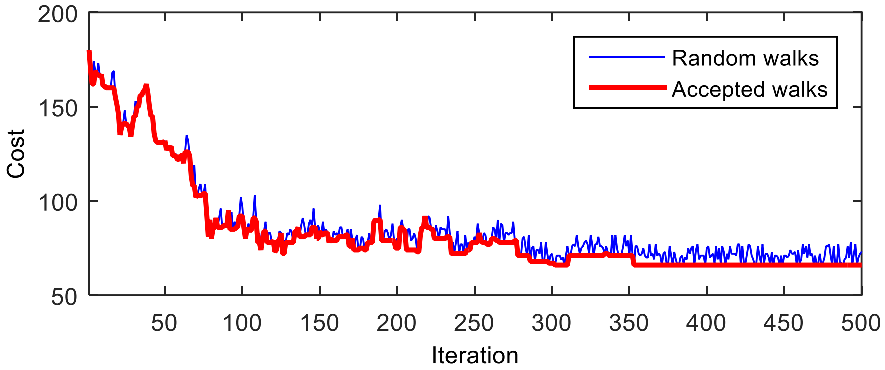
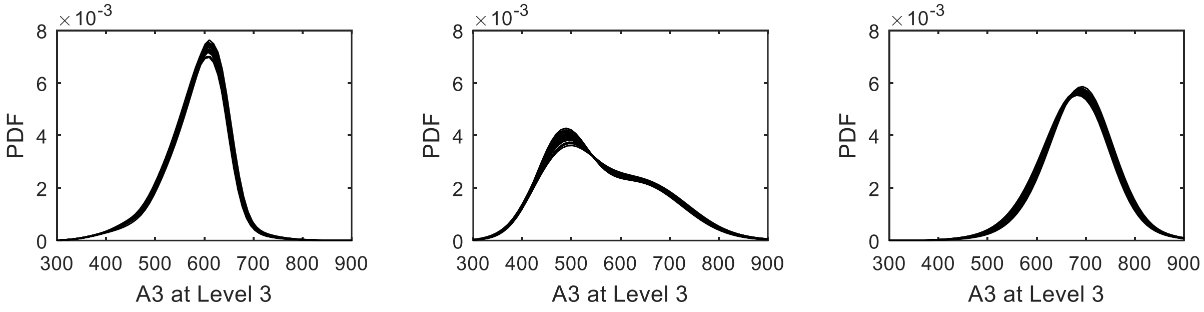


Figure 5.16 Optimization of the multi-level problem based on Eq. (5.18)

As discussed in Section 5.3.1, since the robustness constraints $S_{\boldsymbol{\theta}}^{E(Y)} \geq 0.95, S_{\boldsymbol{\theta}}^{V(Y)} \geq 0.95$ are satisfied, the optimal solution $(N_1, N_2, N_3, N_4) = (11,10,6,3)$ should lead to consistent system response prediction regardless of the true value of model parameters $\boldsymbol{k}$. Similar to the mathematical example in Section 5.5.1, verification of this multi-level test allocation result is shown in Figure 5.17. Figure 5.17 indicates that consistent predictions with three different assumed true values of model parameters.



(a) $\boldsymbol{k} = \{\mathbf{4492}, \mathbf{11183}, \mathbf{9116}\}$ **(b) $\boldsymbol{k} = \{\mathbf{5074}, \mathbf{8760}, \mathbf{7812}\}$** **(c) $\boldsymbol{k} = \{\mathbf{5276}, \mathbf{9883}, \mathbf{9518}\}$**

Figure 5.17 Family of prediction PDFs at $(\boldsymbol{N_1}, \boldsymbol{N_2}, \boldsymbol{N_3}, \boldsymbol{N_4}) = (\mathbf{11}, \mathbf{10}, \mathbf{6}, \mathbf{3})$

5.6 Summary

Test resource allocation aims to optimize the number of each type of test before any actual test is conducted. This research focuses on the proposed robust test resource allocation, which means that the system response prediction is non-sensitive to the variability in the test outcomes so that consistent predictions can be achieved under different test outcomes.

The main challenge for the proposed approach is to quantify the contribution of test outcome uncertainty towards the uncertainty in the prediction. Since test resource allocation is needed before any actual test, this test outcome uncertainty means the uncertainty in the synthetic data. This research analyzes the uncertainty sources in the synthetic data regarding different test conditions and concludes that consistent predictions will be achieved if the contribution of epistemic uncertainty regarding model parameters in the synthetic data can be maximized. This research uses the global sensitivity analysis method Sobol' indices to assess this contribution, so the desired consistent predictions can be guaranteed regardless of the true values of the parameters in the actual tests ($\boldsymbol{\theta}$ for single specimen and $\boldsymbol{P_{\theta}}$ for multiple specimen).

Two cases of optimization are considered in this research: 1) subject to the budget constraint, optimize the number of each type of test to reach the most robust design; or 2) subject to the robustness requirement, find the number of each type of test to minimize the budget. In addition, the proposed approach can be applied in multiple situations: 1) only model calibration tests are performed, or 2) both model calibration and model validation tests are performed. The proposed method results in a discrete stochastic optimization problem, and a simulated annealing algorithm is used to solve this problem.

This research assumes that the test inputs are from a range of values, which represents the uncertainty regarding the test inputs through uniform distributions. Future work will focus on the selection of the best input values (test design) such that the resultant prediction uncertainty can be further reduced. This challenge can be addressed in two ways: 1) optimize the number of tests and test inputs together; or 2) adaptively decide the number of tests and their input conditions based on the observation data as the test campaign progresses.

CHAPTER 6

UNCERTAINTY INTEGRATION IN TIME DEPENDENT STRUCTURAL HEALTH DIAGNOSIS/PROGNOSIS

6.1 Background

In the earlier part of this dissertation, Chapter 4 discussed the response prediction for a time-independent system, especially when test is available at the sub-system/component level; and Chapter 5 discussed test strategy to obtain a robust prediction, but the system of interest is still time-independent. This chapter focuses on time dependent systems, where the prediction is not a single value or probability distribution, but a series of values or probability distributions varying over time. This chapter is organized for a case study of aircraft wing structural health diagnosis and prognosis, but the underlying concepts of dynamic Bayesian network, particle filter, etc., are applicable to other time-dependent systems.

In deciding whether an aircraft is capable of safely performing an upcoming mission, a structural health monitoring (SHM) system is desired to provide the decision-maker with the information on damage state of the aircraft, such as the crack length on the wing or the reliability of a replaceable unit. Information based on fleet statistics is not useful in assessing the health and capability of a particular aircraft, since the damage state varies from aircraft to aircraft due to the variability in manufacturing, material properties, mission history, pilot variability, etc. The data collected in Ref. [106] reveal that at the same operational hours some aircraft has twice the fatigue damage rate compared to others aircrafts. In sum, a SHM system tailored to each individual aircraft is desirable.

One example of an individualized SHM system is the individual aircraft tracking (IAT) program [107] to track the potential fatigue damage in the major airframe structural components such as the wing. A typical IAT program for F-16 [108] utilizes the recorded load history to predict the crack growth and estimate the crack severity index (CSI); then a comparison between the resultant CSI and a baseline condition will classify the aircraft health into three damage severity levels. This IAT system mainly focuses on the variation of load history; other uncertainty sources such as the epistemic uncertainty regarding the true values of geometric or material properties are not considered. A more comprehensive IAT program integrating various uncertainty sources in crack growth prediction is desirable, in order to avoid over- or under-estimating the damage prognosis and achieve a balanced decision-making considering safety, performance and budget.

Therefore, this chapter aims to develop a powerful approach for building a *probabilistic* individual aircraft tracking (PIAT) model. This model is developed to analyze the crack growth on the leading edge of an aircraft wing, as shown in Figure 6.1; but the underlying concepts can be extended to other airframe structural components or the entire airframe. As explained earlier, this PIAT model is supposed to integrate various uncertainty sources over the entire life of aircraft wing. In addition, the PIAT is also desired to achieve the following objectives: 1) integrate heterogeneous information including test data, mathematical models, expert opinions, etc.; 2) fly virtually through the same load history as the actual aircraft wing; 3) reduce the uncertainty in model parameters and track the time-dependent system states using measurement data, i.e., diagnosis; and 4) predict the evolution of damage states if no data is available, i.e., prognosis. An introduction to diagnosis and prognosis can be found in Ref. [109].

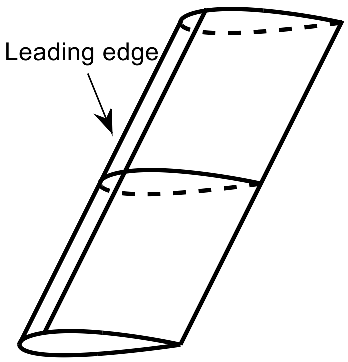


Figure 6.1 Aircraft wing and its leading edge

Bayesian network (BN) is a promising approach to integrate various uncertainty sources and heterogeneous information. Regarding various uncertainty sources, Bayesian network allows different types of random variables, including discrete and continuous variables of different distribution types. Regarding heterogeneous information, Bayesian network is able to incorporate operational data, laboratory data, reliability data, expert opinion, and mathematical models (physics-based as well as empirical) [110].

As explained in Section 2.1, Bayesian network is extended to dynamic Bayesian network (DBN) to track a time-dependent system whose states evolve over time. The ability to track system evolution over time make DBN a suitable methodology to build the PIAT model for diagnosis and prognosis of the aircraft wing.

When data of any node is obtained, the Bayesian network is updated by Bayesian inference thus the uncertainty in the state variables can be reduced. A review of Bayesian inference algorithms for DBN have been given in Section 2.3.2, including Kalman filter [25], extended Kalman filter [25], unscented Kalman filter [111], and particle filter [112] . The Kalman filter gives exact and analytical updating results [25] for a linear Gaussian DBN, which means: 1) the state function and the measurement function are both linear; 2) state variables have a joint Gaussian distribution; and 3) all the noise terms are assumed to be independent zero mean Gaussian variables. If the state function and/or the measurement function are non-linear, the extended Kalman filter linearizes these functions to the first order, and gives analytical updating results. Extended Kalman filter requires computing the Jacobian matrix, which brings computational difficulty in the case of high non-linearity [111].

Another method to handle the non-linear relationships in the DBN is the unscented Kalman filter. Both Kalman filter and extended Kalman filter are purely analytical. In contrast, the unscented Kalman filter uses the method of unscented transform to select several sample points, and propagates them through the non-linear functions. The propagation is used to derive analytical updating results with accuracy to the third order, and the computation of the Jacobian matrix is not required [113]. However, the unscented Kalman filter can encounter ill-conditioning problems in the covariance matrix [113].

Although the extended Kalman filter and unscented Kalman filter provide solutions to non-linear DBN, they still assume that all the state variables are Gaussian. This research aims to develop a generic DBN framework that can handle 1) both discrete and continuous variables; 2) various types of continuous variable distributions; and 3) linear/non-linear functional relationships. In contrast, particle filter (PF) is a sampling-based algorithm, where a particle is sample from the joint distribution of the BN at one time step. The PF is a generic algorithm and fulfills the above requirements [113–115], thus this research chooses PF as the Bayesian inference algorithm for DBN. A brief introduction to the particle filter is given in Section 6.2.1.

The implementation of the PF includes: 1) forward propagation, i.e., sampling of the child nodes based the samples of the parent nodes and their dependence relationships; 2) backward inference, i.e., updating of the current BN to reduce uncertainty. Forward propagation is needed in each time step, while backward inference is needed only if the data of any child node is observed. Due to the complexity of the DBN for a realistic system, the implementation of the PF algorithm is non-trivial. Section 6.2.2 of this research contributes to solve this problem by classifying the random variables in a DBN into five groups so that the required particles can be generated over these groups sequentially.

Compared to the analytical algorithms such as the Kalman filter, PF is more computationally intensive since: 1) the forward propagation proceeds each particle individually; and 2) updating requires extra efforts in computing the likelihood and weights of the particles. This research denote the time step of purely forward propagation as “prognosis step”, and the time step requiring backward inference as “diagnosis step”. Obviously, a diagnosis step is more expensive than a prognosis step. This research also contributes to reduce the computational efforts. Generally the test data in analyzing an airframe component include load data and damage measurement data; and usually load data outnumber damage measurement data significantly. Section 6.2.3 of this section modified the structure of the DBN regarding the node of load and its observation, thus updating is NOT needed if the load is observed but the damage is not. The modified DBN is proved to be equivalent to the original DBN, but reduces the number of the diagnosis steps thus spends much less effort in updating.

In the rest of the paper, Section 6.3 analyzes the uncertainty sources in the fatigue crack growth on an aircraft wing and incorporate them into a DBN. Section 6.4 computes and analyzes the results of diagnosis and prognosis of the aircraft wing. Methods established in Section 6.2 are applied in sections 6.3 and 6.4.

6.2 Diagnosis and Prognosis in the DBN

6.2.1 Introduction to Particle Filter

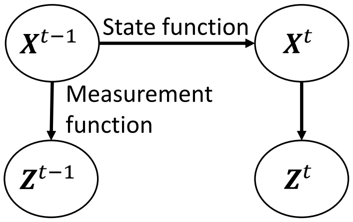


Figure 6.2 A simple DBN

Particle filter (PF) is a general algorithm to track the evolution of the state variables in a DBN. In the simple DBN in Figure 6.2, assume that the state variables $\boldsymbol{X}^t \in \Re^m$ at time t evolves from the state variables $\boldsymbol{X}^{t-1} \in \Re^m$ according to the state function

$$\boldsymbol{X}^t = f(\boldsymbol{X}^{t-1}, \boldsymbol{v}^{t-1}) \qquad (6.1)$$

where $\boldsymbol{v}^{t-1} \in \Re^m$ is the vector of noise terms in the state function. The measurement $\boldsymbol{Z}^t \in \Re^n$ is obtained according to the measurement function

$$\boldsymbol{Z}^t = h(\boldsymbol{X}^t, \boldsymbol{\sigma}^t) \qquad (6.2)$$

where $\boldsymbol{\sigma}^t \in \Re^n$ is the vector of noise terms in the measurement function.

In case that the DBN represented by Eqs. (6.1) and (6.2) is not a linear Gaussian DBN, several particle filter algorithms have been developed to track the evolution of $\boldsymbol{X}^t$ and $\boldsymbol{Z}^t$. The most basic particle filter algorithm is sequential importance sampling (SIS) [112]. The SIS considers the full joint posterior distribution at time step t, $p(\boldsymbol{X}^{0:t}|\boldsymbol{Z}^{1:t})$. This distribution is approximated with a weighted set of particles $\{\boldsymbol{x}_i^{0:t}, \omega_i^t\}_{t=1}^N$. These particles approximate the joint posterior distribution $p(\boldsymbol{X}^{0:t}|\boldsymbol{Z}^{1:t})$ by

$$p(\boldsymbol{X}^{0:t}|\boldsymbol{Z}^{1:t}) \approx \sum_{i=1}^{N} \omega_i^t \delta_{\boldsymbol{x}_i^{0:t}} \tag{6.3}$$

where $\delta_{\boldsymbol{x}_i^{0:t}}$ is a delta function at $\boldsymbol{x}_i^{0:t}$.

In this section, capital letters denote random variables; lower-case letters denote particles, where the superscript i indicates that it is the i-th particle. The subscripts of letters indicate the time step. Thus the state variables at time step t are denoted as $\boldsymbol{X}^t$. At time step t, the i-th particle of $\boldsymbol{X}^t$ is denoted as $\boldsymbol{x}_i^t$, and it is sampled based the current state $\boldsymbol{X}_i^{0:t-1}$ and the observation $\boldsymbol{Z}^{1:t}$ according to a proposal density

$$\boldsymbol{X}_i^t \sim q(\boldsymbol{X}^t \middle| \boldsymbol{X}_i^{0:t-1}, \boldsymbol{Z}^{1:t}) \tag{6.4}$$

In other words, the new state $\boldsymbol{X}_i^t$ of the i-th particle at time step t is sampled from a distribution which takes the current state $\boldsymbol{X}_i^{0:t-1}$ and the observation $\boldsymbol{Z}^{1:t}$ as parameters.

At time step t, the weight ω_i^t is updated from ω_i^{t-1} by

$$\omega_i^t \propto \omega_i^{t-1} \frac{p(\boldsymbol{Z}^t|\boldsymbol{X}_i^t)p(\boldsymbol{X}_i^t|\boldsymbol{X}_i^{t-1})}{q(\boldsymbol{X}_i^t|\boldsymbol{X}_i^{t-1}, \boldsymbol{Z}^t)} \tag{6.5}$$

In addition, the initial state $\boldsymbol{X}_i^0$ are sampled from the joint prior distribution of the state variables, and the initial weight ω_i^0 for each particle is $1/N$.

In practice, iterations of Eqs. (6.4) and (6.5) over time step t may lead to particle degeneracy problem, i.e., only a few particles have significant weights. This problem can be solved by resampling: a new set of N particles is generated from the discrete approximation shown in Eq. (6.3), and the weight of each new particle is set as $1/N$ again.

Some variants of the SIS algorithm have been developed in the literature to simplify its implementation, and a widely used one is the sampling importance resampling (SIR) algorithm [112]. The SIR algorithm 1) takes the state transition distribution $p(\boldsymbol{X}^t|\boldsymbol{X}_i^{t-1})$ as the proposal density distribution $q(\boldsymbol{X}^t|\boldsymbol{X}_i^{0:t-1}, \boldsymbol{Z}^{1:t})$, and 2) conducts resampling at each iteration. Thus Eqs. (6.4) and (6.5) reduce to

$$\boldsymbol{X}_i^t \sim p(\boldsymbol{X}^t|\boldsymbol{X}_i^{t-1}) \tag{6.6}$$

$$\omega_i^t \propto p(\boldsymbol{Z}^t|\boldsymbol{X}_i^t) \tag{6.7}$$

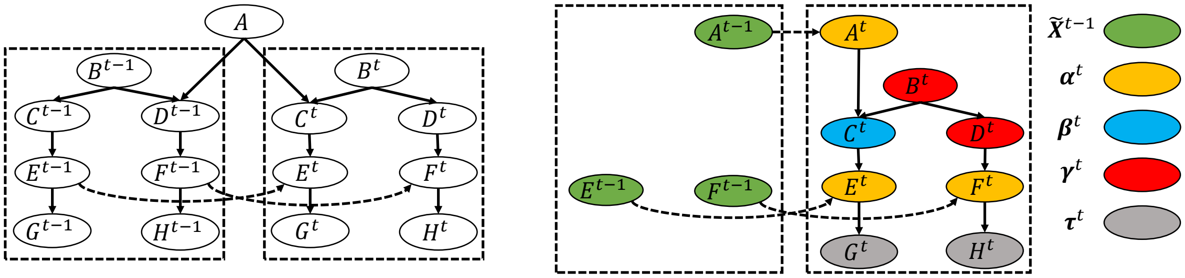
Note that resampling is after the calculation of Eqs. (6.6) and (6.7) at each time step, where new particles of $\boldsymbol{X}_i^t$ are generated and the weight of each new particle is set as $1/N$.

It is straightforward to implement the SIR algorithm, since it only requires sampling from the distribution $p(\boldsymbol{X}^t|\boldsymbol{X}_i^{t-1})$ and evaluating the likelihood $p(\boldsymbol{Z}^t|\boldsymbol{X}_i^t)$. Thus this algorithm is used in this research for aircraft wing structural health diagnosis and prognosis in this research. Other more sophisticated algorithms can be also implemented in the proposed methodology, such as the auxiliary sampling importance resampling filter [116], regularized particle filter [117], and Rao-Blackwellized particle filter [118].

6.2.2 Implementing Particle Filter in DBN

There are two challenges in implementing the SIR algorithm of Eqs. (6.5) and (6.6) to a complex DBN (such as the DBN of aircraft wing in Section 6.3.2). First, in addition to dynamic nodes whose states change over time, static nodes shared by all the time steps are also included. An example of the static node is node A in Figure 6.3(a). The existence of static nodes violates the prerequisite assumption of DBN: one separate BN for each step.

Second, the states of some dynamic nodes depend not only on the previous state of these variables but also on some other variables in the current time step. For example, in Figure 6.3(a) node E^t on E^{t-1} and C^t. Thus at time step t, P^t must be sampled prior to B^t. This requires us to distinguish the parent nodes of each state variable in $\boldsymbol{X}^t$ to implement Eq. (6.6).



(a) DBN **(b) New particle generation**

Figure 6.3 Particle filter for an illustrative DBN

The solutions to these two challenges are explained using an illustrative DBN showed in Figure 6.3. The first challenge can be resolved by separating a static node into two identical nodes. In Figure 6.3, the shared static node A is split to A^{t-1} and A^t. Subscripted by $t-1$ and t, A^{t-1} belongs to the BN at time step $t-1$ and A^t belongs to the BN at time step t. An arrow representing the deterministic relationship $A^{t-1} = A^t$ directs from A^{t-1} to A^t so that these two nodes are identical. In sum, this solution fulfills the assumption of one BN for each time step, and guarantees that the same static node is shared by each time step.

The solution to the second challenge requires several steps in order to realize Eq. (6.6). The nodes in BNs at time step $t-1$ and t are classified into five groups:

1. $\widetilde{\boldsymbol{X}}^{t-1}$: state variables in $\boldsymbol{X}^{t-1}$ with arrows directed to state variables in $\boldsymbol{X}^t$. Among all the nodes in $\boldsymbol{X}^{t-1}$, only $\widetilde{\boldsymbol{X}}^{t-1}$ are the parent nodes of the variables in $\boldsymbol{X}^t$, thus Eq. (6.6) can be written as $\boldsymbol{X}_i^t\sim p(\boldsymbol{X}^t|\widetilde{\boldsymbol{X}}_i^{t-1})$. $\widetilde{\boldsymbol{X}}^{t-1} = \{A^{t-1}, E^{t-1}, F^{t-1}\}$ for the illustrative DBN in Figure 6.3.

2. $\boldsymbol{\alpha}^t$: child nodes of $\widetilde{\boldsymbol{X}}^{t-1}$ in the BN at time step t. The sampling of $\boldsymbol{\alpha}^t$ depends on the value of $\widetilde{\boldsymbol{X}}^{t-1}$ in the previous BN. $\boldsymbol{\alpha}^t = \{A^t, E^t, F^t\}$ for the illustrative DBN in Figure 6.3.
3. $\boldsymbol{\beta}^t$: intermediate nodes of $\boldsymbol{\alpha}^t$. A node in $\boldsymbol{\beta}^t$ has both ancestor and descendant nodes in $\boldsymbol{\alpha}^t$. A node in $\boldsymbol{\beta}^t$ depends on some nodes in $\boldsymbol{\alpha}^t$, and a node in $\boldsymbol{\alpha}^t$ can also depend some nodes in $\boldsymbol{\beta}^t$. In Figure 6.3 we have $\boldsymbol{\beta}^t = C^t$.
4. $\boldsymbol{\gamma}^t$: ancestor nodes of $\boldsymbol{\alpha}^t$ or $\boldsymbol{\beta}^t$ in the BN at time step t. No node in $\boldsymbol{\gamma}^t$ is the descendant node of $\widetilde{\boldsymbol{X}}^{t-1}$, i.e., the sampling of $\boldsymbol{\gamma}^t$ is independent of the previous BN. The distribution of $\boldsymbol{\gamma}^t$ is denoted as $p(\boldsymbol{\gamma}^t)$. The sampling of $\boldsymbol{\alpha}^t$ and $\boldsymbol{\beta}^t$ depends both on $\widetilde{\boldsymbol{X}}^{t-1}$ and $\boldsymbol{\gamma}^t$, which can be expressed by a conditional distribution $p(\boldsymbol{\alpha}^t, \boldsymbol{\beta}^t | \widetilde{\boldsymbol{X}}_i^{t-1}, \boldsymbol{\gamma}_i^t)$. In Figure 6.3, we have $\boldsymbol{\gamma}^t = \{B^t, D^t\}$.
5. $\boldsymbol{\tau}^t$: descendant nodes of $\boldsymbol{\alpha}^t$ or $\boldsymbol{\beta}^t$ in the BN at time step t. The sampling of $\boldsymbol{\tau}^t$ depends on $\boldsymbol{\alpha}^t$ or $\boldsymbol{\beta}^t$, i.e., a conditional probability distribution $\boldsymbol{\tau}_i^t \sim p(\boldsymbol{\tau}^t | \boldsymbol{\alpha}_i^t, \boldsymbol{\beta}^t)$. In Figure 6.3 we have $\boldsymbol{\tau}^t = \{G^t, H^t\}$.

As $\boldsymbol{X}^t$ is denoted as $\{\boldsymbol{\alpha}^t, \boldsymbol{\beta}^t, \boldsymbol{\gamma}^t, \boldsymbol{\tau}^t\}$ based on the classification above, the sampling of $\boldsymbol{X}_i^t$ in Eq. (6.6) is realized sequentially by

$$\begin{gathered} \boldsymbol{\gamma}_i^t \sim p(\boldsymbol{\gamma}^t) \\ \boldsymbol{\alpha}_i^t, \boldsymbol{\beta}_i^t \sim p(\boldsymbol{\alpha}^t, \boldsymbol{\beta}^t | \widetilde{\boldsymbol{X}}_i^{t-1}, \boldsymbol{\gamma}_i^t) \\ \boldsymbol{\tau}_i^t \sim p(\boldsymbol{\tau}^t | \boldsymbol{\alpha}_i^t, \boldsymbol{\beta}_i^t) \end{gathered} \tag{6.8}$$

For the illustrative DBN in Figure 6.3, to generate new particles $\boldsymbol{X}_i^t = \{A_i^t, B_i^t, C_i^t, D_i^t, E_i^t, F_i^t, G_i^t, H_i^t\}$ based on $\widetilde{\boldsymbol{X}}_i^{t-1} = \{A_i^{t-1}, E_i^{t-1}, F_i^{t-1}\}$, $\boldsymbol{\gamma}_i^t = \{B_i^t, D_i^t\}$ is first sampled by $p(\boldsymbol{\gamma}^t) = p(B^t, D^t) = p(B^t)p(D^t | B_i^t)$; then $\boldsymbol{\alpha}_i^t = \{A_i^t, E_i^t, F_i^t\}$ and $\boldsymbol{\beta}_i^t = C_i^t$ are sampled by

$p(\boldsymbol{\alpha}^t, \boldsymbol{\beta}^t | \widetilde{\boldsymbol{X}}_i^{t-1}, \boldsymbol{\gamma}_i^t) = p(A^t | A_i^{t-1}) p(C^t | A_i^t, B_i^t) p(E^t | E_i^{t-1}, C_i^t) p(F^t | F_i^{t-1}, D_i^t)$; finally $\boldsymbol{\tau}^t = \{G^t, H^t\}$ is sampled from $p(\boldsymbol{\tau}^t | \boldsymbol{\alpha}_i^t, \boldsymbol{\beta}_i^t) = p(G^t | E_i^t) p(H^t | F_i^t)$.

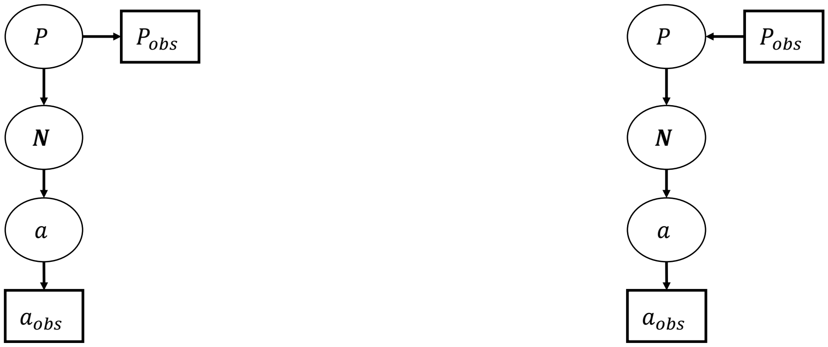
6.2.3 Computation Effort Reduction by Modifying the DBN Structure

As explained in Section 1, the diagnosis step is more expensive than the prognosis step. The prognosis step is purely forward uncertainty propagation and only requires particle generation by Eq. (6.8). In contrast, the diagnosis step requires Bayesian inference thus brings extra computation effort for the likelihood $p(\boldsymbol{Z}^t | \boldsymbol{X}_i^t)$, the weight ω_i^t and the resampling in SIR. The computational cost increase as more diagnosis steps are needed. Diagnosis step happens if and only if any child node is observed. Prognosis step happens in two cases: 1) no observation is available; 2) all the observations are for the root nodes, thus the distribution of other nodes can be obtained by uncertainty propagation using Eq. (6.8) with these root nodes fixed at their observations.

The damage a in an airframe component is caused by the load P applied on it, thus the DBN starts from the node of load P and end at the node of damage a (In crack growth analysis of the aircraft, the damage a is the fatigue crack length).

Generally P and a are observable. Due to the measurement error, the observed data of P is the realization of a new random variable P_{obs} and the observed data of a is the realization of a new random variable a_{obs}. Thus in the BN node P directs to node P_{obs}, indicating a measurement model such as $P_{obs} = P + \epsilon_P$ where ϵ_P is the measurement error; node a directs to node a_{obs}, indicating another measurement model such as $a_{obs} = a + \epsilon_a$ where ϵ_a is the measurement error. The resultant BN at one time step is shown in Figure 6.4(a), where $\boldsymbol{N}$ denotes all the other nodes except for $P, P_{obs}, a,$ and a_{obs}.

If a data point D_P of the load P is observed, in the BN node P_{obs} will be fixed at D_P; similarly, node a_{obs} will be fixed at the data point D_a if the damage a is observed. Since neither node of P_{obs} and a_{obs} is root node, diagnosis of Bayesian inference is needed whenever the data of P and/or a are available. If the full load history is measured, diagnosis of Bayesian inference is conducted in every time step even if the crack length data are sparse. This causes tremendous computational cost.



(a) Original DBN **(2) Modified DBN**

Figure 6.4 Original BN and modified BN

In fact, it can be proved that under certain assumptions (explained below) we can reverse the arrow from P to P_{obs}, i.e., replace $P \rightarrow P_{obs}$ with $P_{obs} \rightarrow P$, and the modified BN is equivalent to the original one. In the modified BN, P_{obs} is a root node so that Bayesian inference is not needed if only the load is observed. In other words, diagnosis is conducted only at limited steps with the crack length observed, which reduces the computational cost significantly. Proof of the equivalence between the original BN and the modified BN is given as follows.

If the load is not observed, the node P_{obs} can be removed in the BN thus the original BN and the modified BN are exactly the same. If the load is observed, two cases need to be considered. In the first case, we assume that both the load and the crack length are observed. In the original BN, if load and crack length data are denoted as D_P and D_a, the posterior distribution over the BN is:

$$
\begin{aligned}
&p(P,\boldsymbol{N},a|P_{obs}=D_P,a_{obs}=D_a)\\
&\propto p(P_{obs}=D_P,a_{obs}=D_a|P,\boldsymbol{N},a)p(P,\boldsymbol{N},a)\\
&=p(a_{obs}=D_a|a)p(P_{obs}=D_P|P)p(P)p(N|P)p(a|\boldsymbol{N})
\end{aligned}
\tag{6.9}
$$

where $p(a_{obs}=D_a|a)p(P_{obs}=D_P|P)$ is the likelihood function and $p(P)p(\boldsymbol{N}|P)p(a|\boldsymbol{N})$ is the prior distribution over $P,\boldsymbol{N},$ and a.

For the modified BN, the posterior distribution is:

$$
\begin{aligned}
&p(P,\boldsymbol{N},a|P_{obs}=D_P,a_{obs}=D_a)\\
&\propto p(P_{obs}=D_P,a_{obs}=D_a|P,\boldsymbol{N},a)p(P,\boldsymbol{N},a)\\
&=p(a_{obs}=D_a|a)p(P_{obs}=D_P|P,\boldsymbol{N},a)p(P,\boldsymbol{N},a)\\
&=p(a_{obs}=D_a|a)p(P|P_{obs}=D_P)p(N|P)p(a|\boldsymbol{N})
\end{aligned}
\tag{6.10}
$$

where $p(a_{obs}=D_a|a)$ is the likelihood function, and $p(P|P_{obs}=D_P)p(N|P)p(a|\boldsymbol{N})$ is the prior distribution over $P,\boldsymbol{N},$ and a is conditioned at $P_{obs}=D_P$. The posteriors in Eqs. (6.9) and (6.10) are equivalent if $p(P_{obs}=D_P|P)p(P)\propto p(P|P_{obs}=D_P)$.

As the measurement noise is generally assumed to be zero mean Gaussian distribution, we have $P_{obs}=P+N(0,\sigma_p)$, which gives $P=P_{obs}-N(0,\sigma_p)$. Then it can be proved that:

$$
(P_{obs}=D_P|P)=p(P|P_{obs}=D_P)=\frac{1}{\sigma_p\sqrt{2\pi}}\exp\left(-\frac{\left(P-D_p\right)^2}{2\sigma_p^2}\right)
\tag{6.11}
$$

Thus the condition of $p(P_{obs}=D_P|P)p(P)\propto p(P|P_{obs}=D_P)$ will be fulfilled if node P in the original BN has a non-informative uniform distribution such that $p(P)$ is a constant.

In the second case, we assume that only the load is observed. The posterior distribution over the original BN is:

$$\begin{aligned} p(P, \boldsymbol{N}, a | P_{obs} = D_P) &\propto p(P_{obs} = D_P | P, \boldsymbol{N}, a) p(P, \boldsymbol{N}, a) \\ &= p(P_{obs} = D_P | P) p(P) p(N|P) p(a|\boldsymbol{N}) \end{aligned} \tag{6.12}$$

The posterior distribution over the modified BN is obtained purely by uncertainty propagation:

$$p(P, \boldsymbol{N}, a | P_{obs} = D_P) = p(P | P_{obs} = D_P) p(N|P) p(a|\boldsymbol{N}) \tag{6.13}$$

Eqs. (6.12) and (6.13) are equivalent under the same assumptions: 1) zero mean Gaussian measurement error for the load; and 2) $p(P)$ is a non-informative uniform prior distribution in the original BN if the load is observed. The first one is a widely used assumption in the literature. The second one requires that load P^t is independent of P^{t-1} if P^t is observed. This is also a reasonable assumption since the observation of the load at time t provides strong information for the true value of the load at time t such that the information from P^{t-1} can be neglected. A time series model giving the CPD of $p(P^t|P^{t-1})$ is still applicable if P^t is not observed.

In sum, this section distinguished the steps of diagnosis and prognosis. The time-consuming diagnosis is required if and only if any child node is observed. This section also showed that under two weak assumptions, the CPD $P \rightarrow P_{obs}$ can be replaced by $P_{obs} \rightarrow P$ such that the number of diagnosis steps can be reduced significantly, i.e., Bayesian updating is required only if crack inspection data is available.

6.3 Dynamic Bayesian Network of Crack Growth on Aircraft Wing

Different fracture mechanics-based fatigue crack growth models have been developed to calculate the propagation of long cracks, including Paris' law [119], modified Paris' law [120], Wheeler's retardation model [121], etc. Generally these models require computing the stress intensity factor K, for which finite element analysis (FEA) is widely used. Two techniques of utilizing the FEA to compute the crack growth can be found in the literature: 1) include the crack geometry into the FEA model and compute the stress intensity factor by the FEA directly; then calculate the crack growth using a crack growth law and adjust the crack geometry by modifying the input file to the FEA model [4,19]; 2) build a FEA model without the crack geometry and compute the nominal stress at the crack; then calculate the stress intensity factor using an analytical formula and subsequent crack growth using a crack growth law [122]. Due to the mesh complexity of the FEA model with the crack geometry, the computational cost of technique 2 is significantly smaller than technique 1. Since crack growth prediction under uncertainty requires numerous runs of the FEA model, technique 2 is applied in this research. Based on technique 2, the rest of this section discusses the uncertainty sources in predicting the fatigue crack growth on the leading edge of an aircraft wing; then all the uncertainty sources are incorporated into the DBN. Note that DBN is still applicable regarding technique 1, and this research selects technique 2 only for higher computational efficiency.

6.3.1 Uncertainty Sources

Uncertainty Sources in the FEA Model and Surrogate Model

Figure 6.5 shows the FEA model of the leading edge of an aircraft wing. Spring and beam elements in Figure 6.5 simulate the connection between the leading edge and the wing body. The load on the leading edge is simulated by connecting the leading edge to an anchor point through rigid bars and applying the load P on the anchor point. A single bolt is assumed to fix the anchor point to the wing body. 15 geometric and material parameters are assumed as random variables in the FEA model:

1. $T_i (i = 1 \text{ to } 7)$: The leading edge is divided into 7 sections along the Y axis, and T_i is the thickness of the i-th section;

2. $K_i (i = 1 \text{ to } 4)$: K_1 and K_2 are the stiffnesses of Spring A and B along the Y axis; K_3 and K_4 are the stiffnesses of Spring A and B along the Z axis;
3. IBP: Inboard beam property, which is the area moment of inertia of Beam A;
4. OBP: Outboard beam property, which is the area moment of inertia of Beam B;
5. TR: Taper ratio, measuring the rate that the leading edge width shrinks from the wing root to the wing tip; here it is defined as the ratio of Beam A length to Beam C length;
6. Y_A: Coordinate of the anchor point along the Y axis; the value of Y_A varies if the bolt is loose.

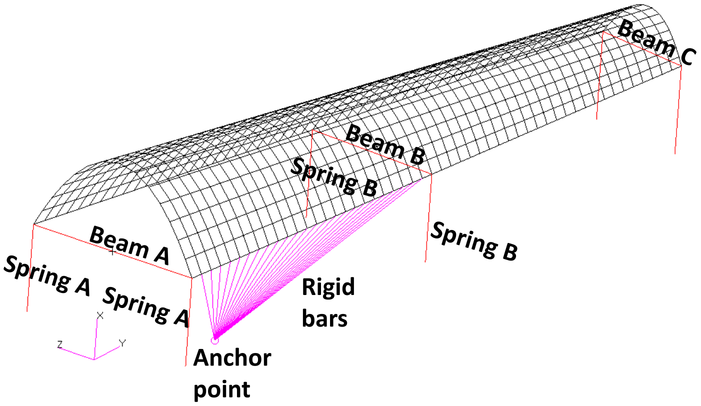


Figure 6.5 Leading edge of an aircraft wing

All the parameters above except for Y_A have deterministic but unknown values thus brings epistemic uncertainty. Prior distributions are assigned to them, and the proposed DBN-based PIAT model seeks to reduce their epistemic uncertainty by Bayesian inference. The value of Y_A changes over time thus the proposed PIAT model needs to track its evolution.

These 15 parameters and the load P are the inputs to the FEA model in Figure 6.5, which computes the nominal stress S at the crack. Probabilistic prediction as well as Bayesian inference require many evaluations of the analysis model. In order to achieve computational efficiency, this research uses a Gaussian process (GP) surrogate model [30,35] to replace the FEA model. Training points are obtained by repeatedly running the FEA model at different combinations of values (DoE points) of the 15 parameters and the load P. At given inputs, the prediction of the GP model is a normal distribution $S\sim N(\mu_{GP},\sigma_{GP}^2)$, which represents the surrogate model uncertainty in computing the stress for a given value of the inputs. This also indicates that these 15 parameters and the load P are the parent nodes of stress S in the DBN, and the corresponding conditional probability distribution (CPD) is given by the GP model prediction $S\sim N(\mu_{GP},\sigma_{GP}^2)$.

Not all the 15 parameters are equally important to the crack growth. Global sensitivity analysis by Sobol' indices [59,61] can be used to assess the contribution of each parameter to the uncertainty in the crack growth. Parameters of low sensitivity can be fixed at their nominal values, thus reducing the computational cost in diagnosis and prognosis.

Crack Growth Model Uncertainty and Damage State Uncertainty

Once the nominal stress at the crack is computed using the GP model, the next step is to compute the stress intensity factor and crack growth. Methods to compute the stress intensity factor for different load conditions and crack shapes are summarized in Ref [123]. The validity of these models is generally problem-dependent. For the sake of illustration, this research assumes a mode I uniaxial crack; thus the range of stress intensity factor in one time step is

$$\Delta K = 1.2F\Delta S\sqrt{\pi a_s} \tag{6.14}$$

where $1.2F$ is the crack shape factor and ΔS is the stress range and a_s is the initial crack length in the current time step. Here F is defined as a multiplier for the shape factor, and the uncertainty in F represents the uncertainty in the shape factor.

Next, for the sake of illustration, this research uses the Paris' law to compute the crack growth Δa in each time step:

$$\frac{\mathrm{d}a}{\mathrm{d}N} = C\Delta K^m \tag{6.15}$$

where C and m are the Paris' law parameters obtained from material component experiments; $\mathrm{d}a/\mathrm{d}N$ is the crack growth rate, and its magnitude is equal to the predicted crack growth Δa in one time step. The crack length after the current time step is $a = a_s + \Delta a$. C and m are assumed to be known constants in this research to keep the focus on other parameters that provide particular challenges to DBN that are addressed in this research; C and m can be easily treated as aleatory or epistemic uncertain quantities and included in the DBN as needed.

The uncertainty sources in the crack growth prediction are the uncertainties in the parameters of Eq. (6.14), which are affected by the damage state, and uncertainties regarding the parameters of Eq. (6.15) (ignored in this research). In this research, two damage states are considered.

1. Bolt looseness (B): For the sake of illustration, assume that bolts are used to fix the anchor point to the wing body. Assume that all the bolts are collectively represented by one notional bolt with equivalent properties. Whether the bolt becomes loose depends on its resistance R and the current load P. A higher P or a lower R leads to a higher probability of loose bolt ($B = 1$). The bolt will stay loose once it becomes loose ($B = 0$). The loose bolt causes uncertainty in the anchor point position (Y_A) thus affecting the nominal stress (S) at the crack location. In addition, Eq. (6.16) is assumed to simulate the degradation of the bolt resistance with time step t. In Eq. (6.16), R_0 is the initial bolt resistance; k is the degradation coefficient and it has a negative value so that $R(t)$ decreases with t.

$$R(t) = R_0 \exp(kt) \tag{6.16}$$

2. Crack tip in elastic zone vs. plastic zone (M): The aircraft wing is mostly elastic ($M = 0$); it is assumed that randomly located plastic zones ($M = 1$) can be caused by accidents such as a dropped hammer; the crack is assumed to start at the elastic zone and there is a finite probability that the crack grows into a plastic zone in any time step; the crack is assumed to stay in the plastic zone once it reaches it. It is assumed that 1) the shape factor multiplier in the elastic zone (F_e) has a known deterministic value obtained from material coupon experiments; 2) the plastic zone retards the crack growth thus the multiplier F_p in the plastic zone is smaller than F_e; and 3) F_p has a deterministic but unknown value, i.e., epistemic uncertainty. This damage state M can be represented by expanding Eq. (6.14) as

$$\Delta K = \begin{cases} 1.2 F_e \Delta\sigma\sqrt{\pi a_s} & \text{if} \quad M = 0 \\ 1.2 F_p \Delta\sigma\sqrt{\pi a_s} & \text{if} \quad M = 1 \end{cases} \tag{6.17}$$

The damage states above bring two new uncertainty sources: 1) whether the damage states have occurred; and 2) uncertainty in the value of F_p. The proposed DBN-based PIAT model is beneficial in tracking the damage states and quantify the uncertainty in F_p. In addition, the damage states are discrete variables, thus requiring a DBN that can handle both discrete and continuous variables.

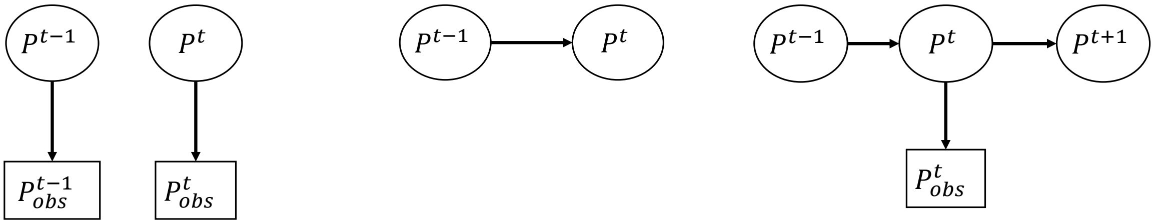
Load Uncertainty

The uncertainty in load P depends on specific cases. In case 1, the load history at each time step is measured by sensors in the aircraft wing. The measured load history can be used to simulate the flight, diagnose damage states, and compute the crack length after the flight. Techniques to measure the load history include flight parameters-based loads monitoring and strain gauge-based loads monitoring [124]. In this case, the uncertainty in the load history is the measurement error.

The numerical example in Section 4 assumes the measurement error as a zero mean Gaussian noise, i.e., $\epsilon_p \sim N(0, \sigma_p^2)$.

In case 2, the PIAT model is used to simulate the future load time history and predict the crack growth. To model this time series input based on the observed load history in earlier flights and capture the uncertainty in the future loading, two types of time domain methods have been developed: time step counting methods and random process methods. The time step counting methods [125] discretize the time series into k levels and extract a counting matrix from the data. The counting matrix is used to generate the load history stochastically. In contrast, one of the random process methods, e.g., the autoregressive moving average (ARMA) [40] model assumes that the input in the current time step is a linear function of 1) its past p values; and 2) the current and past q values of noise terms. Both types of methods can be used in the PIAT model. In case 2, the load uncertainty includes the natural variability in the time series input and epistemic uncertainty due to limited information in building the model.

Different conditions in case 1 and 2 affect the DBN structure. Let P^{t-1} and P^t denote the loads at time $t-1$ and t; and P_{obs}^{t-1} and P_{obs}^t denote their observations at time $t-1$ and t respectively. In the BN of case 1, P^{t-1} and P^t are directly connected to the nodes P_{obs}^{t-1} and P_{obs}^t respectively, giving the conditional probability distributions (CPDs) of $P_{obs}^{t-1} \sim N(P^{t-1}, \sigma_p^2)$ and $P_{obs}^t \sim N(P^t, \sigma_p^2)$. In the BN of case 2, the value of P^{t-1} affects the value of P^t, thus an arrow of CPD defined by the time series model is used to connect them. The node P_{obs}^t or P_{obs}^{t-1} are not necessary since the load is not observed. A hybrid case is also possible, i.e., both case 1 and 2 occur in the DBN. The DBN structures of case 1, case 2, and the hybrid case are shown in Figure 6.6.



(a) Case 1 **(b) Case 2** **(c) Hybrid case**

Figure 6.6 DBN structure for loading history uncertainty

Crack Length Data Uncertainty

The crack length data are assumed to be available from on-ground inspection, which brings two uncertainty sources: measurement error and data sparsity. Similar to the load uncertainty, the measurement error in the crack length data depends on the accuracy of the inspection technique, and is generally assumed to have a zero mean Gaussian distribution $\epsilon_a \sim N(0, \sigma_a^2)$. The proposed methodology can also handle other distributions of measurement error.

Crack length data are rarely available for every time step. Even if one data point is obtained after each mission and applied in the DBN for diagnosis and prognosis, the crack length data are missing during the mission; thus data uncertainty is introduced by data sparseness.

In sum, two data sources are available for the PIAT model of aircraft wing: load history data and crack inspection data. The availability of these data can be quite flexible: 1) load history data can be available at all time steps (case 1 in Figure 6.6), no time step (case 2 in Figure 6.6) and limited time steps (case 3 in Figure 6.6); while crack inspection data are only available at sparse time steps. DBN has the capacity of both Bayesian inference (diagnosis) and uncertainty propagation (prognosis).

6.3.2 DBN Construction

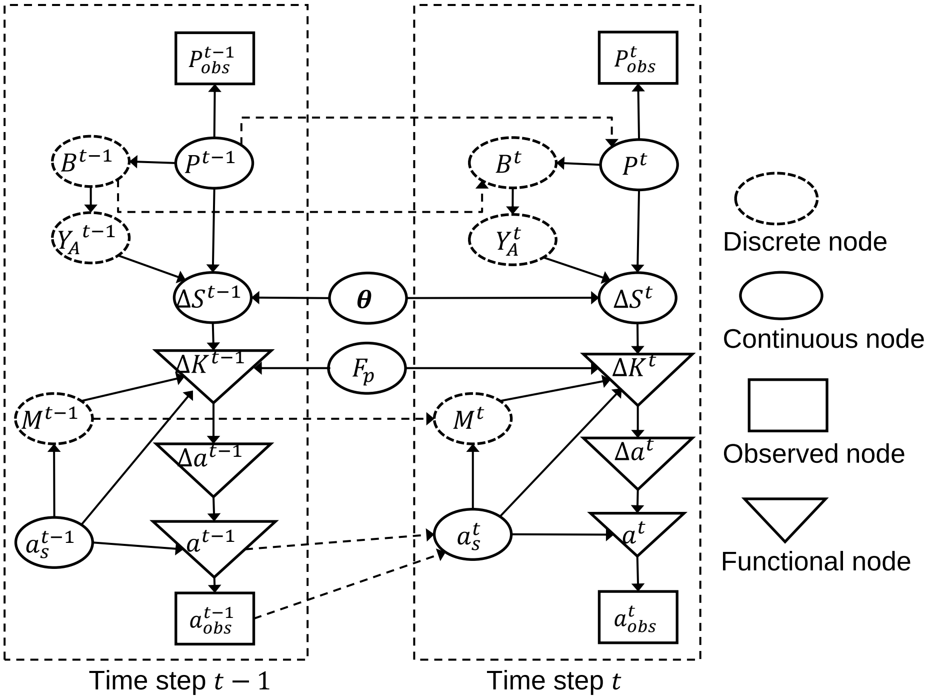


Figure 6.7 Dynamic Bayesian network for crack growth

Table 6.1 Nomenclature for the DBN

P_{obs}	Load observation	ΔK	Stress intensity factor range
P	Load	Δa	Crack growth in current time step
B	Bolt looseness	a	Crack length after current time step
Y_A	Anchor point position	a_{obs}	Crack length observation
ΔS	Stress range	$\boldsymbol{\theta}$	Geometric and material properties
M	Elastic/Plastic zone	F_p	Shape factor in the plastic zone
a_s	Crack length before current time step		

As shown in Figure 6.7, the uncertainty sources identified in Section 6.3.1 are represented by nodes in the DBN; nodes are connected by arrows which represent conditional probability distributions or deterministic functional relations. The superscript $t-1$ or t denotes the time step, and the symbols in Figure 6.7 are explained in Table 6.1.

In Figure 6.7, an elliptical node is a stochastic node, meaning the variable is stochastic for given values of parent nodes, thus the arrows towards it represent a CPD; a triangular node is a functional node, meaning the variable is the result of deterministic calculation for given values of parent nodes thus the arrow towards it represent a deterministic function. In addition, elliptical nodes with solid lines represent continuous variables, whereas elliptical nodes with dashed lines represent discrete variables. The rectangular nodes represent observed variables (e.g., load and crack length). In addition, solid arrows are used within a BN slice, and dashed arrows connect the nodes across different time steps.

In Figure 6.7, node $\boldsymbol{\theta}$ represents all the 15 geometric and material properties (except for Y_A) of the aircraft wing. Each property should be a node in the DBN connected to ΔS. They are depicted as a single node to save space.

Another special node in the DBN is a_s. For the BN in any time step, prior distributions are assigned to all the root nodes first, then uncertainty propagation or Bayesian inference will be conducted. Except for time step 1 where prior distributions are defined by users, BNs at other time steps obtain the prior distributions by propagating the posterior distributions of previous time step through the arrows connecting adjacent BNs. But for a_0^t:

1. If the crack length is not observed at time step $t-1$, its prior distribution is the predicted distribution of a^{t-1}, which means a deterministic functional relationship $a^{t-1} = a_0^t$, thus a^{t-1} directs to a_0^t in Figure 6.7;
2. If the crack length is observed at time step $t-1$, its prior distribution for time step *t* should be defined using this data point. Let $D_{a^{t-1}}$ denote the observed data point value. This research defines the prior distribution of a_0^t as $N(D_{a^{t-1}}, \sigma_a)$, thus a_{obs}^{t-1} also directs to a_0^t in Figure 6.7.

Once the DBN is constructed, diagnosis and prognosis are the next steps in the health monitoring of the aircraft wing. This is explained in Section 6.4.

6.4 Results and Analysis

A numerical example of crack growth on the leading edge of aircraft wing is used to illustrate all the concepts explained in earlier sections. The structure of the aircraft wing has been explained in Section 6.3.2. A time series input of 10,000 steps is applied at the anchor point. The FEA result in Figure 6.8 shows that under the geometric and material property uncertainty and load uncertainty, the location of maximum stress is always around Node 389. Thus we assume that a crack of 0.0588 inch is initialized at Node 389 and grows under the time series loading at the anchor point. A GP surrogate model predicting the stress at Node 389 is built to replace the FEA model.

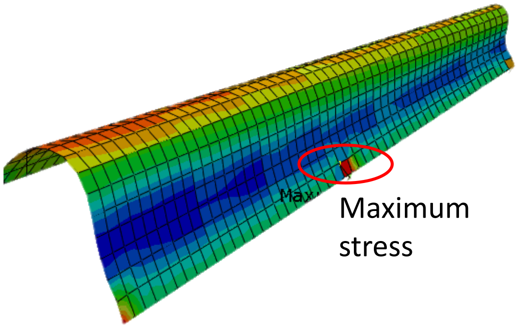


Figure 6.8 Maximum stress in the aircraft wing

It is assumed that the time series input is observed at each step, and that the measurement error is a zero mean Gaussian variable $N(0,0.002^2)$. The observed load history is shown in Figure 6.9. Furthermore, crack length data are assumed to be observed only at time steps 2000, 4000, 5600, 6400, and 6800.

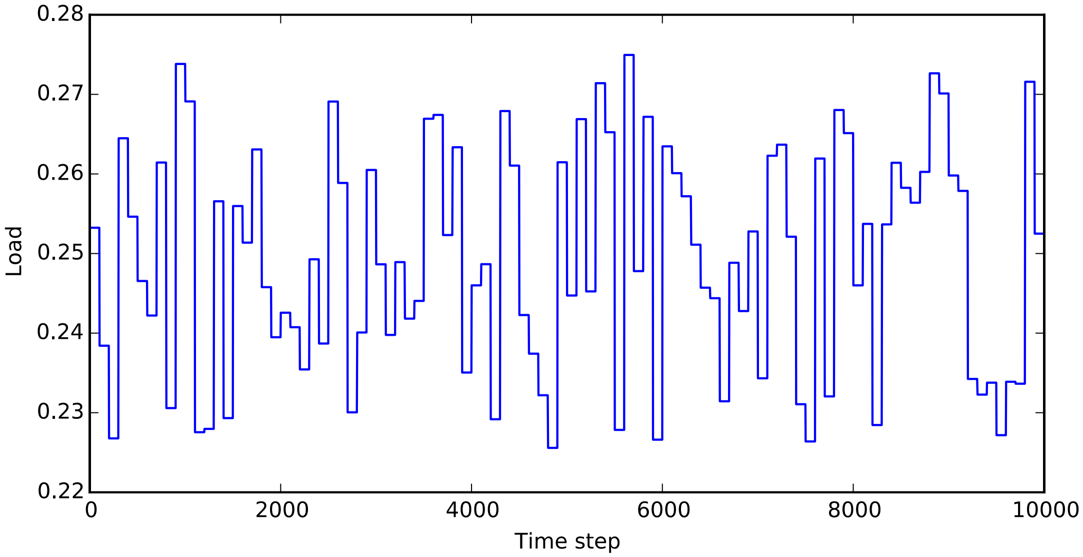


Figure 6.9 Load history observation

As explained in Section 6.3.1, the aircraft wing contains 15 stochastic geometric parameters and one stochastic crack growth model parameter F_p. Except for the anchor point position Y_A, all these parameters are static root nodes in Figure 6.7.

GSA results for the elastic zone ($M = 0$) and plastic zone ($M = 1$) are shown in Table 6.2. In the elastic zone, T_4 is the only significant parameter; in the plastic zone, T_4 and F_p are both significant. The sensitivity index of Y_A is small, indicating that Y_A and its only parent node bolt looseness B can be fixed at nominal values and the crack length data cannot track the evolution of B effectively. In this research, we retain the nodes of B and Y_A in the DBN to quantitatively prove this proposition. The parameters in Table 6.2 except for T_4, F_p and Y_A are fixed at their nominal values.

Table 6.2 GSA results

	Elastic zone		Plastic zone	
Parameters	First-order index	Total effects index	First-order index	Total effects index
T_1	0.000	0.000	0.000	0.000
T_2	0.000	0.000	0.000	0.000
T_3	0.012	0.025	0.001	0.002
T_4	0.875	0.902	0.104	0.277
T_5	0.002	0.010	0.000	0.000
T_6	0.001	0.003	0.000	0.000
T_7	0.000	0.000	0.000	0.000
IBP	0.000	0.000	0.000	0.000
OBP	0.000	0.001	0.000	0.000
K_1	0.000	0.000	0.000	0.000
K_2	0.001	0.002	0.000	0.000
K_3	0.000	0.000	0.000	0.000
K_4	0.000	0.000	0.000	0.001
TR	0.000	0.000	0.000	0.000
Y_A	0.022	0.031	0.002	0.004
F_p	/	/	0.696	0.865

The deterministic relationships (represented by the arrows to deterministic nodes in the DBN) have been discussed in Section 6.3.1. The conditional probability distribution for the continuous node ΔS_t is a Gaussian distribution $N(\mu_{GP}(Y_A^t, P^t, \boldsymbol{\theta}), \sigma_{GP}(Y_A^t, P^t, \boldsymbol{\theta}))$ obtained by the GP surrogate model. Then the DBN for the crack growth is constructed as in Figure 6.7 and used for diagnosis and prognosis.

In this example, since the load P is observed at each time step, which provides strong evidence on the true value of P, the CPD of $p(P^t|P^{t-1})$ can neglected thus the arrow from P^{t-1} to P^t can be removed in the DBN of Figure 6.7. The prior distribution of node P at each time step is assumed to be a uniform distribution $U(P_l, P_u)$, where P_l and P_u are lower and upper bounds based on expert opinion. With these assumptions, the method of replacing $P \rightarrow P_{obs}$ by $P_{obs} \rightarrow P$ in Section 6.2.3 is applied to improve the computational efficiency. This research uses 10^4 particle in the computation of this example and the overall time cost is $T = 11109s$, including: 1) $11102s \approx 3.1hrs$ spent on forward propagation of 10^4 time steps; and 2) $7s$ spent on 4 updating. If the method in Section 6.2.3 is not used so that each time step requires updating, the time spent on updating will be $17500s$ and overall time cost will be $28602s \approx 7.9hrs$. In other words, the proposed method in Section 6.2.3 reduces the time cost by 61%.

The resistance of the bolt decreases as time, as shown in Eq. (6.16). Here we assume that the initial resistance of the bolt is $R_0 = 0.275$; the resistance $R(t)$ reduces to $0.9R_0$ after 10^4 time steps so that the degradation coefficient is $k = -1.0536 \times 10^{-5}$. The conditional probability tables for the discrete nodes B, Y_A and M are assumed as shown in Table 6.3, Table 6.4, and Table 6.5, for the sake of illustration.

Table 6.3 Conditional probability table of B_t

$p(B^t\|P^t, B^{t-1})$	$B^{t-1} = 1$	$B^{t-1} = 0$		
		$P^t < 0.85R(t)$	$0.85R(t) < P^t < 0.95R(t)$	$P^t > 0.95R(t)$
$B_t = 0$	0	1	0.975	0.95
$B_t = 1$	1	0	0.025	0.05

Table 6.4 Conditional probability table of Y_{A_t}

$p(Y_A^t\ \|B^t)$	$Y_A^t = 0$	$Y_A^t = 9.935 \pm 0.5$	$Y_A^t = 9.935 \pm 1.0$	$Y_A^t = 9.935 \pm 1.5$	$Y_A^t = 9.935 \pm 2.0$
$B^t = 0$	1.0	0.0	0.0	0.0	0.0
$B^t = 1$	0.0	0.25	0.125	0.075	0.05

Table 6.5 Conditional probability table of M_t

$p(M^t\|a^t,\ M^{t-1})$	$M^{t-1} = 1$	$M^{t-1} = 0$			
		$a^t < 0.1$	$0.1 < a^t < 0.12$	$0.12 < a^t < 0.15$	$0.15 < a^t$
$M^t = 0$	0	1	0.8	0.6	0.5
$M^t = 1$	1	0	0.2	0.4	0.5

True values are assumed for the model parameters in Table 6.2. Synthetic data for the observed crack length at time steps 2000, 4000, 5600, 6400, and 6800 are generated using the load history in Figure 6.9 and the assumed true values. Using these data, the objectives of this numerical example are to calibrate the static variables T_4 and F_p, track the evolution of the time-dependent damage state variables B and M, and predict the crack length in the future. The results of these calculations are shown in Figures Figure 6.10 to Figure 6.14.

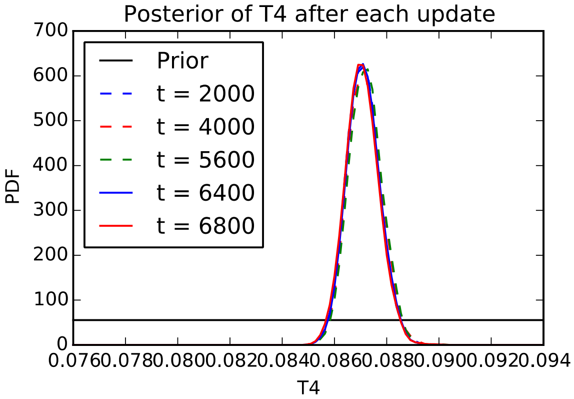


Figure 6.10 Updating of T_4

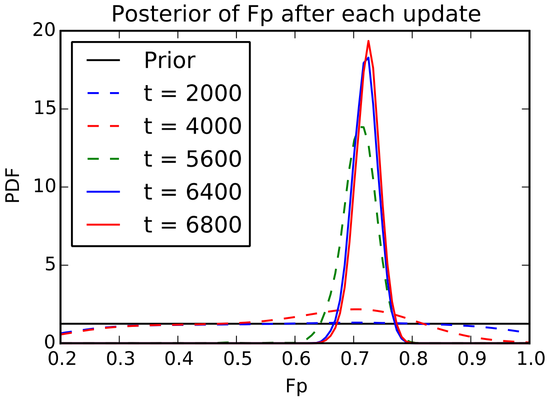


Figure 6.11 Updating of F_p

Figure 6.10 and Figure 6.11 show the updating of T_4 and F_p at each time step of inspection. Due to its high sensitivity in both the elastic zone and plastic zone, the uncertainty of T_4 is reduced significantly just after the Inspection 1 at t=2000 of crack length. In contrast, F_p is not updated at Inspection 1 at t=2000. The reason is that the crack tip has not reached the plastic zone at t=2000 (shown in Figure 6.12) thus the obtained data do not contain information on the parameter F_p of the plastic zone. The uncertainty in F_p is reduced using the data from later inspections, where the crack tip has reached the plastic zone.

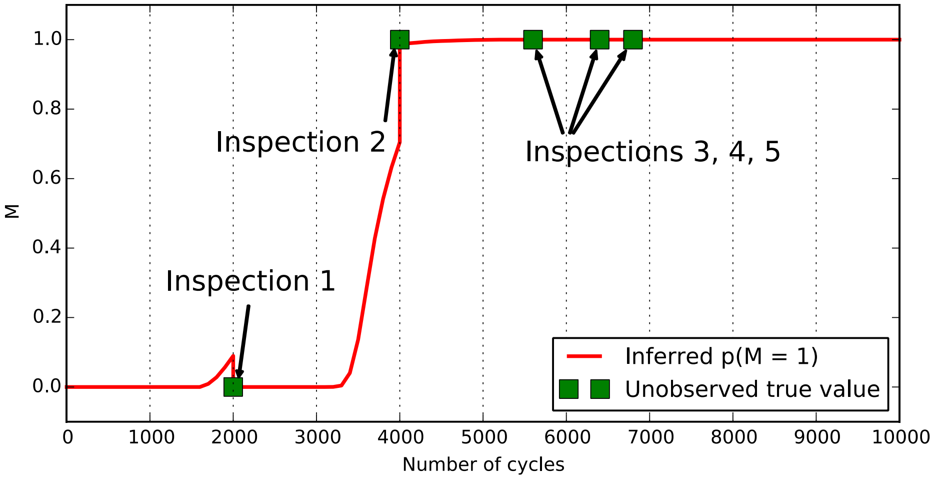


Figure 6.12 Tracking damage state *M* (crack tip in elastic vs. plastic zone)

Figure 6.12 shows the inferred evolution of damage state M. Recall that $M = 0$ indicates that the crack tip is in the elastic zone, whereas $M = 1$ indicates that the crack tip is in the plastic zone. Since the two states of the discrete variable M are 0 and 1, the mean value of the inferred M is $0 \times p(M = 0) + 1 \times p(M = 1)$, i.e., equal to the probability $p(M = 1)$. This probability $p(M = 1)$ increases before Inspection 1 due to the assumed conditional probability distribution $p(M_t|a_t,\ M_{t-1})$ and reaches around 0.1 at $t = 2000$. Then the network was updated by the crack length data from Inspection 1 and $p(M = 1)$ is corrected to 0. The unobserved true value of M is still 0 at t =2000 thus this correction is valid. This reduced $p(M = 1)$ also reduces the effect of the uncertainty in F_p, and thus reduces the uncertainty in the crack length prediction, as shown in Figure 6.14. A similar correction also occurs at Inspection 2, where $p(M = 1)$ is corrected from 0.7 to 1.0. In inspections 3, 4 and 5, Figure 6.12 shows a probability of $p(M = 1) = 1$, i.e., the crack has reached the plastic zone.

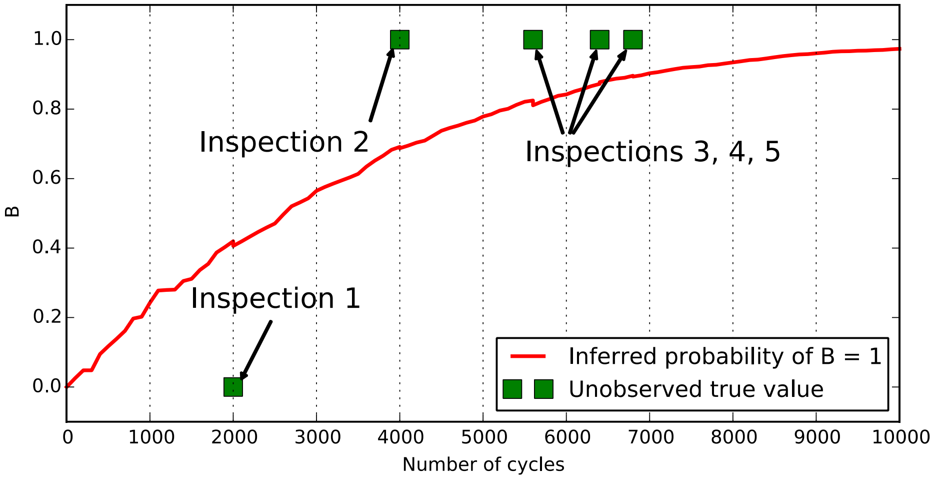


Figure 6.13 Tracking damage state *B* (bolt loosening)

Figure 6.13 shows the inferred evolution of damage state B, where $B = 1$ indicates a loose bolt and $B = 0$ indicates a tight bolt. The probability $p(B = 1)$, which is equal to the mean value of inferred B, increases before any inspection due to the assumed conditional probability distribution $p(B^t|P^t, B^{t-1})$. The curve fluctuates due to the randomness in load P^t. However, $p(B = 1)$ is not corrected significantly by the crack length data in the inspection. This can be explained by the GSA results in Table 6.2. The sensitivity of Y_A with respect to the crack length is negligible, meaning that as the only parent node of Y_A, the bolt looseness B also has negligible influence on the crack growth, and is therefore not updated significantly.

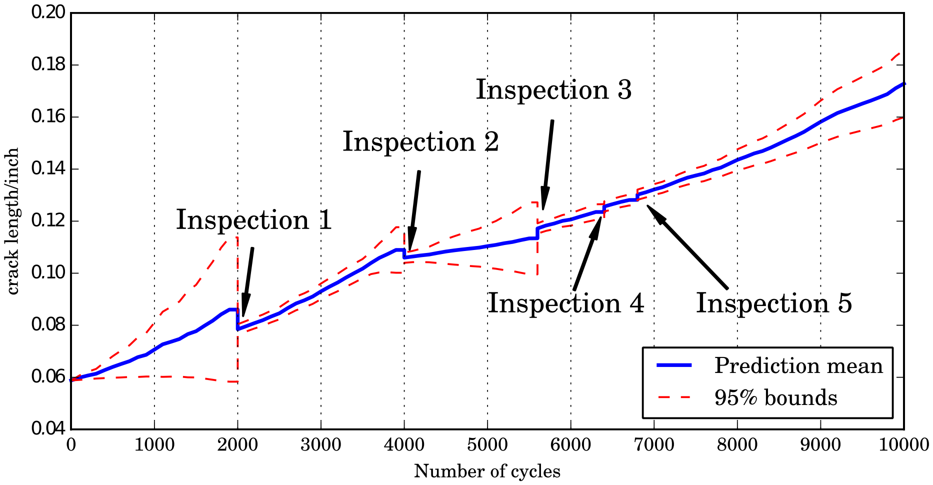


Figure 6.14 Diagnosis and prognosis of crack length

Figure 6.14 shows the diagnosis and prognosis of the crack length. The uncertainty in the crack length is reduced to measurement error at each inspection, and grows between inspections. The uncertainty grows fast (wide 95% bounds) before the first inspection, since the uncertainty propagation is based on prior distributions of T_4 . The uncertainty grows slower between inspections 1 and 2 due to: 1) significantly reduced uncertainty in T_4 at the first inspection, as shown in Figure 6.10; and 2) low probability that the crack has reached the plastic zone as shown in Figure 6.12, i.e., low probability that the uncertainty in F_p is introduced. The uncertainty grows fast between Inspections 2 and 3 since: 1) the crack has reached the plastic zone so that the uncertainty in F_p is introduced; the data from Inspection 2 barely reduces the uncertainty in F_p, as shown in Figure 6.11. The uncertainty grows slower again after Inspection 3 since the uncertainty in F_p has been reduced by the observation data at Inspections 3, as shown in Figure 6.11.

6.5 Summary

Various uncertainty sources affect the health state diagnosis and prognosis of aircraft components. This chapter establishes a framework for probabilistic health diagnosis and prognosis using a dynamic Bayesian network (DBN). This framework is versatile due to the following characteristics: 1) incorporate various aleatory and epistemic uncertainty sources; 2) handle both discrete and continuous variables; 3) allow the continuous variables to have any distribution type; and 4) allow non-linear functional relationships.

Particle filter is used as the Bayesian inference algorithm for the non-linear and non-Gaussian DBN. The implementation of particle filter for this DBN is non-trivial due to 1) the existence of static nodes, which are time-independent variables shared by all the Bayesian networks; and 2) state variables that may have parent nodes across two adjacent Bayesian networks. Therefore, this research classifies the nodes in adjacent Bayesian networks into five groups to facilitate generating new particles based on the particle in the Bayesian network in the previous time instant. The generated new particles are used in Bayesian updating and help to realize the diagnosis.

Prognosis requires no Bayesian inference thus is computationally less demanding than diagnosis. In case that the load is observed at each time step, theoretically Bayesian updating of the DBN is required at each time step, which implies large computational cost. This research shows that the DBN can be modified under reasonable assumptions about measurement error and load observation; as a result, the number of time steps requiring Bayesian updating of the DBN is reduced significantly, thus providing substantial savings in computational effort (61% saving in the numerical example).

All the concepts above are illustrated by a numerical example of fatigue crack growth on the leading edge of an aircraft wing. The results for this example show that the proposed framework has the capabilities to: 1) track the evolution of time-dependent state variables (diagnosis); 2) reduce the uncertainty in time-independent state variables (diagnosis); and 3) probabilistically predict the crack growth in the future (prognosis).

CHAPTER 7

EFFICIENT GLOBAL SENSITIVITY ANALYSIS: A NEW SAMPLE-BASED ALGORITHM TO ESTIMATE THE FIRST-ORDER SOBOL' INDEX

7.1 Background

Section 2.6 provided an introduction to the global sensitivity analysis using Sobol' index. For a deterministic function $Y = F(\boldsymbol{X})$ where the input $\boldsymbol{X} = \{X_1, \dots, X_k\}$ is a vector of mutually independent random variables, the calculation of the first-order Sobol' index S_i is based on the following formula:

$$S_i = \frac{V_{X_i}\left(E_{\boldsymbol{X}_{-i}}(Y|X_i)\right)}{V(Y)} \tag{7.1}$$

where $\boldsymbol{X}_{-i}$ means all the model inputs other than X_i.

As pointed out in Section 2.6, computing S_i based on Eq. (7.1) is expensive since the numerator leads to a double-loop MCS. As shown in Eq. (2.16), the computational cost (number of functional evaluation) is $kn_{dl}^2 + n_{dl}$. This cost increases with n_{dl} and k, and is unaffordable if a single model evaluation is time-consuming or economically expensive, since n_{dl} is often of the order greater than 1000 in many practical applications.

Various algorithms have been proposed to reduce the computational cost of the Sobol' indices. These algorithms can be categorized into analytical methods and sample-based methods. In the analytical methods, the original model $Y = F(\boldsymbol{X})$ is generally approximated by some surrogate model of special form, so that the multi-dimensional integral can be converted into multiple univariate integrals, which can be easily calculated analytically or numerically. Zhang & Pandey [62] approximated the original model $Y = F(\boldsymbol{X})$ by a multiplication of univariate functions; then the univariate integral was calculated by Gaussian quadrature. Sudret [126] proposed that if the original model is approximated by a polynomial chaos expansion (PCE), the Sobol' index can be calculated by post-processing the PCE coefficients. Chen et al. [63] proposed another analytical method for commonly used surrogate models such as the linear regression model, Gaussian process model [30], Gaussian radial basis function model, and MARS model [127]; and analytical solution of the Sobol' index is available if the model inputs are normally or uniformly distributed. Analytical methods reduce the number of model evaluations significantly, but may require: 1) extra approximations and assumptions, and 2) extra computational cost in building the surrogate model.

Compared to the analytical methods, sample-based methods are more widely used [68,86,128–130] in engineering due to their simplicity in implementation. The basic sample-based method for GSA is the double-loop MCS, which has been explained earlier and often has prohibitive computational cost. Various efficient sample-based methods have been developed in the literature to reduce this cost. A brief review of these sample-based methods is given in Section 7.2. To the authors' knowledge, the computational cost (number of model evaluations) of most sample-based methods is proportional to the model input dimension k. Therefore the first objective of this research is to develop a more efficient sample-based method whose computational cost is not proportional to k, but much less.

A key assumption of the Sobol' index is the mutual independence of model inputs. With correlated model inputs, higher-order indices are no longer valid. However, Saltelli [64] pointed out that the first-order index S_i is still an informed choice to rank the importance of correlated model inputs, which has been explained in Section 2.6. Saltelli's paper [67] in 2002 mentioned that there is no alternative to the expensive double-loop MCS to compute S_i with correlated model inputs. The authors have not found any efficient algorithm in more recent studies, either. Thus the second objective of this research is to develop an efficient algorithm that can handle correlated model inputs.

The outline of this section is as follows. Section 7.2 briefly reviews existing sample-based methods for GSA, and discusses their computational cost. Section 7.3 illustrates the proposed modularized sample-based method that reduces the computational cost and handles correlated model inputs. Section 7.4 uses three numerical examples to compare the proposed method with existing methods.

7.2 Literature Review: Sample-based Methods

7.2.1 Sobol's Scheme

Consider a real integrable function $Y = F(\boldsymbol{X})$ where $\boldsymbol{X} = \{X_1, \dots, X_k\}$ is the vector of independent model inputs. Denote $\boldsymbol{Z} = \{Z_1, \dots, Z_k\}$ as the vector of the same independent model inputs, i.e., $Z_i (i = 1 \text{ to } k)$ and X_i are independently and identically distributed (i.i.d.). Sobol' [61] developed the following formula to compute the first-order index:

$$V_i = \int F(\boldsymbol{x}) F(X_i, \boldsymbol{Z}_{-i}) p(\boldsymbol{X}) p(\boldsymbol{Z}_{-i}) \mathrm{d}\boldsymbol{X} \mathrm{d}\boldsymbol{Z}_{-i} - E^2(Y) \tag{7.2}$$

where $p(\cdot)$ denotes the joint probability density function (PDF) of all the arguments, and it is the product of the PDFs of individual arguments under the assumption of independent model inputs. $\boldsymbol{Z}_{-i}$ are all the variables in $\boldsymbol{Z}$ other than Z_i.

Eq. (7.2) leads to the following estimator of V_i:

$$V_i = \frac{1}{n}\sum_{j=1}^{n} F(\boldsymbol{x}^j)F(x_i^j, \boldsymbol{z}_{-i}^j) - \left[\frac{1}{n}\sum_{j=1}^{n} F(\boldsymbol{x}^j)\right]^2 \tag{7.3}$$

Eq. (7.3) requires n_s samples of $\boldsymbol{X}$ and n_s samples of $\boldsymbol{Z}$, which are sampled independently from the distributions of the model inputs. In Eq. (7.3), the superscript j is the index of the samples and the subscript i is the index of model inputs. For example, $\boldsymbol{x}^j$ means the j-th sample of $\boldsymbol{X}$, and $\boldsymbol{z}_{-i}^j$ means the j-th sample of $\boldsymbol{Z}$ except Z_i. In Eq. (7.3), $F(\boldsymbol{x}^j)$ implies n_s model evaluations; $F(x_i^j, \boldsymbol{z}_{-i}^j)$ implies n_s model evaluations for each model input, i.e., kn_s evaluations for all the model inputs. To improve the accuracy, generally another n_s model evaluations are needed over the samples in $\boldsymbol{Z}$, and the results are used to estimate $V(Y)$ together with earlier evaluations over $\boldsymbol{X}$. The first-order index is calculated as $S_i = V_i/V(Y)$. The overall cost for all the first-order indices is $kn_s + 2n_s$.

Eq. (7.3) is the first efficient sample-based method to compute the first-order Sobol' index. Several methods have been proposed to improve its accuracy or reduce computational cost. Homma & Saltelli [131] suggested a more accurate estimator of V_i by using $\frac{1}{n}\sum_{j=1}^{n} F(\boldsymbol{x}^j)F(\boldsymbol{z}^j)$ to calculate $E^2(Y)$ instead of $\left[\frac{1}{n}\sum_{j=1}^{n} F(\boldsymbol{x}^j)\right]^2$. Thus Eq. (7.3) becomes [132]:

$$V_i = \frac{1}{n}\sum_{j=1}^{n} F(\boldsymbol{x}^j)\left[F(x_i^j, \boldsymbol{z}_{-i}^j) - F(\boldsymbol{z}^j)\right] \tag{7.4}$$

Compared to Eq. (7.3), Eq. (7.4) brings no extra model evaluation.

Sobol' & Myshetskaya [133] improved Eq. (7.4) further by replacing $F(\boldsymbol{x}^j)$ with $F(\boldsymbol{x}^j) - c$, where c is a constant equal or close to the true value of $E(Y)$. Thus Eq. (7.4) becomes:

$$V_i = \frac{1}{n}\sum\nolimits_{j=1}^{n}\left[F(\boldsymbol{x}^j) - c\right]\left[F\left(x_i^j, \boldsymbol{z}_{-i}^j\right) - F(\boldsymbol{z}^j)\right] \tag{7.5}$$

Eq. (7.5) brings no extra model evaluation either. In the numerical examples in Section 7.4, we define c as the mean value of Y over all samples of $\boldsymbol{X}$ and $\boldsymbol{Z}$.

In addition, another formula for V_i to improve the accuracy of small S_i is proposed by Owen [134]:

$$V_i = \frac{1}{n}\sum\nolimits_{j=1}^{n}\left[F(\boldsymbol{x}^j) - F\left(w_i^j, \boldsymbol{x}_{-i}^j\right)\right]\left[F\left(x_i^j, \boldsymbol{z}_{-i}^j\right) - F(\boldsymbol{z}^j)\right] \tag{7.6}$$

In Eq. (7.6) another i.i.d of $\boldsymbol{X}$ is denoted as $\boldsymbol{W}$, and a sample set of size n_s is generated for $\boldsymbol{W}$ so that w_i^j is the j-th sample of the i-th model input in this sample set. Eq. (7.6) proves to be more accurate in estimating small S_i; but no accuracy improvement is observed in estimating large S_i [134]. In addition, the term $F(w_i^j, \boldsymbol{x}_{-i}^j)$ in Eq. (7.6) brings n_s more model evaluations to estimate a single S_i.

More sample-based methods derived from Eq. (7.3) can be found in Refs. [67,135–137]. This research does not describe all these methods due to space limitations. Note that all the existing sample-based methods using the Sobol' scheme have a computational cost proportional to model inputs dimension k.

7.2.2 FAST Scheme

The FAST (Fourier amplitude sensitivity test) scheme includes two methods: classical FAST [138] and improved FAST [139] based on random balanced design [140]. The classical FAST was introduced in the 1970s, earlier than the introduction of Sobol' index. However, FAST estimates the equivalent of the first-order index defined in Eq. (7.1); thus the classical FAST is considered as an algorithm to compute the first-order index.

The classical FAST method assumes that any model input $X_i(i = 1 \text{ to } k)$ follows the standard uniform distribution $U(0,1)$, such that the domain of the model inputs is a unit hypercube $C^k = (\boldsymbol{X}|0 \leq X_i \leq 1; i = 1 \text{ to } k)$. This can be satisfied by converting all the model inputs into their CDF space.

Instead of directly generating random samples of $\boldsymbol{X}$ to fill in the sampling space C^k, FAST utilizes a curve to explore it. This curve is defined as:

$$X_i(s) = G_i(\sin \omega_i s) \; \forall \; i = 1 \text{ to } k \tag{7.7}$$

In Eq. (7.7), s varies in $[-\pi, \pi]$; ω_i is the angular frequency of X_i, set as linearly independent positive integers, and detailed strategy to select ω_i can be found in [141]; $G_i(\cdot)$ is a transfer function.

The curve in Eq. (7.7) explores the hypercube C^k as s changes. In other words, by generating samples of s from the uniform distribution $U(-\pi, \pi)$, the corresponding samples of $X_i(i = 1 \text{ to } k)$ can be obtained by Eq. (7.7). The resultant samples of $X_i(i = 1 \text{ to } k)$ should follow the uniform distribution $U(0,1)$, and the samples of X_i and $X_j(i \neq j)$ should be independent. These two objectives are achieved by the designed transfer function $G_i(\cdot)$. Different forms of $G_i(\cdot)$ have been proposed in Refs. [138,142,143].

Substituting Eq. (7.7) into the original model $Y = F(\boldsymbol{X})$ results in a new function of s denoted as $Y = F(\boldsymbol{X}(s))$, which can be expanded into a Fourier series. Then the total variance $V(Y)$ and the output variance caused by X_i itself are:

$$V(Y) = 2\sum_{p=1}^{+\infty}\Lambda_p, \quad V_i = 2\sum_{p=1}^{+\infty}\Lambda_{p\omega_i} \tag{7.8}$$

where p can be any positive integer; Λ_p and $\Lambda_{p\omega_i}$ are the Fourier spectrum at frequency p and $p\omega_i$, respectively. Eq. (7.8) means that V_i is related to the Fourier spectrum at the selected frequency ω_i and its higher harmonics $p\omega_i$.

In numerical computation, n_F samples of s are uniformly generated from $[-\pi, \pi]$, corresponding to n_F underlying samples of $\boldsymbol{X} = \{X_1, \dots, X_k\}$. The model $Y = F(\boldsymbol{X}(s))$ is evaluated at these samples to obtain the model output values, based on which the Fourier spectrum Λ_p can be computed by a numerical integral. Usually $\Lambda_{p\omega_i}$ is computed up to $H\omega_i$, where H is usually set to 4 or 6. Fourier coefficients at frequencies higher than $H\omega_i$ can be ignored in Eq. (7.8).

The computational cost of classical FAST is simply n_F, since the same model evaluation $F(\boldsymbol{X}(s))$ can be used to evaluate different V_i. However, n_F is constrained to a lower limit $n_F \geq 2\max(\omega_i) + 1$ [138]. According to the algorithm in Ref. [141] to select ω_i, $\max(\omega_i)$ increases with the input dimension k, thus the computational cost n_F also increases as k.

The improved FAST combines the classical FAST above with random balanced design. The improved FAST generates n_F samples of s with $\omega_i = 1 \; \forall \; i = 1 \text{ to } k$. Then the model $Y = F(\boldsymbol{X}(s))$ is evaluated n_F times to obtain the corresponding output values, denoted as $Y(s^j), j = 1 \text{ to } n_F$.

To obtain V_i, the output values $Y(s^j)$ are reordered as $Y^R(s^j)$ such that the corresponding values of X_i are ranked in increasing order. Then V_i is calculated in the same way as for the classical FAST approach by computing Fourier spectrum.

The improved FAST has no lower limit of sample size thus its computational cost n_F is not related to the model input dimension k. In addition, this method also achieves better accuracy [54,139,144] than the classical FAST.

7.3 Proposed Method

The motivation of the proposed method is rooted in the following challenge: with the input-output samples regarding a physics/computational model available, can we directly estimate the Sobol' index from the samples without more model evaluations? The intuitive answer should be yes, since the resultant input-output samples contain information about 1) the underlying input-output functional relationship, and 2) the underlying input/output distributions.

One GSA method based on the classical ANOVA using factorial design of experiments was proposed in [145]. If the random samples of each model input are considered as the levels of factors in the factorial design, this method gives the same result as the Sobol' index since the variance decomposition powering the Sobol' index is the same as that used in the classical ANOVA [146]. However, the factorial design in this method requires all possible combinations of the model input samples (levels) [145] and the corresponding model output samples, thus common MCS samples are not applicable.

In this research, a new sample-based method is proposed to resolve this challenge. Instead of modifying or improving the Sobol' scheme or the FAST approach, the proposed method is developed by analyzing the inner and outer loops of MCS in calculating the first-order index.

7.3.1 Algorithm 1

The proposed Algorithm 1 addresses the first-order index expression of Eq. (7.1), whose numerator includes the inner loop $E_{\boldsymbol{X}_{-i}}(Y|X_i)$ and the outer loop $V_{X_i}(\cdot)$.

Consider a model of $Y = F(\boldsymbol{X})$ where $\boldsymbol{X} = \{X_1, \dots, X_k\}$. We divide $\boldsymbol{X}$ into two sets: $\boldsymbol{X} = \{X_i, \boldsymbol{X}_{-i}\}$ where $\boldsymbol{X}_{-i}$ are the inputs other than X_i. $E_{\boldsymbol{X}_{-i}}(Y|X_i)$ is a function of X_i, and it can be proved that the mean value of Y over $= \{X_i, \boldsymbol{X}_{-i}\}$ is equal to the average of $E_{\boldsymbol{X}_{-i}}(Y|X_i)$ over X_i:

$$\begin{aligned} E_{X_i}\left(E_{\boldsymbol{X}_{-i}}(Y|X_i)\right) &= \int E_{\boldsymbol{X}_{-i}}(Y|X_i)\, p(X_i)\mathrm{d}X_i \\ &= \int \left(\int F(X_i, \boldsymbol{X}_{-i}) p(\boldsymbol{X}_{-i}|X_i)\mathrm{d}\boldsymbol{X}_{-i}\right) p(X_i)\mathrm{d}X_i \\ &= \int\int F(X_i, \boldsymbol{X}_{-i}) p(X_i, \boldsymbol{X}_{-i})\mathrm{d}X_i \mathrm{d}\boldsymbol{X}_{-i} = E(Y) \end{aligned} \tag{7.9}$$

Eq. (7.9) is called the law of total expectation and can be found in Ref. [147]. In Eq. (7.9), if X_i is constrained into a closed and bounded interval Φ, the distribution of X_i (and $\boldsymbol{X}_{-i}$ for correlated inputs) will change but Eq. (7.9) is still valid. In this case, based on the extreme value theorem [148], if $E_{\boldsymbol{x}_{-i}}(Y|X_i)$ is a continuous function of X_i in Φ, it must have a maximum value $\max\limits_{X_i \in \Phi}(E_{\boldsymbol{X}_{-i}}(Y|X_i))$ and a minimum value $\min\limits_{X_i \in \Phi}(E_{\boldsymbol{X}_{-i}}(Y|X_i))$ in Φ. The mean value of $E_{\boldsymbol{X}_{-i}}(Y|X_i)$, i.e., $E_\Phi(E_{\boldsymbol{X}_{-i}}(Y|X_i))$, is between these maximum and minimum values. Due to $X_i \in \Phi$, we denote the mean value of Y as $E_\Phi(Y)$. With $E_\Phi(Y) = E_\Phi(E_{\boldsymbol{X}_{-i}}(Y|X_i))$ proved in Eq. (7.9), we obtain

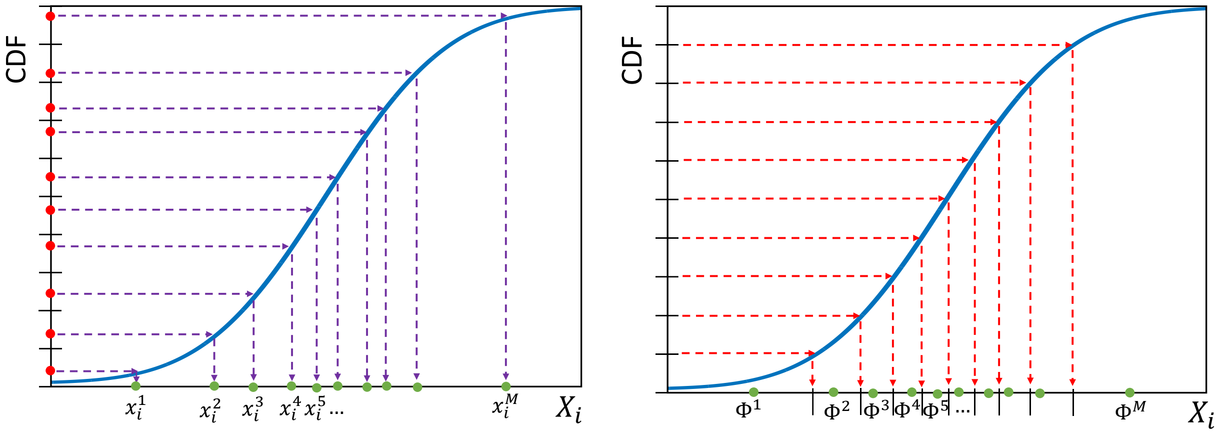
$$\min_{X_i \in \Phi}\left(E_{\boldsymbol{X}_{-i}}(Y|X_i)\right) \le E_\Phi(Y) \le \max_{X_i \in \Phi}\left(E_{\boldsymbol{X}_{-i}}(Y|X_i)\right) \tag{7.10}$$

Furthermore, since $E_{\boldsymbol{X}_{-i}}(Y|X_i)$ is a continuous function in Φ, the intermediate value theorem [148] proves that

$$\exists x_i^* \in \Phi \text{ such that } E_\Phi(Y) = E_{\boldsymbol{X}_{-i}}(Y|x_i^*) \tag{7.11}$$

Eq. (7.11) leads to the proposed Algorithm 1 if we design the interval Φ based on stratified sampling.

Stratified sampling generates samples in equal probability intervals to represent the distribution of a random variable X. Figure 7.1(a) shows one strategy [54] of stratified sampling: 1) divide the CDF of X_i into M intervals such that these intervals have the same length; 2) generate one sample u^l (the red dots in Figure 7.1(a), and $l = 1$ to M) from each CDF interval and obtain samples of X_i (the green dots in Figure 7.1) by CDF inversion $x_i^l = P^{-1}(u^l)$, where $P^{-1}(\cdot)$ is the inverse CDF of X_i. If we take the bounds of these intervals of the CDF as the inputs of $P^{-1}(\cdot)$, the sampling space of x_i is actually divided into M equally probable intervals $\Phi^l(l = 1 \text{ to } M)$, as shown in Figure 7.1(b), and x_i^l is actually a random sample generated within Φ^l.



(a) Stratified sampling

(b) Equally probably intervals of x_i

Figure 7.1 Stratified sampling and equally probably intervals

Consider the inner loop $E_{\boldsymbol{X}_{-i}}(Y|X_i)$ in Eq. (7.1) first. Assuming $\Phi = \Phi^l$, Eq. (7.11) proves that $\exists x_i^{l*} \in \Phi^l$ such that $E_{\boldsymbol{x}_{-i}}\left(Y|x_i^{l*}\right) = E_{\Phi^l}(Y)$, where $E_{\Phi^l}(Y)$ is the mean value of Y with $X_i \in \Phi^l$. In other words, calculating $E_{\Phi^l}(Y)$ is equivalent to fixing X_i at an unknown but existing point $x_i^{l*} \in \Phi^l$ and calculating the conditional mean value $E_{\boldsymbol{X}_{-i}}(Y|X_i = x_i^{l*})$.

The outer loop $V_{X_i}(\cdot)$ requires fixing X_i at different locations, and these selected locations are samples from the distribution of X_i. Based on stratified sampling, the set of these unknown but existing points $\boldsymbol{x}_i^* = \{x_i^{1*}, \dots, x_i^{M*}\}$ from the equally probable intervals $\boldsymbol{\Phi} = \{\Phi^1, \dots, \Phi^M\}$ can represent the distribution of X_i. As $E_{\Phi^l}(Y) = E_{\boldsymbol{X}_{-i}}(Y|X_i = x_i^{l*})$, the computation of S_i in the proposed Algorithm 1 is expressed as

$$S_i = \frac{V_{\boldsymbol{\Phi}}\left(E_{\Phi^l}(y)\right)}{V(y)} \tag{7.12}$$

where it numerator is the variance of $\{E_{\Phi^1}(Y), E_{\Phi^2}(Y), \dots, E_{\Phi^M}(Y)\}$. The steps to realize Algorithm 1 are listed in Section 7.3.3.

7.3.2 Algorithm 2

Based on the law of total variance

$$V(Y) = E_{X_i}\left(V_{\boldsymbol{X}_{-i}}(Y|X_i)\right) + V_{X_i}\left(E_{\boldsymbol{X}_{-i}}(Y|X_i)\right) \tag{7.13}$$

Eq. (7.1) can be rewritten as

$$S_i = 1 - \frac{E_{X_i}\left(V_{\boldsymbol{X}_{-i}}(Y|X_i)\right)}{V(Y)} \tag{7.14}$$

The proposed Algorithm 2 is regarding this equivalent first-order Sobol' index expression in Eq. (7.14), whose numerator implies an expensive double-loop Monte Carlo simulation including the inner loop $V_{\boldsymbol{X}_{-i}}(Y|X_i)$ and the outer loop $E_{X_i}(\cdot)$. Its inner loop part $V_{\boldsymbol{X}_{-i}}(Y|X_i)$ is a function of X_i. Assume $X_i \in \Phi$, where Φ can be the entire sampling space of X_i or only a small interval. Based on the extreme value theorem, $V_{\boldsymbol{X}_{-i}}(Y|X_i)$ must have a maximum value and a minimum value in

Φ. The mean value of $V_{\boldsymbol{X}_{-i}}(Y|X_i)$, i.e., $E_\Phi(V_{\boldsymbol{X}_{-i}}(Y|X_i))$ for $X_i \in \Phi$, is between these maximum and minimum values:

$$\min_{X_i \in \Phi}(V_{\boldsymbol{X}_{-i}}(Y|X_i)) \leq E_\Phi(V_{\boldsymbol{X}_{-i}}(Y|X_i)) \leq \max_{X_i \in \Phi}(V_{\boldsymbol{X}_{-i}}(Y|X_i)) \tag{7.15}$$

Then the intermediate value theorem proves that

$$\exists x_i^\# \in \Phi \text{ s.t.} V_{\boldsymbol{X}_{-i}}\left(Y|x_i^\#\right) = E_\Phi(V_{\boldsymbol{X}_{-i}}(Y|X_i)) \tag{7.16}$$

With $X_i \in \Phi$, we rewrite the law of total variance in Eq. (7.13) as:

$$V_\Phi(Y) = E_\Phi\left(V_{\boldsymbol{X}_{-i}}(Y|X_i)\right) + V_\Phi\left(E_{\boldsymbol{X}_{-i}}(Y|X_i)\right) \tag{7.17}$$

where the subscript Φ means all the terms are constrained to $X_i \in \Phi$. Substituting Eq. (7.17) into Eq. (7.16) and assuming $\Phi = \Phi^l$ as one of the equally probable intervals in stratifying sampling, we can have

$$\exists x_i^{l\#} \in \Phi^l \text{ s.t.} V_{\boldsymbol{X}_{-i}}\left(Y|x_i^{l\#}\right) = V_{\Phi^l}(Y) - V_{\Phi^l}(E_{\boldsymbol{X}_{-i}}(Y|X_i)) \tag{7.18}$$

where $x_i^{l\#}$ is an unknown but existing point in Φ^l. Note that now $V_{\Phi^l}(Y)$ is the variance of Y given $X_i \in \Phi^l$ and $V_\Phi(E_{\boldsymbol{X}_{-i}}(Y|X_i))$ is the variance of $E_{\boldsymbol{X}_{-i}}(Y|X_i)$ given $X_i \in \Phi^l$.

The outer loop $E_{X_i}(\cdot)$ requires fixing X_i at different locations, and these selected locations are samples from the distribution of X_i. Based on stratified sampling, the set of these unknown but existing points $\boldsymbol{x}_i^\# = \left\{x_i^{1\#}, \ldots, x_i^{M\#}\right\}$ from the equally probable intervals $\boldsymbol{\Phi} = \{\Phi^1, \ldots, \Phi^M\}$ can represent the distribution of X_i. Based on Eqs. (7.14) and (7.18), computation of S_i is expressed as

$$
\begin{aligned}
S_i = 1 - \frac{E_{\boldsymbol{\Phi}}\left(V_{\boldsymbol{X}_{-i}}\left(Y|x_i^{l\#}\right)\right)}{V(Y)} &= 1 - \frac{E_{\boldsymbol{\Phi}}\left(V_{\Phi^l}(Y) - V_{\Phi^l}(E_{\boldsymbol{X}_{-i}}(Y|X_i))\right)}{V(Y)} \\
&= 1 - \frac{E_{\boldsymbol{\Phi}}\left(V_{\Phi^l}(Y)\right)}{V(Y)} + \frac{E_{\boldsymbol{\Phi}}\left(V_{\Phi^l}(E_{\boldsymbol{X}_{-i}}(Y|X_i))\right)}{V(Y)}
\end{aligned}
\tag{7.19}
$$

On the right-hand side of Eq. (7.19), the first term is a known constant 1; the second term can be directly computed using the Monte Carlo samples, following the steps given later in Figure 7.2; the third term is still a challenge but we can prove that this term can be ignored by rewriting it as:

$$
\frac{E_{\boldsymbol{\Phi}}\left(V_{\Phi^l}(E_{\boldsymbol{X}_{-i}}(Y|X_i))\right)}{V(Y)} = E_{\boldsymbol{\Phi}}\left(\frac{V_{\Phi^l}(E_{\boldsymbol{X}_{-i}}(Y|X_i))}{V_{\Phi^l}(Y)} \frac{V_{\Phi^l}(Y)}{V(Y)}\right) \tag{7.20}
$$

In Eq. (7.20), the term $V_{\Phi^l}(E_{\boldsymbol{X}_{-i}}(Y|X_i))/V_{\Phi^l}(Y) = S_i^{\Phi^l}$ is nothing but the first-order sensitivity of X_i as it is restricted to the interval Φ^l. We always have $S_i^{\Phi^l} < S_i$ since the uncertainty of X_i has been reduced significantly by restricting it in Φ^l such that its sensitivity index will be much lower. The other term in Eq. (7.20) $V_{\Phi^l}(Y)/V(Y)$ is the ratio of 1) the variance of Y as X_i is restricted to the interval Φ^l and 2) the overall variance of Y.

If X_i has a high sensitivity index S_i close to one, restricting it to Φ^l will reduce the variance of y significantly such that $V_{\Phi^l}(Y)/V(Y)$ will be close to zero; meanwhile $S_i^{\Phi^l}$ will be also smaller than S_i. Overall, their product will be close to zero.

If X_i has a low sensitivity index S_i closer to zero, restricting it to Φ^l will NOT reduce the variance of Y significantly such that $V_{\Phi^l}(Y)/V(Y)$ will be close to 1; however, we always have $S_i^{\Phi^l} < S_i$ so that $S_i^{\Phi^l}$ is closer to zero. Overall, their product will be close to zero.

In sum, no matter whether S_i is closer to zero or one, Eq. (7.20) is always a small value close to zero, and this value will reduce further as the number of intervals M increases, since in that case Φ will be narrower so that . Thus Eq. (7.19) can be approximated as

$$S_i \approx 1 - \frac{E_{\mathbf{\Phi}}\left(V_{\Phi^l}(Y)\right)}{V(Y)} \tag{7.21}$$

Eq. (7.21) is the proposed Algorithm 2, and the steps to realize it are listed in Section 7.3.3.

7.3.3 Implementation and Benefits of the Proposed Method

The innovation in the proposed methods is that the inner loop $E_{\boldsymbol{X}_{-i}}(Y|X_i)$ or $V_{\boldsymbol{X}_{-i}}(Y|X_i)$ is not conditioned on an explicit sample of X_i selected by the user, but on an unknown but existing point. The first-order index S_i is obtained without knowing the value of this existing point. The benefits of the proposed methods can be observed from the following steps to realize Eqs. (7.12) and (7.21):

1. Generate n_M random samples of $\boldsymbol{X}$;
2. Obtain corresponding values of Y by evaluating $Y = F(\boldsymbol{X})$, and estimate $V(Y)$ using all samples of Y;
3. Divide the domain of X_i into M equally probable intervals, as shown in Figure 7.1;
4. Assign the samples of Y into divided intervals based on one-to-one mapping between the samples of X_i and samples of Y;
5. For Algorithm 1, estimate $E_{\Phi^l}(Y)$ as the sampling mean of Y in each interval; for Algorithm 2, estimate $V_{\Phi^l}(Y)$ as the sampling variance of Y in each interval;
6. For Algorithm 1, estimate $V_{\mathbf{\Phi}}(E_{\Phi^l}(Y))$ as the sampling variance of $E_{\Phi^l}(Y)$ in step 5; for Algorithm 2, estimate $E_{\mathbf{\Phi}}(V_{\Phi^l}(Y))$ as the sampling mean of $V_{\Phi^l}(Y)$ in step 5;
7. $S_i = V_{\mathbf{\Phi}}(E_{\Phi^l}(Y))/V(Y)$ for Algorithm 1 and $S_i = 1 - E_{\mathbf{\Phi}}(V_{\Phi^l}(Y))/V(Y)$ for Algorithm 2.

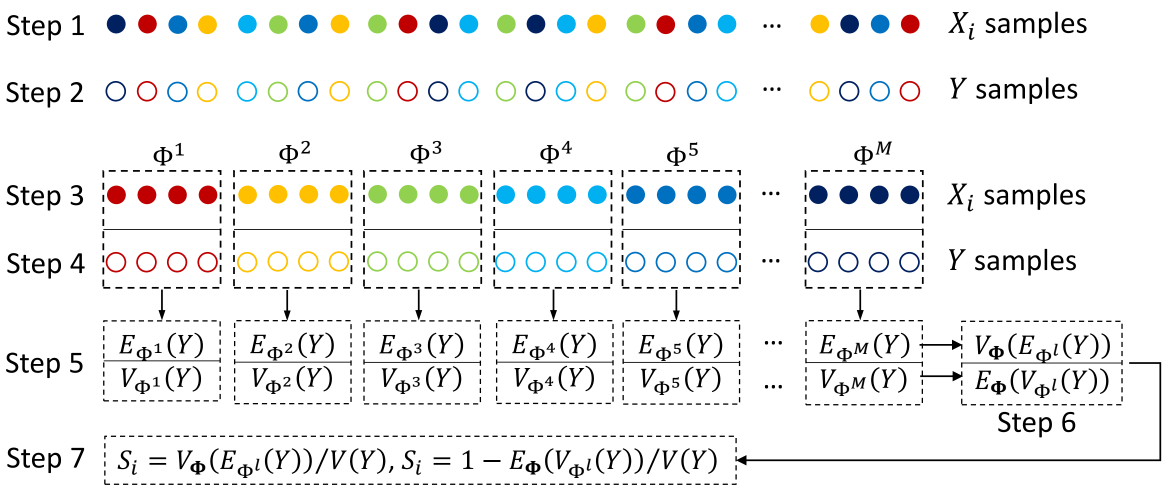


Figure 7.2 Steps to realize the proposed method

The steps to realize the proposed method are also illustrated in Figure 7.2, where samples in different equally probable intervals are represented in different colors. These steps indicate that the proposed methods are modularized in two aspects. First, Steps 3 and 4 show that the samples of $\boldsymbol{X}_{-i}$ are not used in calculating the index S_i for X_i, so that index calculations for different model inputs are separated. Therefore the calculation of S_i purely depends on the samples of X_i and Y, and can be achieved even if the samples of $\boldsymbol{X}_{-i}$ are missing. Second, model inputs sampling, model evaluation, and index calculation are separate processes. The computational cost of most existing sample-based methods is proportional to the model inputs dimension k because each input needs new samples to calculate its Sobol' index. In comparison, the computational cost of the proposed method is not proportional to k because each input uses the same samples to calculate its Sobol' index. Therefore in the proposed method the accuracy of the resultant Sobol' index only relies on the number of samples n_M and the selected value of M, but not dependent on k.

Another benefit brought by this modularization is that the input-output samples in step 1 can be from other uncertainty quantification activities. It provides a solution of sensitivity analysis when input-output samples are available but the underlying model is not available or too expensive for re-running.

One very important benefit of the proposed methods is that the derivation of these proposed algorithms does not assume independent model inputs. Thus the proposed methods can handle both independent and correlated model inputs. To the authors' knowledge, the proposed method is the only available alternative so far to the costly double-loop MCS method when the model inputs are correlated.

7.3.4 Accuracy Comparison: Algorithm 1 vs. Algorithm 2

For a given set of input-output samples, the factor affecting the implementation of the proposed method is M, the number of equally probable intervals used to stratify the samples. This section identifies the effect of M on the proposed algorithms and compares their accuracy. The algorithm found to be better will then be used to compare against existing methods.

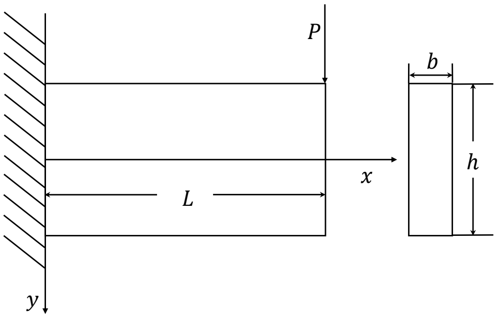


Figure 7.3 Thick cantilever beam

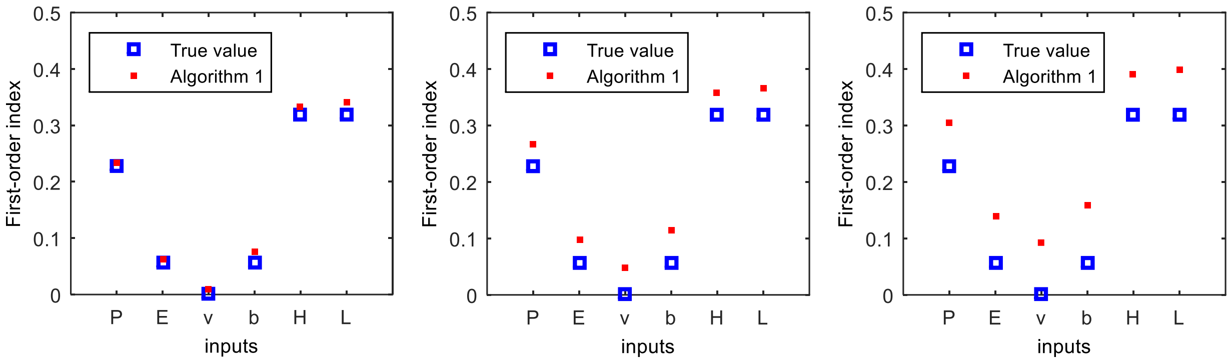
First, Algorithms 1 and 2 are compared by an illustrative example of a thick cantilever beam shown in Figure 7.3. This example computes the beam's tip deflection along the y-axis using the Timoshenko beam theory [149]:

$$u = \frac{P}{6EI}\left[(4+5\nu)\frac{h^2L}{4} + 2L^3\right] \tag{7.22}$$

where $I = bh^3/12$. The statistics of other model inputs in Eq. (7.22) are listed in Table 7.1.

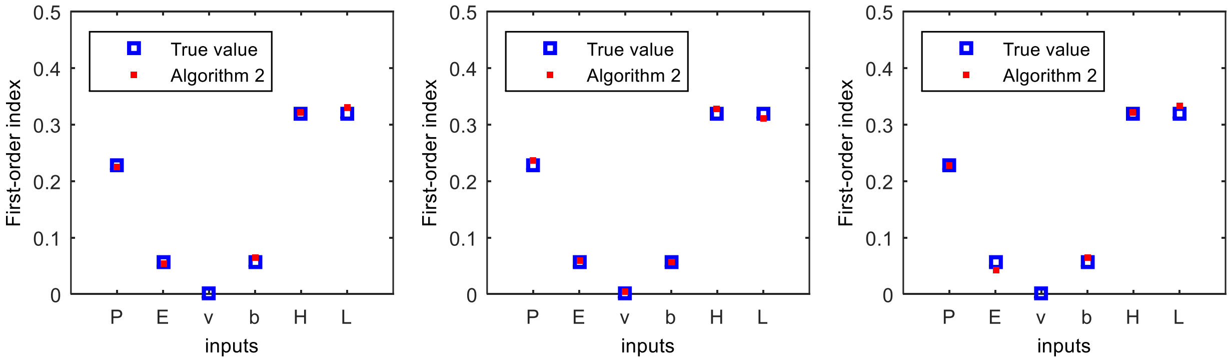
Table 7.1 Statistics of model inputs in the cantilever beam example

Input	Load	Young's modulus	Poisson's ratio	Width	Height	Length
Symbol/Unit	P/kN	E/GPa	ν	b/mm	h/mm	L/mm
Distribution type	Normal	Lognormal	Lognormal	Lognormal	Lognormal	Lognormal
Mean	2.5	200	0.225	1.0	3.0	3.5
COV	0.1	0.05	0.05	0.05	0.05	0.05



(a) $\boldsymbol{M = 100}$ **(b) $\boldsymbol{M = 500}$** **(c) $\boldsymbol{M = 1000}$**

Figure 7.4 Algorithm 1, Cantilever beam example



(a) $\boldsymbol{M = 100}$ **(b) $\boldsymbol{M = 500}$** **(c) $\boldsymbol{M = 1000}$**

Figure 7.5 Algorithm 2, Cantilever beam example

The proposed two algorithms are used to calculate the first-order index using 10^4 MCS samples. Results for different interval numbers M are shown in Figure 7.4 and Figure 7.5, where the "True value" is estimated by the costly double-loop MCS method with $n_{dl} = 10^4$.

Figure 7.4 clearly shows that Algorithm 1 tends to overestimate the first-order index, especially at large M values; while Figure 7.5 shows that Algorithm 2 is more robust and reveals adequate accuracy at different values of M. This observation can be explained by analyzing the numerical errors in implementing Algorithms 1 and 2.

In Algorithm 1, assume that the true mean value of Y in the l-th interval Φ^l is $\mu^l(i = 1 \text{ to } M)$ while the estimated sampling mean value is $\bar{y}^l$. We denote $\bar{y}^l = \mu^l + d^l$ where d^l is the bias due to limited samples in Φ^l. At given M, the best estimate for $V_{X_i}(E_{\boldsymbol{X}_{-i}}(Y|X_i))$ is

$$\hat{V} = \frac{1}{M-1}\sum_{l=1}^{M}(\mu^l - \bar{\mu})^2 \tag{7.23}$$

where $\bar{\mu}$ is the mean value of μ^l, i.e., $\bar{\mu} = (\sum_{l=1}^{M}\mu^l)/M$. This $\hat{V}$ approximates the desired $V_{X_i}(E_{\boldsymbol{X}_{-i}}(Y|X_i))$ well if M is large enough. However, $\hat{V}$ can be obtained only if $\mu^l\ \forall\ l = 1$ to M is estimated correctly, which cannot be achieved due to the numerical errors. Denoting $\bar{\bar{y}} = (\sum_{l=1}^{M}\bar{y}^l)/M$ and $\bar{d} = (\sum_{l=1}^{M}d^l)/M$, the actual estimate of $V_{X_i}(E_{\boldsymbol{X}_{-i}}(Y|X_i))$ is

$$\tilde{V} = \frac{1}{M-1}\sum_{l=1}^{M}(\bar{y}^l - \bar{\bar{y}})^2 = \frac{1}{M-1}\sum_{l=1}^{M}\left(\mu^l + d^l - \bar{\mu} - \bar{d}\right)^2 \tag{7.24}$$

The bias of $\tilde{V}$ from $\hat{V}$ is

$$\begin{aligned}
\tilde{V} - \hat{V} &= \frac{1}{M-1}\left[\sum_{l=1}^{M}\left(\mu^l + d^l - \bar{\mu} - \bar{d}\right)^2 - \sum_{l=1}^{M}\left(\mu^l - \bar{\mu}\right)^2\right] \\
&= \frac{1}{M-1}\left[\sum_{l=1}^{M}\left(2\mu^l + d^l - 2\bar{\mu} - \bar{d}\right)\left(d^l - \bar{d}\right)\right] \\
&= \frac{2\left(\mu^l - \bar{\mu}\right)}{M-1}\sum_{l=1}^{M}(d^l - \bar{d}) + \frac{1}{M-1}\sum_{l=1}^{M}\left(d^l - \bar{d}\right)^2 \\
&= \frac{1}{M-1}\sum_{l=1}^{M}\left(d^l - \bar{d}\right)^2 > 0
\end{aligned} \tag{7.25}$$

Eq. (7.25) clearly indicates that $\tilde{V}$ is a positively biased estimate of $\hat{V}$, where $\hat{V}$ is used to approximate the desired term $V_{X_i}(E_{\boldsymbol{X}_{-i}}(Y|X_i))$. Due to the square term in the last line of Eq. (7.25), this bias tends to increase as M becomes large. This explains why Algorithm 1 overestimates the first-order indices in Figure 7.4 and why this overestimation increases with M.

In Algorithm 2, assume that the true variance of y in the l-th interval Φ^l is $S^l(i = 1 \text{ to } M)$ while the estimated sampling mean value is V^l. We denote $V^l = S^l + \delta^l$ where δ^l is the bias due to limited sample in Φ^l. At given M, the best estimate of $E_{X_i}(V_{\boldsymbol{X}_{-i}}(Y|X_i))$ is $\hat{E} = (\sum_{l=1}^{M} S^l)/M$ while the actual estimate is $\tilde{E} = (\sum_{l=1}^{M} V^l)/M$. The bias of $\tilde{E}$ from $\hat{E}$ is

$$\tilde{E} - \hat{E} = \frac{1}{M}\sum_{l=1}^{M}(V^l - S^l) = \frac{1}{M}\sum_{l=1}^{M}\delta^l \tag{7.26}$$

The bias from Eq. (7.26) is around zero since δ^l can be randomly positive or negative. Then $\tilde{E}$ is an unbiased estimate of $\hat{E}$, where $\hat{E}$ is used to approximate the desired term $E_{X_i}(V_{\boldsymbol{X}_{-i}}(Y|X_i))$. This explains why Algorithm 2 estimates the first-order index accurately at different values of M. In conclusion, Algorithm 2 is more accurate and robust than Algorithm 1. Note that Eqs.(7.25) and (7.26) which compare the accuracy of Algorithms 1 and 2 are general; the cantilever beam example was only for illustrative purposes.

In this cantilever beam example, Figure 7.5 proves the robustness of Algorithm 2 at different values of M. Section 7.4.4 will give a detailed discussion on the selection of M based on another three numerical examples and provide an empirical instruction in selecting M.

7.3.5 Extension of the Proposed Method

Theoretically, the proposed Algorithms 1 or 2 can be extended to estimate higher-order Sobol' index. For example, the formula for the second-order Sobol' index of X_i and X_j $(i \neq j)$ is[54]

$$S_{ij} = \frac{V\left(E\left(Y|X_i, X_j\right)\right)}{V(Y)} - S_i - S_j \tag{7.27}$$

where S_i and S_j are given by the proposed method.

Similar to Algorithm 1, $V(E(Y|X_i, X_j))$ in Eq. (7.27) can be estimated by: 1) dividing X_i and X_j into equally probable intervals; and 2) $V(E(Y|X_i, X_j)) = V_{\mathbf{\Phi}}(E_{\Phi^l}(Y))$ where $\Phi^l(l = 1 \text{ to } M)$ represents a 2-dimensional equally probable interval, instead of a 1-dimensional interval in calculating S_i. In general, Sobol' index of order D requires calculating $V(E(Y|X_{i_1}, X_{i_2}, \dots, X_{i_D}))$, meaning that Φ^l is an interval in a D-dimensional sampling space. The required number of intervals to fill this space, i.e., the value of M, increases with D. For a given number of samples, this means less samples in a single interval and increased numerical error in estimating $V_{\mathbf{\Phi}}(E_{\Phi^l}(Y))$. In the worst case, some intervals may not contain any sample at all. In conclusion, extending the proposed method to higher-order indices is theoretically possible, but much larger numbers of samples are needed for accurate results. Therefore, this research only focuses on the first-order index.

7.3.6 Summary

Section 7.3 proposed two new algorithms to calculate the first-order Sobol' index. The main innovation is that the conditional mean value $V_{\boldsymbol{X}_{-i}}(Y|X_i)$ or the conditional variance $E_{\boldsymbol{X}_{-i}}(Y|X_i)$ is no more conditioned on a user-defined location but an unknown existing location of X_i. This innovation enables the proposed algorithms to directly estimate the Sobol' index from the input-output samples, and reuse the same samples to compute the indices of different input. This section also proves that Algorithm 2 is more accurate and robust than Algorithm 1. Therefore in the next section Algorithm 2 is selected to compete with existing methods. The proposed method has the following advantages:

1. Less computational effort than most existing sample-based methods in Section 7.2, since its computational cost is not proportional to the model input dimension.
2. Handling correlated model inputs, which is an advantage over both the existing sample-based methods and the existing analytical methods such as the M-DRM algorithm in [62].
3. Capability to compute the first-order index if input-output samples have been generated but the underlying model is not available or too expensive for re-running, and this is also an advantage over both the existing sample-based methods and the existing analytical methods. In fact, the first-order Sobol' index of X_i can be computed by the proposed method even if the samples of $\boldsymbol{X}_{-i}$ are missing.

The only parameter to be tuned in the proposed method is the number of equally probable intervals. The selection of this parameter will be discussed in Section 7.4.4.

Note that analytical methods such as the M-DRM algorithm [62] are more efficient and use less model evaluations than the proposed method. However, these methods need a mathematical model that connects the input to the output so that the users can run the functions at some specific values; whereas our proposed method works directly with the input-output samples, which might have been collected from tests or field observations. As pointed out in the abstract, the main focus of this research is to extract Sobol' index from the samples directly. Another difference is that the analytical methods need independent inputs, whereas our method is applicable also with correlated inputs.

7.4 Numerical Examples

The objective of this section is to compare the performance of the proposed method against existing sample-based methods. The proposed Algorithm 2 is used in the comparison since Section 7.3.4 has shown that this algorithm is more accurate than Algorithm 1. Three numerical examples are used for comparison: 1) a low-dimensional classical non-smooth function; 2) a high-dimensional linear function; and 3) a cantilever beam problem with correlated model inputs. The comparison is conducted under the same computational cost, i.e., the same number of model evaluations.

For examples 1 and 2, the selected existing methods are 1) Sobol' method in Eq. (7.5), and 2) improved FAST. As a representative of the Sobol' scheme, Sobol' method in Eq. (7.5) is selected due to its higher accuracy than Eqs. (7.3) and (7.4), and lower computational cost than Eq. (7.6). The improved FAST method is selected due to its advantage of higher accuracy and lower cost than the classical FAST.

For example 3, the selected existing method is the costly double-loop MCS since other advanced methods are only suitable for independent model inputs.

Except the improved FAST, other methods (Sobol' method, double-loop MCS, and proposed method) in this section require random samples of model inputs. To achieve a comparison of best possible performance, this section uses Latin hypercube sampling to generate these random samples. Latin hypercube sampling fills the model input sampling space more evenly and improves the computational accuracy at given cost [55,139,140].

7.4.1 Example 1: Non-smooth function

The classical non-smooth function proposed by Sobol' [61] and widely used in the literature [79,133,142,143] is considered in Example 1, as

$$Y = \prod_{i=1}^{k} \frac{|4X_i - 2 + a_i|}{a_i + 1} \tag{7.28}$$

where X_i $(i = 1 \text{ to } k)$ are independent model inputs, each following a standard uniform distribution $U(0,1)$; and a_i $(i = 1 \text{ to } k)$ are user-defined constants. An analytical expression of the first-order index is available for this function:

$$\begin{gathered} V(y) = -1 + \prod_{i=1}^{k} \left[\frac{1}{3(a_i + 1)^2} + 1\right] \\ S_i = \frac{1}{V(y)} \cdot \frac{1}{3(a_i + 1)^2} \end{gathered} \tag{7.29}$$

Eq. (7.29) indicates that a smaller value of a_i corresponds to a larger first-order index. Here we define a 4-dimentional function ($k = 4$) with $a_i = i$ such that $S_1 > S_2 > S_3 > S_4$.

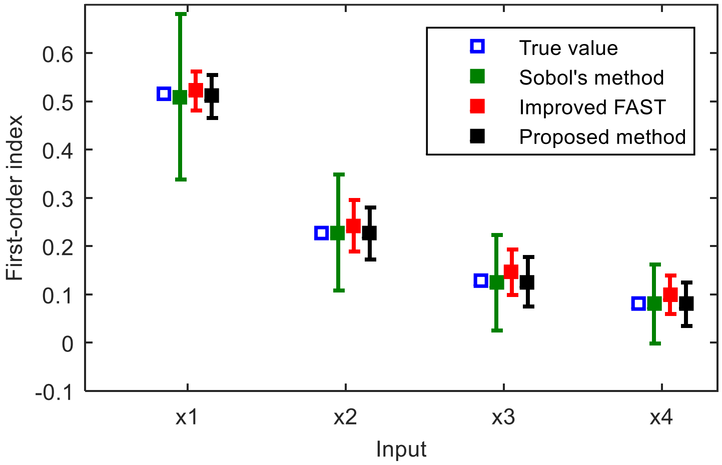


Figure 7.6 First-order index of the non-smooth function

Comparison of the three methods is shown in Figure 7.6. The true values are based on Eq. (7.29). And the 95% confidence intervals for the three methods are based on 1000 runs. For each method, a single run should spend the same computational cost of model evaluations to achieve a fair comparison. The computational cost of Sobol' method is $kn_s + 2n_s$ where $k = 4$ in this example and n_s is number of samples to calculate a single index S_i. Here we use $n_s = 100$ thus the computational cost of the Sobol' method is 600 model evaluations. To achieve a fair comparison, we also use n_M =600 samples in the improved FAST method and the proposed method. In this example the number of equally probable intervals is $M = [\sqrt{n_M}] = 24$, i.e., the square root of n_M rounded to the nearest integer. A detailed discussion on the selection of M can be found in Section 7.4.4.

Sobol' method is expected to be less accurate, since it will only use 100 samples to compute the first-order index of each individual variable, but the other two methods use all the 600 samples to compute the first-order index of each individual variable. This is confirmed by the wider confidence interval for the Sobol' method in Figure 7.6.

In contrast to the Sobol' method, the improved FAST and the proposed method reduce the confidence interval by over 50%. However, the improved FAST tends to slightly overestimate the first-order indices in this example. This is probably due to the limited Fourier spectrum order ($H = 6$ here). The indices estimated by the proposed method (Algorithm 2) show excellent agreement with the true values.

Note that analytical methods may solve the same problem using less functional evaluations. For instance, the M-DRM algorithm [62] can compute the Sobol' indices of a higher order ($k = 8$) non-smooth function with only 81 model evaluations. The example here is to test the validity of the proposed method and prove its advantage in reducing computational cost and improving accuracy compared to other sample-based methods. Compared to analytical methods, the advantages of the proposed methods are: 1) no approximation in the model of interest; 2) calculation of Sobol' indices if input-output samples are available but the model is not; and 3) handling problems with correlated model inputs.

7.4.2 Example 2: High-dimensional Linear Function

The computational cost in the improved FAST and the proposed method is not proportional to the model input dimension. This advantage is more prominent in high-dimensional problems. Consider a 50-dimensional linear function $Y = \sum_{i=1}^{50} b_i X_i$ where $b_i = 1 + i/50$ and X_i are independent model inputs of standard normal distribution. For this example, the true value of the first-order index has analytical solution $S_i = b_i^2 / \sum_{i=1}^{50} b_i^2$.

The results of the three methods are shown in Figure 7.7, where the 95% confidence intervals for the three methods are based on 1000 runs. For each method, a single run should use the same computational cost of model evaluations to achieve a fair comparison. The computational cost of Sobol' method is $kn_s + 2n_s$ where $k = 50$ in this example and n_s is number of samples to calculate a single index S_i. Here we use $n_s = 200$ thus the computational cost of the Sobol' method is 10400 model evaluations. To achieve a fair comparison, we also use n_M =10400 samples in the improved FAST method and the proposed method. Similar to the non-smooth function example, the number of equally probable intervals is $M = [\sqrt{n_M}] = 102$. A detailed discussion on the selection of M can be found in Section 7.4.4.

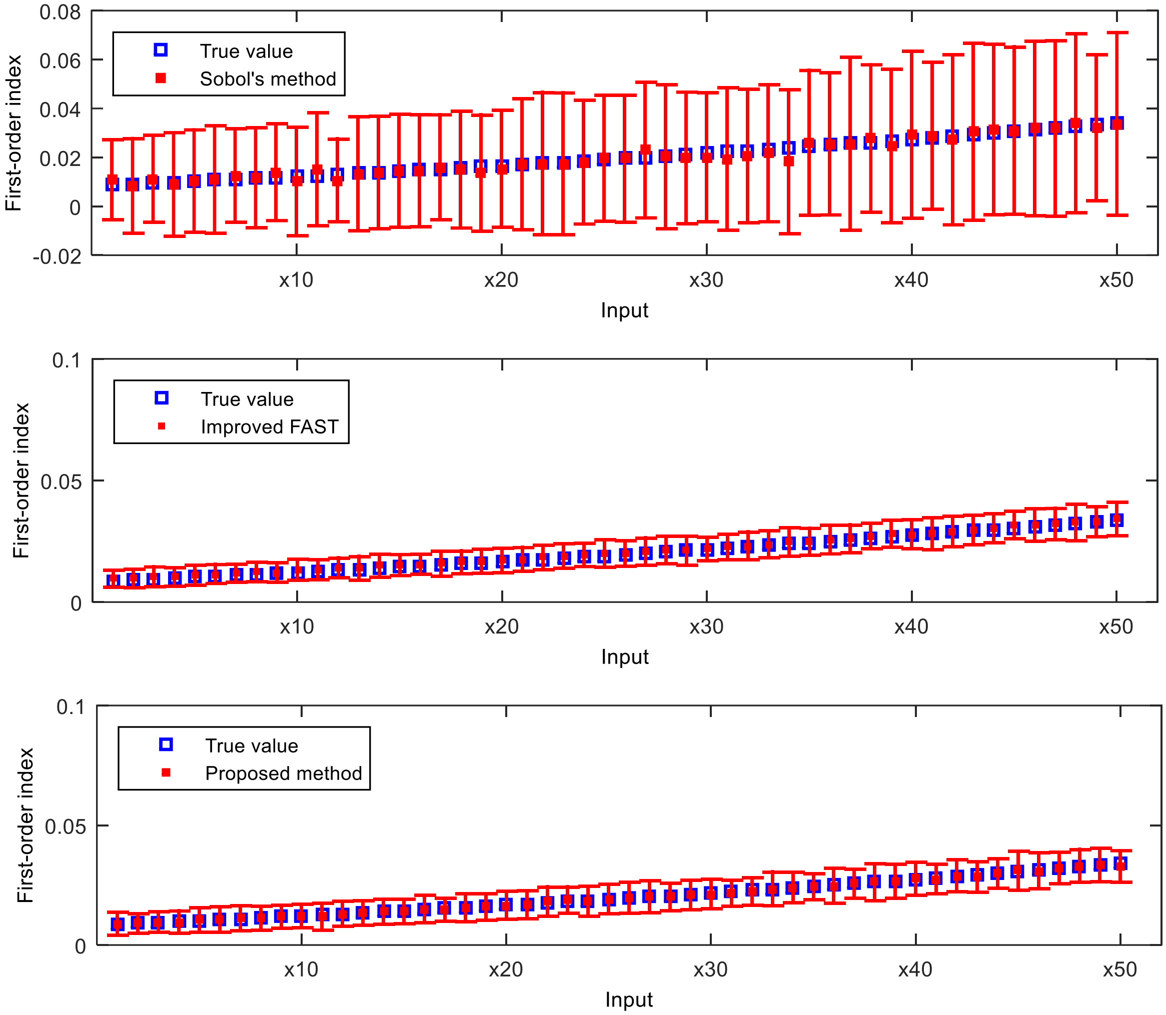


Figure 7.7 First-order index of the linear function

In Figure 7.7, the improved FAST and the proposed method show comparable performance. The improved FAST method still slightly overestimates the first-order indices in this example. The improved FAST method and the proposed method reduce the confidence interval width by around 80% in contrast to Sobol' method.

7.4.3 Example 3: Cantilever Beam with Correlated Inputs

Examples 1 and 2 show that the improved FAST and the proposed method perform equally well and outperform the Sobol' method. Model inputs are independent in examples 1 and 2. However, advanced methods such as the improved FAST method are no more valid for correlated model inputs and the costly double-loop MCS is the only existing option. The proposed method provides an alternative to compute the first-order index with correlated model inputs. The example in this section illustrates this unique benefit of the proposed method. Other benefits have been discussed in Section 7.3.3.

Consider the cantilever beam example in Section 7.3.4 again. Here the model inputs are assumed to follow correlated normal distributions. Mean values and standard deviations of the model inputs are listed in Table 7.2 and their correlation matrix is shown in Table 7.3.

Table 7.2 Statistics of model inputs

Model input	P/kN	E/GPa	ν	b/mm	h/mm	L/mm
Mean value	2.5	200	0.225	1.0	3	3.5
Standard deviation	0.25	20	0.0225	0.1	0.3	0.35

Table 7.3 Correlation matrix of model inputs

Model input	P/kN	E/GPa	ν	b/mm	h/mm	L/mm
P/kN	1.000	0.174	0.451	0.082	-0.134	0.004
E/GPa	0.174	1.000	-0.800	0.059	-0.125	-0.082
ν	0.451	-0.800	1.000	-0.004	0.033	0.080
b/mm	0.082	0.059	-0.004	1.000	-0.105	-0.400
h/mm	-0.134	-0.125	0.033	-0.105	1.000	0.279
L/mm	0.004	-0.082	0.080	-0.400	0.279	1.000

The results of the double-loop MCS method and the proposed method are shown in Figure 7.8, where the 95% confidence intervals for the two methods are based on 1000 runs. For each method, a single run should use the same computational cost of model evaluations to achieve a fair comparison. The computational cost of the double-loop MCS method is $kn_{dl}^2 + n_{dl}$ where $k = 6$ in this example and n_{dl} is number of samples to calculate a single index S_i. Here we use $n_{dl} = 50$ thus the computational cost of the double-loop MCS method is 15050 model evaluations. To achieve the fair comparison, we also use $n_M =$15050 MCS samples in the proposed method.

Similar to the other two examples, the number of equally probable intervals is $M = [\sqrt{n_M}] = 123$. A detailed discussion on the selection of M can be found in Section 7.4.4.

The true values in Figure 7.8 are approximated by an extreme expensive double-loop MCS with $n_{dl} = 10^4$, whose total cost is more than 6×10^8 model evaluations. Figure 7.8 shows that: 1) the proposed method is very accurate for correlated model inputs; and 2) compared to the double-loop MCS, the proposed method narrows the confidence intervals by 80%~95% for the same number of model evaluations.

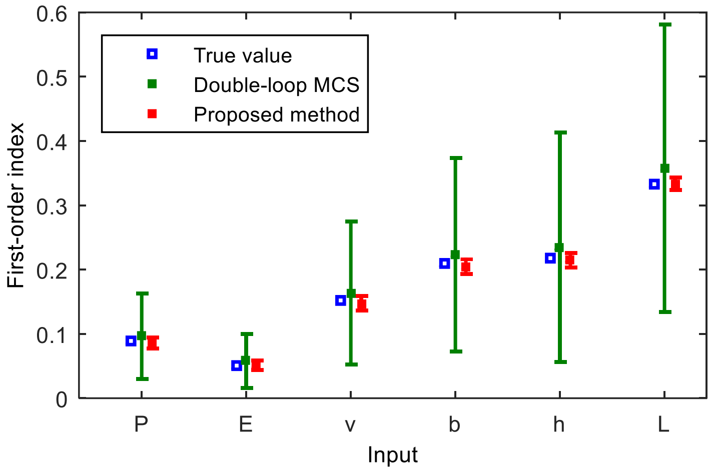


Figure 7.8 First-order index of the cantilever beam example with correlated inputs

7.4.4 Discussion: Selection of M

At a given number of input-output samples (n_M is fixed), the only parameter to be tuned in the proposed method is M, the number of equally probably intervals. A lager M tends to improve the accuracy in the outer loop of $E_{\mathbf{\Phi}}(V_{\Phi^l}(Y))$ in Eq. (7.21); but also reduce the accuracy in the inner loop since the average number of samples $n = n_M/M$ to compute $V_{\Phi^l}(Y)$ in each individual interval will be decreased. Therefore a tradeoff between M and n is to be decided. This section aims to compare different selections of M using the three numerical examples above and provides an heuristic guideline in selecting M. This discussion constitutes of the following steps:

1. Set different values of M. The medium value is $M = [\sqrt{n_M}]$ to achieve a balance of $M = n$; the lowest values is 5 meaning only 5 intervals; and the highest value is $M = n_M/5$ meaning only around 5 samples in each interval.
2. Calculate the confidence intervals (CI) of the first-order indices at different values of M.
3. Compare the accuracy at different values of M based on the location and width of the CIs.

The comparison for the non-smooth function example is shown in Figure 7.9. As explained in Section 7.4.1, the total number of input-output samples is $n_M = 600$. As shown in the legend of Figure 7.9, 5 values of M are used where the medium value is $M = [\sqrt{n_M}] = 24$. Figure 7.9 indicates that: 1) the result by $M = 5$ & $n = 120$ is biased from the true value, especially for X_1; 2) the result by $M = 120$ & $n = 5$ has wider CIs than others; and 3) the results by other values of M are comparable good. In sum, this example requires $M \geq 10$ and $n \geq 10$.

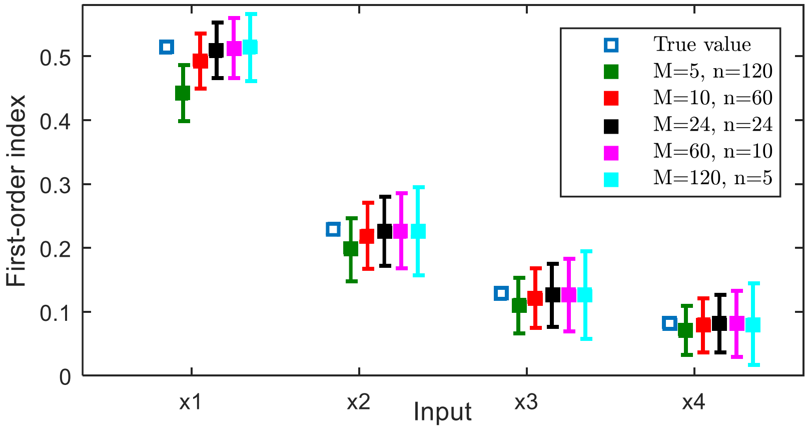


Figure 7.9 Selection of *M* in the non-smooth function example

The comparison for the high-dimensional linear function example is shown in Figure 7.10, and only the last five inputs are included due to limited space. As explained in Section 7.4.2, the total number of input-output samples is $n_M = 10400$. As shown in the legend of Figure 7.10, 7 values of M are used where the medium value is $M = [\sqrt{n_M}] = 102$. Figure 7.10 indicates that: 1) the result by $M = 5$ is biased from the true value significantly; 2) the result by $M = 10$ is biased slightly but still acceptable; 3) the results by $n = 5$ have significant wider CIs; 4) the result by $n = 10$ also have wider CIs but still acceptable; and 5) the results by other values of M are comparable. In sum, this example requires $M \geq 10$ & $n \geq 10$ but $M \geq 50$ & $n \geq 50$ is recommended.

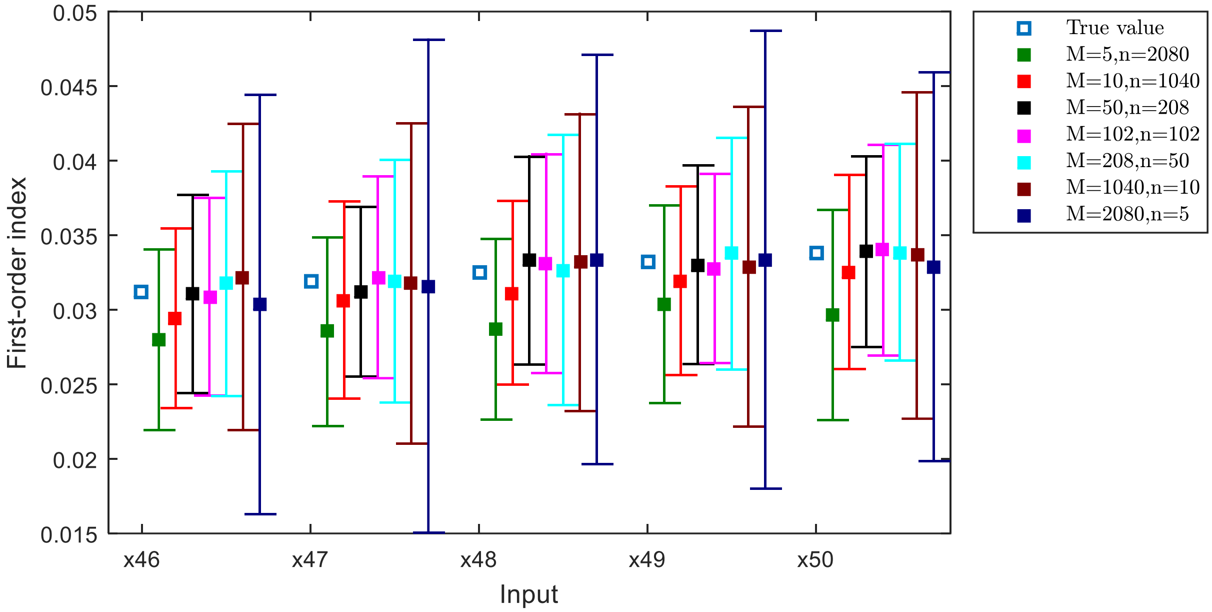


Figure 7.10 Selection of *M* in high-dimensional linear function example

The comparison for the cantilever beam with correlated inputs is shown in Figure 7.11, and only the inputs with first-order index larger than 0.1 is listed. As explained in Section 7.4.3, the total number of input-output samples is $n_M = 15050$. As shown in the legend of Figure 7.11, 7 values of M are used where the medium value is $M = [\sqrt{n_M}] = 123$. Figure 7.11 indicates that: 1) the results by $M = 5,10$ or $n = 5,10$ are biased from the true value, especially for L; and 2) the results by other values of M are comparable, and the result by $M = n = 123$ is slightly better. In sum, this example requires $M \geq 50$ and $n \geq 50$.

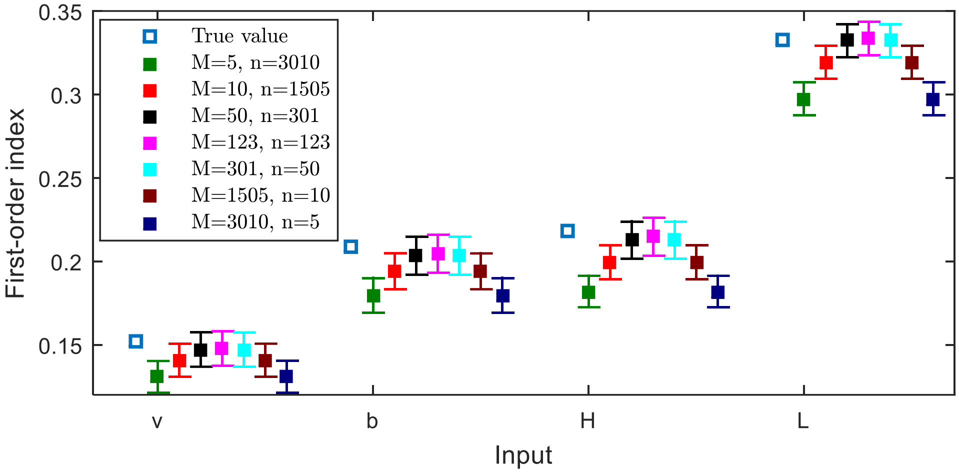


Figure 7.11 Selection of *M* in the cantilever beam example

Based on the comparisons above, the authors conclude that the minimum requirement for the proposed algorithm is $M \geq 10 \,\&\, n \geq 10$; but $M \geq 50 \,\&\, n \geq 50$ is recommended. Actually a simple strategy is $M = [\sqrt{n_M}]$ to achieve a balance of $M = n$, and this strategy has been used in all the examples in this research. Note that this guidance is purely heuristic, and formally optimizing the value of M may be explored in future.

7.4.5 Example 4: Input-Output Function NOT Available

The proposed algorithm can estimate the first-order Sobol' index as long as the input-output samples have already been collected, even if the underlying function is NOT available or cannot be re-evaluated. This situation often happens in the industry when an analyst supplies only the input-output data, but does not provide the computational model due to proprietary reasons. The reason that we used computational models in the earlier examples was to be able to compare the accuracy of our method with existing methods. Here we demonstrate the case of sensitivity analysis with only input-output data, assuming the computational model is not available.

We generated 2500 Monte Carlo input-output samples using the Ishigami function $Y = \sin X_1 + a\sin^2 X_2 + bX_3^4 \sin X_1$ [126]. These samples can be downloaded via the URL https://github.com/VandyChris/Global-Sensitivity-Analysis/blob/master/Ishigami.csv. Now suppose that only these samples are available, and that the actual function is not available, and even the distributions of $X_i (i = 1,2,3)$ are not known. In this situation, the proposed algorithm can directly estimate the first-order Sobol' indices from the available samples following the steps in Figure 7.2, but the existing algorithms discussed in Section 7.2 cannot. Using the proposed Algorithm 2 and $M = 50$, the result is obtained as shown in Table 4. (Of course, if the function is known, then it is possible to verify the accuracy of this result. We have verified that the result using the above analytical function is exactly the same. However, the purpose of this example is to demonstrate that the proposed method can calculate the first-order Sobol' indices using only the input-output samples. An alternative approach is to build a regression model based on the samples, and then use the regression model for GSA using any of the other existing methods; in that case, the regression error should also be accounted for).

Table 7.4 First-order Sobol' index

Variable	X_1	X_2	X_3
First-order index	0.42	0.23	0.00

7.5 Summary

This chapter focused on directly extracting first-order Sobol' indices from Monte Carlo samples. To solve this problem, this research showed that the conditional variance and mean in the expression of the first-order Sobol' index can be computed at an unknown but existing location of model inputs, instead of an explicit user-defined location. This concept leads to the proposed method which is modularized in two aspects: 1) separate the index calculations for different model inputs; and 2) model inputs sampling, model evaluation, and index calculation are separate processes. The modularization brings several benefits: 1) The computational cost of the proposed method is not proportional to the number of model inputs; 2) The proposed method can be used when only legacy input-output data or field data are available but the underlying model is not available, which is our main focus; 3) The calculation of S_i purely depends on the samples of X_i and y, and can be achieved even if the samples of $\boldsymbol{X}_{-i}$ are missing; and most importantly 4) The proposed method is able to compute the first-order index with correlated model inputs.

The proposed method includes two algorithms. Algorithm 1 computes the inner loop $E_{\boldsymbol{X}_{-i}}(Y|X_i)$ first and then the outer loop $V_{X_i}(\cdot)$; while Algorithm 2 computes the inner loop $V_{\boldsymbol{X}_{-i}}(Y|X_i)$ first and the then outer loop $E_{X_i}(\cdot)$. Section 7.3.4 proves that Algorithm 2 provides higher accuracy while Algorithm 1 is positively biased due to numerical error.

Algorithm 2 is used in two numerical examples with independent model inputs to compare with two existing methods: 1) the widely used Sobol' method, and 2) the improved FAST method. The latter one also has a computational cost that is not proportional to the model inputs dimension; and it is the best previously available algorithm for independent model inputs to the authors' knowledge. The results show that the proposed method has comparable accuracy with improved FAST and outperforms the Sobol' method. Algorithm 2 is also used in a third numerical example with correlated model inputs and seen to significantly outperform the double-loop MCS method; the improved FAST method cannot handle correlated model inputs.

The benefits brought by the proposed method imply strong promise for practical implementation such as test design [48,89], dimension reduction, feature selection, etc. Nowadays in areas such as transportation and social networks, obtaining data can be much easier than extracting the underlying models. Since the proposed method is highly efficient and only requires data, it is especially useful in ranking and identifying important variables, no matter whether the variables are correlated or not.

CHAPTER 8

GLOBAL SENSITIVITY ANALYSIS OF A BAYESIAN NETWORK

8.1 Background

In a Bayesian network, how a node of interest is affected by the observation at another node is of interest in both forward propagation and backward inference. However, two challenges in the application of Bayesian network are: 1) if the calculation is sample-based, a high-dimensional network (the number of nodes is large) will encounter the problem of computational efficiency, especially when the network includes some time-consuming computational models; 2) before the inference, efficacy of the observation to reduce the uncertainty in the state variables of interest is unknown. The second challenge is also financially important, since we do not want to invest our limited budget to measure some variables that are not useful in uncertainty reduction.

The global sensitivity analysis (GSA) of Bayesian network proposed in this chapter aims to solve the two challenges above by calculating the first-order Sobol' index of node X_1 with respect to another node of interest X_N. In forward propagation where X_1 is the ancestor node of X_N, a low index of X_1 indicates that X_1 is not significantly contributing to the uncertainty in X_N, thus we can simply fix X_1 at a deterministic value and reduce the dimension of the network. In backward inference where X_1 is the child node of X_N, a low sensitivity index of X_1 indicates that the observation of node X_1 will not significantly reduce the uncertainty in $X_N;$ thus we should measure another node with a higher Sobol' index in order to effectively calibrate X_N and reduce its uncertainty.

The desired GSA for a Bayesian network confronts two challenges of feasibility and affordability. First, the computation of the Sobol' index requires a deterministic function [150] but the Bayesian network is a stochastic model, i.e., it has probabilistic relationships among the nodes. And the required deterministic function mapping X_1 (and some other variables) to the node of interest X_N is unestablished. Proof of the existence and the establishment of this deterministic function needs to be solved.

Second, using the existing algorithms, the computation of the Sobol' index can be expensive even if the deterministic function is established. However, in Bayesian network, the prior samples of the node of interest X_N and the observation node X_1 is easy to obtain. Thus the new sample-based algorithm proposed in Section 7.3 which directly estimate the first-order Sobol' index turns out to be an ideal algorithm for the sensitivity analysis of Bayesian network. This section will also extend the proposed algorithm in Section 7.3 to estimate the variance reduction ratio (VRR) of the node of interest at a given value of the observation node.

The rest of the chapter is organized as follows. Section 8.2 uses the auxiliary variable method to convert the path between node X_1 and node X_N to a deterministic function, thus making the Sobol' index computation feasible for a Bayesian network. An introduction to the auxiliary variable method can be found in Section 2.7. Section 8.3 extends the proposed algorithm in Section 7.3 to estimate the uncertainty reduction of the node of interest when another node is fixed at the observation. This extension only uses the prior distribution samples, and no Bayesian inference effort is required. Thus this extension provides quantitative guidance for effective observation and updating, i.e., deciding which node is the most effective observation node in reducing the uncertainty in the node of interest. Section 8.4 illustrates the proposed method using two examples, including a time-independent static Bayesian network and a time-dependent dynamic Bayesian network.

8.2 Feasibility and Affordability of GSA for a Bayesian Network

8.2.1 Deterministic Function for a Directed Path

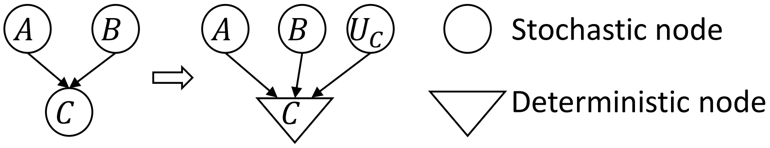


Figure 8.1 Auxiliary variable for a CPD

The auxiliary variable method have been extended to any variable whose distribution is conditioned on other variables [59,86], i.e., to any conditional probability distribution (CPD) in the Bayesian network. Assume that the distribution of a random variable C depends on the value of two other random variables A and B by a CPD $p(C|A,B)$. Then the variability in $p(C|A,B)$ can be captured by a single auxiliary variable U_C, which is the CDF value of $p(C|A,B)$. Thus the uncertainty in variable C is caused by two components: 1) the uncertainty due to the parent nodes A,B; and 2) the uncertainty expressed by the CPD at given values of A and B. The introduced auxiliary variable captures the later part. As shown in Figure 8.1, A, B and C constitute a simple Bayesian network. The introduced auxiliary variable U_C converts C to be a deterministic node, which means the value of C is fixed once the value of its parent nodes $\{A,B,U_C\}$ is given. Finally this auxiliary variable build a deterministic function $C=\mathcal{P}^{-1}(U_C|A,B)$, where $\mathcal{P}^{-1}(\cdot)$ is the inverse CDF of the CPD $p(C|A,B)$.

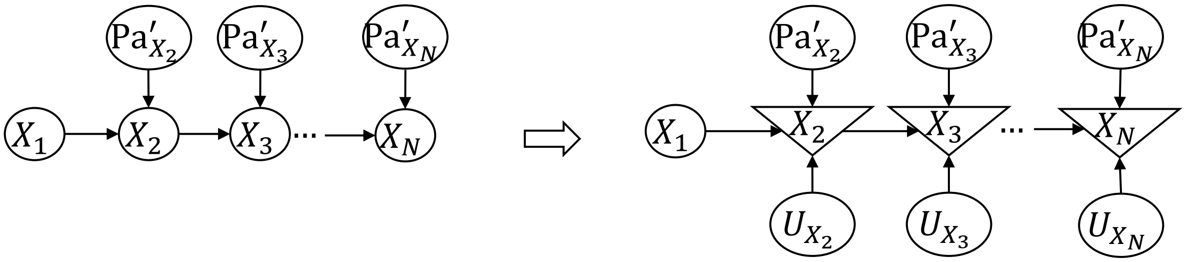


Figure 8.2 Deterministic function for the path $X_1 \to X_N$

The auxiliary variable method can be further extended to a directed path in a Bayesian network, as follows. In a Bayesian network, a node X_1 is called the ancestor node of node X_N if it leads a directed path $X_1 \to X_2 \ldots \to X_N$ to node X_N. For example, in Figure 8.3 node E has the ancestor nodes A, B, C and D. As shown in Figure 8.2, by introducing auxiliary variables to each CPD in this directed path, a deterministic function mapping X_1 to X_N is established

$$\begin{cases} X_2 = \mathcal{P}^{-1}\left(U_{X_2} | \mathrm{Pa}'_{X_2}, X_1\right) \\ X_3 = \mathcal{P}^{-1}\left(U_{X_3} | \mathrm{Pa}'_{X_3}, X_2\right) \\ \cdots \\ X_N = \mathcal{P}^{-1}\left(U_{X_N} | \mathrm{Pa}'_{X_N}, X_{N-1}\right) \end{cases} \tag{8.1}$$

where $\mathcal{P}^{-1}(U_{X_i} | \mathrm{Pa}'_{X_i}, X_{i-1})$ for $i = 2$ to N is the inverse CDF of the CPD $p(X_i | \mathrm{Pa}'_{X_i}, X_{i-1})$, and U_{X_i} is the auxiliary variable introduced for this CPD, and Pa'_{X_i} represents the parent nodes of X_i that are not in this path (Note that another notation P_V is used later, which means all the parents node of V, i.e., $\mathrm{Pa}_{X_i} = \{\mathrm{Pa}'_{X_i}, X_{i-1}\}$ in Figure 8.2. The inputs of Eq. (8.1) are $\{X_1, X_i, \mathrm{Pa}'_{X_i}, U_{X_i}\}$ for $i = 2$ to N, thus Eq. (8.1) can be also denoted as a deterministic function $f: \{X_1, X_i, \mathrm{Pa}'_{X_i}, U_{X_i}\} \to X_N$.

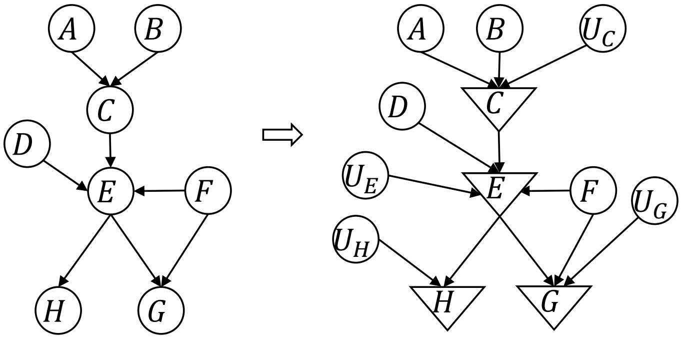


Figure 8.3 Auxiliary variable for a Bayesian network

The deterministic function established in Eq. (8.1) can be illustrated by a simple Bayesian network in Figure 8.3, which introduces an auxiliary variable to each CPD so that all the child nodes are converted to deterministic nodes. Based on Eq. (8.1), A is an ancestor node of C via the directed path $A \to C$, thus the deterministic function mapping A to C is

$$C = \mathcal{P}^{-1}(U_C|A, B) \tag{8.2}$$

And A is also an ancestor node of G via the directed path $A \to C \to E \to G$, thus the deterministic function mapping A to G is

$$\begin{cases} C = \mathcal{P}^{-1}(U_C|A, B) \\ E = \mathcal{P}^{-1}(U_E|C, D, F) \\ G = \mathcal{P}^{-1}(U_G|E, F) \end{cases} \tag{8.3}$$

8.2.2 Deterministic Function for an Undirected Path

As explained in Section 8.2.1, the directed path from X_1 to X_N requires that all the arcs are directed towards X_N. In comparison, an undirected path $X_1 - X_2 - \cdots - X_N$ (where the arc "$-$" is still directed, either "→" or "←") only requires all the adjacent nodes in the path are connected by arcs, regardless of the direction of the arcs. The deterministic function established in Eq. (8.1) for the directed path can be also extended to the undirected path based on the theorem of Arc Reversal [151].

Theorem 1. Arc Reversal. Given that there is an arc (V_1, V_2) from node V_1 to node V_2, but no other directed path from V_1 to V_2, arc (V_1, V_2) can be replaced by arc (V_2, V_1). Afterwards, both nodes inherit each other's parent nodes.

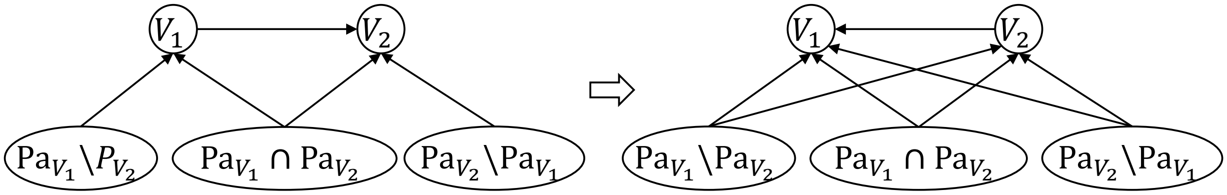

V_1
V_2
$\mathrm{Pa}_{V_1}\backslash P_{V_2}$
$\mathrm{Pa}_{V_1} \cap \mathrm{Pa}_{V_2}$
$\mathrm{Pa}_{V_2}\backslash \mathrm{Pa}_{V_1}$
V_1
V_2
$\mathrm{Pa}_{V_1}\backslash \mathrm{Pa}_{V_2}$
$\mathrm{Pa}_{V_1} \cap \mathrm{Pa}_{V_2}$
$\mathrm{Pa}_{V_2}\backslash \mathrm{Pa}_{V_1}$

Figure 8.4 Arc Reversal [151]

This theorem is illustrated in Figure 8.4. Here Pa_{V_1} indicates the parent nodes of V_1, and Pa_{V_2} indicates the parent nodes of V_2. In addition, $\mathrm{Pa}_{V_1}\backslash\mathrm{Pa}_{V_2}$ are the nodes which are the parents of V_1 but not the parents of V_2, and correspondingly $\mathrm{Pa}_{V_2}\backslash\mathrm{Pa}_{V_1}$ are the nodes which are the parents of V_2 but not the parents of V_1; and $\mathrm{Pa}_{V_1} \cap \mathrm{Pa}_{V_2}$ are the shared parents of V_1 and V_2. Figure 8.4 shows that after reversing the arc between V_1 and V_2, extra arcs $(\mathrm{Pa}_{V_1}\backslash\mathrm{Pa}_{V_2}, V_2)$ and $(\mathrm{Pa}_{V_2}\backslash\mathrm{Pa}_{V_1}, V_1)$ are also derived based on Ref. [151] and added the new BN to guarantee that the new BN after arc reversal is mathematically equivalent to the original BN. The CPDs also need to be redefined, and the derivation of the new CPDs can be also found in Ref [151]. However, note that the proposed method in this research do NOT need to derive these new CPDs. The main focus of this section is to illustrate the possibility of arc reversal and prove the existence of the deterministic function mapping X_1 to X_N even if the path between them is undirected.

With respect to the undirected path between X_1 and X_N, Theorem 1 proves that the arc (X_i, X_j) between two adjacent nodes X_i and X_j ($i = j + 1$ or $j - 1$ so that they are adjacent) can be reversed, as long as there is no other directed path from X_i to X_j. If all the arcs towards X_1 can be reversed, this undirected path will be converted to a directed path from X_1 to X_N so that a deterministic function mapping X_1 to X_N exists based on Eq. (8.1). In Figure 8.3, the undirected path $H \leftarrow E \rightarrow G$ can be converted to a directed path $H \rightarrow E \rightarrow G$ by reserving the arc (E, H); then a deterministic function mapping H to G can be constructed using auxiliary variables.

Furthermore, a directed path from the node of interest X_N to X_1 can be even converted to another directed path from X_1 to X_N by reversing all the arcs, so that a deterministic function mapping X_1 to X_N exists. For example, the directed path $A \rightarrow C \rightarrow E \rightarrow G$ in Figure 8.3 can be converted to $G \rightarrow E \rightarrow C \rightarrow A$ so that a deterministic function mapping G to A can be constructed using auxiliary variables.

8.2.3 Affordability of GSA for A Bayesian Network

For two arbitrary nodes X_1 and X_N, Sections 8.2.1 and 8.2.2 explained the possibility to build a deterministic function mapping X_1 to X_N as long as Theorem 1 of Arc Reversal is satisfied. This process is illustrated in Figure 8.5. Thus we can conduct the GSA on Eq. (8.1) and compute the first-order Sobol' index S_{X_1} for X_1. As explained earlier, S_{X_1} is the average ratio of the reduced variance of X_N by fixing X_1. If an observation of X_1 is used in the subsequent Bayesian inference to update the network, this Sobol' index S_{X_1} provides an assessment of two aspects before the updating: 1) identifiability of X_N, i.e., whether X_1 has a low sensitivity such that fixing X_1 at its observation does not identify the value of X_N; and 2) quantification of the expected uncertainty reduction of X_N in the updating.

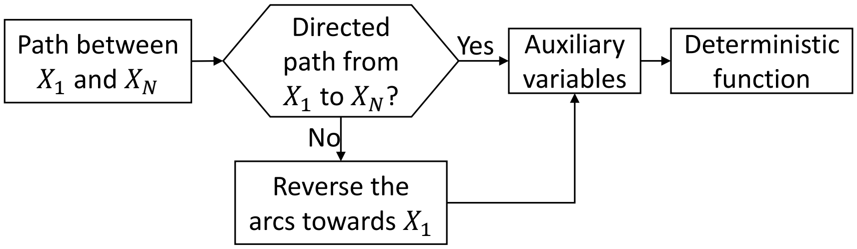

Path between X_1 and X_N
Directed path from X_1 to X_N?
Yes
No
Auxiliary variables
Deterministic function
Reverse the arcs towards X_1

Figure 8.5 Deterministic function for a path

However, the computation of S_{X_1} is non-trivial. First, building the deterministic function explicitly can be complicated in either forward propagation or backward inference. In the case of forward prorogation, an ancestor node is observed and the posterior distribution of the descendant node is of interest, i.e. the path is $X_1 \rightarrow \cdots \rightarrow X_N$. The deterministic function mapping X_1 to X_N can be established by Eq. (8.1). However, the effort to build this function becomes intensive if the path is long so that many nodes and auxiliary variables will be involved in Eq. (8.1). In the case of backward inference, which is more common in Bayesian network, a descendant node is observed and the posterior distribution of an ancestor node is of interest, i.e., the path is $X_N \rightarrow \cdots \rightarrow X_1$. To build the required deterministic function mapping X_1 to X_N, all arcs in the path need to be reversed, and this brings extra computational effort to modify the structure of the Bayesian network and derive new CPDs.

Second, even with the deterministic function established, calculating the sensitivity index also needs intensive effort. The inputs of the deterministic function includes the nodes in the Bayesian network, so the correlation between them is unavoidable. As mentioned in Saltelli's paper [67], since current efficient algorithms for Sobol' index usually require uncorrelated inputs, the expensive double-loop MCS is the only choice.

The efficient sample-based algorithm proposed in Section 7.3 can directly extracts the first-order Sobol' index from the Monte Carlo samples, and turns out to be an ideal algorithm for the sensitivity analysis of Bayesian network. For a Bayesian network, samples from its joint prior distribution are easy to obtain. If the Bayesian network has been established, these samples can be easily generated based on all CPDs, and these samples may be used again in the subsequent updating; if the Bayesian network is to be learned from the data, these data are actually the prior samples needed. We will also see that explicitly establishing the deterministic function is not necessary, and the expensive double-loop MCS method is also avoided. Based on Section 7.3, the sensitivity of the node of interest X_1 to the observation node X_N is

$$S_{X_1} = 1 - \frac{E_{\mathbf{\Phi}}\left(V_{\Phi^l}(X_N)\right)}{V(X_N)} \tag{8.4}$$

Now the sensitivity analysis of Bayesian network becomes straightforward since its feasibility has been proved and an efficient algorithm making use of the prior distribution samples has been developed. To implement Eq. (8.4) to calculate the first-order Sobol' index of X_1, we only need to:

1. Obtain the samples from the joint distribution of X_1 and X_N;
2. Use the samples of X_1 as input samples (X_i sample in Figure 7.2) and the samples of X_N as output samples (Y sample in Figure 7.2);
3. Follow the steps in Figure 7.2 to calculate the first-order Sobol' index of X_1.

These three steps above are straightforward and do not require intensive computational effort. Thus the affordability of the proposed sensitivity analysis for the Bayesian network has been solved.

8.3 Variance Reduction Prediction in Bayesian Inference

The resultant index S_{X_1} from Eq. (8.4) is the average ratio of the variance reduction of X_N by fixing X_1 at its observation; and this is an average over all possible values of X_1. This is an informative estimate before the updating if the value of the observation is NOT known.

If the value of the observation of X_1 is known, this variance reduction ratio (VRR) estimate can be further improved by identifying the equally probably interval where the observation is located and computing the local variance. Denote $\widehat{\Phi}$ as the equally probable interval that contains the observation $\hat{X}_1$, i.e., $\hat{X}_1 \in \widehat{\Phi}$.

The improved estimate is

$$\mathrm{VRR} \approx 1 - \frac{V_{\widehat{\Phi}}(X_N)}{V(X_N)} \tag{8.5}$$

Compared to Eq. (8.4) which computes the average VRR of X_N over all possible values of X_1, Eq. (8.5) estimates the VRR of X_N at a specific value of X_1. The accuracy of Eq. (8.5) will be higher if 1) $\widehat{\Phi}$ is narrower so that $\hat{X}_1$ is closer to $X_1^{\#}$ and 2) more samples of X_N are assigned to $\widehat{\Phi}$ so that $V_{\widehat{\Phi}}(X_N)$ is a better estimate of $V(X_N|X_1 = X_1^{\#})$.

8.4 Numerical Examples

8.4.1 Structural Dynamics Problem

A structural dynamics problem provided by Sandia National Laboratories is used to illustrate the proposed method, and more details on this problem can be found in Ref. [68,79,91,130]. As shown in Figure 8.6, the system of interest contains three mass-spring-damper components in series; and these components are mounted on a beam supported by a hinge at one end and a spring at the other end; and a sinusoidal force input $P = 3000\sin(350t)$ is applied on the beam.

This system has three model parameters of spring stiffnesses $\boldsymbol{k} = (k_1, k_2, k_3)$ and they are assumed to have unknown true values to be calibrated. The prior distribution of k_i is assumed to be Gaussian with a coefficient of variation of 10% and mean values of $\mu_{k_1} = 5000$, $\mu_{k_2} = 10000$, and $\mu_{k_3} = 9000$.

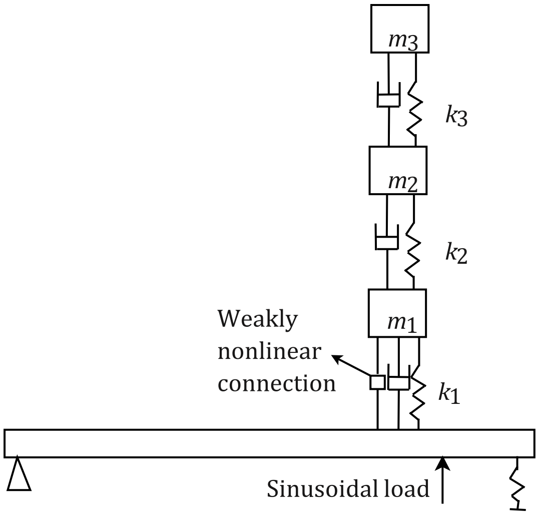


Figure 8.6 Beam with mass-spring-damper

The quantity to be measured for model calibration is the maximum acceleration A_3 in the 3rd mass m_3. A computational model $A_3 = F(\boldsymbol{k})$ based on finite element analysis has been provided by Sandia National Laboratories [91].

To improve the computational efficiency, a Gaussian process (GP) [30,152] surrogate model $A_3 = \mathrm{GP}(\boldsymbol{k})$ is constructed to replace the expensive dynamics computational model. The prediction of the GP model is a Gaussian distribution $N\big(\mu(\boldsymbol{k}), \sigma^2(\boldsymbol{k})\big)$, thus a CPD is given by the GP model.

The observation variable is denoted as D and we have $D = A_3 + \epsilon_m$ where ϵ_m is the measurement error with a zero-mean Gaussian distribution $\epsilon_m \sim N(0, \sigma_m^2)$. Thus another CPD is given by the measurement error. In this example, σ_m is another parameter to be calibrated and we assign a non-informative uniform prior distribution $U(150,250)$ to it.

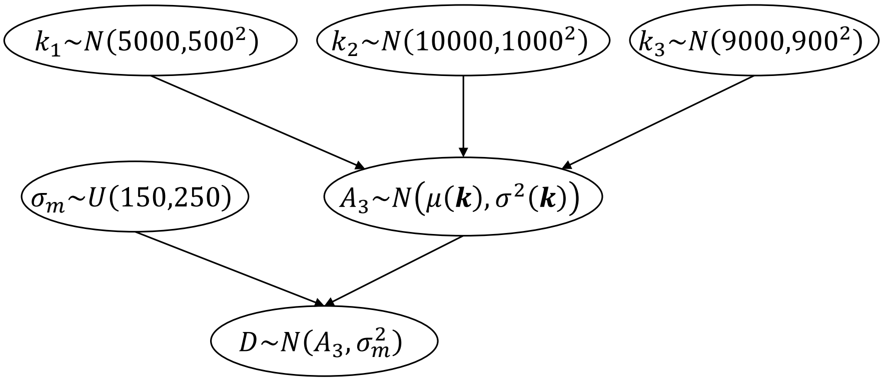

$k_1 \sim N(5000,500^2)$
$k_2 \sim N(10000,1000^2)$
$k_3 \sim N(9000,900^2)$
$\sigma_m \sim U(150,250)$
$A_3 \sim N(\mu(\boldsymbol{k}),\sigma^2(\boldsymbol{k}))$
$D \sim N(A_3,\sigma_m^2)$

Figure 8.7 Bayesian network of Example 1

A Bayesian network is established for this model calibration problem, as shown in Figure 8.7. In this example, we are interested in 1) calculating the first-order Sobol' index of the calibration parameters $\{k_1,k_2,k_3,\sigma_m\}$ with respect to the observation variable D, and 2) predicting the variance reduction ratio (VRR) of the calibration parameters at a given observation.

Table 8.1 First-order Sobol' index, Example 1

Parameter	k_1	k_2	k_3	σ_m
First-order Sobol' index	0.50	0.02	0.11	0.00

As samples are generated from the joint prior distribution of this network, the first-order Sobol' indices of $\{k_1,k_2,k_3,\sigma_m\}$ are obtained by considering the calibration parameter as X_N and the observation variable D as X_1 in Eq. (8.5). The results are listed in Table 8.1. From this table, we conclude that the variance of k_1 will reduce by 50% on average due to calibration; the variance reduction of k_3 is 11% on average; while the variance of k_2 and σ_m will not be reduced significantly by calibration. This is a very valuable insight. Thus if we want to reduce the uncertainty in k_2, we need to observe another quantity. In the latter computation of VRR at specific observations of A_3, we focus on k_1 and k_3.

Table 8.2 Variance reduction ratio at specific observations of A_3

Data point	k_1		k_3	
	Proposed method (Bayesian inference NOT needed)	Sample-based (Bayesian inference needed)	Proposed method	Sample-based
3900	49.9%	47.0%	3.2%	4.0%
4000	45.2%	44.2%	13.6%	15.4%
4100	38.3%	42.4%	23.6%	19.7%
4200	43.5%	44.8%	20.7%	25.5%
4300	48.2%	54.2%	35.2%	31.9%
4400	60.5%	63.2%	41.7%	38.9%
4500	69.6%	69.1%	43.3%	44.7%

Table 8.1 shows the average variance reduction ratio (VRR). Now assume that the specific observed value of A_3 is known (a synthetic data point). Based on Eq. (8.5), we predict the VRR of k_1 and k_3 for this specific observation, as shown in Table 8.2, where the "Sample-based" method mean we finish the Bayesian inference and compute the VRR based on the samples of the posterior distributions. In comparison, the proposed method only uses the samples from the prior distribution, and no actual Bayesian inference effort is required.

We also implement the Bayesian inference using the rejection sampling (RS) algorithm [17] to generate 2×10^4 samples from the posterior distributions of the calibration parameters. Figure 8.8 shows the PDFs of these posterior distributions at data point 4500. We re-calculate the VRR by comparing the variances of the posterior samples and the prior samples. As shown in Table 8.2, our earlier predictions are close to the sample-based results. This verifies the effectiveness of the proposed method. Note that these two results are not exactly the same due to: 1) the numerical error in computing the output variance within the equally probable interval that contains the data point ($V_{\widehat{\Phi}}(X_N)$ in Eq. (8.5)); 2) the approximation of $\hat{X}_1 \approx X_1^{\#}$ in Eq. (8.5); and 3) the numerical error in the RS.

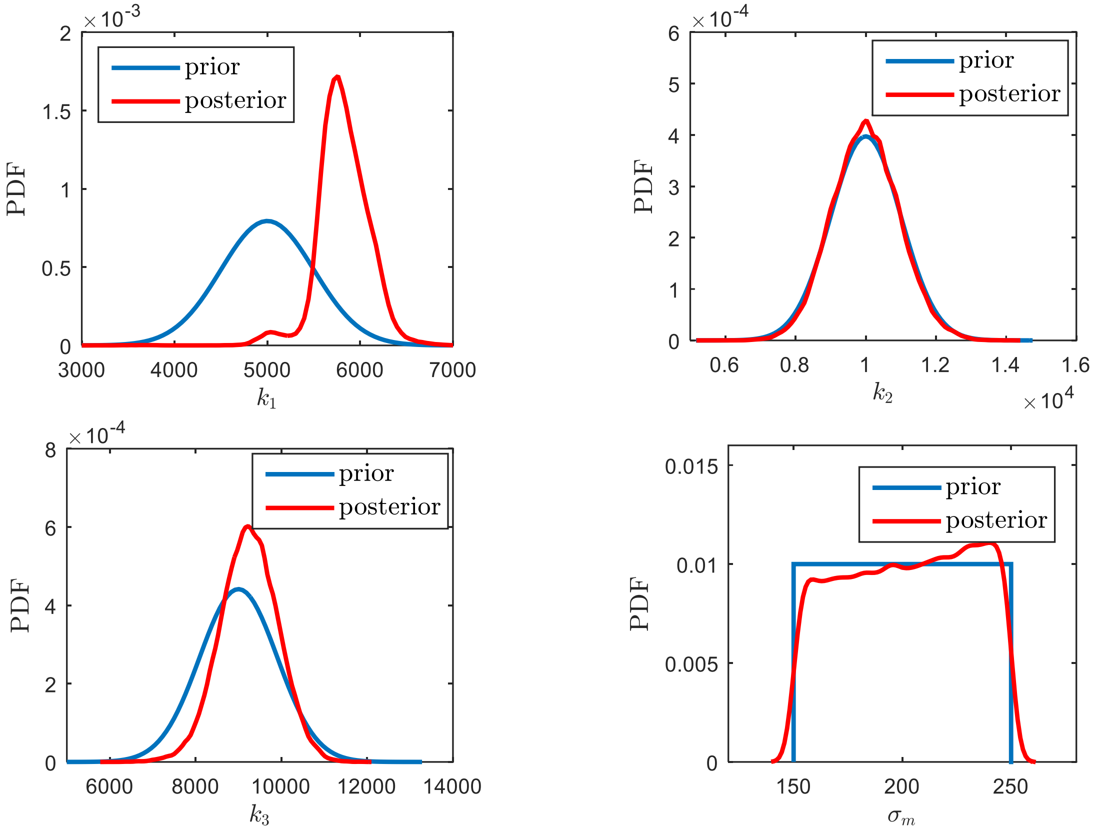


Figure 8.8 Posterior distributions at observation value of $A_3 = 4500$

In summary, this example verified the effectiveness of the proposed method to predict the variance reduction ratio before conducting the Bayesian updating. Thus the proposed method provides valuable guidance for selecting observation nodes; for example, the subsequent updating, nodes such as k_2 and σ_m cannot be updated by observing A_3 data.

8.4.2 Example of a Dynamic Bayesian Network

This example applies the proposed method to a mathematical example of a dynamic Bayesian network, as shown in Figure 8.9. The CPDs of this dynamic Bayesian network are as follows.

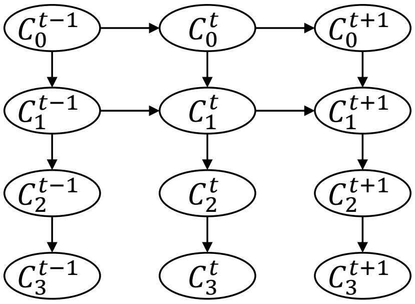


Figure 8.9 Dynamic Bayesian network of Example 2

The root node C_0 has an unknown true value to be calibrated, so that C_0 is a static node and $C_0^t = C_0^{t+1}$. The prior distribution of C_0 is $N(2{,}0.5^2)$. C_1 and C_2 are two dynamic state variables, and their states are to be tracked. At $t = 1$ the CPD of the child node C_1 is $C_1^1 \sim N({C_0^1}^2 + 10, 1^2)$; at $t > 1$ the CPD of C_1 is $C_1^t \sim N({C_0^t}^2 + 0.9C_1^{t-1} + 1, 1^2)$, thus the distribution of C_1 depends on its previous value and the value of C_0. C_2 is the child node of C_1 and its CPD is $C_2^t \sim N({C_1^t}^2, 5^2)$. In this problem the observation node is C_3 and its CPD is $C_3^t \sim N(C_2^t, (C_2^t/20)^2)$, i.e., the value of C_2^t plus a measurement error of zero mean Gaussian distribution. This example considers the first 30 steps of this dynamic Bayesian network. Assuming the true value of C_0 is 2.5, the synthetic data of the observable node C_3 is generated at each step, as shown in Figure 8.10.

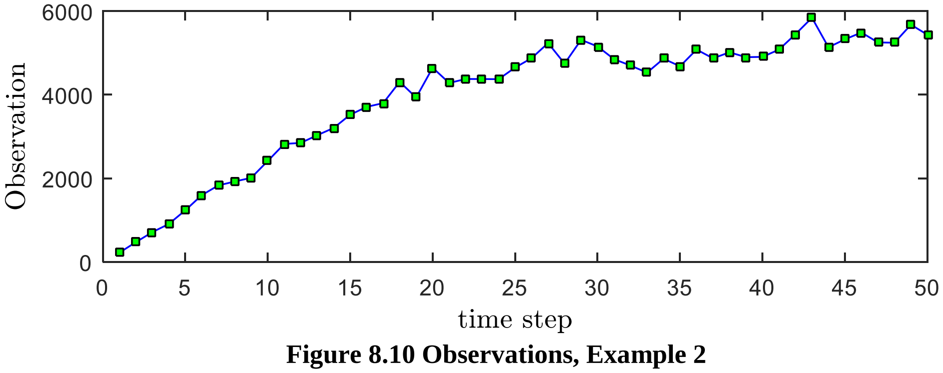


Figure 8.10 Observations, Example 2

A widely-used particle filter method named sequential importance resampling (SIR) algorithm [112] is applied in this example to track the state variables. Here a particle is a sample from the joint distribution of the state variables. This SIR algorithm propagates the particles of the posterior distribution at time step $t-1$ to time step t to obtain the particles of the prior distribution of time step t. The likelihoods of these particles are calculated and normalized as the weights for them. Then the particles are resampled based on the weight terms and the resultant new particles represent the posterior joint distribution of the state variables in time step t.

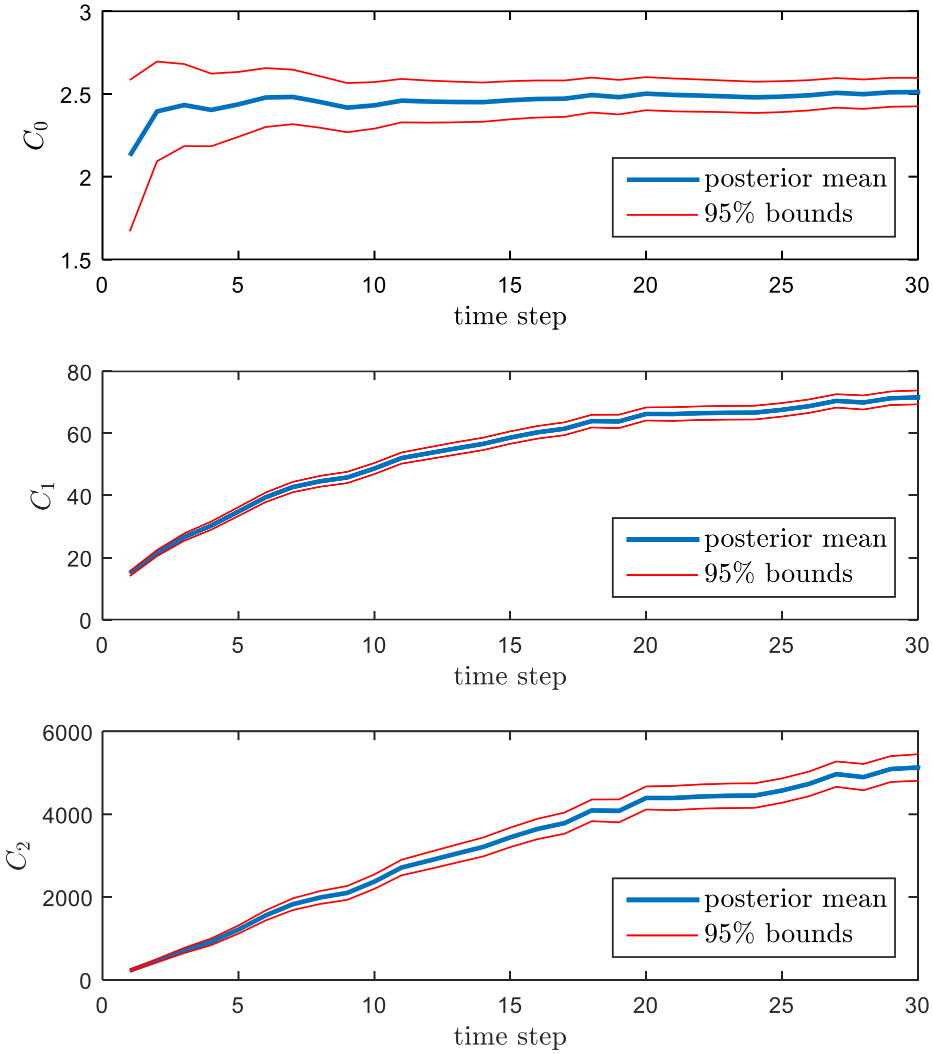


Figure 8.11 Posterior distribution of state variables

The number of particles in this example is 50,000. The mean value and 95% bounds of the posterior distributions of the state variables are shown in Figure 8.11. The uncertainty of C_0 reduces and its posterior distribution approximates to its true value 2.5, but this uncertainty reduction is not significant after step 20.

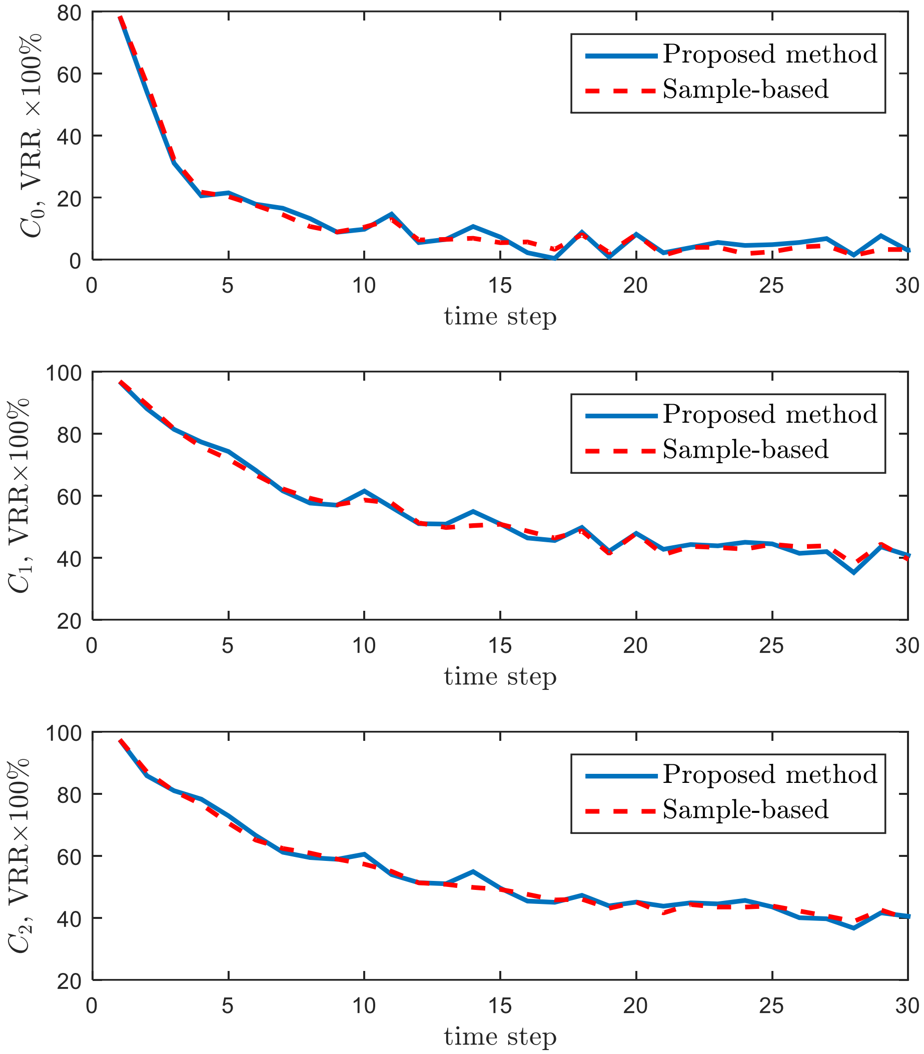


Figure 8.12 Variance reduction ratio (VRR) of the state variables

At each step, before the calculation of the likelihoods and the particle resampling, we apply the proposed method using the particles of the prior joint distribution of the state variables. The variance reduction ratio (VRR) of each state variable is predicted by the proposed method of Eq. (8.5) using the prior samples of the state variables. This VRR is also calculated by the prior/posterior samples at each step. Figure 8.12 shows that the results from these two methods are consistent so that the proposed method is verified. Note that the proposed method uses the prior samples and the observation data; while the sample-based method needs both the prior and posterior sample. In other words, the proposed method can be applied before Bayesian inference, but the sample-based method happens after the Bayesian inference has been done.

In this example, the CPD of C_0 is $C_0^t = C_0^{t+1}$ so the uncertainty of C_0 will not be enlarged in the propagation from time step $t-1$ to time step t. However, its uncertainty is reduced by the updating in each time step. Figure 8.11 shows that this uncertainty reduction is significant for the first 5 times steps so that the VRR in Figure 8.12 has a large value before time step 5; this uncertainty reduction is negligible after time step 20 so that the value of VRR is closer to zero after time step 20.

In comparison, Figure 8.12 shows that the uncertainty in the posterior distributions of C_1 and C_2 are not reducing, even if their VRR values in Figure 8.12 are always significant. The reason is that the uncertainty of C_1 and C_2 are enlarged in the propagation from time step $t-1$ to time step t, so their prior distributions at t have more uncertainty than the posterior distribution at $t-1$. The uncertainty in the prior at t is reduced by the updating, but the posterior uncertainty at t may not be smaller than the posterior uncertainty at $t-1$ if the uncertainty reduction by the updating cannot outperform the uncertainty enlargement by the propagation. Note that the VRR in Figure 8.12 is the variance reduction with respect to the prior/posterior distribution at the same time step, not the variance reduction for adjacent posterior distributions.

In summary, this example extended the proposed sensitivity analysis to a dynamic Bayesian network and verified its validity. The proposed method is capable to predict the variance reduction of each state variable before updating.

8.5 Summary

In a Bayesian network, how a node of interest is affected by fixing another node at some value is of prominent interest. The proposed GSA for Bayesian network calculates the first-order Sobol' index of any node X_1 with respect to any other node of interest X_N. In forward propagation where X_1 is the ancestor node of X_N, a low index of X_1 indicates that X_1 is not significantly contributing to the uncertainty in X_N so that we can simply fix X_1 at a deterministic value. In backward inference where X_1 is the descendant node of X_N, a low sensitivity index of X_1 indicates that X_N cannot be updated by observing X_1; thus we should measure another node of higher Sobol' index in order to calibrate X_N and reduce its uncertainty.

The proposed GSA for Bayesian network is realized in two steps. First, an auxiliary variable method is used to convert the path between node X_1 and node X_N to a deterministic function thus making the Sobol' index computation feasible for a Bayesian network. If the path from X_1 to X_N is not a directed path form, the theorem of arc reversal is used to transform it to the desired directed path so that the auxiliary variable method can still be used to build the deterministic function. Second, this research proposed an efficient algorithm to directly estimate the Sobol' index from Monte Carlo samples of the prior distribution of the Bayesian network, so that the proposed GSA for the Bayesian network is computationally affordable. The resultant Sobol' index is the average variance reduction ratio across all possible observations of X_1. The proposed algorithm can also give an accurate prediction of the uncertainty reduction of the node of interest purely by using the prior distribution samples when the value of the observation is known, thus providing an informative guidance before the updating.

CHAPTER 9

FAST INFERENCE ALGORITHM FOR NON-LINEAR BAYESIAN NETWORKS WITH CONTINUOUS VARIABLES

9.1 Background

The research on BN includes two main topics: inference and learning, and this dissertation focuses on inference, which aims to estimate the posterior distribution of the state variables based on evidence. An introduction to Bayesian inference has been provided in Section 2.2. One main challenge in Bayesian inference is time cost. This challenge is more severe in time dependent problem of dynamic Bayesian network, where inference in real time may be required. Therefore, this chapter aims to develop a fast inference algorithm.

A quick recap of Bayesian inference is given here to facilitate further development of this chapter.

As shown in Section 2.2, the inference in static BN is to calculate the posterior probability distribution $p(\boldsymbol{X}|\boldsymbol{Y}=\boldsymbol{y})$, where X is the vector of state variables for inference, and $\boldsymbol{y}$ is the measurement of the observation variables $\boldsymbol{Y}$. The inference is based on Bayes' theorem:

$$p(\boldsymbol{X}|\boldsymbol{Y}=\boldsymbol{y}) \propto p(\boldsymbol{X})p(\boldsymbol{y}|\boldsymbol{X}) \tag{9.1}$$

where $p(\boldsymbol{X})$ and $p(\boldsymbol{X}|\boldsymbol{Y}=\boldsymbol{y})$ are the prior and posterior distributions of state variables $\boldsymbol{X}$, and $p(\boldsymbol{y}|\boldsymbol{X})$ is the likelihood function of $\boldsymbol{X}$.

In a dynamic Bayesian network (DBN), inference is to estimate the probability $p(\boldsymbol{X}^t|\boldsymbol{y}^{1:t})$, i.e., the posterior distribution of the state variables in the current time instant given observations in the past and current time instants. The inference in a DBN is based on:

$$p(\boldsymbol{X}^t|\boldsymbol{y}^{1:t}) \propto p(\boldsymbol{X}^t|\boldsymbol{y}^{1:t-1})p(\boldsymbol{y}^t|\boldsymbol{X}^t) \tag{9.2}$$

The detailed derivation of Eq. (9.2) can be found in Eq. (2.6). Similar to Eq. (9.1), Eq. (9.2) also include two components:

1. The prior distribution $p(\boldsymbol{X}^t|\boldsymbol{y}^{1:t-1})$ at time t;
2. The likelihood function $p(\boldsymbol{y}^t|\boldsymbol{X}^t)$ (based on Eq. (2.5)), which only utilizes the observation at time t.

In Eq. (9.1) for static BN and Eq. (9.2) for DBN, the product of the prior distribution and the likelihood function is only proportional to but not equal to the posterior distribution. Thus a specific inference algorithm, either exact or approximate, is required to calculate the PDF/PMF value of the posterior distribution or generate random samples representing the posterior distribution. A literature review on inference algorithms has been provided in Section 2.3, where we can find that fast, analytical inference algorithms for static/dynamic BN with discrete variables have been well-developed in the literature, but the current algorithms for static/dynamic BN with continuous variables are either time-consuming or restricted to specific CPDs and/or BN topology. This research aims to develop a more general approximate inference algorithm for static/dynamic BN with continuous variables. The main concept of the proposed algorithm is to utilize the auxiliary variable method based on the probability integral transform [65][59] to collapse a complex BN of arbitrary topology to a two-layered BN so that the unscented Kalman filter (UKF) can be used for inference. The proposed algorithm is analytical and fast, and applicable to static/dynamic BN of any topology and CPDs as long as the assumption of Gaussian posterior distribution is acceptable.

The rest of the chapter is organized as follows. Section 9.2 gives a brief introduction of the unscented Kalman filter, which is used in the proposed method; Section 9.3 develops the proposed method; and Section 9.4 provides two numerical examples.

9.2 Introduction to Unscented Kalman Filter

9.2.1 Kalman Filter

Kalman filter [25] is an exact inference algorithm for linear Gaussian dynamic system, which means: 1) the state function and the measurement function are both linear; 2) state variables have a joint Gaussian distribution; and 3) all the noise terms are assumed to be independent zero mean Gaussian variables. The state function is

$$\boldsymbol{X}^{t+1} = \boldsymbol{A}^{t}\boldsymbol{X}^{t} + \boldsymbol{v}^{t} \tag{9.3}$$

where $\boldsymbol{A}^{t} \in \mathbb{R}^{N_X \times N_X}$ is the state transition matrix, and $\boldsymbol{B}^{t} \in \mathbb{R}^{N_X \times N_X}$ is the control-input matrix applied to the control vector $\boldsymbol{u}^{t} \in \mathbb{R}^{N_X}$, and $\boldsymbol{v}^{t} \sim N(0, \boldsymbol{Q}^{t})$ is the zero-mean Gaussian noise where $\boldsymbol{Q}^{t} \in \mathbb{R}^{N_X \times N_X}$ is the covariance matrix.

The measurement function is

$$\boldsymbol{Y}^{t} = \boldsymbol{H}^{t}\boldsymbol{X}^{t} + \boldsymbol{\sigma}^{t} \tag{9.4}$$

where $\boldsymbol{H}^{t} \in \mathbb{R}^{N_Y \times N_X}$ is the observation transition matrix, and $\boldsymbol{\sigma}^{t} \sim N(0, \boldsymbol{R}^{t})$ is the zero-mean Gaussian noise where $\boldsymbol{R}^{t} \in \mathbb{R}^{N_Y \times N_Y}$ is the covariance matrix.

The Kalman filter algorithm computes the posterior distribution $p(\boldsymbol{X}^t|\boldsymbol{y}^{1:t-1})$ at given $\boldsymbol{A}^t, \boldsymbol{B}^t, \boldsymbol{Q}^t, \boldsymbol{H}^t$ and $\boldsymbol{R}^t$. Five functions have been derived for this objective, which can be found in Ref. [25].

An extended Kalman filter or an unscented Kalman filter may be used when the state function and/or the measurement function are non-linear. In this case the state function is:

$$\boldsymbol{X}^{t+1} = f(\boldsymbol{X}^t, \boldsymbol{v}^t) \tag{9.5}$$

and the measurement function is:

$$\boldsymbol{Y}^t = h(\boldsymbol{X}^t, \boldsymbol{n}^t) \tag{9.6}$$

The functions $f(\cdot)$ in Eq. (9.5) and $h(\cdot)$ in Eq. (9.6) are non-linear functions, in contrast to the linear functions in Eqs. (9.3) and (9.4).

The main concept of the extended Kalman filter is to linearize $f(\cdot)$ and $h(\cdot)$ to the first order, so that inference results can be obtained following the five equations of Kalman filter. The details of the extended Kalman filter can be found in Ref. [25]. However, this “first-order” approximation in the extended Kalman filter can introduce large errors into the mean and covariance of the posterior distribution [153], and calculation of the Jacobian matrix also brings computational difficulty in the case of high non-linearity [111].

In contrast, the main concept of unscented Kalman filter is to calculate the output mean and variance of $f(\cdot)$ and $h(\cdot)$ using the method of unscented transform, where several sigma points are selected and propagated through the non-linear functions. The unscented Kalman filter avoids calculating the Jocobian matrix, and has been reported to be more accurate than the extended Kalman filter [111,153,154]. An introduction of the unscented Kalman filter is given latter.

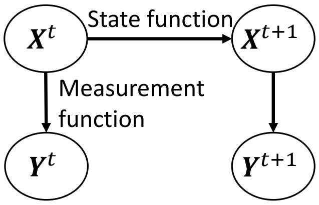


Figure 9.1 Underlying DBN of Kalman filter

Note that these three types of Kalman filters above are NOT proposed for DBN but for a dynamic system which can be depicted by the state function and measurement function. But this dynamic system has an underlying DBN as shown in Figure 9.1 (the same as Figure 2.5). This DBN has two layers: Layer 1 is for state variables $\boldsymbol{X}^t$ and Layer 2 is for observation variables $\boldsymbol{Z}^t$. Theoretically, the three types of Kalman filters are applicable for any DBN if it has the topology in Figure 9.1 so that the CPDs from $\boldsymbol{X}^t$ to $\boldsymbol{Z}^t$ can be represented by a measurement function and the CPDs from $\boldsymbol{X}^{t-1}$ to $\boldsymbol{X}^t$ can be represented by a state function. The basic Kalman filter is adequate if both the state function and measurement function are linear and the noise terms are zero-mean Gaussian variables; otherwise the extended Kalman filter or unscented Kalman filter is required.

However, DBNs with more than two layers cannot be updated by Kalman filters since the CPDs between two layers of state variables are missing in the dynamic system of state/measurement functions. An example of a DBN where a Kalman filters cannot be used is shown in Figure 9.2, for which the CPD $P(\boldsymbol{X}_2^t|\boldsymbol{X}_1^t)$ is missing in the dynamic system.

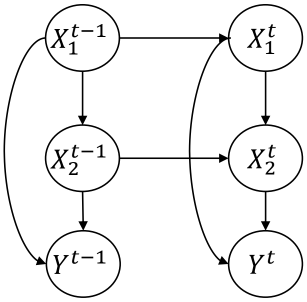


Figure 9.2 A DBN where the Kalman filter cannot be used

This research aims to collapse a DBN of more than two layers to an equivalent DBN of two layers so that the unscented Kalman filter can be applied. The unscented Kalman filter is selected here since it can handle non-linear problems and has been shown to have better accuracy than the extended Kalman filter. The proposed algorithm is developed in Section 9.3.

9.2.2 Unscented Transform

Unscented transform is the basis for unscented Kalman filter. For a non-linear function $\boldsymbol{Y} = G(\boldsymbol{X})$ where the outputs $\boldsymbol{Y} \in R^K$ and the inputs $\boldsymbol{X} \in R^L$, the unscented transform (UT) is a method to calculate the mean and covariance matrix of $\boldsymbol{Y}$ with limited function evaluations. This section introduces the "scaled unscented transform" in Ref. [153,155] which improves the original unscented transform in Ref. [111] so that the calculated covariance matrix is guaranteed to be positive semi-definite.

Assuming $\boldsymbol{X}$ has the mean vector $\overline{\boldsymbol{X}}$ and covariance matrix $\boldsymbol{P_X}$, $2L + 1$ sigma points of $\boldsymbol{X}$ are generated as

$$\begin{gathered}
\boldsymbol{\chi}_0 = \overline{\boldsymbol{X}} \\
\boldsymbol{\chi}_i = \overline{\boldsymbol{X}} + \left(\sqrt{(L+\lambda)\boldsymbol{P_X}}\right)_i \text{ for } i = 1, \dots, L \\
\boldsymbol{\chi}_i = \overline{\boldsymbol{X}} - \left(\sqrt{(L+\lambda)\boldsymbol{P_X}}\right)_{i-L} \text{ for } i = L+1, \dots, 2L
\end{gathered} \tag{9.7}$$

The weights for these sigma points are shown in Eq. (9.8), where $W_0^{(m)}$ and $W_i^{(m)}$ are for computing the mean vector of $\boldsymbol{Y}$, and $W_0^{(c)}$ and $W_i^{(c)}$ are for computing the covariance matrix of $\boldsymbol{Y}$.

$$W_0^{(m)} = \frac{\lambda}{(L+\lambda)}$$

$$W_0^{(c)} = \frac{\lambda}{(L+\lambda)} + (1 - \alpha^2 + \beta) \tag{9.8}$$

$$W_i^{(m)} = W_i^{(c)} = \frac{1}{2(L+\lambda)}, i = 1, \ldots, 2L$$

In Eqs. (9.7) and (9.8), $0 \leq \alpha \leq 1$ determines the spread of the sigma points around $\overline{\boldsymbol{X}}$ and is usually set to a small positive value such as 1e-3; $\beta \geq 0$ is a non-negative weighting term to incorporate knowledge of the higher order moments of $\boldsymbol{X}$ and the optimal choice is $\beta = 2$ is $\boldsymbol{X}$ has a Gaussian distribution; $\lambda = \alpha^2(L+\kappa) - L$ is a scaling parameter where κ is usually set to zero; $\left(\sqrt{(L+\lambda)\boldsymbol{P_X}}\right)_i$ is the i-th row of the matrix square root.

Then the sigma points in Eq. (9.7) are propagated through $\boldsymbol{Y} = g(\boldsymbol{X})$ to obtain the sigma points of $\boldsymbol{Y}$:

$$\boldsymbol{\mathcal{Y}}_i = g(\boldsymbol{\chi}_i) \text{ for } i = 0, \ldots, 2L \tag{9.9}$$

And the mean vector and covariance matrix of $\boldsymbol{Y}$ are approximated as

$$\overline{\boldsymbol{Y}} \approx \sum_{i=0}^{2L} W_i^{(m)} \boldsymbol{\mathcal{Y}}_i$$

$$\boldsymbol{P_Y} \approx \sum_{i=0}^{2L} W_i^{(c)} (\boldsymbol{\mathcal{Y}}_i - \overline{\boldsymbol{Y}})(\boldsymbol{\mathcal{Y}}_i - \overline{\boldsymbol{Y}})^T \tag{9.10}$$

The UT is accurate to the third order for Gaussian inputs, and to at least the second order for non-Gaussian inputs [153,155], thus resultant unscented Kalman filter proves to be more accurate than the extended Kalman filter which linearize the state/measurement functions to the first order. UT also avoids computing the Jocobian matrix which may bring computational difficulty in the case of high non-linearity [155].

9.2.3 Unscented Kalman Filter

The UKF consists of two parts: time update and measurement update. Time update means propagating to next time step and measurement update means inference using observation data. UKF considers the state/measurement function in Eqs. (5.5) and (5.6) as a single function with $\boldsymbol{X}_a^t = \begin{bmatrix} \boldsymbol{X}^t \\ \boldsymbol{v}^t \\ \boldsymbol{\sigma}^t \end{bmatrix}$ as inputs and $\begin{bmatrix} \boldsymbol{X}^{t+1} \\ \boldsymbol{y}^{t+1} \end{bmatrix}$ as outputs. Here the mean and covariance of the inputs are

$$\bar{\boldsymbol{X}}_a^t = \left[\bar{\boldsymbol{X}}_a^{t\,T}, \boldsymbol{0}, \boldsymbol{0} \right]^T, \qquad \boldsymbol{P}_a^t = \begin{bmatrix} \boldsymbol{P}^t & 0 & 0 \\ 0 & \boldsymbol{Q}^t & 0 \\ 0 & 0 & \boldsymbol{R}^t \end{bmatrix} \tag{9.11}$$

where the superscript T means transpose. The sigma points of $\boldsymbol{X}_a^t$ and the corresponding weights $W_i^{(c)}$ and $W_i^{(m)}$can be generated by Eqs. (9.7) and (9.8), where L is the dimension of $\boldsymbol{X}_a^t$. Here a sigma point of $\boldsymbol{X}_a^t$ is in the format

$$\boldsymbol{\chi}_a^{i,t} = \begin{bmatrix} \boldsymbol{\chi}_X^{i,t} \\ \boldsymbol{\chi}_v^{i,t} \\ \boldsymbol{\chi}_\sigma^{i,t} \end{bmatrix} \text{ for} i = \ 0,1, \dots, 2L \tag{9.12}$$

where the three components correspond to $\boldsymbol{X}^t$, $\boldsymbol{v}^t$ and $\boldsymbol{\sigma}^t$.

Time update aims to propagate $\boldsymbol{X}^t$ to time step $t+1$ and obtain the prior distribution of $\boldsymbol{X}^{t+1}$. Note that the distribution of $\boldsymbol{X}^t$ is the posterior which has been updated using data $\boldsymbol{y}^t$, or simply the prior of $\boldsymbol{X}^t$ if $\boldsymbol{y}^t$ is missing. Based on the UT, the sigma points of $\boldsymbol{X}^{t+1}$ are

$$\boldsymbol{\chi}_X^{i,t+1} = f\left(\boldsymbol{\chi}_X^{i,t}, \boldsymbol{\chi}_v^{i,t}\right) \tag{9.13}$$

where $f(\cdot)$ is the state function. Thus the prior mean and covariance of $\boldsymbol{X}^{t+1}$ can be obtained by Eq. (9.10) as

$$\begin{gathered}\overline{\boldsymbol{X}}^{t+1} = \sum\nolimits_{i=0}^{2L} W_i^{(m)} \boldsymbol{\chi}_X^{i,t+1} \\ \boldsymbol{P}_{t+1} = \sum\nolimits_{i=0}^{2L} W_i^{(c)} \left(\boldsymbol{\chi}_X^{i,t+1} - \overline{\boldsymbol{X}}^{t+1}\right)\left(\boldsymbol{\chi}_X^{i,t+1} - \overline{\boldsymbol{X}}^{t+1}\right)^T\end{gathered} \tag{9.14}$$

Measurement update aims to calculate the posterior distribution of $\boldsymbol{X}^t$ using data $\boldsymbol{Y}^t = \boldsymbol{y}^t$. Based on the UT, the sigma points of $\boldsymbol{Y}^t$ are

$$\boldsymbol{\chi}_Y^{i,t} = h\left(\boldsymbol{\chi}_X^{i,t}, \boldsymbol{\chi}_\sigma^{i,t}\right) \tag{9.15}$$

Thus the mean and covariance of $\boldsymbol{Y}^t$ are

$$\begin{gathered}\overline{\boldsymbol{Y}}^{t} = \sum\nolimits_{i=0}^{2L} W_i^{(m)} \boldsymbol{\chi}_Y^{i,t} \\ \boldsymbol{P}_Y^t = \sum\nolimits_{i=0}^{2L} W_i^{(c)} \left(\boldsymbol{\chi}_Y^{i,t} - \overline{\boldsymbol{Y}}^{t}\right)\left(\boldsymbol{\chi}_Y^{i,t} - \overline{\boldsymbol{Y}}^{t}\right)^T\end{gathered} \tag{9.16}$$

and the covariance of $\boldsymbol{X}^t$ and $\boldsymbol{Y}^t$ is

$$\boldsymbol{P}_{XY}^t = \sum\nolimits_{i=0}^{2L} W_i^{(c)} \left(\boldsymbol{\chi}_X^{i,t} - \overline{\boldsymbol{X}}^{t}\right)\left(\boldsymbol{\chi}_Y^{i,t} - \overline{\boldsymbol{Y}}^{t}\right)^T \tag{9.17}$$

Thus the Kalman gain $\boldsymbol{K}^t$, the posterior mean $\overline{\boldsymbol{X}}^{t''}$ and covariance $\boldsymbol{P}^{t''}$ of $\boldsymbol{X}^t$ are

$$\boldsymbol{K}^t = \boldsymbol{P}_{XY}^t \boldsymbol{P}_Y^{t\,-1}$$

$$\bar{\boldsymbol{X}}^{t''} = \bar{\boldsymbol{X}}^t - \boldsymbol{K}^t(\boldsymbol{y}^t - \bar{\boldsymbol{Y}}^t) \tag{9.18}$$

$$\boldsymbol{P}^{t''} = \boldsymbol{P}^t - \boldsymbol{K}^t \boldsymbol{P}_Y^t \boldsymbol{K}^{t^T}$$

The obtained $\bar{\boldsymbol{X}}^{t''}$ and $\boldsymbol{P}^{t''}$ can be used as $\boldsymbol{X}^t$ and $\boldsymbol{P}^t$ in the time update of Eqs. (9.11) to (9.14) and propagate to time step $t + 1$.

It is clear that the time update and measurement update are two distinct calculations in UKF. For the dynamic system depicted by Eqs. (9.5) and (9.6), these two parts are iteratively implemented, as shown in Figure 9.3.

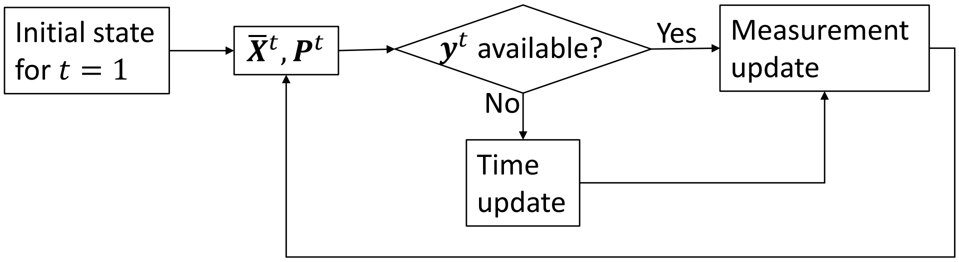

Initial state for $t = 1$
$\bar{\boldsymbol{X}}^t, \boldsymbol{P}^t$
$\boldsymbol{y}^t$ available?
Yes
No
Measurement update
Time update

Figure 9.3 Flowchart to implement the unscented Kalman filter

If the DBN has the topology shown in Figure 9.1 so that it can be depicted as a dynamic system in Eqs. (9.5) and (9.6), the UKF can be applied. In addition, if a static BN can be depicted by the measurement function in Eq. (9.6), we can simply use the measurement update part of UKF as the Bayesian inference algorithm for this static BN.

However, just like the Kalman filter and extended Kalman filter, UKF also assumes that that all the state variables are Gaussian. In some problems, this assumption might cause large error if the distribution of state variables is highly nonlinear, e.g., multi-modal [156].

9.3 Proposed Fast Inference Algorithm

UKF can be a fast inference algorithm for DBN if the DBN has the two-layer topology shown in Figure 9.1, provided the assumption of Gaussian distributions for the state variables is acceptable. For a DBN of more than two layers such as the one in Figure 9.2, UKF is not applicable. This section proposes a network collapsing method to covert a DBN of arbitrary topology to an equivalent DBN of two layers, so that the usage of UKF can be extended to a DBN of any topology.

9.3.1 Network Collapsing and Bayesian Inference

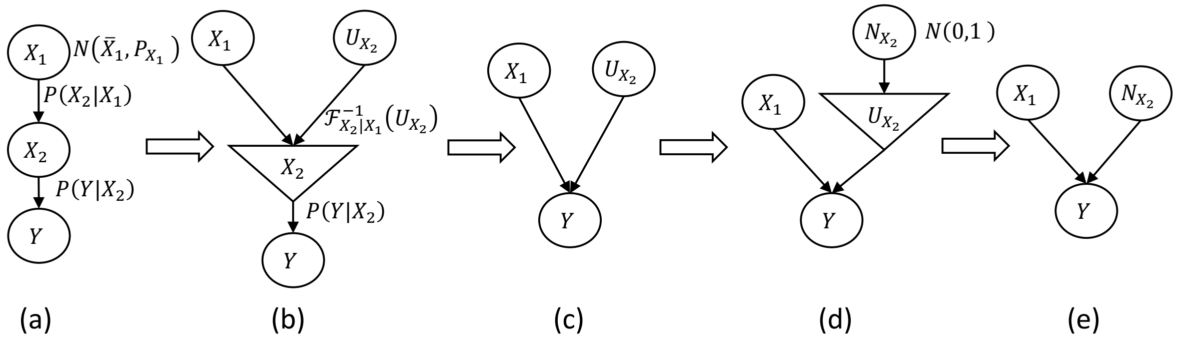


Figure 9.4 Network collapsing: Example 1

The basic concept of the proposed network collapsing method is to introduce an auxiliary variable to each CPD within the state variables, and this concept can be explained by the collapsing of the three-layer BN in Figure 9.4. In Figure 9.4(a), the state variables X_1 and X_2 are at layer 1 and 2 respectively, and layer 3 is the observation variable Y. X_1 has a Gaussian prior $N(\bar{X}_1, P_{X_1})$; the CPD between the state variables is $p(X_2|X_1)$; and the CPD for the observation variable is $p(Y|X_2)$. We also assume that the format of $p(Y|X_2)$ is a measurement function similar to Eq. (9.6):

$$Y = h(X_2, \sigma) \tag{9.19}$$

where σ is the noise term of the zero-mean Gaussian distribution $\sigma \sim N(0, R)$, so that Y is still a random variable at given X_2.

Figure 9.4(b) introduces an auxiliary variable U_{X_2} to the CPD $p(X_2|X_1)$, thus X_2 now is a functional node such that

$$X_2 = \mathcal{P}_{X_2|X_1}^{-1}\left(U_{X_2}\right) \tag{9.20}$$

where $\mathcal{P}_{X_2|X_1}^{-1}(\cdot)$ is the inverse CDF of $p(X_2|X_1)$. This functional node does not count as a layer thus the BN has been collapsed to two layers, as shown in Figure 9.4(c). By substituting Eq. (9.20) into Eq. (9.19), this BN can be expressed by a single measurement function

$$Y = h\left(\mathcal{P}_{X_2|X_1}^{-1}\left(U_{X_2}\right), \sigma\right) = h\left(X_1, U_{X_2}, \sigma\right) \tag{9.21}$$

Now the state variables are X_1 and U_{X_2} in the first (upper) layer, and the observation node is Y in the second (lower) layer. However, UKF is still not applicable to Eq. (9.21) since U_{X_2} has a uniform distribution $U(0,1)$ while the UKF requires all state variables to have Gaussian distributions. As shown in Figure 9.4(d), we can handle this problem by introducing another standard Gaussian variable $N_{X_2} \sim N(0,1)$ where we have

$$U_{X_2} = \Phi\left(N_{X_2}\right) \tag{9.22}$$

where $\Phi(\cdot)$ is the CDF function of standard Gaussian distribution. Now U_{X2} is converted to a functional node which does not count as a layer. The final collapsed BN is shown in Figure 9.4(e), which can be expressed by a measurement function

$$Y = h\left(\mathcal{P}_{X_2|X_1}^{-1}\left(\Phi\left(N_{X_2}\right)\right), n\right) = h\left(X_1, N_{X_2}, n\right) \tag{9.23}$$

where both of the state variables X_1 and N_{X_2} have Gaussian distribution so that the measurement update part of UKF can be applied for Bayesian inference if Y is observed.

A more complex BN example is shown in Figure 9.5. The state variables are $\boldsymbol{X} = \{A, B, C, D, E, F\}$ and the observation nodes are $\boldsymbol{Y} = [Y_1, Y_2]^T$. The root nodes $\{A, B, D, F\}$ have Gaussian distributions; the CPDs between the state variables are $p(C|A, B)$ and $p(E|C, D, F)$; and we assume the CPDs for Y_1 and Y_2 can be expressed by a measurement function

$$\boldsymbol{Y} = h(E, F, \boldsymbol{\sigma}) \tag{9.24}$$

where $\boldsymbol{\sigma} \in \mathbb{R}^2$ is the measurement noise of zero-mean Gaussian distribution $N(\boldsymbol{0}, \boldsymbol{R})$.

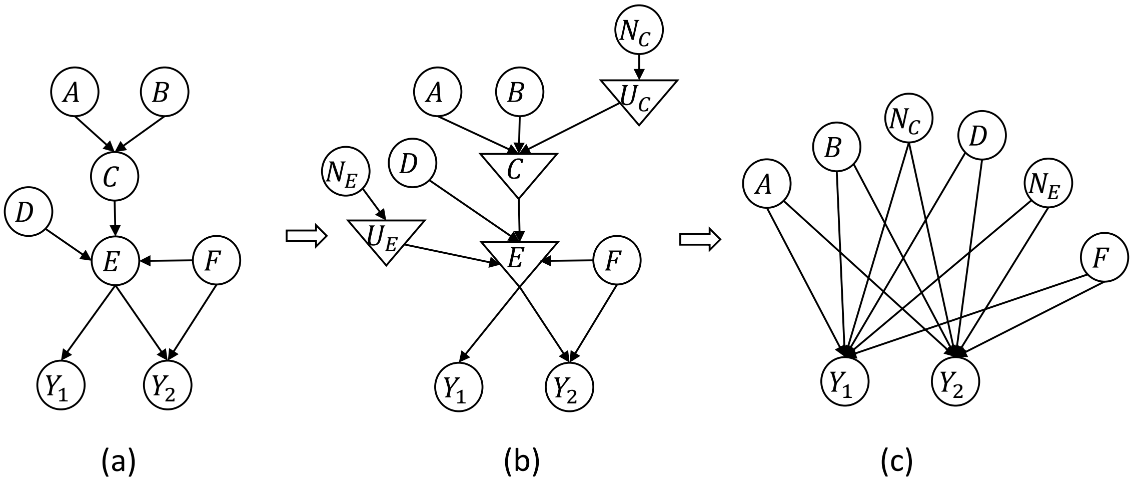


Figure 9.5 Network collapsing: Example 2

Similar to the example in Figure 9.4, we collapse the BN into two layers by introducing auxiliary variable U_C and U_E for the CPDs between the state variables:

$$C = \mathcal{P}^{-1}_{C|A,B}(U_C), E = \mathcal{P}^{-1}_{E|C,D,F}(U_E) \tag{9.25}$$

And standard Gaussian variables N_C and N_E are introduced to U_C and U_E so that all the state variables in the transformed network Figure 9.5(b) have Gaussian distributions:

$$U_C = \Phi(N_C), U_E = \Phi(N_E) \tag{9.26}$$

Finally the BN is collapsed to two layers in Figure 9.5(c). By substituting Eqs. (9.25) and (9.26) into Eq. (9.24), we can express the collapsed BN by a single measurement function

$$\boldsymbol{Y} = h\left(\mathcal{P}^{-1}_{E|\mathcal{P}^{-1}_{C|A,B}(\Phi(N_C)),D,F}\big(\Phi(N_E)\big), F, \boldsymbol{\sigma}\right) = h(A, B, N_C, D, N_E, F, \boldsymbol{\sigma}) \tag{9.27}$$

In Eq. (9.27), all the state variables $\{A, B, N_C, D, N_E, F\}$ have Gaussian distribution so that the measurement update part of UKF can be applied for Bayesian inference if $\boldsymbol{Y} = \{Y_1, Y_2\}$ are observed.

In general, in a BN of arbitrary topology, if the CPD between the observation variables $\boldsymbol{Y}$ and their parent state variables $\mathrm{Pa}(\boldsymbol{Y})$ is

$$\boldsymbol{Y} = h(\mathrm{Pa}(\boldsymbol{Y}), \boldsymbol{\sigma}) \tag{9.28}$$

Then by introducing auxiliary variables for the CPDs between state variables and standard Gaussian variables to convert these auxiliary variables to functional nodes, this BN of arbitrary topology can be collapsed to a two-layer BN expressed by a single measurement function of the form

$$\boldsymbol{Y} = h(\boldsymbol{X}_r, \boldsymbol{N}, \boldsymbol{\sigma}) \tag{9.29}$$

where $\boldsymbol{X}_r$ are the root nodes without any parents node and assumed to have Gaussian distribution; $\boldsymbol{N}$ is the standard Gaussian variables introduced for the auxiliary variables, and $\boldsymbol{n}$ is the measurement noise. The state variables in Eq. (9.29) are $\boldsymbol{X}_c = [\boldsymbol{X}_r^T, \boldsymbol{N}^T]^T$.

When $\boldsymbol{Y}$ are observed, the measurement update part of UKF in Eq. (9.15) to Eq. (9.18) can be implemented as follows:

1. The sigma points $\boldsymbol{\chi}_X^{i,t}, \boldsymbol{\chi}_\sigma^{i,t}$ in Eq. (9.15) are generated by Eq. (9.7) where L is the dimension of $\boldsymbol{X}_c = [\boldsymbol{X}_r^T, \boldsymbol{N}^T]^T$; the weights of the sigma points are calculated by Eq. (9.8);
2. The sigma points $\boldsymbol{\chi}_Y^{i,t}$ are calculated by propagating the sigma points $\boldsymbol{\chi}_X^{i,t}, \boldsymbol{\chi}_\sigma^{i,t}$ through Eq. (9.29);

3. The posterior distribution of $\boldsymbol{X}_c = [\boldsymbol{X}_r^T, \boldsymbol{N}^T]^T$ is calculated using Eqs. (9.16) to (9.18).

9.3.2 Posterior Distribution of Collapsed Nodes

The objective in Bayesian inference is to obtain the joint posterior distribution of all the state variables. In Eq. (9.29), the Bayesian inference using Eq. (9.15) to Eq. (9.18) results in the posterior of $\boldsymbol{X}_r$ and $\boldsymbol{N}$ where $\boldsymbol{X}_r$ are the root nodes and $\boldsymbol{N}$ are the introduced standard Gaussian variables for collapsing the network into two layers. However, the posteriors of the non-root state variables $\boldsymbol{X}_{nr}$ are missing. For example, the posterior of X_2 is missing for the network in Figure 9.4, and the posteriors of C and E are missing for the network in Figure 9.5.

This problem can be solved by another unscented transform. A non-root state variable X^r is the output of a deterministic function of

$$X_{nr} = \mathcal{P}^{-1}_{X_{nr}|\mathrm{Pa}(X_{nr})}\left(\Phi\left(N_{X_{nr}}\right)\right) \tag{9.30}$$

where $\mathrm{Pa}(X_{nr})$ denotes the parent nodes of X_{nr}; and $N_{X_{nr}}$ is the introduced standard Gaussian variable for X_{nr}. Eq. (9.30) is actually a function $f\colon \left(\mathrm{Pa}(X_{nr}), N_{X_{nr}}\right) \to X_{nr}$. Using the posterior mean and variable of $\mathrm{Pa}(X_{nr})$ and $N_{X_{nr}}$, the posterior mean and variance of X_{nr} can be computed by another unscented transform in Eq. (9.7) to Eq. (9.10).

However, the covariance within $\boldsymbol{X}_{nr}$ and the covariance between $\boldsymbol{X}_r$ and $\boldsymbol{X}_{nr}$ are missing if Eq. (9.30) is built separately for each non-root state variable. To obtain covariance of $\boldsymbol{X} = [\boldsymbol{X}_{nr}^T, \boldsymbol{X}_r^T]^T$, a multi-output function $f\colon (\boldsymbol{X}_r, \boldsymbol{N}) \to \boldsymbol{X}$ can be constructed:

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{X}_r \\ \boldsymbol{X}_{nr} \end{bmatrix} = \begin{bmatrix} \boldsymbol{X}^r \\ \mathcal{P}^{-1}_{X_{nr,1}|\mathrm{Pa}(X_{nr,1})}\left(\Phi\left(N_{X_{nr,1}}\right)\right) \\ \mathcal{P}^{-1}_{X_{nr,2}|\mathrm{Pa}(X_{nr,2})}\left(\Phi\left(N_{X_{nr,2}}\right)\right) \\ \cdots \\ \mathcal{P}^{-1}_{X_{nr,d}|\mathrm{Pa}(X_{nr,d})}\left(\Phi\left(N_{X_{nr,d}}\right)\right) \end{bmatrix} \tag{9.31}$$

where $\boldsymbol{X}_{nr} = \left[X_{nr,1}, X_{nr,2}, \dots, X_{nr,d}\right]^T$ and d is the size of $\boldsymbol{X}_{nr}$. Note that the root nodes $\boldsymbol{X}_r$ are also part of the outputs of Eq. (9.31), so that the full covariance matrix of $\boldsymbol{X} = [\boldsymbol{X}_{nr}^T, \boldsymbol{X}_r^T]^T$ can be computed. Based on Eq. (9.31), the corresponding functions for the examples in Figure 9.4 and Figure 9.5 are

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} X_1 \\ \mathcal{P}^{-1}_{X_2|X_1}\left(\Phi\left(N_{X_2}\right)\right) \end{bmatrix}, \qquad \begin{bmatrix} A \\ B \\ C \\ D \\ E \\ F \end{bmatrix} = \begin{bmatrix} A \\ B \\ \mathcal{P}^{-1}_{C|A,B}\left(\Phi(N_C)\right) \\ D \\ E \\ \mathcal{P}^{-1}_{E|C,D,F}\left(\Phi(N_E)\right) \end{bmatrix} \tag{9.32}$$

9.3.3 Forward Propagation to Next Time Step

The proposed method in Sections 9.3.2 and 9.3.3 can be used for the Bayesian inference in a static BN or within one time instant of a DBN. In a DBN, if the state function is $\boldsymbol{X}^{t+1} = f(\boldsymbol{X}^t, \boldsymbol{v}^t)$, where $\boldsymbol{v}^t$ denotes the process noise, the propagation (time update) from time step t to time step $t+1$ can be done by the unscented transform in Eq. (9.10) or Eq. (9.14). The sigma points of $\left[\boldsymbol{X}^{t^T}, \boldsymbol{v}^{t^T}\right]^T$ and the weights are generated by their mean value vector $\left[\overline{\boldsymbol{X}}^{t^T}, \boldsymbol{0}\right]^T$ and covariance matrix $\begin{bmatrix} \boldsymbol{P}^t & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q}^t \end{bmatrix}$. Here $\boldsymbol{P}^t$ is the covariance matrix of $\boldsymbol{X}^t$ obtained posterior covariance of $\boldsymbol{X}^t$ from Eq. (9.31) if data $\boldsymbol{y}_t$ are available, or the prior covariance of $\boldsymbol{X}^t$ propagated from time step $t-1$ if $\boldsymbol{y}^{\boldsymbol{t}}$ are NOT available; and $\boldsymbol{Q}^t$ is the covariance matrix of process noise $\boldsymbol{v}^t$.

Note that the introduced standard Gaussian variables $\boldsymbol{N}$ in Eq. (9.29) are not propagated to time step $t+1$. Therefore, when data $\boldsymbol{y}^{t+1}$ are available, new auxiliary variables and new standard Gaussian variables $\boldsymbol{N}$ are required for the Bayesian inference for time step $t+1$. This introduction of new auxiliary variables and standard Gaussian variables at each observation time step does not significantly increase the computational effort, since the backpropagation in Bayesian inference is only done one step at a time; thus what is needed is proper book-keeping at each time step to use the realizations of the appropriate auxiliary variables and standard Gaussian variables corresponding to each time step.

9.3.4 Summary

The proposed method in this section is a fast Bayesian inference algorithm for the BN of arbitrary topology, as long as the assumption that all the state variables has a joint Gaussian distribution is acceptable.

In the proposed method, an auxiliary variable and a corresponding standard Gaussian variable are introduced to each CPD between the state variables, so that the original BN can be collapsed to a two-layer BN thus the measurement update part of the UKF can be applied for Bayesian inference, as shown in Section 9.3.1. The posterior distribution of the collapsed nodes can be recovered by another unscented transform, as shown in Section 9.3.2. And for DBN, the resultant posterior of the state variables can be propagated to the next time step via another unscented transform, as shown in Section 9.3.3. The proposed method is applied iteratively for a DBN to track the evolution of state variables along with time.

9.4 Numerical Examples

9.4.1 Mathematical Example

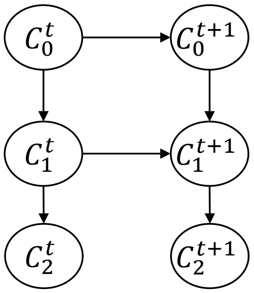


Figure 9.6 DBN of the mathematical example

In this example, the designed DBN for illustration is shown in Figure 9.6, where C_0^t and C_1^t are unknown state variables to be tracked, and C_2^t is the observation node to be measured in each time step. We assume that this DBN has 20 steps in total, i.e., $t = 1$ to 20. At $t = 1$, the prior distribution of the root node $C_0^{t=1}$ and the subsequent CPDs are defined as

$$C_0^{t=1}\sim N(2,0.5^2),\quad C_1^{t=1}\sim N(C_0^{t=1}+10,1^2),\quad C_2^{t=1}\sim N\left({C_1^{t=1}}^{1.2},5^2\right) \tag{9.33}$$

At $t > 1$, the CPDs are

$$C_0^t = C_0^{t-1},\quad C_1^t\sim N(C_0^t + 0.9 * C_1^{t-1} + 1,1^2),\quad C_2^t\sim N\left({C_1^t}^{1.2},5^2\right) \tag{9.34}$$

Eqs. (9.33) and (9.34) shows that the true value of C_0^t is invariant with time, but the true value of C_1^t is changing with time. The CPD for the observation node C_t^2 is always $N\left({C_1^t}^{1.2},5^2\right)$, which can be also expressed as a measurement function $C_2^t = {C_1^t}^{1.2} + \sigma_t$, where measurement noise $\sigma_t\sim N(0,5^2)$.

The measurement data of C_2^t is shown in Figure 9.7. These data are actually synthetic thus the true values of C_0^t and C_1^t in each step are known. We will compare these true values with the Bayesian inference results using the measurement data of C_2^t.

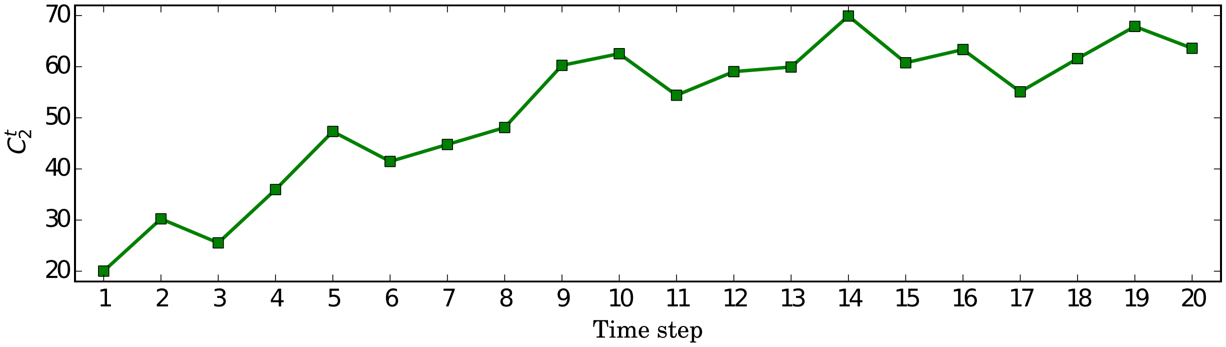


Figure 9.7 Measurement data of C_2^t

The proposed method in Section 9.3 is applied to calibrate the true value of C_0^t and track the history of C_1^t, and compare with the true values. In addition, particle filter also is used to solve the same problem and compare with the proposed method. These comparisons are shown in Figure 9.8 and Figure 9.9. The error bars in these two values indicates the 95% confidence interval of the posterior distribution, so that a narrower error bar indicates lower uncertainty in the posterior distribution.

Figure 9.8 shows that the true value of C_0^t is 2.5; and both particle filter and the proposed method give posterior distributions approaching the value with decreasing uncertainty. Figure 9.9 shows that both particle filter and the proposed method succeed in tracking the history of C_1^t.

The Bayesian inference results by particle filter and the proposed method are highly close so that their error bars are almost overlapping in Figure 9.8 and Figure 9.9. However, the proposed method is computationally much more efficient than particle filter: in a PC of Intel i7 CPU and 16 GB RAM, the time cost of particle filter is 10 seconds with 5000 particles, while the time cost of the proposed method is less than 1 second.

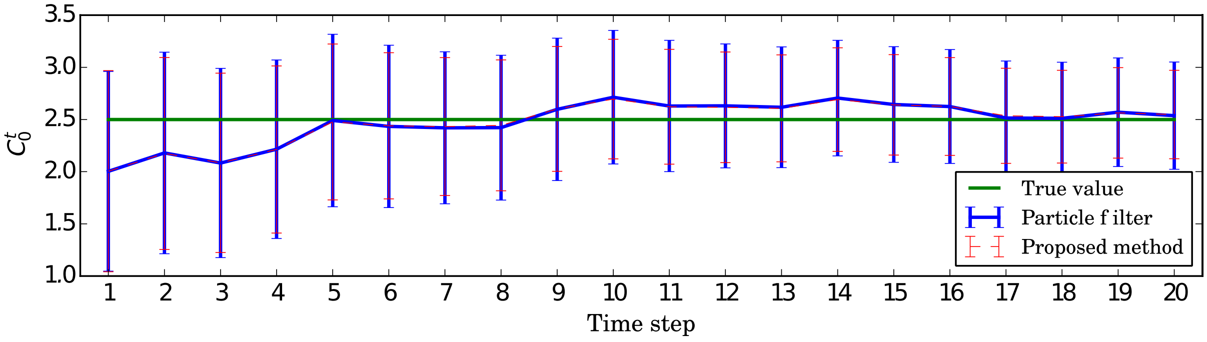


Figure 9.8 Bayesian inference for C_0^t

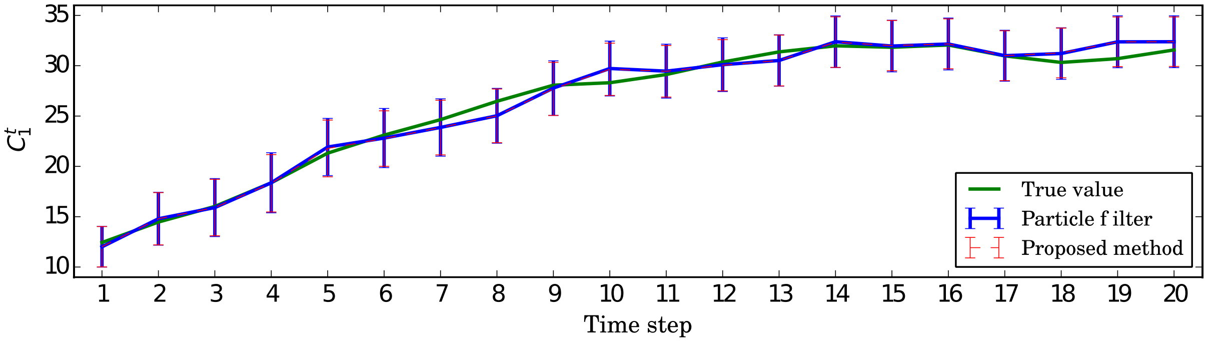


Figure 9.9 Bayesian inference for C_1^t

9.4.2 Crack Growth Example

Here we use the crack growth problem in Chapter 6 as another example. The discrete variables are ignored (fixed) since the proposed algorithm in this section is only applicable to DBN of continuous variables. The geometry and the FEA model of this problem can be found in Figure 6.1 and Figure 6.8, respectively. A Gaussian process (GP) surrogate model has been established to replace it. At given θ and P, the prediction of ΔS is a Gaussian variable $\Delta S \sim N\left(\mu(\theta, P), \sigma^2(\theta, P)\right)$. This actually constructs a CPD in the subsequent Bayesian network, where the parents nodes are θ and P, the child node is ΔS.

This examples assumes that a crack has been initialized in the location of maximum stress. For the sake of illustration, this example assumes a mode I uniaxial crack; thus the range of stress intensity factor in one time step is

$$\Delta K = 1.2F\Delta S\sqrt{\pi a_0} \tag{9.35}$$

where $1.2F$ is the crack shape factor and ΔS is the stress range and a_0 is the initial crack length in the current time step. Here F is defined as a multiplier for the shape factor, and the uncertainty in F represents the uncertainty in the shape factor. The output of Eq. (9.36), ΔK, is a functional node in the subsequent Bayesian network. In addition, this example assumes that the prior distribution of the initial crack length at $t = 1$ is $N(0.0588{,}0.0005^2)$.

Next, this section still uses the Paris' law to compute the long crack growth Δa in each time step:

$$\frac{\mathrm{d}a}{\mathrm{d}N} = C\Delta K^m \tag{9.36}$$

where $C = 1.51 \times 10^{-9}$ and $m = 3.7$ are the Paris' law parameters obtained from material coupon experiments; C and m are assumed to be known constants in this example but can be be easily treated as random variables of unknown true values and included in the Bayesian network if needed; $\mathrm{d}a/\mathrm{d}N$ is the crack growth rate, and its magnitude is equal to the predicted crack growth Δa in one time step. The crack length after the current time step is $a = a_0 + \Delta a$. In the subsequent Bayesian network Δa and a are functional nodes.

In this example, the crack length a is measurable with measurement error $N(0, \sigma_a^2)$ where $\sigma_a = 10^{-4}$; the load P is also measurable with measurement error $N(0, \sigma_P^2)$ where $\sigma_P = 0.002$. Thus the two observation variables in the Bayesian network are a_{obs} and P_{obs}, where the corresponding CPDs are $a_{obs} \sim N(a, \sigma_a^2)$ and $P_{obs} \sim N(P, \sigma_P^2)$.

A Bayesian network is established for this example, as shown in the left half of Figure 9.10 (functional nodes are denoted by triangles). At $t = 1$, the prior distribution of the root nodes are assumed to be $F^{t=1} \sim N(0.8, 0.08^2), \theta^{t=1} \sim N(0.08, 0.008^2), P^{t=1} \sim N(0.25, 0.01^2)$, and $a_0^{t=1} \sim N(0.0588, 0.0005^2)$.

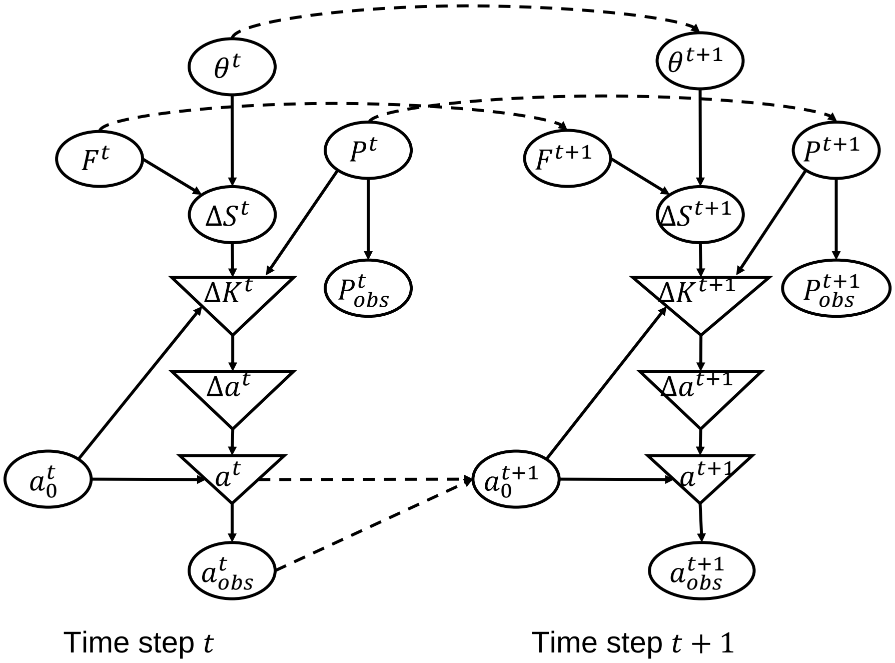


Figure 9.10 Dynamic Bayesian network of Example 2

In this time dependent problem, the state function, which is the transition from t to $t+1$, is also required. This example models the load P as a first-order auto-regressive model $P^{t+1} = 0.2 + 0.2P^t + \epsilon^t$, where the white noise term is $\epsilon_t \sim N(0, 0.01^2)$. a_0^{t+1} in Figure 9.10 is the initial crack length at time step $t+1$. If a^t is not measured, we define $a_0^{t+1} = a^t$, thus the posterior distribution of a^t is the prior distribution of a_0^{t+1}; if a^t is measured, we define $a_0^{t+1} = a_{obs}^t + N(0, \sigma_a^2)$, i.e., the measurement value plus measurement noise. Since θ and F are time-invariant model parameters of unknown true values, the state function for this example is

$$\begin{bmatrix} \theta_{t+1} \\ F_{t+1} \\ P_{t+1} \\ a_0^{t+1} \end{bmatrix} = \begin{bmatrix} \theta^t \\ F^t \\ 0.2 + 0.2P^t + \epsilon^t \\ a^t \text{ if } a^t \text{ not measured; } a_{obs}^t + N(0, \sigma_a^2) \text{ if } a^t \text{ measured} \end{bmatrix} \tag{37}$$

The objective of this example is to predict the crack growth in 20,000 times steps and calibrate the true values of model parameters θ and F. This example assumes that the data of P_{obs} are only available at the first 10,000 time steps; while the data of a_{obs} are only available at the following five time steps: $t = 2000,4000,6000,8000,10000$. All of these data are synthetic and they are generated based on the dynamic Bayesian network in Figure 9.10 using assumed true value of $\theta = 0.0877, F = 0.75, a_0^{t=1} = 0.0588$.

The solution of this example follows the flowchart in Figure 9.3: if data of a_{obs} and/or P_{obs} are available, posterior distributions of state variables are obtained before propagating them to the next time step; otherwise, the prior distribution of state variables are directly propagated to the next time step.

Note that the data in this example are synthetic, thus the true values of θ and F are known, and the true value of crack length in each step is also known. The results by the proposed method is to be compared with these true values to verify the proposed method. The results of the proposed method and the comparison to the true values are shown in Figure 9.11, Figure 9.12 and Figure 9.13.

Figure 9.11 shows the Bayesian inference of F by the proposed method. Recall that the prior distribution of F is $N(0.8,0.08^2)$ and its true value is 0.75. In Figure 9.11 the final posterior distribution of F is $N(0.749,0.026^2)$. The posterior mean is very close to the true value of 0.75, and the posterior standard deviation is also reduced by 67.5% compared to the prior. Thus it is clear that the posterior distribution of F converges to the true value.

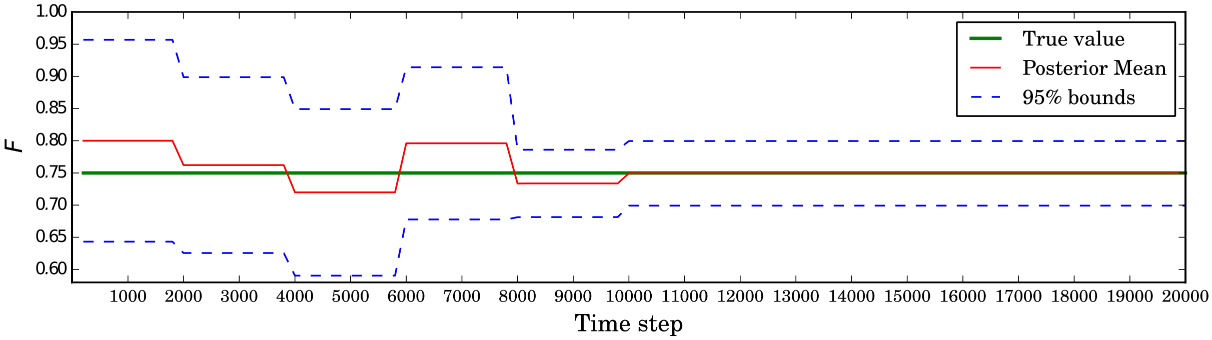


Figure 9.11 Bayesian inference of *F*

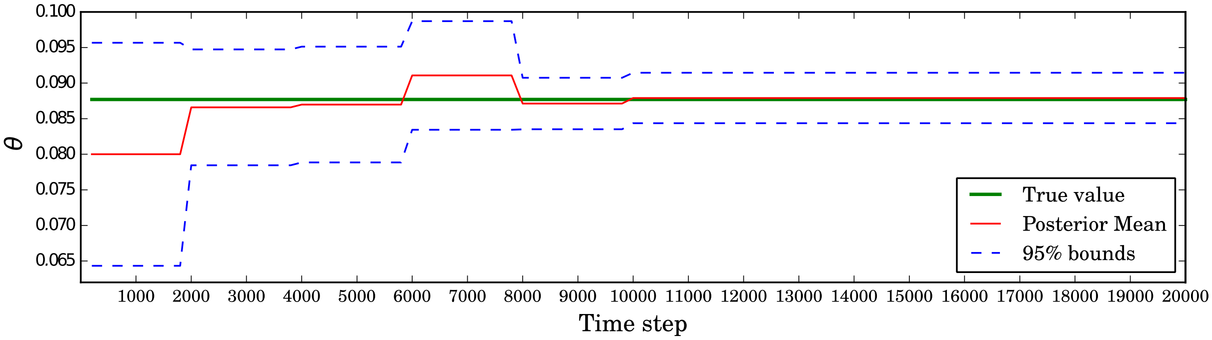


Figure 9.12 Bayesian inference of *θ*

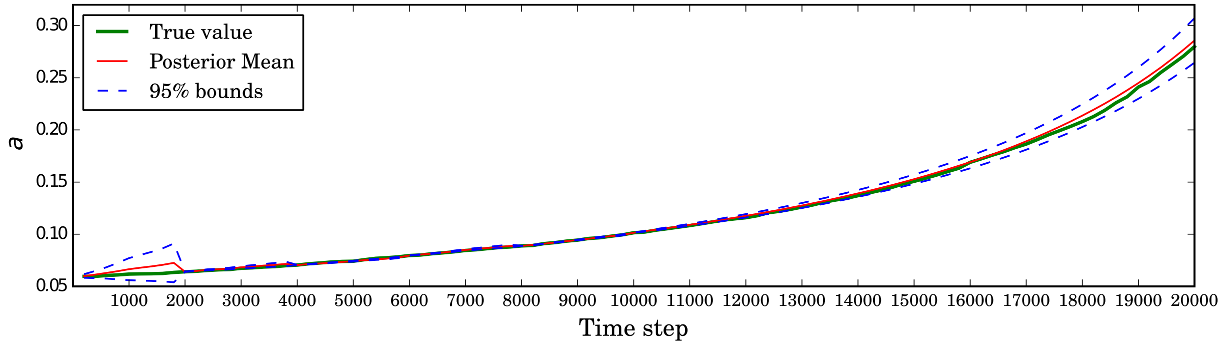


Figure 9.13 Crack growth prediction

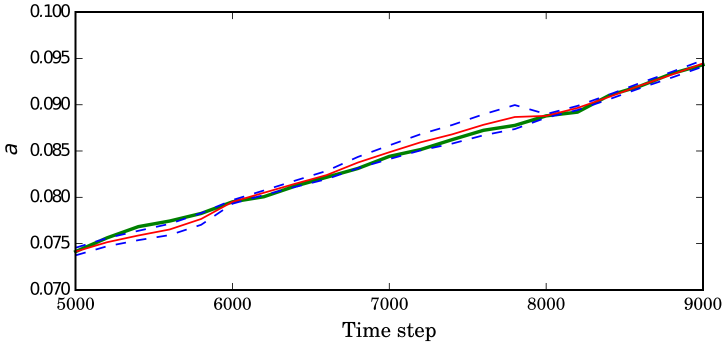


Figure 9.14 Enlarge view of Figure 9.14 from step 5000 to 9000

Similarly, Figure 9.12 also shows that the posterior distribution of θ converge to the true value. Recall that the prior distribution of θ is $N(0.08{,}0.008^2)$ and its true value is 0.0877. In Figure 9.12 the final posterior distribution of θ is $N(0.0879{,}0.0018^2)$. The posterior mean is very close to the true value, and the posterior standard deviation is also reduced by 77.4% compared to the prior.

It is observed that the distribution of F and θ are updated only at the steps of $t = 2000{,}4000{,}6000{,}8000{,}1000$, i.e., only when the measurement data on a_{obs} are available. This is because 1) in time update, the propagation from t to $t+1$ does not change the distributions of F and θ since the state function indicates that $F^{t+1} = F^t$ and $\theta^{t+1} = \theta^t$; and 2) the node P_{obs} is d-separated [157] to F and θ in the Bayesian network, i.e., P_{obs} is independent of F and θ and the data on P_{obs} cannot update the distributions of F and θ during the measurement update. Therefore, only the data of a_{obs} can update and change the distribution of F and θ.

The crack growth prediction is shown in Figure 9.13, which can be divided into three stages:

1. Stage 1 is before the first observation of a_{obs} at $t = 2000$. This stage is purely uncertainty propagation. Due to the large uncertainty in F and θ, the uncertainty of a accumulates fast during this stage and the mean prediction also deviates from the true value.
2. Stage 2 ranges from the first observation of a_{obs} at $t = 2000$ to the last observation of a_{os} at $t = 10000$. This stage includes both uncertainty propagation and Bayesian inference. The prediction matches the true value well and the prediction uncertainty is small due to two reasons: 1) the uncertainty in F and θ are reduced by Bayesian inference using observation data, as shown in Figure 9.11 and Figure 9.12; 2) the observation is used to construct the prior distribution with measurement error, where the observation matches the true value well so constructed prior matches the true value well, and the measurement error is small so that the prior uncertainty is low.

3. Stage 3 is after $t = 10000$. This stage is purely uncertainty propagation. The uncertainty of a accumulates with time, but this accumulation is slow since the uncertainty in F and θ have been reduced significantly during Stage 2. The prediction mean value still matches the true value well since the posterior distributions of F and θ have closely approached their true values.

9.5 Summary

For a non-linear and/or non-Gaussian Bayesian network (BN) with continuous variables, existing inference algorithms are sample-based, such as MCMC for static BN and particle filter for dynamic BN. These sample-based methods are very time-consuming. This research proposed an approximate analytical inference algorithm to obtain the posterior distributions of state variables efficiently. First, this research proposed a network collapsing technique based on the concept of auxiliary variable to convert a multi-layer BN to an equivalent simple two-layer BN. Then unscented Kalman filter is applied to the collapsed BN so that the posterior distributions of state variables can be obtained analytically and efficiently. The proposed method is also able to retrieve the posterior distributions of state variables that are hidden during the collapsing process. In the case of a dynamic BN, the proposed method is also able to propagate the state variables to the next time step analytically using an unscented transform.

The proposed method can be applied to both static and dynamic Bayesian networks, and its main advantage is high computational efficiency. For a static BN where only a single inference is needed so that computational time is not a concern, the computational efficiency of the proposed method may not be a significant advantage. However, the proposed method is particularly suitable for a dynamic BN, where inference and uncertainty propagation are required recursively so that computational effort is a significant concern. Two examples of dynamic BN have been provided in this research to illustrate the proposed method.

CHAPTER 10

CONCLUSION

10.1 Summary of Accomplishments

The overall goal of the research in this dissertation is to develop a versatile and efficient framework for system diagnosis and prognosis under aleatory and epistemic uncertainty. In this research, both time independent and time dependent systems are considered. This target is approached by carrying out studies regarding the state-of-the-art uncertainty quantification and integration techniques. The Bayesian network is utilized as the platform that integrate various sources of uncertainty, and the global sensitivity is utilize the tool for dimension reduction and optimization. The accomplishments and innovations of this dissertation are outlined as follows.

1. **Global sensitivity analysis (GSA) incorporating epistemic uncertainty and time series input**

Chapter 3 developed a novel computational framework to compute the Sobol' indices that quantify the relative contributions of various uncertainty sources towards the system response prediction uncertainty. In the presence of both aleatory and epistemic uncertainty, two challenges were addressed in this research for the model-based computation of the Sobol' indices: due to data uncertainty, input distributions are not precisely known; and due to model uncertainty, the model output is uncertain even for a fixed realization of the input. An auxiliary variable method based on the probability integral transform was introduced to distinguish and represent each uncertainty source explicitly, whether aleatory or epistemic. The auxiliary variables facilitate building a deterministic relationship between the uncertainty sources and the output, which is needed in the

Sobol’ indices computation. The proposed framework was developed for two types of model inputs: random variable input and time series input. A Bayesian autoregressive moving average (ARMA) approach was chosen to model the time series input due to its capability to represent both natural variability and epistemic uncertainty due to limited data. A novel controlled-seed computational technique based on pseudo-random number generation was proposed to efficiently represent the natural variability in the time series input. This controlled-seed method significantly accelerates the Sobol’ indices computation under time series input, and makes it computationally affordable.

2. **System response prediction in multi-level problem with calibration, validation, and relevance analysis**

Chapter 4 proposed a methodology to quantify the uncertainty in the system level prediction by integrating calibration, validation and sensitivity analysis at different levels. The proposed approach considers the validity of the models used for parameter estimation at lower levels, as well as the relevance at the lower level to the prediction at the system level. The model validity is evaluated using a model reliability metric, and models with multivariate output are considered. The relevance is quantified by comparing Sobol’ indices at the lower level and system level, thus measuring the extent to which a lower level test represents the characteristics of the system so that the calibration results can be reliably used in the system level. Finally the results of calibration, validation and relevance analysis are integrated in a roll-up method to predict the system output.

3. **GSA-based resource allocation for robust predictions**

Chapter 5 achieved “robust” test resource allocation, which means that the system response prediction is insensitive to the variability in test outcomes therefore consistent predictions can be achieved under different test outcomes. This research analyzed the uncertainty sources in the generation of synthetic data regarding different test conditions, and found that this objective can be achieved if the contribution of model parameter uncertainty in the synthetic data can be maximized. Global sensitivity analysis (Sobol' index) was used to assess this contribution, and to formulate an optimization problem to achieve the desired consistent prediction. A simulated annealing algorithm was applied to solve this optimization problem. The proposed method is suitable either when only model calibration is considered or when both model calibration and model validation are considered.

4. **Structural health diagnosis and prognosis for time dependent system using dynamic Bayesian network**

Chapter 6 used the concept of dynamic Bayesian networks (DBN) to build a versatile probabilistic model for diagnosis and prognosis, and illustrated the proposed method by an aircraft wing fatigue crack growth example. The proposed method integrates physics models and various aleatory and epistemic uncertainty sources in fatigue crack growth prediction. In diagnosis, the DBN is utilized to track the evolution of the time-dependent variables (dynamic nodes) and calibrate the time-independent variables (static nodes); in prognosis, the DBN is used for probabilistic prediction of crack growth in future loading time steps. This research also proposed a modification of the DBN structure, which does not affect the diagnosis results but reduces time cost significantly by avoiding Bayesian updating with load data. By using particle filtering as the Bayesian inference algorithm for the DBN, the proposed approach handles both discrete and continuous variables of various distribution types, and non-linear relationships between nodes. Challenges in implementing the particle filter in DBN where 1) both dynamic and static nodes exist, and 2) a state variable may have parent nodes across two adjacent Bayesian networks, are also resolved.

5. An efficient sample-based method to estimate the first-order Sobol' index

Chapter 7 developed a new method to directly estimate the first-order Sobol' index based only on available input-output samples, even if the underlying model is unavailable. The innovation is that the conditional variance and mean in the formula of the first-order index are calculated at an unknown but existing location of model inputs, instead of an explicit user-defined location. The proposed method is modular in two aspects: 1) index calculations for different model inputs are separate and use the same set of samples; and 2) model input sampling, model evaluation, and index calculation are separate. Due to this modularization, the proposed method is capable to compute the first-order index if only input-output samples are available but the underlying model is unavailable, and its computational cost is not proportional to the dimension of the model inputs. In addition, the proposed method can also estimate the first-order index with correlated model inputs. Considering that the first-order index is a desired metric to rank model inputs but current methods can only handle independent model inputs, the proposed method contributes to filling this gap.

6. Global sensitivity analysis of a Bayesian network

Chapter 8 extended the use of global sensitivity analysis (GSA) to Bayesian networks in order to calculate the Sobol' sensitivity index of a node with respect to the node of interest. The desired GSA for Bayesian network addresses two challenges. First, the computation of the Sobol' index requires a deterministic input-output function while the Bayesian network has probabilistic relationships between nodes. Second, the computation of the Sobol' index can be expensive, especially if the model inputs are correlated, which is common in a Bayesian network. To solve the first challenge, this research used an auxiliary variable method to convert the path between two nodes in the Bayesian network to a deterministic function, thus making the Sobol' index computation feasible in a Bayesian network. To solve the second challenge, this research used the algorithm proposed in Chapter 7 to directly estimate the first-order Sobol' index from Monte Carlo samples of the prior distribution of the Bayesian network, so that the proposed GSA for Bayesian network is computationally affordable. Before collecting observation, the proposed algorithm can predict the uncertainty reduction of the node of interest purely using the prior distribution samples, thus providing quantitative guidance for effective observation and updating.

7. **An efficient approximate inference algorithm for Bayesian networks with continuous variables**

The inference in a Bayesian network with continuous variables is still challenging if the BN is non-linear and/or non-Gaussian. Chapter 9 proposed a network collapsing technique based on the concept of probability integral transform to convert a multi-layer BN to an equivalent simple two-layer BN, so that the unscented Kalman filter can be applied to the collapsed BN and the posterior distributions of state variables can be obtained analytically. For dynamic BN, the proposed method is also able to propagate the state variables to the next time step analytically using the unscented transform, based on the assumption that the posterior distributions of state variables are Gaussian. Thus the proposed method achieves a very fast approximate solution, making it particularly suitable for dynamic BN where inference and uncertainty propagation are required over many time steps.

10.2 Future Works

Section 10.1 listed the accomplishments of this dissertation, and several potential future works may be pursued.

For the resource allocation in Chapter 5, future work will focus on the selection of the best input values (test design) such that the resultant prediction uncertainty can be further reduced. This challenge can be addressed in two ways: 1) optimize the number of tests and test inputs together; or 2) adaptively decide the number of tests and their input conditions based on the observation data as the test campaign progresses.

For the uncertainty integration of time dependent system in Chapter 6, model validation is necessary to assess the quality of the prediction, thus future work needs to focus on model validation for time-dependent systems. Quantification of the prognosis uncertainty is necessary to assist decision making under uncertainty.

For the global sensitivity analysis of the Bayesian network in Chapter 8, the limitation of the proposed method at present is that currently it only considers a single observation, thus an extension to the case of multiple observations needs to be addressed in future work.

For the fast Bayesian inference algorithm in Chapter 9, the proposed method is only applicable for a BN with continuous variables. Future research is needed to develop fast inference algorithms for hybrid BN of both discrete and continuous variables.

10.3 Concluding Remarks

This dissertation focused on developing a framework of system response prediction under uncertainty. The proposed methods address several challenges in uncertainty integration and system response prediction, including: 1) considering both aleatory and epistemic uncertainty; 2) solving multi-level problems where the prediction needs to be extrapolated from lower level of complexity to the system level of interest; and 3) the prediction for time dependent system under uncertainty. In this dissertation, two major mathematical tools, namely sensitivity analysis (GSA) using Sobol' index and the Bayesian network were considered, and this dissertation also contributed to new developments regarding these tools.

All of these new developments above cover various scenarios in the system response prediction, and will be of high value to the decision makers. The new developments in the global sensitivity analysis allows us to reduce the dimension of the system of interest under various types of inputs, no matter whether the system is time-dependent or time-independent. The new developments in Bayesian network allow us to track the status of a complex time-dependent system in real time. All of these new developments helps predict the system response and evolution, thus show a great potential to be used in an industrial problems.

REFERENCES

[1] Aughenbaugh JM, Paredis CJJ. Probability bounds analysis as a general approach to sensitivity analysis in decision making under uncertainty. 2007.

[2] Ferson S, Troy Tucker W. Sensitivity analysis using probability bounding. Reliab Eng Syst Saf 2006;91:1435–42. doi:http://dx.doi.org/10.1016/j.ress.2005.11.052.

[3] Oberguggenberger M, King J, Schmelzer B. Classical and imprecise probability methods for sensitivity analysis in engineering: A case study. Int J Approx Reason 2009;50:680–93. doi:http://dx.doi.org/10.1016/j.ijar.2008.09.004.

[4] Ling Y, Mahadevan S. Integration of structural health monitoring and fatigue damage prognosis. Mech Syst Signal Process 2012;28:89–104. doi:10.1016/j.ymssp.2011.10.001.

[5] Sankararaman S, Ling Y, Mahadevan S. Uncertainty quantification and model validation of fatigue crack growth prediction. Eng Fract Mech 2011;78:1487–504. doi:10.1016/j.engfracmech.2011.02.017.

[6] Helman P, Veroff R, Atlas SR, Willman C. A Bayesian Network Classification Methodology for Gene Expression Data. J Comput Biol 2004;11:581–615. doi:10.1089/cmb.2004.11.581.

[7] Friedman N, Geiger D, Goldszmidt M. Bayesian Network Classifiers. Mach Learn 1997;29:131–63. doi:10.1023/A:1007465528199.

[8] Korb KB, Nicholson AE. Bayesian artificial intelligence. CRC press; 2010.

[9] Poole D. Probabilistic Horn abduction and Bayesian networks. Artif Intell 1993;64:81–129. doi:10.1016/0004-3702(93)90061-F.

[10] Friedman N, Murphy K, Russell S. Learning the structure of dynamic probabilistic

networks. Proc. Fourteenth Conf. Uncertain. Artif. Intell., 1998, p. 139–47.

[11] Lauritzen SL, Spiegelhalter DJ. Local Computations with Probabilities on Graphical Structures and Their Application to Expert Systems. J R Stat Soc Ser B 1988;50:157–224.

[12] Zhang NL, Poole D. A simple approach to Bayesian network computations. Proc. Tenth Can. Conf. Artif. Intell., 1994.

[13] Shachter RD. Intelligent Probabilistic Inference. In: Lemmer JF, N. KL, editors. Uncertain. Artif. Intell., Amsterdam, Holland: 1986, p. 371–82.

[14] Darwiche A. A differential approach to inference in Bayesian networks. J ACM 2003;50:280–305. doi:10.1145/765568.765570.

[15] Murphy KP, Weiss Y, Jordan MI. Loopy belief propagation for approximate inference: an empirical study 1999:467–75.

[16] Eaton ML. Multivariate statistics: a vector space approach. vol. 198. Wiley New York; 1983.

[17] Henrion M. Propagation of uncertainty by probabilistic logic sampling in Bayes' networks. Uncertain. Artif. Intell., vol. 2, 1988, p. 149–64.

[18] Cheng J, Druzdzel MJ. AIS-BN: An Adaptive Importance Sampling Algorithm for Evidential Reasoning in Large Bayesian Networks. J Artif Intell Res 2000;13:155–88. doi:10.1613/jair.764.

[19] Shi P, Mahadevan S. Corrosion fatigue and multiple site damage reliability analysis. Int J Fatigue 2003;25:457–69. doi:10.1016/S0142-1123(03)00020-3.

[20] Levy BC, Benveniste A, Nikoukhah R. High-level primitives for recursive maximum likelihood estimation. IEEE Trans Automat Contr 1996;41:1125–45.

doi:10.1109/9.533675.

[21] Zweig G. A forward-backward algorithm for inference in Bayesian networks and an empirical comparison with HMMs 1996.

[22] Murphy KP. Dynamic bayesian networks: representation, inference and learning. University of California, Berkeley, 2002.

[23] Boyen X, Koller D. Tractable inference for complex stochastic processes 1998:33–42.

[24] Murphy K, Weiss Y. The factored frontier algorithm for approximate inference in DBNs 2001:378–85.

[25] Welch G, Bishop G, Hill C. An introduction to the Kalman filter. Chapel Hill: 2000.

[26] Absi NG, Mahadevan S. Calibration of System Parameters under Model Uncertainty. IMAC XXXII, Orlando, FL: 2014.

[27] Liang B, Mahadevan S. Error and Uncertainty Quantification and Sensitivity Analysis in Mechanics Computational Models. Int J Uncertain Quantif 2011;1:147–61. doi:10.1615/IntJUncertaintyQuantification.v1.i2.30.

[28] Richards SA. Completed Richardson extrapolation in space and time. Commun Numer Methods Eng 1997;13:573–82. doi:10.1002/(SICI)1099-0887(199707)13:7<573::AID-CNM84>3.0.CO;2-6.

[29] Rangavajhala S, Sura VS, Hombal VK, Mahadevan S. Discretization Error Estimation in Multidisciplinary Simulations. AIAA J 2011;49:2673–83. doi:10.2514/1.J051085.

[30] Xu P, Su X, Mahadevan S, Li C, Deng Y. A non-parametric method to determine basic probability assignment for classification problems. Appl Intell 2014. doi:10.1007/s10489-014-0546-9.

[31] Ling Y. Uncertainty quantification in time-dependent reliability analysis. Vanderbilt University, 2013.

[32] Kennedy MC, O’Hagan A. Bayesian calibration of computer models. J R Stat Soc 2001;63:425–64. doi:10.1111/1467-9868.00294.

[33] Rajashekhar MR, Ellingwood BR. A new look at the response surface approach. Struct Saf 1993;12:205–20.

[34] Ghanem R, Spanos PD. Polynomial Chaos in Stochastic Finite Elements. J Appl Mech 1990;57(1):197–202. doi:10.1115/1.2888303.

[35] Rasmussen CE, Williams CKI. Gaussian Processes for Machine Learning. MIT Press; 2006.

[36] Haldar A, Mahadevan S. Probability, reliability, and statistical methods in engineering design. John Wiley & Sons; 2000.

[37] Halpern EF, Weinstein MC, Hunink MGM, Gazelle GS. Representing Both First- and Second-order Uncertainties by Monte Carlo Simulation for Groups of Patients. Med Decis Mak 2000;20:314–22. doi:10.1177/0272989X0002000308.

[38] Dowling NE. Fatigue failure predictions for complicated stress-strain histories. 1971.

[39] Rychlik I. Simulation of load sequences from rainflow matrices: Markov method. Int J Fatigue 1996;18:429–38.

[40] Box GEP, Jenkins GM, Reinsel GC. Time series analysis: forecasting and control. Hoboken, NJ: John Wiley & Sons; 1976. doi:10.1002/9781118619193.

[41] Trucano TG, Swiler LP, Igusa T, Oberkampf WL, Pilch M. Calibration, validation, and sensitivity analysis: What’s what. Reliab Eng Syst Saf 2006;91:1331–57. doi:10.1016/j.ress.2005.11.031.

[42] Oberkampf WL, Barone MF. Measures of agreement between computation and experiment: Validation metrics. J Comput Phys 2006;217:5–36. doi:10.1016/j.jcp.2006.03.037.

[43] Oberkampf WL, Trucano TGG. Verification and validation in computational fluid dynamics. Prog Aerosp Sci 2002;38:209–72. doi:10.1016/S0376-0421(02)00005-2.

[44] Roache PJ. Fundamentals of verification and validation. Hermosa Press; 2009.

[45] Oberkampf WL, Roy CCJ. Verification and validation in scientific computing. Cambridge University Press; 2010.

[46] Hills RG. Roll-up of Validation Results to a Target Application. Sandia Natl Lab Rep SAND2013-7424 2013.

[47] O'Hagan A. Fractional Bayes Factors for Model Comparison. J R Stat Soc 1995;57:99–138.

[48] Mullins J, Li C, Mahadevan S, Urbina A. Optimal Selection of Calibration and Validation Test Samples Under Uncertainty. In: Atamturktur HS, Moaveni B, Papadimitriou C, Schoenherr T, editors. Model Valid. Uncertain. Quantif. Vol. 3 SE - 39, Springer International Publishing; 2014, p. 391–401. doi:10.1007/978-3-319-04552-8_39.

[49] Mahadevan S, Rebba R. Validation of reliability computational models using Bayes networks. Reliab Eng Syst Saf 2005;87:223–32. doi:10.1016/j.ress.2004.05.001.

[50] Ferson S, Oberkampf WL, Ginzburg L. Model validation and predictive capability for the thermal challenge problem. Comput Methods Appl Mech Eng 2008;197:2408–30. doi:10.1016/j.cma.2007.07.030.

[51] Ferson S, Oberkampf WL, Ginzburg L. Validation of imprecise probability models. Int J Reliab Saf 2009;3:3–22.

[52] Rebba R, Mahadevan S. Computational methods for model reliability assessment. Reliab

Eng Syst Saf 2008;93:1197–207. doi:10.1016/j.ress.2007.08.001.

[53] Sankararaman S, Mahadevan S. Assessing the Reliability of Computational Models under Uncertainty. 54th AIAA/ASME/ASCE/AHS/ASC Struct. Struct. Dyn. Mater. Conf., Reston, Virginia: American Institute of Aeronautics and Astronautics; 2013, p. 1–8. doi:10.2514/6.2013-1873.

[54] Saltelli A, Ratto M, Andres T, Campolongo F, Cariboni J, Gatelli D, et al. Global sensitivity analysis: the primer. John Wiley & Sons; 2008.

[55] Helton JC, Johnson JD, Sallaberry CJ, Storlie CB. Survey of sampling-based methods for uncertainty and sensitivity analysis. Reliab Eng Syst Saf 2006;91:1175–209. doi:10.1016/j.ress.2005.11.017.

[56] Hu Z, Du X. Mixed Efficient Global Optimization for Time-Dependent Reliability Analysis. J Mech Des 2015;137:051401. doi:10.1115/1.4029520.

[57] Hu Z, Du X. Time-dependent reliability analysis with joint upcrossing rates. Struct Multidiscip Optim 2013;48:893–907. doi:10.1007/s00158-013-0937-2.

[58] Li C, Mahadevan S. Robust Test Resource Allocation using Global Sensitivity Analysis. 18th AIAA Non-Deterministic Approaches Conf., Reston, Virginia: American Institute of Aeronautics and Astronautics; 2016. doi:10.2514/6.2016-0952.

[59] Li C, Mahadevan S. Relative contributions of aleatory and epistemic uncertainty sources in time series prediction. Int J Fatigue 2016;82:474–86. doi:10.1016/j.ijfatigue.2015.09.002.

[60] Nannapaneni S, Mahadevan S. Uncertainty quantification in performance evaluation of manufacturing processes. 2014 IEEE Int. Conf. Big Data (Big Data), IEEE; 2014, p. 996–1005. doi:10.1109/BigData.2014.7004333.

[61] Sobol’ IM. Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. Math Comput Simul 2001;55:271–80. doi:10.1016/S0378-4754(00)00270-6.

[62] Zhang X, Pandey MD. An effective approximation for variance-based global sensitivity analysis. Reliab Eng Syst Saf 2014;121:164–74. doi:10.1016/j.ress.2013.07.010.

[63] Chen W, Jin R, Sudjianto A. Analytical Variance-Based Global Sensitivity Analysis in Simulation-Based Design Under Uncertainty. J Mech Des 2005;127:875. doi:10.1115/1.1904642.

[64] Saltelli A, Tarantola S. On the Relative Importance of Input Factors in Mathematical Models. J Am Stat Assoc 2002;97:702–9. doi:10.1198/016214502388618447.

[65] Sankararaman S, Mahadevan S. Separating the contributions of variability and parameter uncertainty in probability distributions. Reliab Eng Syst Saf 2013;112:187–99. doi:10.1016/j.ress.2012.11.024.

[66] Angus JE. The probability integral transform and related results. SIAM Rev 1994;36:652–4.

[67] Saltelli A. Making best use of model evaluations to compute sensitivity indices. Comput Phys Commun 2002;145:280–97. doi:10.1016/S0010-4655(02)00280-1.

[68] Li C, Mahadevan S. Uncertainty Quantification and Output Prediction in Multi-level Problems. 16th AIAA Non-Deterministic Approaches Conf., Reston, Virginia: American Institute of Aeronautics and Astronautics; 2014. doi:10.2514/6.2014-0124.

[69] Uturbey W. Identification of ARMA Models by Bayesian Methods Applied to Streamflow Data. 2006 Int Conf Probabilistic Methods Appl to Power Syst 2006:1–7.

doi:10.1109/PMAPS.2006.360240.

[70] Ljung GM, Box GEP. On a Measure of Lack of Fit in Time Series Models. Biometrika 1978;65:297. doi:10.2307/2335207.

[71] Ben-Haim Y. Fatigue Lifetime with Load Uncertainty Represented by Convex Model. J Eng Mech 1994;120:445–62. doi:10.1061/(ASCE)0733-9399(1994)120:3(445).

[72] Echard B, Gayton N, Bignonnet a. A reliability analysis method for fatigue design. Int J Fatigue 2014;59:292–300. doi:10.1016/j.ijfatigue.2013.08.004.

[73] Huang S, Mahadevan S, Rebba R. Collocation-based stochastic finite element analysis for random field problems. Probabilistic Eng Mech 2007;22:194–205. doi:10.1016/j.probengmech.2006.11.004.

[74] Xi Z, Youn BD, Hu C. Random Field Characterization Considering Statistical Dependence for Probability Analysis and Design. J Mech Des 2010;132:101008. doi:10.1115/1.4002293.

[75] Matsumoto M, Nishimura T. Mersenne Twister: A 623-dimensionally Equidistributed Uniform Pseudo-random Number Generator. ACM Trans Model Comput Simul 1998;8:3–30. doi:10.1145/272991.272995.

[76] L'ecuyer P. Good parameters and implementations for combined multiple recursive random number generators. Oper Res 1999;47:159–64.

[77] Matteis A, Pagnutti S. Long-range correlation analysis of the Wichmann-Hill random number generator. Stat Comput 1993;3:67–70. doi:10.1007/BF00153065.

[78] Tierney L. Markov chains for exploring posterior distributions. Ann Stat 1994;22:1701–28.

[79] Mullins J, Li C, Sankararaman S, Mahadevan S, Urbina A. Probabilistic Integration of

Validation and Calibration Results for Prediction Level Uncertainty Quantification: Application to Structural Dynamics. 54th AIAA/ASME/ASCE/AHS/ASC Struct. Struct. Dyn. Mater. Conf., Reston, Virginia: American Institute of Aeronautics and Astronautics; 2013. doi:10.2514/6.2013-1872.

[80] Sankararaman S, Mahadevan S. Comprehensive framework for integration of calibration, verification and validation. 53rd AIAA/ASME/ASCE/AHS/ASC Struct. Struct. Dyn. Mater. Conf., Honolulu, Hawaii: 2012, p. 1–12.

[81] Arendt PD, Apley DW, Chen W. Quantification of Model Uncertainty: Calibration, Model Discrepancy, and Identifiability. J Mech Des 2012;134:100908. doi:10.1115/1.4007390.

[82] Liu F, Bayarri MJ, Berger JO. Modularization in Bayesian analysis, with emphasis on analysis of computer models. Bayesian Anal 2009;4:119–50.

[83] Jeffreys H. An invariant form for the prior probability in estimation problems. Proc R Soc Lond A Math Phys Sci 1946;186:453–61.

[84] Cha S. Comprehensive survey on distance/similarity measures between probability density functions. Int J Math Model Methods Appl Sci 2007;1.

[85] De Maesschalck R, Jouan-Rimbaud D, Massart DL. The Mahalanobis distance. Chemom Intell Lab Syst 2000;50:1–18. doi:10.1016/S0169-7439(99)00047-7.

[86] Li C, Mahadevan S. Global Sensitivity Analysis for System Response Prediction Using Auxiliary Variable Method. 17th AIAA Non-Deterministic Approaches Conf., Reston, Virginia: American Institute of Aeronautics and Astronautics; 2015. doi:10.2514/6.2015-0661.

[87] Singhal A. Modern information retrieval: A brief overview. IEEE Data Eng Bull

2001;24:35–43.

[88] Van Horn KS. Constructing a logic of plausible inference: a guide to Cox's theorem. Int J Approx Reason 2003;34:3–24. doi:10.1016/S0888-613X(03)00051-3.

[89] Li C, Mahadevan S. Sensitivity Analysis for Test Resource Allocation. In: Atamturktur HS, Moaveni B, Papadimitriou C, Schoenherr T, editors. Model Valid. Uncertain. Quantif. Vol. 3 SE - 14, Springer International Publishing; 2015, p. 143–50. doi:10.1007/978-3-319-15224-0_14.

[90] Rosenblatt M. Remarks on some nonparametric estimates of a density function. Ann Math Stat 1956;27:832–7.

[91] Red-Horse JR, Paez TL. Sandia National Laboratories Validation Workshop: Structural dynamics application. Comput Methods Appl Mech Eng 2008;197:2578–84. doi:10.1016/j.cma.2007.09.031.

[92] Chopra AK. Dynamics of Structures: Theory and Applications to Earthquake Engineering. 4th ed. Prentice Hall; 2011.

[93] Conti S, O'Hagan A. Bayesian emulation of complex multi-output and dynamic computer models. J Stat Plan Inference 2010;140:640–51. doi:10.1016/j.jspi.2009.08.006.

[94] Ainsworth M, Oden JTT. A posteriori error estimation in finite element analysis. Comput Methods Appl Mech Eng 1997;142:1–88. doi:10.1016/S0045-7825(96)01107-3.

[95] Sankararaman S, McLemore K, Mahadevan S, Bradford SC, Peterson LD. Test Resource Allocation in Hierarchical Systems Using Bayesian Networks. AIAA J 2013;51:537–50. doi:10.2514/1.J051542.

[96] Li C, Mahadevan S. Uncertainty Quantification and Output Prediction in Multi-level

Problems. 16th AIAA Non-Deterministic Approaches Conf., National Harbor, Maryland: American Institute of Aeronautics and Astronautics; 2014. doi:doi:10.2514/6.2014-0124.

[97] Li C, Mahadevan S. Role of Calibration, Validation, and Relevance in Multi-level Uncertainty Integration. Reliab Eng Syst Saf 2015.

[98] Urbina A. Uncertainty Quantification and Decision Making In Hierarchical Development of Computational Models. Vanderbilty University, 2009.

[99] Vanlier J, Tiemann CA, Hilbers PAJ, van Riel NAW. A Bayesian approach to targeted experiment design. Bioinformatics 2012;28:1136–42. doi:10.1093/bioinformatics/bts092.

[100] Coles D, Prange M. Toward efficient computation of the expected relative entropy for nonlinear experimental design. Inverse Probl 2012;28:055019. doi:10.1088/0266-5611/28/5/055019.

[101] Sebastiani P, Wynn HP. Maximum entropy sampling and optimal Bayesian experimental design. J R Stat Soc 2013;62:145–57.

[102] Terejanu G, Upadhyay RR, Miki K. Bayesian experimental design for the active nitridation of graphite by atomic nitrogen. Exp Therm Fluid Sci 2012;36:178–93. doi:10.1016/j.expthermflusci.2011.09.012.

[103] Schrijver A. Theory of linear and integer programming. John Wiley & Sons; 1998.

[104] Alrefaei MH, Andradóttir S. A Simulated Annealing Algorithm with Constant Temperature for Discrete Stochastic Optimization. Manage Sci 1999;45:748–64. doi:10.1287/mnsc.45.5.748.

[105] Sankararaman S, Mahadevan S. Integration of Model Verification, Validation, and Calibration for Uncertainty Quantification in Engineering Systems. Reliab Eng Syst Saf

2015;138:194–209. doi:10.1016/j.ress.2015.01.023.

[106] Aktepe B, Molent L. Management of airframe fatigue through individual aircraft loads monitoring programs. Int. Aerosp. Congr. Adelaide, 1999.

[107] Lee H, Cho H, Park S. Review of the F-16 Individual Aircraft Tracking Program. J Aircr 2012;49:1398–405. doi:10.2514/1.C031692.

[108] Lee H, Park S, Kim H. Estimation of Aircraft Structural Fatigue Life Using the Crack Severity Index Methodology. J Aircr 2010;47:1672–8. doi:10.2514/1.C000250.

[109] Jardine AKS, Lin D, Banjevic D. A review on machinery diagnostics and prognostics implementing condition-based maintenance. Mech Syst Signal Process 2006;20:1483–510. doi:10.1016/j.ymssp.2005.09.012.

[110] Bartram G, Mahadevan S. Integration of heterogeneous information in SHM models. Struct Control Heal Monit 2014;21:403–22. doi:10.1002/stc.1572.

[111] Julier S, Uhlmann J. A new extension of the Kalman filter to nonlinear systems. Proc. 11th Int. Symp. Aerospace/Defense Sensing, Simul. Control., vol. 3, 1997.

[112] Arulampalam MS, Maskell S, Gordon N, Clapp T. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. IEEE Trans Signal Process 2002;50:174–88. doi:10.1109/78.978374.

[113] Mihaylova L, Boel R, Hegyi A. Freeway traffic estimation within particle filtering framework. Automatica 2007;43:290–300. doi:10.1016/j.automatica.2006.08.023.

[114] Roychoudhury I, Biswas G, Koutsoukos X. Comprehensive diagnosis of continuous systems using dynamic bayes nets. Proc. 19th Int. Work. Princ. Diagnosis, 2008, p. 151–8.

[115] Chen H, Chang KC. K-nearest neighbor particle filters for dynamic hybrid Bayesian

networks. IEEE Trans Aerosp Electron Syst 2008;44:1091–101. doi:10.1109/TAES.2008.4655366.

[116] Pitt M, Shephard N. Filtering via simulation: Auxiliary particle filters. J Am Stat Assoc 1999;94:590–9.

[117] Musso C, Oudjane N, Le Gland F. Improving regularised particle filters. Seq. Monte Carlo methods Pract., Springer; 2001, p. 247–71.

[118] Doucet A, De Freitas N, Murphy K, Russell S. Rao-Blackwellised particle filtering for dynamic Bayesian networks. Proc. Sixt. Conf. Uncertain. Artif. Intell., 2000, p. 176–83.

[119] Paris PC, Erdogan F. A critical analysis of crack propagation laws. J Fluids Eng 1963;85:528–33.

[120] Donahue RJ, Clark HM, Atanmo P, Kumble R, McEvily AJ. Crack opening displacement and the rate of fatigue crack growth. Int J Fract Mech 1972;8:209–19. doi:10.1007/BF00703882.

[121] Yuen B, Taheri F. Proposed modifications to the Wheeler retardation model for multiple overloading fatigue life prediction. Int J Fatigue 2006;28:1803–19. doi:10.1016/j.ijfatigue.2005.12.007.

[122] Jesus AMP de, Pereira RMG. FEM Analysis of Riveted Connections aiming Fatigue and Fracture Assessments. Proc. Iber. Conf. Fract. Struct. Integr., 2010.

[123] Liu Y, Mahadevan S. Threshold stress intensity factor and crack growth rate prediction under mixed-mode loading. Eng Fract Mech 2007;74:332–45. doi:10.1016/j.engfracmech.2006.06.003.

[124] Staszewski W, Boller C, Tomlinson GR. Health monitoring of aerospace structures: smart

sensor technologies and signal processing. John Wiley &amp; Sons; 2004.

[125] ASTM E1049-85, Standard Practices for Cycle Counting in Fatigue Analysis 2005.

[126] Sudret B. Global sensitivity analysis using polynomial chaos expansions. Reliab Eng Syst Saf 2008;93:964–79. doi:10.1016/j.ress.2007.04.002.

[127] Friedman JH. Multivariate Adaptive Regression Splines. Ann Stat 1991;19:1–67.

[128] Mara T, Joseph O. Comparison of some efficient methods to evaluate the main effect of computer model factors. J Stat Comput Simul 2008;78:167–78. doi:10.1080/10629360600964454.

[129] Tissot J, Prieur C. A randomized Orthogonal Array-based procedure for the estimation of first- and second-order Sobol ' indices. J Stat Comput Simul 2014:1–24.

[130] Li C, Mahadevan S. Role of calibration, validation, and relevance in multi-level uncertainty integration. Reliab Eng Syst Saf 2016;148:32–43. doi:10.1016/j.ress.2015.11.013.

[131] Homma T, Saltelli A. Importance measures in global sensitivity analysis of nonlinear models. Reliab Eng Syst Saf 1996;52:1–17. doi:10.1016/0951-8320(96)00002-6.

[132] Sobol' IM, Tarantola S, Gatelli D, Kucherenko SS, Mauntz W. Estimating the approximation error when fixing unessential factors in global sensitivity analysis. Reliab Eng Syst Saf 2007;92:957–60. doi:10.1016/j.ress.2006.07.001.

[133] Sobol' IM, Myshetskaya EE. Monte Carlo estimators for small sensitivity indices. Monte Carlo Methods Appl 2008;13:455–65.

[134] Owen A. Better Estimation of Small Sobol Sensitivity Indices. ACM Trans Model Comput Simul 2013;23:11. doi:10.1145/2457459.2457460.

[135] Wu Q-L, Cournède P-H, Mathieu A. An efficient computational method for global

sensitivity analysis and its application to tree growth modelling. Reliab Eng Syst Saf 2012;107:35–43. doi:10.1016/j.ress.2011.07.001.

[136] Janon A, Klein T, Lagnoux AA, Nodet M, Prieur C. Asymptotic normality and efficiency of two Sobol index estimators. ESAIM Probab Stat 2014;18:342–64.

[137] Weirs VG, Kamm JR, Swiler LP, Tarantola S, Ratto M, Adams BM, et al. Sensitivity analysis techniques applied to a system of hyperbolic conservation laws. Reliab Eng Syst Saf 2012;107:157–70. doi:10.1016/j.ress.2011.12.008.

[138] Saltelli A, Tarantola S, Chan K. A quantitative model-independent method for global sensitivity analysis of model output. Technometrics 1999;41:39–56. doi:10.1080/00401706.1999.10485594.

[139] Tarantola S, Koda M. Random balance designs for the estimation of first order global sensitivity indices. Procedia - Soc Behav Sci 2006;2:7753–4. doi:10.1016/j.sbspro.2010.05.212.

[140] Satterthwaite FE. Random Balance Experimentation. Technometrics 1959.

[141] Cukier RI, Levine HB, Shuler KE. Nonlinear sensitivity analysis of multiparameter model systems. J Comput Phys 1978;26:1–42. doi:10.1016/0021-9991(78)90097-9.

[142] Cukier RI. Study of the sensitivity of coupled reaction systems to uncertainties in rate coefficients. I Theory. J Chem Phys 1973;59:3873. doi:10.1063/1.1680571.

[143] Koda M, Mcrae GJ, Seinfeld JH. Automatic sensitivity analysis of kinetic mechanisms. Int J Chem Kinet 1979;11:427–44. doi:10.1002/kin.550110408.

[144] Hu Z, Mahadevan S. Global sensitivity analysis-enhanced surrogate (GSAS) modeling for reliability analysis. Struct Multidiscip Optim 2015. doi:10.1007/s00158-015-1347-4.

[145] Ginot V, Gaba S, Beaudouin R, Aries F, Monod H. Combined use of local and ANOVA-based global sensitivity analyses for the investigation of a stochastic dynamic model: Application to the case study of an individual-based model of a fish population. Ecol Modell 2006;193:479–91. doi:10.1016/j.ecolmodel.2005.08.025.

[146] Archer G, Saltelli A, Sobol' IM. Sensitivity measures, ANOVA-like techniques and the use of bootstrap. J Stat Comput Simul 1997;58:99–120.

[147] Weiss NA, Holmes PT, Hardy M. A course in probability. Pearson Addison Wesley; 2006.

[148] Keisler HJ. Elementary calculus: An infinitesimal approach. Courier Corporation; 2012.

[149] Timoshenko SP, Goodier JN. Theory of Elasticity. 3rd ed. New York: McGraw-Hill; 1970. doi:10.1115/1.3408648.

[150] Marrel A, Iooss B, Laurent B, Roustant O. Calculations of Sobol indices for the Gaussian process metamodel. Reliab Eng Syst Saf 2009;94:742–51. doi:10.1016/j.ress.2008.07.008.

[151] Shachter RD. Evaluating influence diagrams. Oper Res 1986;34:871–82.

[152] Pan F, Zhu P, Chen W, Li C. Application of conservative surrogate to reliability based vehicle design for crashworthiness. J Shanghai Jiaotong Univ 2013;18:159–65. doi:10.1007/s12204-012-1240-x.

[153] Wan E a. a, Van Der Merwe R. The unscented Kalman filter for nonlinear estimation. Adapt. Syst. Signal Process. Commun. Control Symp. 2000. AS-SPCC. IEEE 2000, 2000, p. 153–8. doi:10.1109/ASSPCC.2000.882463.

[154] Liu X, Niranjan M. State and parameter estimation of the heat shock response system using Kalman and particle filters. Bioinformatics 2012;28:1501–7. doi:10.1093/bioinformatics/bts161.

[155] Julier SJ. The scaled unscented transformation. Proc. 2002 Am. Control Conf. (IEEE Cat. No.CH37301), vol. 6, IEEE; 2002, p. 4555–9 vol.6. doi:10.1109/ACC.2002.1025369.

[156] Merwe R Van Der. Sigma-Point Kalman Filters for Probabilitic Inference in Dynamic State-Space Models. Tech. reportn Proc. Work. Adv. Mach. Learn., 2003.

[157] Koller D. Object-Oriented Bayesian Networks. Conf. Uncertain. Artif. Intell., Providence, Rhode Island: 1997, p. 302–13.