Uncertainty Aggregation in System Performance Assessment

By

Saideep Nannapaneni

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

September 30, 2017

Nashville, Tennessee

Approved:

Sankaran Mahadevan, PhD

Abhishek Dubey, PhD

Douglas E. Adams, PhD

Hiba Baroud, PhD

Sudarsan Rachuri, PhD

To my Parents and Family

ACKNOWLEDGEMENTS

This dissertation work would not have been possible if not for the guidance, encouragement and support of several people and agencies. I start with my primary doctoral advisor, Prof. Sankaran Mahadevan; his tremendous wealth of knowledge and mentoring have transformed me from being a fresh undergraduate student to an academic professional, equipping me with the necessary skills to approach my professional career. I express my gratitude to my secondary doctoral advisor, Dr. Abhishek Dubey for training me in the area of Cyber-Physical systems, in which I had no prior knowledge. I sincerely appreciate them for always taking time from their always-busy schedules to interact with me whenever I had issues with my research work. I acknowledge other members of my dissertation committee – Prof. Douglas Adams, Dr. Hiba Baroud, and Dr. Sudarsan Rachuri for their constructive criticism and suggestions that helped improve my doctoral work. I also extend my appreciation to Prof. Prodyot K. Basu and Dr. Ravindra Duddu for the productive discussions I had with them, which provided me with guidance and advice regarding my doctoral training and professional life.

I am grateful for the research collaboration I had with several personnel in the Department of Civil & Environmental Engineering at Vanderbilt University: Dr. Zhen Hu, Dr. Chenzhao Li, Dr. Chen Liang, Dr. You Ling, and Dr. Guowei Cai. I express my appreciation to Dr. Ronay Ak, Dr. Anantha Narayanan, Dr. David Lechevalier and Dr. Tina Lee at the National Institute of Standards and Technology with whom I had fruitful academic interactions regarding the use of Bayesian networks for performance evaluation in manufacturing systems. I would like to thank Dr. Youngwon Shin and Dr. Rumi Ghosh for mentoring me during the summer internships I had with General Electric Global Research Center and Robert Bosch Research and Technology Center respectively. These internships have helped me obtain industrial perspectives of my doctoral research work, particularly the use of Bayesian networks for knowledge discovery in manufacturing systems and Dynamic Bayesian networks for diagnosis, prognosis and health

management in aerospace systems. I extend my appreciation to faculty with the Department of Civil Engineering at my alma mater, the Indian Institute of Technology Madras, for providing me with the necessary academic foundation, training and support to pursue doctoral-level education.

The financial support for the research presented in this dissertation was from (1) National Institute of Standards and Technology (Cooperative Agreements No. 70NANB14H036 and 70NANB16H297), (2) Air Force Office of Scientific Research (Grant No. FA9550-15-1-0018, Technical Monitor: Dr. Ben Smarslok), (3) Siemens Corporate Technology, and (4) Teaching Assistantship at Vanderbilt University. The support is deeply appreciated.

In addition, I thank by colleagues - Dr. Abhinav Subramanian, Dao Ao, Dr. Thushara Gunda, Ghina Nakad Absi, Eric Vanderhorn, Xiaoge Zhang, Paromita Nath, Dr. Erin DeCarlo, Yanqing Bao, Kyle Neal, Dr. Pranay Seshadri, Sarah Miele, Dr. Josh Mullins, Dr. Shuhai Zhang, Dr. Xiang Zhang, Anup Aryal, Thushara De Silva Manikkuwahandi, Dr. Chih-Chao Yang, Dr. Wenbiao Chen, Dr. Yi Ern Cheah and many others, in no particular order, for their interactions and friendship, without which this doctoral work would not have been possible. Finally, I express my appreciation for the several programs and activities organized at the Vanderbilt Recreation and Wellness Center, especially the Badminton and Yoga sessions, for keeping me sane and calm during the struggling times I came across during my doctoral training.

TABLE OF CONTENTS

Page

Chapter

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

1.1 Overview

Probabilistic performance assessment evaluates a system’s capability to accomplish the required functions under uncertainty. The results of probabilistic performance assessment can be used to support decision making such as data collection, design optimization and operational risk management. There are two different approaches for probabilistic performance assessment – (1) physics model-based, and (2) test-based or data-driven. In the physics model-based approach, models are constructed based on first-principles to explain the real-world phenomena, which are used for performance assessment. In the data-driven approach, the performance of the system is estimated using experimental or operational data. Performance assessment is affected by different types of aleatory (natural variability) and epistemic (lack of knowledge) uncertainty sources. Lack of sufficient data or knowledge causes epistemic uncertainty, both in model inputs (statistical uncertainty) and models (model uncertainty). Therefore, methods for the systematic incorporation of various sources of epistemic uncertainty in performance assessment need to be investigated.

Systems may consist of a single component or may be composed of multiple components (subsystems). A conceptual representation of a single-component system (M) along with the inputs ($\boldsymbol{X}$) and outputs ($\boldsymbol{Y}$) is provided in Fig. 1.1. When multiple components are composed to form a complex system, interactions arise between them, which need to be quantified and included in performance assessment. In addition, each component is associated with different sources of aleatory and epistemic uncertainty; therefore, a framework is needed for dependence learning, uncertainty aggregation, and performance assessment.

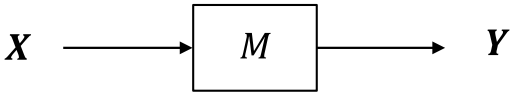


Fig. 1.1. Conceptual representation of a single-component system along with the inputs and outputs

Bayesian networks have shown much effectiveness for uncertainty aggregation across complex networks in many application domains, such as information retrieval, data fusion and engineering decision-making [1], computational biology and bioinformatics [2], epidemiology [3], and civil infrastructure networks [4]. BNs can be constructed either using physics-models or data or their combination. Current implementations of learning in hybrid Bayesian networks involve an assumption of conditional linear Gaussian distributions [5] or discretization of continuous variables into discrete variables. Several discretization strategies have been developed such as uniform distributions [6], Gaussian mixture model (GMM) [7] and truncated exponentials [8]. The discretization process is arbitrary and dependent on the analyst or expert opinion. Therefore, techniques for learning hybrid Bayesian networks that eliminate the discretization of continuous variables need to be investigated.

Systems can often be decomposed into multiple hierarchical levels. An illustration of a hierarchical system in provided in Fig. 1.2. For example, in a production network, a factory can be divided into several lines, where each line undertakes a fraction of the overall manufacturing process. Each line may consist of several unit processes, where each unit process accomplishes a specific task. In some cases, a unit process can also be further divided into several lower level unit processes. Each unit process can be associated with a single machine or a machine assembly. Hierarchical Bayesian networks (HBNs) [9], which are extensions of Bayesian networks, can be used to model hierarchical systems, and estimate the uncertainty in the overall performance assessment. Segmented learning can decompose the learning effort into several smaller learning efforts, thus making the computation affordable; however, interactions between different segments need to be carefully learned. As the network becomes larger, dimension reduction techniques are also necessary to eliminate unnecessary computational complexity.

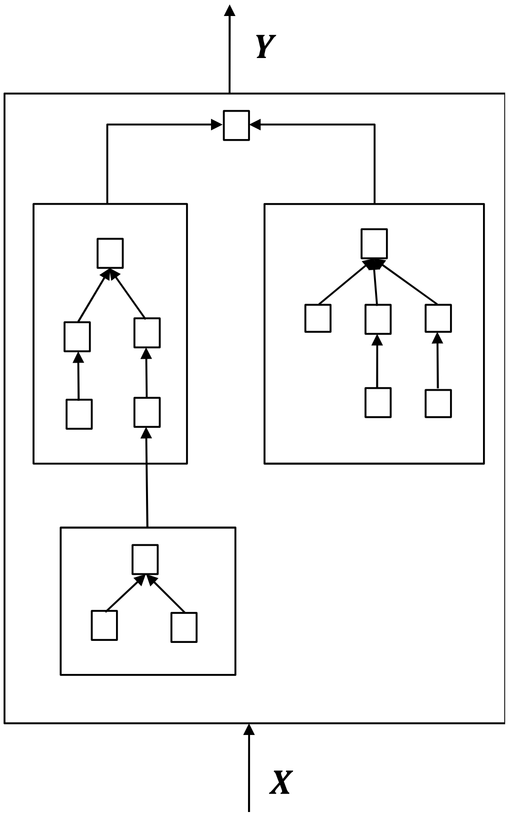


Fig. 1.2 Conceptual representation of a hierarchical system

The interactions between components or between subsystems may be one-way or two-way, and may be time-dependent or time-dependent; such features may or may not be revealed depending on the level of resolution at which the system is analyzed. So far, we considered systems with one-directional and time-independent interactions between several subsystems. The interactions are assumed time-independent due to the assumed low resolution in time. In some systems, the different subsystems may operate with a time lag (e.g., multiple processes in a manufacturing network).

In some systems with multiple subsystems, there exists two-directional interactions between them; these interactions can occur with a time lag or occur simultaneously. In addition, the individual subsystems can be computational or physical in nature. An example of a system with interactions between subsystems with a time lag, with both computational and physical subsystems is a cyber-physical system (CPS), which is a control system, where the computational subsystem (software) controls a physical subsystem. Examples of systems where interactions occur simultaneously include multi-physics systems such as an aircraft wing where fluid and elastic analyses occur simultaneously, or a multi-component system with the same physics such as a sandwich composite beam (shown in Fig. 1.3), i.e., a beam with three layers in which the material of the middle layer is different from the material used in the top and bottom layers. For the given load, there exists interactions between the different material media in order to converge to same displacement response due to the compatibility requirements at the material interface. In this dissertation, we consider performance assessment of both cyber-physical (interactions with a time lag) and multi-physics (simultaneous interactions) systems under uncertainty.

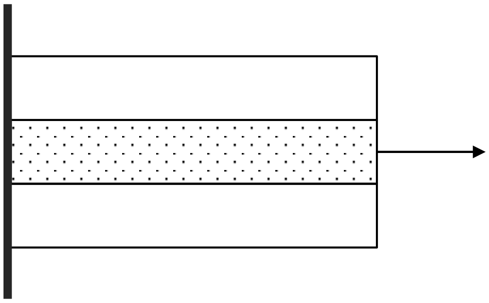


Fig. 1.3. A cantilever sandwich composite beam subjected to a tensile load

In a CPS, a software subsystem controls a physical subsystem (also referred to as plant) by collecting sensor data regarding the system parameters and implementing the necessary actuation. The interactions in a generic CPS occurs at two levels: (1) interactions between individual subsystems (plant, cyber, actuation, sensors), and (2) interactions between computing nodes, if a distributed computational subsystem is considered. Fig. 1.4 shows the interactions between several subsystems and between computational nodes (C_1, C_2 and C_3). The different types of interactions between computational nodes are discussed in Section 5.3. Each of the subsystems is associated with different uncertainty sources. Physical and actuation subsystems may be associated with model uncertainty and uncertainty in the inputs; sensors may be associated with sensor measurement uncertainty and software subsystem may be associated with uncertainty in communication between several software nodes.

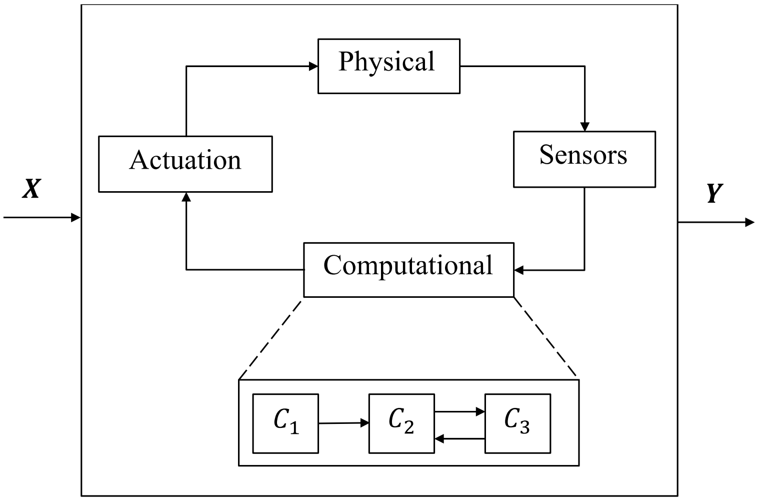


Fig. 1.4. Two-level interactions between several subsystems and computational nodes in a cyber-physical system

In the control system discussed above, the interactions among the subsystems occur with a time lag, i.e., each subsystem takes a finite amount of time to complete its task. As a result, the interactions are one-directional when time lag is considered. Therefore, uncertainty aggregation approaches to quantify the uncertainty in the QoI of a CPS are needed.

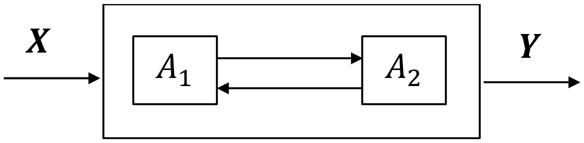


Fig. 1.5. Conceptual representation of a two-discipline multi-physics system

For analysis of multi-physics systems, where interactions between multiple subsystems occur simultaneously, models may be employed for each individual physics phenomenon and are run iteratively until the results from individual models are compatible. This results in a high computational expense as the individual models are run multiple times to reach compatibility. Fig. 1.5 presents a conceptual representation of two-discipline coupled system, where A_1 and A_2 represent individual disciplines. For example, consider the aero-elastic response of an aircraft wing: the deformation is affected by aerodynamic pressure, and the aerodynamic pressure is affected by the deformation. A computational fluid dynamics (CFD) model is employed for aerodynamics analysis, and a finite element (FE) model is employed for deformation analysis.

The computational effort further increases when uncertainty is considered, since the multidisciplinary analysis has to be repeated at multiple realizations of the uncertain variables. Current techniques combat the high computational expense in the performance assessment of multidisciplinary systems by the use of inexpensive surrogates to replace the individual disciplinary models [10] and by the use of decoupled analyses, where the bi-directional interaction between individual disciplines is approximated with a one-directional interaction, using first-order approximations [11]. Even though the surrogates that replace the individual disciplinary models are inexpensive, they need to be run multiple times at several realizations of the uncertain variables. In the case of decoupled approaches, the individual disciplinary models can be non-linear and in such cases, the first order approximations may not be accurate. Therefore, performance assessment techniques that overcome the repeated runs of the surrogates and the first order approximations of the decoupled approaches need to be investigated.

Based on the above discussion, the following research objectives are proposed to overcome the shortcomings identified above, for performance assessment under uncertainty in time-independent and time-varying coupled systems.

1.2 Research objectives

1. The first objective investigates a Bayesian framework for the inclusion of multiple sources of aleatory and epistemic uncertainty in model-based performance assessment. The developed techniques are demonstrated for two mechanical load-bearing systems.

2. The second objective investigates hierarchical Bayesian networks for uncertainty aggregation in multi-level systems for performance assessment. Techniques for hierarchical Bayesian network learning, efficient dimension reduction and model calibration are developed. These techniques are illustrated using an injection molding process.
3. The third objective investigates dynamic Bayesian networks for performance assessment in time-varying coupled systems by aggregating uncertainty from several subsystems that are performing tasks in a time sequence. These techniques are illustrated with a smart indoor heating system and a smart manufacturing process.
4. The fourth objective investigates Bayesian network approach for computationally efficient performance assessment in coupled multi-physics systems. The proposed techniques are demonstrated for performance assessment of an aircraft wing.

1.3 Organization of the dissertation

The remaining chapters of this dissertation are organized as follows to tackle the objectives identified in Section 1.2.

Chapter 2 provides a brief introduction and review of the several concepts, which are later used to develop the solution methodologies to meet the objectives. The concepts that are briefly discussed are: (1) Types of uncertainty sources (aleatory, statistical and model), (2) Reliability analysis for performance assessment under uncertainty (both component and system), (3) Different types of surrogate modeling techniques, (4) Variance-based global sensitivity analysis, (5) Auxiliary variable approach (to enable global sensitivity analysis under epistemic uncertainty), (6) Bayesian network and its variants (Hierarchical Bayesian network and Dynamic Bayesian network), and (7) Bayesian inference techniques in both static and dynamic Bayesian networks.

Chapter 3 develops a model-based performance assessment (reliability analysis) framework for single-disciplinary physical systems, considering both aleatory and epistemic (statistical and data) uncertainty sources. A surrogate model of the performance limit state is constructed and used for computationally efficient reliability analysis. The additional uncertainty introduced by the surrogate model and by Monte Carlo sampling is also incorporated in the framework.

Chapter 4 develops a Hierarchical Bayesian network methodology to enable uncertainty quantification in multi-level systems, and illustrates the methodology with an example of a manufacturing network. A new learning algorithm for Bayesian networks with both discrete and continuous variables is developed to overcome the discretization approaches discussed in Section 1.1. A segmented approach is developed for constructing a HBN and using it for Bayesian inference. In addition, an efficient dimension-reduction approach is developed based on global-sensitivity analysis.

Chapter 5 develops a model-based performance evaluation framework for a dynamic system with multiple subsystems operating in sequence, and illustrated with two examples of control systems (indoor thermal control, and a manufacturing process). A two-level dynamic Bayesian network is developed to aggregate the uncertainty arising from several subsystems. The higher level DBN considers the coupling across the subsystems whereas the lower-level DBN considers the interactions within the subsystem. The structure of the lower-level DBN depends on the architecture within the subsystem, namely synchronous or asynchronous; both types of architecture are considered, and their extension to complex architectures is detailed. The use of the proposed performance evaluation methodology for design-time and run-time decision-making is illustrated.

Chapter 6 discusses a Bayesian network-based approach for computationally efficient performance assessment of multi-physics system and systems with coupled subsystems undergoing the same physics. The developed Bayesian network method does not make any first-order assumptions made in the existing decoupled approaches and does not require repeated evaluations for uncertainty analysis, contrary to the existing surrogate-based approaches. Using the Bayesian network approach, the values of system QoI can be obtained simultaneously for several realizations of the uncertain variables at interdisciplinary compatibility. The extension of performance assessment framework to carry out design optimization is also illustrated.

Chapter 7 concludes the dissertation, with a summary of intellectual contributions made and possible extensions of the reported research.

CHAPTER 2

BACKGROUND CONCEPTS

2.1. Types of uncertainty

Uncertainty in variables has generally been classified into two types: aleatory (natural variability) and epistemic (lack of knowledge) [12]. Aleatory uncertainty represents the inherent randomness in a quantity; this uncertainty is irreducible and is typically represented in a probabilistic framework through a probability distribution. On the contrary, epistemic uncertainty refers to uncertainty regarding a quantity due to lack of knowledge. Epistemic uncertainty can be divided into two categories – statistical uncertainty and model uncertainty [13]. Statistical uncertainty stems from inadequacies in the available data (e.g., sparse, imprecise, qualitative, missing, or erroneous) which results in uncertainty regarding the probability distributions of the input random variables or the precise values of deterministic inputs. Model uncertainty is due to uncertainty in model parameters, numerical solution errors, and model form assumptions [14]. Numerical solution errors may include discretization error, round-off error, truncation error, surrogate model error and Monte Carlo sampling error [15]. The model form error and numerical solution errors can together be referred to as model discrepancy. In the rest of this section, we discuss the approaches for the representation of epistemic uncertainty.

2.1.1. Representation of statistical uncertainty

Several theories, both probabilistic and non-probabilistic, have been used to represent this type of epistemic uncertainty. Some of the approaches include interval analysis [16], convex models [17], fuzzy sets and possibility theory [18], evidence theory [19], Bayesian probability theory [20] and imprecise probabilities [21]. This dissertation uses a Bayesian probabilistic approach to model the epistemic uncertainty about the input random variables. A random variable can be represented either by using a parametric (e.g., normal) or a non-parametric distribution. In this work, we adopt the parametric representation of a random variable. A parametric distribution is associated with a distribution type and distribution parameters. If the distribution type of an input variable X is known but the distribution parameters are uncertain, then X can be represented by a distributional p-box. If the distribution type is also uncertain, then X may be represented by a free p-box [22].

Distribution parameter uncertainty: In the presence of sparse point data on X, two approaches may be used to construct the probability distributions of distribution parameters Θ (using a Bayesian perspective). The first approach is to use resampling methods such as Jack-knife and Bootstrap [23] to generate multiple values of Θ that are used to construct their distributions; the second approach is to use a likelihood-based representation of the available data to construct distributions of Θ using Bayes' theorem [24]. The likelihood-based approach can be extended to accommodate interval data and to construct parametric as well as non-parametric distributions [25].

Let a dataset D for a variable X consist of n point data p_i $(i = 1\ to\ n)$ and m interval data $[a_j, b_j](j = 1\ to\ m)$. The likelihood function for the distribution parameters Θ can be constructed as

$$L(\theta) = \prod_{i=1}^{n} f_X(x = p_i|\theta) \prod_{j=1}^{m} \left[F_X\left(x = b_j\middle|\theta\right) - F_X(x = a_j|\theta)\right] \tag{2.1}$$

where $f_X(x)$ and $F_X(x)$ represent the PDF and CDF of variable X respectively. After constructing the likelihood function, the distributions of the distribution parameters are obtained using Bayes' theorem as

$$f''_\Theta(\theta) = \frac{L(\theta)f'_\Theta(\theta)}{\int_{\Omega^\Theta} L(\theta)f'_\Theta(\theta)d\theta} \tag{2.2}$$

where $f_\Theta{}'(\theta)$ and $f_\Theta{}''(\theta)$ refer to the joint prior and posterior distributions of the distribution parameters. Ω^Θ refers to the domain of the Θ. Sometimes, point data and/or interval data may be directly available on the distribution parameters (e.g., in the reliability estimation of mechanical components, point and interval data on failure rates may be available from several sources). In this case, a non-parametric distribution can be constructed to express the uncertainty in the parameters. A detailed procedure for the construction of a non-parametric distribution is described in [25]. Only important steps are presented here for the sake of completeness.

Let the data available on Θ consist of r point data, θ_p^i $(i = 1\ to\ r)$ and s interval data $\left[\theta_a^j, \theta_b^j\right]$ $(j = 1\ to\ s)$. From the available data, the range of Θ is obtained by observing the maximum and minimum values. The domain is then discretized into Q points, given by $\theta_1, \theta_2, \dots \theta_Q$ and the PDF values at these discretized points are represented by $\alpha_1, \alpha_2, \dots \alpha_Q$. Using$(\theta_k, \alpha_k)(k = 1\ to\ Q)$, the probability density function can be constructed through an interpolation technique (e.g., linear, spline-based or Gaussian-process interpolation) over these Q points. The likelihood function, in this case, can be defined as

$$L(\alpha) = \prod_{i=1}^{r} f_\Theta(\theta = \theta_p^i|\alpha) \prod_{j=1}^{s}\left[F_\Theta\left(\theta = \theta_b^i|\alpha\right) - F_\Theta(\theta = \theta_a^i|\alpha)\right] \tag{2.3}$$

The values of $\alpha_k(k = 1\ to\ Q)$ can be obtained by maximizing the likelihood function, subject to the following constraints.

(1) $\alpha_k \geq 0$

(2) $f_\Theta(\theta) \geq 0$ for all values of θ

(3) $\int_{\Omega^\Theta} f_\Theta(\theta) = 1$

The likelihood approach helps to construct either parametric or non-parametric distributions of a variable or its distribution parameters (Eqs. 2.1 and 2.3) and either of these options can be used depending on the available data and analysis requirements.

Distribution type uncertainty: The first step in the probabilistic representation of an uncertain variable is the selection of a distribution type, if a parametric approach is used. In the presence of sparse data, when the distribution type of a variable is not known precisely, a set of candidate distribution types can be obtained through a combination of prior knowledge and results from statistical goodness-of-fit tests. With the list of candidate distribution types, one could either construct a composite distribution, i.e., a weighted average of all the candidate distribution types using Bayesian model averaging (BMA) [26], or select a single distribution type that best explains the observations. The weights for averaging or selection can be computed using Bayesian hypothesis testing (BHT) [27] by comparing the likelihoods of possible distribution types. Given a dataset D, the ratio of the posterior probabilities of two distribution types $f_X(x|\theta_c)$ and $f_X(x|\theta_d)$, is calculated as

$$\frac{\Pr(f_X(x|\theta_c)|D)}{\Pr(f_X(x|\theta_d)|D)} = \frac{\Pr(D|f_X(x|\theta_c))}{\Pr(D|f_X(x|\theta_d))}\,\frac{\Pr(f_X(x|\theta_c))}{\Pr(f_X(x|\theta_d))} \tag{2.4}$$

where $\Pr(f_X(x|\theta_c))$ and $\Pr(f_X(x|\theta_c))$ refer to the prior probabilities of the two distribution types and $\frac{\mathbf{Pr}(\boldsymbol{D}|\boldsymbol{f_X}(\boldsymbol{x}|\boldsymbol{\theta_c}))}{\mathbf{Pr}(\boldsymbol{D}|\boldsymbol{f_X}(\boldsymbol{x}|\boldsymbol{\theta_d}))}$ refers to the ratio of likelihoods (also referred to as Bayes factor B). The Bayes factor is a quantitative measure of extent of data support for $f_X(x|\theta_c)$ relative to support for $f_X(x|\theta_d)$. If $B > 1$, the dataset D supports $f_X(x|\theta_c)$ over $f_X(x|\theta_d)$. When BMA is used, a composite distribution can be constructed as

$$f_X(x|\theta) = \sum_{k=1}^{N} w_k\, f_X(x|\theta_k) \tag{2.5}$$

where $f_X(x|\theta)$ represents the composite distribution, $f_X(x|\theta_k)$ $(k = 1\ to\ N)$ refers to each of the N possible distribution types with $w_k, \boldsymbol{\theta}_k$ representing their weights and distribution parameters. Note that the weights are proportional to the posterior probabilities calculated using Eq. (2.4). In general, adequate data may not be always available to characterize the correlations between the input variables. Similar to the distribution parameters, parametric or non-parametric distributions may be used to represent the uncertainty regarding correlation coefficients. In the case of a parametric approach, bounded distributions such as a beta distribution may be used because correlation coefficients lie between -1 and 1. In a non-parametric approach, Eq. (2.3) may be used to construct a non-parametric PDF to represent the uncertainty regarding the correlation coefficients.

2.1.2. Representation of model uncertainty

As discussed at the beginning of this section, there are three sources of model uncertainty – model parameter uncertainty, solution approximation errors and model form errors. The model error $\boldsymbol{\epsilon_m}$ is composed of solution approximation errors and model form error. Methods for the quantification of model parameter uncertainty, model form error, and solution approximation errors (discretization error, surrogate model bias error and Monte Carlo sampling error) are discussed below.

Model parameter uncertainty represents the uncertainty in the model parameters due to either natural variability or limited data or both. The three possible scenarios of model parameter uncertainty are – (1) model parameter is deterministic but unknown (epistemic uncertainty), (2) model parameter is stochastic with known distribution parameters (aleatory uncertainty), and (3) model parameter is stochastic with unknown distribution parameters (aleatory and epistemic uncertainty). If a model parameter is deterministic but unknown, it can be estimated using available data using least squares, maximum likelihood or Bayesian calibration.

Model parameters that are associated with aleatory uncertainty (probability distributions) and with fixed distribution parameters, can be treated similar to input variables for reliability analysis and the techniques used for the quantification of uncertainty in the inputs (parametric and non-parametric approaches, described in Section 2.1.1) can also be used for model parameters. If the distribution parameters of model parameters are unknown (both aleatory and epistemic uncertainty), then one of the three aforementioned calibration techniques can be used to estimate the distribution parameters using available data. Among these, Bayesian calibration can explicitly quantify the uncertainty in the model parameters or their distribution parameters.

Model discrepancy, in this discussion, represents the combined error introduced due to the assumptions and simplifications made in building a model (model form error) as well as the errors that arise in the methodology adopted in solving the model equations (numerical solution errors). Different types of numerical solution errors exist such as discretization error, round-off error, and truncation error. Suppose $g_{obs}(X), g_{model}(X)$ and $\delta(X)$ represent the observations, simulation model prediction and model discrepancy respectively. For a given $X = x$, the three quantities are related as $g_{obs}(x) = g_{model}(x) + \delta(x) + \epsilon_{obs}(x)$. Here, $\epsilon_{obs}(x)$ refers to the observation (or experimental) error, which may or may not be dependent on the input. In the model calibration framework developed by Kennedy and O'Hagan [28], the quantification of the model discrepancy is performed together with the calibration of model parameters. Kennedy and O'Hagan proposed two approaches for model calibration – (1) fully Bayesian, and (2) modular Bayesian. A good overview of both fully and marginal Bayesian approaches is available in [29]. In the fully Bayesian approach, the model parameters of simulation model and the hyper parameters are calibrated together by obtaining joint posterior distributions. Using the joint posterior distributions, marginal distributions can be obtained.

The model calibration using the modular Bayesian approach can be summarized in the following steps [29]: (1) Estimation of MLE (maximum likelihood estimates) of the hyper parameters of the simulation model GP surrogate using the simulation data, (2) Estimation of MLE of hyper parameters of the model discrepancy GP surrogate using the experimental data, simulation data, and hyper parameters of the simulation model GP surrogate, (3) Calibration of model parameters based on the estimated hyper parameters of the GP surrogates, and (4) Prediction, where the overall prediction is marginalized over the model parameters. The model prediction at any input, when conditioned on the model parameters and MLE of hyper parameters of GP surrogates, follows a Gaussian distribution. However, the unconditional prediction at any input is obtained by marginalizing over the posterior distributions of the model parameters; in this case, the model prediction is not Gaussian. When the overall prediction is marginalized over the model parameters, an expected value for the reliability estimate is obtained. The above marginalization results in the overall prediction to not being Gaussian. However, since the goal in this work is to quantify the uncertainty in the reliability estimate (as opposed to an expected value), marginalization over model parameters is not performed and the model parameters (with their posterior distributions) are treated just like any stochastic inputs. In this case, the prediction (for a given realization of system input and model parameter) will be Gaussian. As stated in [29], the separation of both the GP models for calibration is intuitive and therefore, the modular Bayesian approach for calibration is adopted.

The outputs of the model calibration analysis using the KOH framework are – (1) data on the inputs, simulation output and observations, which are used for model calibration, (2) a GP model for the simulation model, (3) a GP model for the model discrepancy, and (4) Posterior distribution of the model parameters. The above four elements are later used in the construction of a limit-state surrogate (discussed in Section 2.3) for reliability analysis in Chapter 3.

2.2. Reliability analysis

Here, we briefly discuss the definitions for performance assessment, namely reliability analysis, with respect to a single performance criterion or limit state (referred to as component reliability analysis) and with respect to multiple criteria or limit states (referred to as system reliability analysis).

2.2.1. Component reliability analysis

Consider a single performance function of a system mathematically expressed as $g(\boldsymbol{X})$, where $\boldsymbol{X} = [X_1, X_2 \dots X_n]$ is a vector of input random variables, such that $g(\boldsymbol{X}) > 0$ refers to satisfactory performance, $g(\boldsymbol{X}) < 0$ refers to failure, and $g(\boldsymbol{X}) = 0$ is referred to as the limit state . The component failure probability ($\boldsymbol{p_f^c}$) is given as

$$p_f^c = \Pr(g(\boldsymbol{X}) \leq 0) \tag{2.6}$$

In surrogate-based reliability analysis using Monte Carlo sampling (MCS), the component failure probability can be calculated as

$$p_f^c = \frac{\sum_{j=1}^{n_{MCS}} I\left(\hat{g}\left(\boldsymbol{x}^{(j)}\right) \leq 0\right)}{n_{MCS}} \tag{2.7}$$

where $\boldsymbol{x}$ is a realization of random variables $\boldsymbol{X}$, n_{MCS} is the number of MCS samples, $\hat{g}(\boldsymbol{X})$ is the surrogate, $\hat{g}\left(\boldsymbol{x}^{(j)}\right)$ is the surrogate prediction at the *j*th sampling point $x^{(j)}$, and $I\left(\hat{g}\left(\boldsymbol{x}^{(j)}\right) \leq 0\right)$ is a failure indicator function defined as

$$I(\text{event}) = \begin{cases} 1, \text{ if an event is true} \\ 0, \text{ otherwise} \end{cases} \tag{2.8}$$

2.2.2 System reliability analysis

When system performance is assessed with respect to multiple criteria, system failure may occur through the union, intersection, or a combination of unions and intersections of component failures [30]. Let $g_i(\boldsymbol{X}), i = 1,2 \dots m$ be the individual limit-state functions. The failure probability of the union of individual failures (referred to as a series combination) is given by

$$p_f^s = \Pr\left(\bigcup_{i=1}^{m} g_i(\boldsymbol{X}) \leq 0\right) \tag{2.9}$$

where p_f^s is the system failure probability and $\Pr(.)$ is the probability operator. The system failure probability, using a surrogate and MCS, can be calculated as

$$p_f^s = \frac{\sum_{j=1}^{n_{MCS}} I\left(\bigcup_{i=1}^{m} \hat{g}_i\left(\boldsymbol{x}^{(j)}\right) \leq 0\right)}{n_{MCS}} \tag{2.10}$$

where $\hat{g}_i\left(\boldsymbol{x}^{(j)}\right)$ is the surrogate prediction of the ith limit-state function at the jth sampling point $x^{(j)}$. Eq. (2.10) can be re-written as

$$p_f^s = \frac{\sum_{j=1}^{n_{MCS}} I\left(\min\left(\hat{g}_i\left(\boldsymbol{x}^{(j)}\right)\right) \leq 0\right)}{n_{MCS}} = \frac{\sum_{j=1}^{n_{MCS}} I\left(\hat{g}_{min}\left(\boldsymbol{x}^{(j)}\right) \leq 0\right)}{n_{MCS}} \tag{2.11}$$

Similarly, the failure probability of the intersection of individual failures (referred to as a parallel combination) is given by

$$p_f^s = \Pr\left(\bigcap_{i=1}^{m} g_i(\boldsymbol{X}) \leq 0\right) \tag{2.12}$$

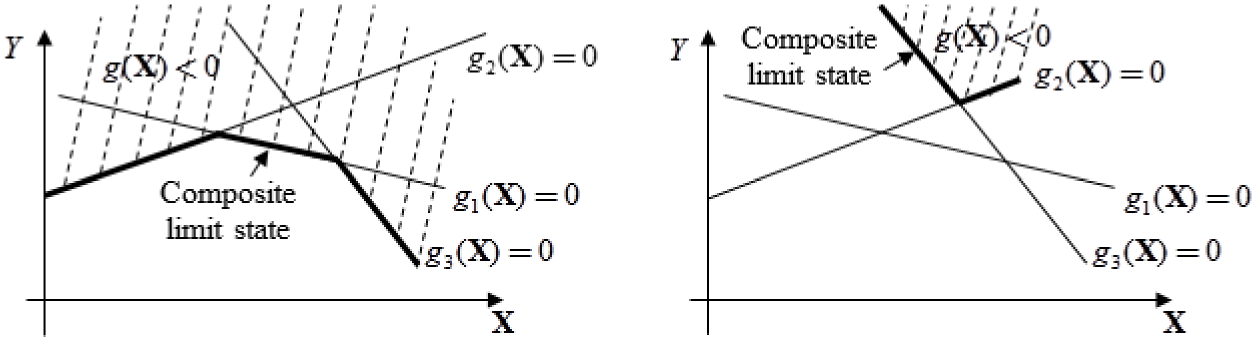
Similar to the series system, the failure probability of a parallel system can be calculated using MCS as

$$p_f^s = \frac{\sum_{j=1}^{n_{MCS}} I\left(\bigcap_{i=1}^{m} \hat{g}_i\left(\boldsymbol{x}^{(j)}\right) \leq 0\right)}{n_{MCS}} \tag{2.13}$$

The above equation can be written as

$$p_f^s = \frac{\sum_{j=1}^{n_{MCS}} I\left(\max\left(\hat{g}_i\left(\boldsymbol{x}^{(j)}\right)\right) \leq 0\right)}{n_{MCS}} = \frac{\sum_{j=1}^{n_{MCS}} I\left(\hat{g}_{max}\left(\boldsymbol{x}^{(j)}\right) \leq 0\right)}{n_{MCS}} \tag{2.14}$$

Fig. 2.1 provides graphical illustrations of composite limit states for series and parallel combinations of three individual limit states, $g_i(\boldsymbol{X}) = 0, i = 1,2,..m$. When a system failure is defined through a mixture of series and parallel combinations of component failures, the system reliability can be estimated by the following steps [31]: (1) decompose the combined system into a set of mutually exclusive series combinations, (2) compute the reliability of each series combination, and (3) compute the system reliability using the reliabilities of each of the mutually exclusive series combinations.



(a) (b)

Fig. 2.1. Composite limit state of three individual limit states

2.2.3. Reliability analysis errors

In addition to the uncertainty sources mentioned in Section 2.1, other sources such as the uncertainty in the surrogate model prediction and Monte Carlo sampling error exist during prediction analysis (such as uncertainty quantification and reliability analysis). For the sake of simplicity, these two errors are referred to as reliability analysis errors in this work.

Uncertainty in surrogate model prediction: In this work, the variance (and not bias) associated with the prediction of a surrogate is called surrogate uncertainty. For example, the calibration framework by Kennedy and O'Hagan (referred to as KOH) [28] constructs a Gaussian Process surrogate model to replace the expensive physics model. The prediction of a Gaussian process model is not a deterministic quantity but a Gaussian distribution, with parameters dependent on the input. The uncertainty in the surrogate model prediction (i.e., prediction variance) represents another uncertainty source to be considered for a comprehensive reliability analysis. Since a surrogate (GP model) is employed in this work for reliability analysis, the details about surrogate uncertainty and its impact on the reliability estimate are discussed later, in Section 3.2.

Monte Carlo Simulation (MCS) error: MCS error represents the error due to the use of limited number of Monte Carlo samples for uncertainty propagation. The MCS error is quantified as the difference between the empirical CDF (constructed using Monte Carlo samples after uncertainty propagation) and the true CDF of the output quantity of interest [32].

Consider the model $Y = g(\boldsymbol{X})$ and let N_s samples be generated from the input and propagated through the model to obtain the samples of the output Y. Let $\bar{F}_Y(y)$ represent the empirical CDF constructed from N_s samples and $F_Y(y)$ be the true CDF. If n_y samples are less than a value of $Y = y^*$, then n_y follows a binomial distribution $n_y{\sim}B(N_s, F_Y(y))$ considering the value of each sample as the result of a Bernoulli trial [31]. As N_s becomes larger, the binomial distribution can be approximated with a Gaussian distribution. A review of several empirical rules that have been proposed in the literature for approximating a binomial distribution with a Gaussian distribution is provided in [33]. The most commonly used rule, as stated in [33], is $N_s F_Y(y) > 5$ and $N_s(1 - F_Y(y)) > 5$. Let us consider this case. A large of Monte Carlo samples are generally used in surrogate-based methods since it is computationally inexpensive. Even if we use about 100,000 samples (which is a common number in surrogate-based methods), the threshold $F_Y(y)$ value is $\frac{5}{100000} = 0.00005$, which covers more than 99.99% of the domain.

Since the CDF value from the Monte Carlo output samples is given by $\bar{F}_Y(y) = \frac{n_y}{N_s}$, we have

$$\bar{F}_Y(y) \sim N(F_Y(y), \sqrt{\frac{(F_Y(y)\big(1 - F_Y(y)\big)}{N_s}} \tag{2.15}$$

Therefore, the MCS error associated with $\bar{F}_Y(y)$ can be expressed as a Gaussian random variable with mean and standard deviation as given in Eq. (2.15). Since the true CDF ($F_Y(y)$ is unknown, Eq. (2.15) cannot be used directly. However, confidence intervals for $F_Y(y)$ can be estimated given the empirical

CDF $\bar{F}_Y(y)$, the number of samples used N_s and the degree of accuracy γ as $\left[\bar{F}_Y(y) + \frac{z_{\frac{\gamma}{2}}^2}{2\times n_{MCS}} \pm \sqrt{\frac{p_f^s(j)\,(1-p_f^s(j))}{N_s} + \frac{z_{\frac{\gamma}{2}}^2}{4\times n_{MCS}^2}}\right] \times \frac{1}{1+\frac{z_{\frac{\gamma}{2}}^2}{n_{MCS}}}$, where $z_{\frac{\gamma}{2}}$ refers to the $1-\frac{\gamma}{2}$ quantile of the standard normal distribution [34]. It should be noted that the true CDF $F_Y(y)$ is a fixed quantity but unknown; therefore, it is an epistemic source of uncertainty and quantified using confidence intervals. For given values of $\bar{F}_Y(y)$ and N_s, we can estimate the percentile values of $F_Y(y)$ by varying the accuracy parameter γ. From the percentile values, the entire CDF can be numerically constructed which can then be used to obtain a PDF.

For illustration, let l, u represent the lower and upper bounds of confidence intervals corresponding to accuracy parameter γ. Therefore, $Pr(F_Y(y) < u) = 1-\frac{\gamma}{2}$ and $Pr(F_Y(y) < l) = \frac{\gamma}{2}$. Hence, the CDF values at $F_Y(y) = l, u$ are $\frac{\gamma}{2}$ and $1-\frac{\gamma}{2}$ respectively. Following the same procedure at multiple values of γ, the CDF values at the corresponding lower and upper bounds of the confidence intervals can be obtained which can be used to construct the CDF of $F_Y(y)$. After identifying and quantifying several uncertainty sources (statistical and model uncertainty, reliability analysis errors), their aggregated effect on the reliability estimate is detailed in Chapter 3.

2.3 Surrogate modeling

If the computational model for $g(\boldsymbol{X})$ is expensive, two categories of surrogates have been used in the reliability analysis literature – general-purpose surrogates that estimate the output for any given input i.e. $g(\boldsymbol{X})$, and limit-state surrogates that particularly model the failure limit state i.e. $g(\boldsymbol{X}) - g_o = 0$.

2.3.1 General purpose surrogate

The goal of a general-purpose surrogate is prediction of response at particular input values within the domain in which the surrogate is constructed. The first step in building a surrogate is the selection of the input variable ranges. After selecting the input ranges, training points are generated in such a way that they cover all regions of the input domain. Typically, several design of experiments (DoE) techniques [35] such as Latin hypercube sampling (LHS), full-factorial design and stratified sampling are used to generate training points for building a surrogate.

In this work, we use a Gaussian process surrogate for its flexibility to capture non-linear variations of the original function for which the surrogate is constructed. A Gaussian process (GP) model represents the function outputs at several inputs using a multivariate Gaussian distribution. Let $\boldsymbol{x_T}$ and y_T represent the training inputs and their corresponding outputs. Let $\boldsymbol{x_P}$ represent the new inputs at which their model prediction (y_P) are required. A Gaussian process is parameterized through a trend function and a covariance function, parameterized through a covariance kernel. For illustration, we define a Gaussian process below with one input and one output.

$$G(X) = h(X)^T\beta + P(X) \tag{2.16}$$

In the above equation, $G(X)$ is the approximation to the original model, $h(X)^T\beta$ is the trend function and $P(X)$ represents a zero-mean Gaussian process. The trend function is typically a parametric function such as a linear or a quadratic function; this captures the overall behavior of the original function whereas $P(X)$ captures the errors between the trend and the original function. The covariance between the outputs is represented through the assumed kernel function. Several types of kernels are available such as linear, squared-exponential and Matern kernels [36]. The squared-exponential kernel (given below) represents the most-widely used covariance kernel.

$$C(y_i, y_j) = \sigma^2 \times \exp\left(-\lambda\left(x_i - x_j\right)^2\right) \tag{2.17}$$

In Eq. (2.17), σ and λ are referred to as process variance and length-scale parameter; these parameters are estimated using the training points. $C(y_i, y_j)$ represents the covariance between outputs y_i and y_j at two inputs, x_i and x_j. Given the training points, the estimated parameters (σ, λ) and the new inputs, x_P, the corresponding model predictions y_P can be calculated as

$$\Pr(y_P | x_T, y_T, \lambda, \sigma) \sim N(m, S) \tag{2.18}$$

As the function outputs are represented through a multivariate Gaussian distribution, the conditional distribution over a subset of those outputs also represents a multivariate Gaussian distribution. In the above equation, m and S represent the mean and covariance functions of y_P; the expressions of them are given below as

$$m = \ h(x)^T\beta + K_{PT}K_{TT}^{-1}(y_T - h(x_P)^T\beta) \tag{2.19}$$

$$S = K_{PP} - K_{PT}K_{TT}^{-1}K_{TP} \tag{2.20}$$

In Eqs. (2.19) and (2.20), K_{TT} and K_{PP} represent the covariance matrices between the training and prediction points. K_{TP} represents the covariance matrix between the training and prediction points, and K_{PT} is the transpose of K_{TP}. More details regarding Gaussian Process models are available in [36]. After considering a general-purpose surrogate, we consider a limit-state surrogate below.

2.3.2. Limit state surrogate

The limit state surrogate is primarily useful for reliability analysis (prediction around the limit state) and not for general-purpose prediction in the input domain. For a general-purpose surrogate, the training points are selected using DoE techniques such that the entire input ranges are covered. However, a limit state surrogate requires training points along the failure limit state, as its goal is to accurately model the limit state. Several techniques have been developed in the literature such as EGRA [37], META-IS [38] and AK-MCS [39] that proposed several criteria for selecting training points. In this work, we follow the EGRA (Efficient Global Reliability Analysis) approach for selecting training points, as discussed below.

In EGRA, a Gaussian Process surrogate is constructed to approximate the limit state equation, $g_{ext}(\boldsymbol{X}) = g_o$, i.e., we are building a surrogate for the classifier or boundary between failure and safety. (Note that $g_{ext}(\boldsymbol{X}) = g(\boldsymbol{X})$ for a component, $g_{ext}(\boldsymbol{X}) = g_{min}(\boldsymbol{X})$ for a series system; and $g_{ext}(\boldsymbol{X}) = g_{max}(\boldsymbol{X})$ for a parallel system). Using the idea of the Expected Improvement (EI) [40], EGRA adaptively selects training points close to the limit state to accurately model $g_{ext}(\boldsymbol{X})$. The selection of training points is based on a learning function called the Expected Feasibility Function (EFF) defined as [37]

$$\begin{aligned} EFF(\boldsymbol{x}) = \left(\mu_g(\boldsymbol{x}) - e\right)\left[2\Phi\left(\frac{e - \mu_g(\boldsymbol{x})}{\sigma_g(x)}\right) - \Phi\left(\frac{e^L - \mu_g(\boldsymbol{x})}{\sigma_g(x)}\right) - \Phi\left(\frac{e^U - \mu_g(\boldsymbol{x})}{\sigma_g(x)}\right)\right] \\ - \sigma_g(\boldsymbol{x})\left[2\Phi\left(\frac{e - \mu_g(\boldsymbol{x})}{\sigma_g(\boldsymbol{x})}\right) - \Phi\left(\frac{e^L - \mu_g(\boldsymbol{x})}{\sigma_g(\boldsymbol{x})}\right) - \Phi\left(\frac{e^U - \mu_g(\boldsymbol{x})}{\sigma_g(\boldsymbol{x})}\right)\right] \\ - \left[\Phi\left(\frac{e^L - \mu_g(\boldsymbol{x})}{\sigma_g(\boldsymbol{x})}\right) - \Phi\left(\frac{e^U - \mu_g(\boldsymbol{x})}{\sigma_g(\boldsymbol{x})}\right)\right] \end{aligned} \tag{2.21}$$

where e is the failure threshold, $e^U = e + \epsilon$, $e^L = e - \epsilon$, $\mu_g(\boldsymbol{x})$, $\sigma_g(\boldsymbol{x})$ are the mean and standard deviation of the GP prediction at point $\boldsymbol{X} = \boldsymbol{x}$, ϵ is usually chosen as $2\sigma_g(\boldsymbol{x})$ [13], and Φ and ϕ are the CDF and PDF of a standard Gaussian variable, respectively. In EGRA, a new training point is identified by maximizing the EFF as $\boldsymbol{x}^* = Max(EFF(\boldsymbol{x}))$. More details about EGRA are available in [41]. EGRA and other similar methods that focus on limit state surrogates such as META-IS [38] and AK-MCS [39] have so far concentrated on reliability analysis with only aleatory uncertainty. In practical applications, several sources of epistemic uncertainty may be involved in the reliability analysis; therefore, Chapter 3 considers the inclusion of epistemic uncertainty in the limit state surrogate construction. In addition, the uncertainty in the reliability estimate computed using a limit state surrogate due to reliability analysis errors is detailed in Chapter 3.

2.4 Global sensitivity analysis

Sensitivity analysis, in general, estimates the influence of model inputs or model components to some feature of interest in the model output. Two types of sensitivity analysis are available – local and global [32] . The local sensitivity index of a quantity (input variable, model parameter or model error term) measures the variation of the model prediction when the variable is fixed at a single value whereas the global sensitivity index measures the variability of model prediction when the quantity is varied over its probability distribution. Variance has been commonly used as the measure of model output variability. Global sensitivity analysis (GSA) is used in this work as it considers the entire probability distribution of a variable in assessing the uncertainty contribution to the output.

Consider a model with $\boldsymbol{n}$ random input variables X_1 , $X_2, \ldots, X_n$ given by

$$Y = G(X_1, X_2, X_3, \ldots, X_n) \tag{2.22}$$

Variance-based GSA is based on the variance decomposition theorem, where the variance in an output variable is decomposed as

$$Var(Y) = E_{X_i}\left(Var_{X_{\sim i}}(Y|X_i)\right) + Var_{X_i}(E_{X_{\sim i}}(Y|X_i)) \tag{2.23}$$

where Y represents the output variable (model prediction), X_i is the variable for which the sensitivity measures are computed, and $X_{\sim i}$ represents all the other variables excluding X_i. Two indices are commonly computed for each variable – main effect, also called the first-order effect, and total effect. The first-order effect index (S_i^I) quantifies the individual contribution from the uncertainty in an input variable

X_i (without considering its interaction with other variables), to the overall uncertainty in the output variable and is given by

$$S_i^I = \frac{Var_{X_i}\left(E_{X_{\sim i}}(Y|X_i)\right)}{Var(Y)} \tag{2.24}$$

where $E_{X_{\sim i}}(Y|X_i)$ calculates the expected value of the model output Y when X_i is fixed at a specific value, and Var_{X_i} computes the variance of this expected value when the randomness in X_i is included. The contribution from the variable X_i including its interaction with all other variables is quantified by the total effects index, computed as

$$S_i^T = 1 - \frac{E_{X_{\sim i}}\left(Var_{X_i}(Y|X_{\sim i})\right)}{Var(Y)} \tag{2.25}$$

where $Var_{X_i}(Y|X_{\sim i})$ computes the variance of Y when all the input variables except for X_i are fixed at specific values, and $E_{X_{\sim i}}$ calculates the expected value of this variance considering the randomness in $X_{\sim i}$. Note that superscripts I and T on S_i in Eq. (2.24) and (2.25) are used to represent individual effects and total effects respectively.

The above described sensitivity analysis approach can also be extended to include model parameters, $\boldsymbol{\Theta_M}$. Consider a function $Y = G(\boldsymbol{X}, \boldsymbol{\Theta_M})$ where $G(.)$ represents a deterministic function connecting the inputs and model parameters to the output Y. In such cases, for the purpose of sensitivity analysis, the model parameters can also be treated in the same manner at the input variables $\boldsymbol{X}$ [42]. One of the assumptions made in the development of these indices (Eqs. 2.24 and 2.25) is the independence between several inputs and model parameters [43]. In the presence of correlated inputs (or model parameters), the first order index can still be used to measure the relative importance of several inputs (or model parameters) to the output; however the total effects index is no longer applicable [44].

Computation of sensitivity indices (Eqs 2.24 and 2.25) can be computationally expensive as it involves a double-loop Monte Carlo, where in the case of the first order index, $E_{X_{\sim i}}(Y|X_i)$ is calculated for k_1 realizations of X_{-i} and in the outer loop, $Var_{X_i}\left(E_{X_{\sim i}}(Y|X_i)\right)$ is computed for k_2 realizations of X_i. For computation of the total variance in $Y(Var(Y))$, i.e., the denominator in Eq. 2.24, assume k_3 realizations of the inputs (or model parameters) are considered. Therefore, the total number of model evaluations required can be computed as $C = nk_1k_2 + k_3$, where C represents the total number of model evaluations, n is the number of inputs (or model parameters). For sampling-based methods, the number of realizations that need to be considered is typically very high, in the order of 10^3 and may increase depending on the problem. For a ten-dimensional problem, the number of model evaluations can be in the order of 10^7.

Several algorithms have been developed in the literature to combat the high computational expense involved with the computation of sensitivity indices. Some of the techniques include Sobol's scheme [45], Fourier Amplitude Sensitivity Test (FAST) [46] and its variant, improved FAST [47], Importance sampling and kernel regression [48] and a modularized approach for sensitivity analysis [49]. In this dissertation, we use the modularized approach for computation of first order sensitivity indices, as it uses the input-output samples from the computation of $Var(Y)$ to estimate the first-order indices. The steps in the algorithm are given below.

1. The domain of an input variable (or model parameter) whose first order index is required, is discretized into M equally probable intervals.
2. The samples of the output Y corresponding to the each of the intervals are obtained.
3. Using the samples in each interval, the mean value of Y in each interval is calculated. The mean value corresponds to $E(Y|X_i)$

4. Using the mean values obtained in the previous step for several intervals, the variance of all the mean values is computed. This provides us with the numerator term $Var_{X_i}\left(E_{X_{\sim i}}(Y|X_i)\right)$ in Eq. 2.24.
5. First order index of X_i is thus equal to the ratio of output from Step 4 and the overall variance computed ahead of Step 1 (those samples are used to carry out Steps 1-4).

The above five steps are repeated to compute first-order sensitivity indices of all input variables (or model parameters). Therefore, the overall computational expense is equal to k_3 and does not vary with the number of input variables. The computational expense is significantly lower than the double-loop approach, which is equal to $nk_1k_2 + k_3$, from the discussion above.

As sensitivity analysis provides the relative contributions of several inputs to the variance in the system output, the inputs that do not have significant contributions can be assumed deterministic at their nominal values; this process can be used for dimension reduction. More details regarding the application of sensitivity analysis for dimension reduction are provided in Section 4.2.4.

2.5 Auxiliary variable approach

Section 2.4 discussed sensitivity analysis when the distributions of inputs (or model parameters) are known, i.e., the distribution parameters are known deterministically. There may arise some cases where the distribution parameters of some inputs are not precisely due to sparse, uncertain or imprecise data as discussed in Section 2.1.1. When the distribution parameters are unknown, then the distribution parameters are also represented using probability distributions in a Bayesian probability framework (Section 2.1.1). It is desirable to estimate the sensitivity indices of the distribution parameters to identify their contributions to the variance of the output. However, for a given value of the distribution parameter, the random variable X is represented by a probability density function (PDF). As X is represented by a

PDF, the system output is also represented by a PDF in the presence of a deterministic function between the inputs and the output. Therefore, the relationship between the distribution parameters and the model output is not deterministic but stochastic in nature. In such scenarios where stochastic relationship exist, the auxiliary variable approach [50] can be used to devise a deterministic relationship between the distribution parameters and the model output.

Let X represent a random variable with $\mathbf{\Theta}_X$ as its distribution parameters. When the values of $\mathbf{\Theta}_X$ are known, then the random variable X can be represented using a single probability distribution. However, if $\mathbf{\Theta}_X$ is uncertain, then X has a different probability distribution for each realization of the distribution parameters ($\mathbf{\Theta}_X = \mathbf{\Theta}_X^*$). When multiple realizations of $\mathbf{\Theta}_X$ are considered, then X is represented using a family of distributions, as shown in Fig. 2.2.

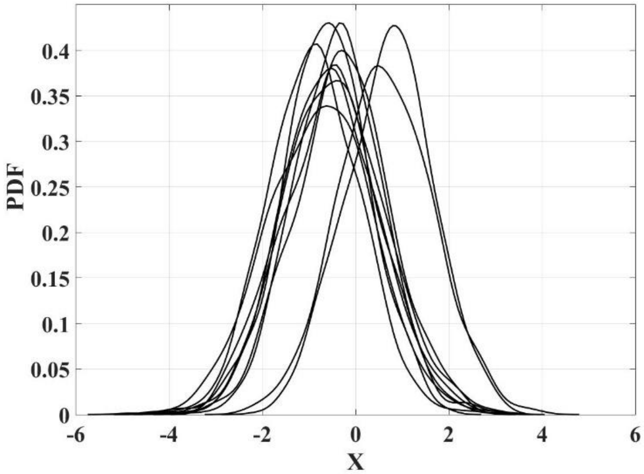


Fig. 2.2. Illustration of a family of PDFs due to parameter uncertainty

If $f(.)$ represents a PDF (probability density function), then $f(X|\mathbf{\Theta}_X = \mathbf{\Theta}_X^*)$ is a single PDF whereas $f(X)$ is represented as a family of PDFs. $f(X|\mathbf{\Theta}_X = \mathbf{\Theta}_X^*)$ represents the variability in X when the parameters are fixed, i.e., $\mathbf{\Theta}_X = \mathbf{\Theta}_X^*$. Therefore, the overall uncertainty in X is due to two uncertainty sources – (1) uncertainty regarding the distribution parameters (epistemic uncertainty), and (2) variability in X when the distribution parameters are fixed (aleatory uncertainty).

It is desirable to quantify the contributions of each of the uncertainty sources to the overall uncertainty as it helps in the decision-making process, i.e., if the contribution of epistemic uncertainty is larger compared to the aleatory uncertainty, then additional data may be collected to reduce the uncertainty in the distribution parameters. In the auxiliary variable approach [50], the auxiliary variable corresponds to the variability in X for fixed distribution parameters. If U_X represents the auxiliary variable, then it can be defined as $U_X = F(X|\mathbf{\Theta}_X)$, where $F(.)$ represents the CDF of X. Thus, for a realization of U_X, denoted as U_X^*, a sample of X, denoted as x, can be computed as

$$x = F^{-1}(U_X = U_X^* | \mathbf{\Theta}_X = \mathbf{\Theta}_X^*) \tag{2.26}$$

In Eq. (2.26), $F^{-1}(.)$ represents the inverse CDF of X. It should be noted that U_X follows a uniform distribution between 0 and 1, as it represents the CDF of X, when conditioned on a set of distribution parameters, $\mathbf{\Theta}_X = \mathbf{\Theta}_X^*$. For several realizations of U_X and $\mathbf{\Theta}_X$, several values of the random variable X can be obtained using the inverse CDF, and thus sensitivity analysis can be carried out.

To differentiate between the inputs with and without uncertainty distribution parameters, we divide the inputs into two non-intersecting subsets $\boldsymbol{X}_p$ and $\boldsymbol{X}_{np}$. Therefore, $Y = G(X_p, X_{np}, \Theta_M)$ is a deterministic relationship. Using the auxiliary variable representation, an input $\in X_p$can be replaced by an auxiliary variable and its distribution parameters using Eq. 2.26. Therefore, a deterministic relationship between the inputs and the output, after considering uncertain distribution parameters can be given as

$$Y = H\big(U_p, \Theta_p, X_{np}, \Theta_M\big) \tag{2.27}$$

where U_p and Θ_p correspond to the auxiliary variables and distribution parameters of X_p. $H(.)$ represents the deterministic relationship between the uncertain distribution parameters and the model output. The functions $H(.)$ and $G(.)$ can be related using the inverse CDF in Eq. 2.26 as $H\left(U_p, \Theta_p, X_{np}, \Theta_M\right) = G(F^{-1}\left(U_p \middle| \Theta_p\right), X_{np}, \Theta_M)$. In some cases, it might be possible that the relationship between the inputs X and Y is not deterministic but stochastic in nature. An example of such a case is when a Gaussian process model is used, which provides a Gaussian distribution as an output with parameters dependent on the input. Presence of a stochastic model is not conducive for carrying out sensitivity analysis. Therefore, we again use the auxiliary variable for representing the uncertainty in the model output, which enables us to obtain a deterministic relationship and to carry out sensitivity analysis.

Similar to Eq. (2.26), the model output can be represented in terms of an auxiliary variable and its distribution parameters as $Y = F^{-1}(U_Y|\Theta_Y)$. The distribution parameters Θ_Y (which represent the mean and a standard deviation when a Gaussian process is used) are dependent on the inputs (or distribution parameters of inputs or model parameters) as $\Theta_Y = J(U_p, \Theta_p, X_{np}, \Theta_M)$. Therefore, the deterministic relationship (represented as K) between the model output and the several types of inputs (model inputs, model parameters and distribution parameters of model inputs) in the presence of a stochastic model can be given as

$$Y = K(U_Y, U_p, \Theta_p, X_{np}, \Theta_M) \tag{2.28}$$

where $K\left(U_Y, U_p, \Theta_p, X_{np}, \Theta_M\right) = F^{-1}(U_Y|J(U_p, \Theta_p, X_{np}, \Theta_M))$. Though the auxiliary variable was initially developed to represent the aleatory uncertainty at a given realization of the distribution parameters, it is later extended to handle uncertainty in the model prediction (a source of epistemic uncertainty). Thus, the auxiliary variable can be used to represent both aleatory and epistemic uncertainty sources. More details regarding the use of auxiliary variable to handle different uncertainty sources are given in [51]. In this dissertation, we use the auxiliary variable to represent the uncertainty in the model discrepancy prediction, as it is modeled using a Gaussian process. The auxiliary variable-based model discrepancy representation is then used in the construction of a limit state surrogate, which is used for reliability analysis. More details are provided in Section 3.2.

2.6 Bayesian networks

The discussion thus far considered uncertainty representation in variables and their propagation through a single physics-based model to obtain the uncertainty in the output QoI. For a system with multiple components, physics-based models may not be available for all the components but some data might be available. In addition, the uncertainty from several individual models may combine in a linear or a non-linear manner to obtain the uncertainty in the system QoI. In such cases, where a combination of physics-based models and data are available to represent a system, Bayesian networks offer a systematic and a rigorous approach to fuse the information from multiple models and aggregate several uncertainty sources arising from multiple models to estimate the uncertainty in the system QoI. Also, in the presence of any new information regarding the system, a Bayesian network facilitates the inference of parameters corresponding to several models simultaneously. In this dissertation, we consider multi-level systems, time varying coupled systems and coupled multi-physics systems each consisting of several components. Therefore, a brief introduction to Bayesian networks is given below.

A Bayesian network (BN) represents the joint probability distribution of a set of random variables through a directed acyclic graphical model. The graphical model consists of nodes and directed arcs where nodes represent random variables and arcs represent the dependence between the nodes. Using the directional information from the graphical model, the joint probability of the random variables is represented as a product of conditional and marginal probability distributions. The joint probability of $\boldsymbol{n}$ random variables, $\boldsymbol{X} = \{X_1, X_2 \dots X_n\}$ can be represented as

$$f(\boldsymbol{X}) = \prod_{i=1}^{n} f(X_i|\Pi_{X_i}) \tag{2.29}$$

where Π_{X_i} represents the set of parent nodes of X_i, i.e., nodes from which the arcs direct to X_i and $f(X_i|\Pi_{X_i})$ represents the conditional probability distribution of X_i conditioned on its parent nodes. If X_i has no parent nodes (also referred to as root nodes), then $f(X_i|\Pi_{X_i})$ represents the marginal distribution of X_i. For illustration, consider a 5-node Bayesian network shown in Fig. 2.3.

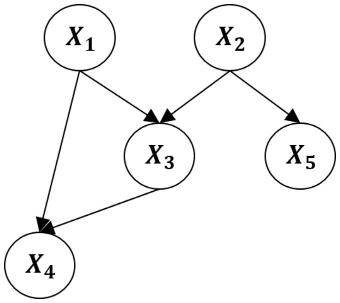


Fig. 2.3. An illustrative Bayesian network.

Using the dependence information from Fig. 2.3, the joint probability distribution of X_1, X_2, X_3, X_4 and X_5 can be described as

$$f(X_1, X_2, X_3, X_4, X_5) = f(X_1)P(X_2)P(X_3|X_1, X_2)P(X_4|X_3, X_1)P(X_5|X_2) \tag{2.30}$$

Techniques for the construction of Bayesian networks are discussed below in Section 2.6.1. Bayesian networks are primarily used to update our knowledge on a subset of random variables when another subset of random variables is observed. Depending upon the network complexity, several analytical and approximate techniques have been developed to perform Bayesian Inference; these techniques are discussed in Section 2.7.

2.6.1 Bayesian network construction

The techniques for constructing a BN can be broadly classified into three types: (1) physics-based, (2) data-driven, and (3) hybrid approaches. The physics-based approach relies on mathematical models that represent the relationships between the system variables. The data-driven approach uses the available data to learn the BN structure and conditional probabilities using learning algorithms discussed later in this section. In some cases, mathematical models might be available for some segments of the system and data is available for other segments. In such a scenario, a hybrid approach is taken, where physics-based equations are used to model some dependence relations whereas the remaining relationships are learned from the available data. The BN is constructed in two stages – (1) a partial BN is obtained using the available physics-based models, and (2) the BN constructed in step 1 is used as a prior for learning the remaining dependence relations using the available data.

2.6.2. Learning Bayesian networks from data

The goal of the learning algorithms is to identify a BN that best describes the available data. The learning process involves two tasks: structure learning and parameter learning. Structure learning involves finding a graphical structure that best represents the dependence between nodes based on available data. Parameter learning involves quantification of dependence among the nodes by estimating the parameters of the conditional probability distributions/tables [52].

The structure learning algorithms can be broadly divided into three categories: (1) constraint-based, (2) score-based, and (3) hybrid [52]. Constraint-based methods employ conditional independence tests to learn an optimal BN [52]. A commonly used conditional independence test is the mutual information test. The expressions for mutual information ($I_{X,Y}$) in the case of discrete variables and continuous variables are given in Eqs. (2.29, 2.30) respectively, as

$$I_{X,Y} = \sum_{Y}\sum_{X} Pr(x,y)\, log\left(\frac{Pr(x,y)}{Pr(x)P(y)}\right) \tag{2.31}$$

$$I_{X,Y} = \int_{Y}\int_{X} Pr(x,y)\, log\left(\frac{Pr(x,y)}{Pr(x)P(y)}\right) \tag{2.32}$$

where $Pr(x,y)$ represents the joint probability distribution of X and Y, and $Pr(x)$ and $Pr(y)$ represent the marginal distributions of X and $Y,$ respectively. Some other conditional independence tests that are used include linear correlation and conditional correlation for continuous variables, and G-test and Chi-square test [23] for discrete variables.

In score-based learning, every candidate BN structure is assigned a network score based on the goodness-of-fit for available data and a set of heuristic optimization techniques are used to obtain the BN that optimizes the defined score. Some commonly used metrics that are used for scoring Bayesian networks include Log-likelihood, Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), Minimum Description Length (MDL) and Bayesian Dirichlet Equivalence (BDe) [22]. A commonly used scoring metric is Bayesian Dirichlet equivalence (BDe). The BDe based scoring criterion maximizes the posterior probability of a network-structure given data and is given as

$$Pr(G|D) \propto Pr(D|G)\, P(G) = P\,r(G) \int Pr(D|G,\Theta)\, Pr(\Theta|G)\, d\Theta \tag{2.33}$$

In Eq. (2.33), G, D represent the structure of the BN and available data, respectively. Θ represents the parameters of the conditional probability distributions.

Hybrid algorithms employ both conditional independence tests and network scores for learning the BN structure. The conditional independence tests are first used to reduce the space of candidate BN structures and score-based methods are then used to obtain the optimal BN structure among them [52]. Parameter learning algorithms estimate the parameters of the conditional probability distributions from available data using the maximum-likelihood approach [53]. We use the concept of Bayesian networks for computationally efficient performance assessment framework in multi-level and multi-physics systems detailed in Chapters 4 and 6 respectively.

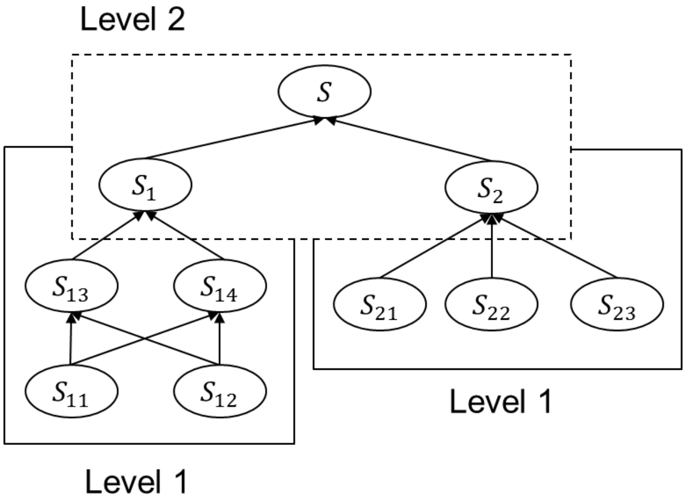


Fig. 2.4. A conceptual representation of a hierarchical Bayesian network

2.6.3 Hierarchical Bayesian networks

A hierarchical Bayesian network (HBN) can be considered as a Bayesian network where each node may be connected to another lower-level Bayesian network forming a hierarchy. Any number of levels are possible i.e., a node in a lower level BN can also represent a further lower level Bayesian network. The focus in this work is to compose the BN of a hierarchical system (such as a manufacturing network) by integrating the BNs of subsystems at multiple levels (see Chapter 4). Fig. 2.4 shows a simple representation of a hierarchical Bayesian network. In Fig. 2.4, the HBN consists of two levels – Level 1 and Level 2. The root variables in the Level 2 BN ($\boldsymbol{S_1}$ and $\boldsymbol{S_2}$) are connected to lower-level BNs. Chapter 4 discusses HBNs for performance assessment in multi-level systems, with multiple possible levels of hierarchy.

2.6.4 Dynamic Bayesian networks

Dynamic Bayesian Networks belong to a class of state-space models used to model the time-dependent behavior of dynamic systems. A DBN can be regarded as a composition of two BNs. (1) a BN connecting the variables in a single time instant, also referred to as a Static BN, and (2) a BN connecting variables across time instants, and also referred to as a Transitional BN; these BNs are shown in Fig. 2.5. We consider DBNs with the Markov assumption, i.e., the state variables in the current time step are only dependent on the state variables in the previous time step and the inputs in the current time step, resulting in a 2-slice DBN [54]:

$$\boldsymbol{P^{t+1}} = G(\boldsymbol{P^{t}}, \boldsymbol{v^{t+1}}) \tag{2.34}$$

$$\boldsymbol{Q^{t}} = H(\boldsymbol{P^{t}}) \tag{2.35}$$

In the above equations, $\boldsymbol{P^{t}}$ and $\boldsymbol{P^{t+1}}$ represent the state variables in two time steps. Similarly, $\boldsymbol{Q^{t}}$ and $\boldsymbol{Q^{t+1}}$ represent observation variables at two time steps. The evolution to $\boldsymbol{P^{t+1}}$ from $\boldsymbol{P^{t}}$ can be represented in Eq. (2.34). $\boldsymbol{v^{t}}$ and $\boldsymbol{v^{t+1}}$ refer to system inputs at time t and $t+1$ respectively. Eq. (2.35) represents the relationship connecting observation variables $\boldsymbol{Q^{t}}$ and state variables $\boldsymbol{P^{t}}$. Probabilistic modeling of systems where Markov assumption does not hold good is not considered in this research.

The techniques used for BN learning (Section 2.6.2) can be extended for learning DBNs [54]. As discussed above, a DBN consists of two BNs – static BN and a transitional BN. First, the static BN can be learnt using a combination of available physics-based models or data or their combination (hybrid approach). We discussed above that the state variables in one time step are dependent on the state variables in the previous time step and the inputs in the current time step. As the variables in the transitional BN are known, the learning techniques (model, data or hybrid) can again be used for learning the transitional BN. Thus, the learning of a DBN reduced to learning two BNs.

Dynamic Bayesian networks have been used for real-time health monitoring, diagnosis, prognosis and remaining useful life (RUL) estimation for a variety of mechanical [55], manufacturing [56], infrastructural [57] and aerospace [58] engineering systems. In this dissertation, we use DBNs for modeling coupled and time-varying systems (such as cyber-physical systems) for the purpose of their performance assessment.

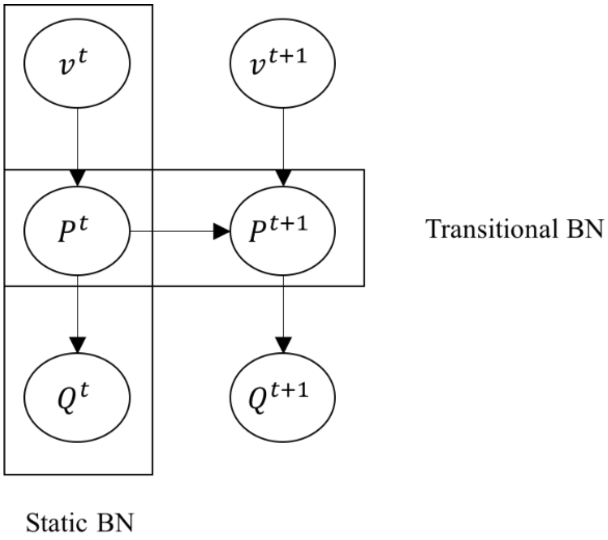


Fig. 2.5. A conceptual DBN between two consecutive time steps

2.7 Bayesian inference

In this section, we discuss the implementation of Bayesian inference in both static and dynamic Bayesian networks.

2.7.1 Inference in a static Bayesian network

The Bayesian network constructed using the techniques in Section 2.6 can now be used for inferring the posterior distributions of unobserved variables (denoted as $\boldsymbol{X_{unobs}}$) using any data ($\boldsymbol{D}$) on the observed variables ($\boldsymbol{X_{obs}}$) via the Bayes' theorem as

$$f(\boldsymbol{X_{unobs}}|\boldsymbol{X_{obs}} = \boldsymbol{D}) = \frac{f(\boldsymbol{X_{obs}} = \boldsymbol{D}|\boldsymbol{X_{unobs}})\boldsymbol{f}(\boldsymbol{X_{unobs}})}{\int f(\boldsymbol{X_{unobs}}, \boldsymbol{X_{obs}} = \boldsymbol{D})\ \boldsymbol{dX_{unobs}}} \tag{2.36}$$

In Eq. (2.36), the terms $f(\boldsymbol{X_{unobs}}|\boldsymbol{X_{obs}} = \boldsymbol{D})$, $f(\boldsymbol{X_{unobs}})$ and $f(\boldsymbol{X_{obs}} = \boldsymbol{D}|\boldsymbol{X_{unobs}})$ represent the posterior distributions of unobserved variables, their prior distributions and the likelihood function of observed variables. The denominator term, $\int f(\boldsymbol{X_{unobs}}, \boldsymbol{X_{obs}} = \boldsymbol{D})\ \boldsymbol{dX_{unobs}}$, refers to the probability of observed variables to be equal to the data; this is a deterministic value that can be computed by marginalizing over the unobserved variables.

Updating in a generic Bayesian network can be computationally intractable due to high-dimensional integration, i.e., exact inference is prohibitively expensive. However, exact algorithms are available for updating in special classes of Bayesian networks such as discrete Bayesian networks, conditional linear Gaussian networks, networks where conditional dependence relationships are modeled using mixtures of truncated exponentials or truncated polynomials [59]. Some of the exact inference algorithms include Junction Tree[53], Variable Elimination [53] and Differential Approach methods [60]. When exact updating techniques are not available, sampling-based (approximate) techniques are used to obtain the posterior distributions. In the case of forward uncertainty propagation when the observed variables are simply the inputs, then the posterior distributions of unobserved (output) variables can be obtained using Monte Carlo sampling over the conditional dependence relationships. For the inverse problem of Bayesian inference, techniques such as Markov Chain Monte Carlo (MCMC) methods [27], Variational methods [28], Bootstrap filters [29] and Approximate Bayesian Inference (ABC) [30] can be used to obtain the posterior distributions. Among these approximate methods, the MCMC-based methods are widely used. MCMC is not a single algorithm but a class of algorithms that use Markov Chain techniques to achieve an approximation to the underlying true posterior distribution. Algorithms such as Metropolis-Hastings [61], Gibbs sampling [62] and Slice sampling [63] fall under the class of MCMC algorithms. In this dissertation, we used the Metropolis-Hastings algorithm for Bayesian inference. A brief introduction to it is given below.

As mentioned above, the Metropolis-Hastings (MH) algorithm can be used to obtain an approximation of posterior distributions of variables when the analytical solution is computationally expensive. The MH algorithm generates a Markov chain of samples, i.e., the next sample is dependent on the previous sample (Markov assumption). When the Markov chain converges, the samples in the Markov chain after convergence are used to construct the posterior distributions. A Markov chain is assumed to converge when there exist no trend on the sequence of samples and no correlation between successive samples. The MH algorithm requires three inputs – (1) a function $h(x)$, that is proportional to the posterior distribution, (2) a proposal density function, $j(x^*|x_i)$, which is used to generate samples in the Markov chain and (3) an initialization point of the Markov chain, x_0.

Let C_X represent the Markov chain of samples. Initially, $C_X = \{x_o\}$ as x_o is the initial sample. The following steps are repeated for the generation of samples in the Markov chain.

1. Using the i^{th} sample x_i, a candidate sample x^* is generated according to the proposal density function, $j(x^*|x_i)$. Next, we evaluate if this sample needs to be accepted as part of the Markov chain. We calculate the acceptance ratio, as $r = \min\left(1, \frac{h(x^*)}{h(x_1)}\right)$.

2. The acceptance ratio is then compared against the random uniform sample, u, generated between 0 and 1.

3. If $u < r$, then the candidate sample is accepted to the Markov chain, i.e., $x_{i+1} = x^*$. On the contrary, if $u > r$, then the previous sample x_i is repeated, i.e., $x_{i+1} = x_i$.

The samples from the converged segment of the Markov chain are used to construct the posterior distributions using techniques such as kernel density estimation. More details regarding the MH algorithm are given in [61]. The MH algorithm is later used in Chapter 4 for inference in hierarchical Bayesian networks. After considering inference technique in a static BN, we now consider inference in a DBN.

2.7.2 Inference in a dynamic Bayesian network

In a DBN, the state variables need to be estimated using the data on observation variables at every time step. As DBNs are used in real-time analysis, the inference techniques used need to be computationally inexpensive. The commonly used MCMC methods can be computationally demanding and may not be suited for real-time analysis. Therefore, several exact and approximate inference techniques have been developed to estimate the state variables in real-time. Different techniques are available for inference in discrete, continuous and hybrid (both discrete and continuous) variable DBNs. In this work, we do not consider discrete variable DBNs but consider only continuous and hybrid DBNs. A list of possible inference techniques for discrete DBNs are available in [42]. One of the simplest formulations of a DBN is a linear Gaussian system, with linear functions between the state and observation variables, and between state variables in successive time steps, and the observation data is associated with Gaussian noise (zero mean and fixed covariance). For such linear Gaussian systems, the Kalman filter technique [64] provides exact and analytical inference of state variables using data on observation variables.

For non-linear Gaussian systems, variants of Kalman filter such as Extended Kalman filter (EKF) [65] and Unscented Kalman filter (UKF) [66] have been developed. These methods are approximate but computationally inexpensive. The extended Kalman filter implements local linear approximations, which helps to obtain analytical inference. The unscented Kalman filter is based on the concept of unscented transform, where the state and observation variables are approximated using Gaussian distributions, and their first and second order moments are obtained with “sigma” points and their associated weights. These sigma points are also used to obtain the covariance between the state and observation variables. After obtaining Gaussian approximations of the state and observation variables, and the covariance, analytical inference (similar to that used in Kalman filter) is used to infer the state variables according to the observation data. More details regarding the use of EKF and UKF are given in [67].

Apart from Kalman filter and its variants, several sampling-based inference techniques are available for generic non-linear and non-Gaussian noise systems such as particle filtering methods. Similar to MCMC, particle filtering is not a single algorithm but a class of algorithms; some of the algorithms include Sequential Importance Sampling (SIS), Sequential Importance Resampling (SIR) and Rao-Blackwellized Particle Filter [67]. In this work, we use the SIR algorithm; the steps in the algorithm are given below.

1. Generate N samples of the state variables at the current time step, P_k^t, $k = 1,2..N$

2. Compute the likelihood of each of the N particles by propagating them through the static BN and by using the observation data.

3. Compute weights for each particle as being proportional to their likelihood measures

4. Resample the generated N values of the state variables according to the computed weights and obtain N values; these values can be used to construct the posterior distributions of the state variables. These samples can then be used to obtain the prior distributions of the state variables in the next time step by propagating them through the transitional BN. This process can be repeated for several time steps as required for the analysis. The SIR algorithm-based filtering is used in Chapter 5 for dynamic updating of model parameters in coupled and time-varying systems.

CHAPTER 3

PERFORMAMCE ASSESSMENT IN SINGLE-COMPONENT SYSTEMS

3.1. Introduction

Performance assessment under uncertainty can be considered as a generalization of reliability analysis, which computes the probability that a pre-defined performance function is greater than a threshold. Techniques for reliability analysis can be categorized into analytical methods and simulation-based methods [68]. Analytical methods such as First Order Reliability Method (FORM) and Second Order Reliability Method (SORM) approaches employ first order and second order approximations of the limit state [31]. First-order and second-order bounds for system reliability estimates have been proposed based on first-order and second-order approximations of the limit states [69, 70]. The FORM and SORM-based methods become inaccurate when the limit-states are highly nonlinear. Monte Carlo sampling (MCS) approaches can be accurate but computationally expensive. To reduce the computational effort, surrogate-based reliability analysis methods have been developed [71–73] . If the computational model is expensive, two categories of surrogates – general purpose and limit state have been used in the reliability analysis literature. A brief introduction to these types of surrogates was provided in Section 2.3.

In the context of general purpose surrogate (commonly known as response surface), Faravelli [74], Choi et al [75] used a polynomial expansion response surface model, Papadrakikis et al [76] used a neural network based surrogate, Dubourg et al [77], Kaymaz [78] used a Kriging (or Gaussian process) surrogate. In the case of limit state surrogates (which are basically classifiers), Bichon et al [37] and Echard et al [39] used a Gaussian process (GP) surrogate while Song et al [79] used a support vector machine (SVM)-based surrogate. In this work, the limit state surrogates (classifiers) are considered. The surrogate-based reliability methods have mainly considered only aleatory uncertainty (natural variability) so far. However, practical reliability analysis is affected by many sources of epistemic uncertainty (lack of knowledge); therefore, this work investigates approaches to include such uncertainty sources in surrogate-based reliability analysis. Different types of epistemic uncertainty sources and their quantification are discussed in Section 2.1. In recent years, efforts have been made to account for both aleatory and epistemic uncertainty within reliability analysis, such as the auxiliary variable approach [13], the conditional reliability index method [80], and the Bayesian network approach [69]. These methods have only concentrated on component reliability analysis (i.e. a single limit state) considering a few sources of epistemic uncertainty.

Therefore, this chapter seeks to incorporate all the discussed epistemic uncertainty sources in Sections 2.1and 2.2.3, i.e., statistical uncertainty (distribution parameter and distribution type), model uncertainty (model parameter uncertainty and model discrepancy) and reliability analysis (surrogate uncertainty and Monte Carlo sampling error) in limit state surrogate-based reliability analysis.

Earlier studies such as [37, 39, 77, 78] considered construction of the limit state surrogate for the physics-based simulation model without any model discrepancy. In this work, the model discrepancy is also considered in constructing the limit state surrogate. However, construction of any surrogate requires point values of paired input-output data whereas in the presence of model discrepancy, the output at any input is a PDF. This problem is overcome in this work by using an auxiliary approach, which allows for a one-to-one relationship between input and output. This work uses a GP surrogate to *model the limit state* following the Efficient Global Reliability Analysis (EGRA) approach proposed by Bichon et al [37]. However, the techniques proposed in this work are not limited to GP models and can be extended to any surrogate. The GP surrogate of the simulation model obtained from the KOH framework-based calibration analysis (discussed in Section 2.1.2) is further refined to construct the limit state surrogate by adding more training points close to the limit state. The training points are adaptively selected by maximizing a learning function called the Expected Feasibility Function (EFF). More details about the EFF and selection of training points are provided in Section 2.3.2.

In the presence of statistical uncertainty, an input variable can be represented through a family of PDFs, where each PDF corresponds to a realization of the distribution parameters and a distribution type. The sampling of the input variable can be done through a nested sampling approach where the distribution type and parameters are sampled in the outer loop, and the samples of the input variable are generated in the inner loop. This nested-loop procedure is computationally expensive; therefore a faster single loop sampling approach using the probability integral transform [24] is used here.

The major contribution is a unified framework connecting the model calibration analysis to constructing a limit state surrogate and estimating the uncertainty in reliability analysis by incorporating different sources of epistemic uncertainty. The basic contributions include: (1) The use of the auxiliary variable approach to represent the model discrepancy for its inclusion in limit state surrogate refinement, and (2) Quantification of different types of epistemic uncertainty and their incorporation in reliability analysis to quantify the uncertainty in the reliability estimate.

3.2. Proposed methodology

An overview of the proposed methodology for surrogate-based reliability estimation by including different types of epistemic uncertainty is presented in Fig. 3.1.

The overall approach can be divided into two stages as shown in Fig. 3.1 – (1) construction of a surrogate, and (2) use of the surrogate for reliability analysis. Uncertainty sources such as statistical uncertainty in the inputs, uncertain model parameters, and model discrepancy influence the surrogate construction, thereby affecting the reliability estimate, whereas the surrogate uncertainty and MCS error do not influence the surrogate construction but only affect the reliability estimate. Details regarding the handling of different sources of epistemic uncertainty in these two stages are discussed in this section.

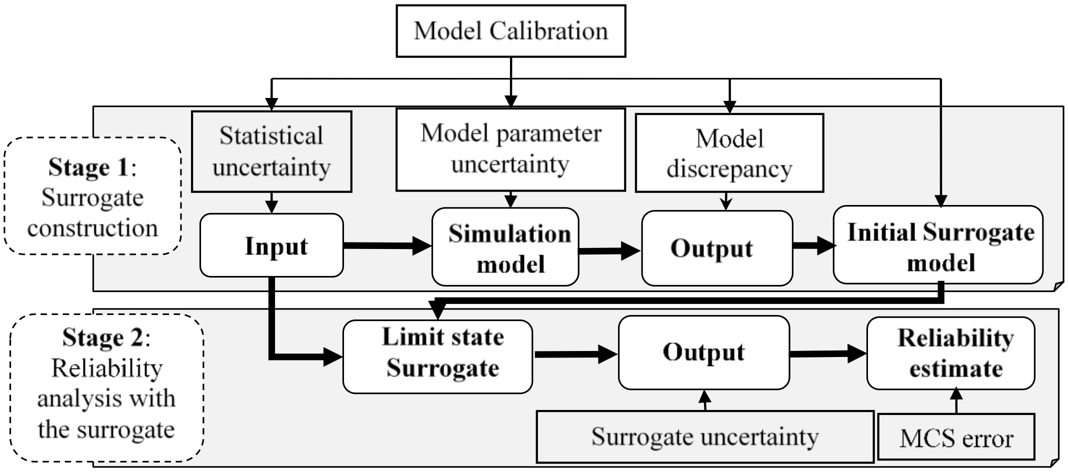


Fig. 3.1. Flow chart of proposed framework for reliability analysis under uncertainty

3.2.1. Limit state surrogate construction including epistemic uncertainty

First, the different inputs to be included in the surrogate are discussed which is followed by the surrogate construction.

A. Inputs to the surrogate

In this work, the inputs for the surrogate model include the inputs for the simulation model ($\boldsymbol{X}$), the uncertain model parameters ($\boldsymbol{\Psi}$) and the model discrepancy. Consider a system with m limit state functions given by $g_i(\boldsymbol{X},\boldsymbol{\Psi}) = g_{model,i}(\boldsymbol{X},\boldsymbol{\Psi}) + \widehat{\delta_\iota}(\boldsymbol{X})$ where $g_{model,i}(\boldsymbol{X},\boldsymbol{\Psi})$ and $\widehat{\delta_\iota}(\boldsymbol{X})$ are the simulation model and the model discrepancy of the ith limit state function respectively. Since the KOH calibration framework is used, the model discrepancy is not modeled as a function of model parameters. However, including the model parameters in the model discrepancy term might be a more rigorous. The quantification of the model discrepancy is by using the simulation model. The performance of the overall model (simulation model and model discrepancy) might be satisfactory over the entire range of the inputs but the main idea of the limit state surrogate is to model the limit state perfectly and is not concerned about the its performance in the interior domain of the inputs (away from limit state). Since the limit state model governs the reliability estimate (and its uncertainty), we wish to model it as precisely as possible. In system reliability, it is possible in some cases that we may have some calibration data on a few limit states but not a lot of data on the remaining limit states. Therefore, the construction of the limit state surrogate refines the system limit state by obtaining training points considering all the individual limit states. The focus of this work is to quantify the uncertainty in the reliability prediction by investigating various sources of epistemic uncertainty. It is based on the assumption that the model calibration has already been performed. We obtain a GP surrogate for the model discrepancy from calibration analysis.

The key idea is to include the model discrepancy terms in the construction of the limit state surrogate and then use the surrogate for reliability analysis. The limit-state function $\boldsymbol{g_{ext}}(\boldsymbol{X},\boldsymbol{\Psi})$ for surrogate-based reliability analysis is formulated as

$$\boldsymbol{g_{ext}}(\boldsymbol{X},\boldsymbol{\Psi}) = \begin{cases} g_{model}(\boldsymbol{X},\boldsymbol{\Psi}) + \hat{\delta}(\boldsymbol{X}), \quad \text{for a component} \\ Min\left\{g_{model,i}(\boldsymbol{X},\boldsymbol{\Psi}) + \ \widehat{\delta_\iota}(\boldsymbol{X})\right\}, \text{for a series system} \\ Max\left\{g_{model,i}(\boldsymbol{X},\boldsymbol{\Psi}) + \ \widehat{\delta_\iota}(\boldsymbol{X})\right\}, \text{for a parallel system} \end{cases} \tag{3.1}$$

Note that the model discrepancy, $\widehat{\delta_\iota}(\boldsymbol{X})$, is random at any given point $\boldsymbol{X} = \boldsymbol{x}$, which results in uncertainty in the response $g_{ext}(\boldsymbol{X},\boldsymbol{\Psi})$. Directly constructing a surrogate for the implicit response given in Eq. (3.1) is not practical due to randomness in $\widehat{\delta_\iota}(\boldsymbol{X})$. An explicit representation of variability in $\widehat{\delta_\iota}(\boldsymbol{X})$ is required to formulate Eq. (3.3) as a deterministic function; this challenge is addressed using the auxiliary variable method, briefly described in Section 2.5.

The auxiliary variable approach provides a deterministic relationship between $(\boldsymbol{\Theta},\boldsymbol{u})$ and $\boldsymbol{X}$. Here, $\boldsymbol{\Theta}$ and $\boldsymbol{u}$ represent the distribution parameters and auxiliary variables respectively. In this work, we have extended the concept of the auxiliary variable further, to represent the epistemic uncertainty in the prediction of model discrepancy and in the reliability estimate. In general, whenever there is a stochastic mapping, i.e., mapping of a single value to a probability distribution, the auxiliary variable can be used to convert the stochastic mapping to a deterministic mapping. An auxiliary variable is used to represent the variability in the prediction of model discrepancy in order to build the limit state surrogate. Since a GP is used to model $\widehat{\delta_i}(\boldsymbol{X})$, for any given $\boldsymbol{X} = \boldsymbol{x}$, the model discrepancy follows a Gaussian distribution given by $\widehat{\delta_i}(\boldsymbol{X}) \sim N(\boldsymbol{\mu}_{\delta_i(x)}, \ \boldsymbol{\sigma}_{\delta_i(x)})$. The PDF of $\widehat{\delta_i}(\boldsymbol{X})$ is $f_\delta(\delta_i(\boldsymbol{x})|\boldsymbol{\mu} = \boldsymbol{\mu}_{\delta_i(x)}, \boldsymbol{\sigma} = \boldsymbol{\sigma}_{\delta_i(x)})$ and the CDF value is given by

$$u_{MD,i} = F_\delta\big(\delta_i(\boldsymbol{x}, \boldsymbol{\psi})\big|\boldsymbol{\mu} = \boldsymbol{\mu}_{\delta_i(x)}, \boldsymbol{\sigma} = \boldsymbol{\sigma}_{\delta_i(x)}\big) = \int_{-\infty}^{\boldsymbol{\delta_x}} f_\delta\big(\boldsymbol{w}\big|\boldsymbol{\mu} = \boldsymbol{\mu}_{\delta_i(x)}, \boldsymbol{\sigma} = \boldsymbol{\sigma}_{\delta_i(x)}\big)\boldsymbol{dw} \qquad (3.2)$$

Using the auxiliary variable, a realization of $\delta_i(\boldsymbol{x}, \boldsymbol{\psi})$ for a given $U_{MD,i} = u_{MD,i}$, $\boldsymbol{\mu} = \boldsymbol{\mu}_{\delta_i(x)}, \boldsymbol{\sigma} = \boldsymbol{\sigma}_{\delta_i(x)}$, can be computed by

$$\delta_i(\boldsymbol{x}) = F_{\delta_i(\boldsymbol{x})}^{-1}(U_{MD,i} = u_{MD,i}|\boldsymbol{\mu} = \boldsymbol{\mu}_{\delta_i(x)}, \boldsymbol{\sigma} = \boldsymbol{\sigma}_{\delta_i(x)}) \qquad (3.3)$$

where $F_{\delta_i(x)}^{-1}$ is the inverse Gaussian CDF of $\boldsymbol{\delta_i}(\boldsymbol{x})$. Combining Eqs. (3.1) and (3.3), we have

$$g_{ext}(\boldsymbol{X}, \boldsymbol{\Psi}, \mathbf{U}) = \begin{cases} g_{model}(\boldsymbol{X}, \boldsymbol{\Psi}) + \boldsymbol{F}_{\delta(x)}^{-1}(\boldsymbol{U_{MD}}|\boldsymbol{\mu} = \boldsymbol{\mu}_{\delta(x)}, \boldsymbol{\sigma} = \boldsymbol{\sigma}_{\delta(x)}), & \text{for a component} \\ Min\ \left\{g_{model,i}(\boldsymbol{X}, \boldsymbol{\Psi}) + \ \boldsymbol{F}_{\boldsymbol{\delta_i}(x)}^{-1}(\boldsymbol{U_{MD,i}}|\boldsymbol{\mu} = \boldsymbol{\mu}_{\boldsymbol{\delta_i}(x)}, \boldsymbol{\sigma} = \boldsymbol{\sigma}_{\boldsymbol{\delta_i}(x)})\right\}, & \text{for a series system} \\ Max\ \left\{g_{model,i}(\boldsymbol{X}, \boldsymbol{\Psi}) + \ \boldsymbol{F}_{\boldsymbol{\delta_i}(x)}^{-1}(\boldsymbol{U_{MD,i}}|\boldsymbol{\mu} = \boldsymbol{\mu}_{\boldsymbol{\delta_i}(x)}, \boldsymbol{\sigma} = \boldsymbol{\sigma}_{\boldsymbol{\delta_i}(x)})\right\}, & \text{for a parallel system} \end{cases} \qquad (3.4)$$

Thus, the original model with stochastic output is mapped to a deterministic model, which can then be used to build a surrogate for reliability analysis. Initial GP surrogates (for simulation model and model discrepancy) are obtained from the KOH calibration framework as shown in Stage 1 of Fig. 3.1. These surrogates can directly be used for reliability analysis. However, if better accuracy is desired, then the GP surrogate of the simulation model can be further refined around the limit state (Stage 2 in Fig. 3.1). This local refinement requires additional training points and these are obtained by evaluating the simulation model. After constructing the limit state surrogate, it is directly used for reliability analysis without any further runs of the original model.

B. Surrogate construction

As stated in Section 2.3.2, the training points for the construction of the limit state surrogate are adaptively selected by maximizing a learning function called the Expected Feasibility Function (EFF). Several optimization techniques (gradient-based, sampling-based) are available for maximizing the EFF. In this work, a Monte Carlo sampling-based technique is implemented, as demonstrated in the Adaptive-Kriging Monte Carlo Simulation (AK-MCS) method [39]. The key idea in MCS-based optimization is to generate a pool of samples of the inputs from their corresponding PDFs and choose the sample from the pool that maximizes the EFF. The epistemic uncertainty in the inputs of the limit state surrogate can be included in the surrogate modeling construction through the sampling of input variables, which is discussed below.

Inputs to the simulation model (***X***): Quantification of uncertainty in the distribution parameters and distribution type of random input variables was discussed in Section 2.1. For a realization of distribution parameters and distribution type, the input variable is represented by a PDF; therefore, for multiple realizations of distribution parameters and type, the input variable is represented through a family of PDFs. The traditional approach for sampling of an input variable with uncertain distribution parameters and distribution type is through a nested double-loop procedure where the distribution parameters and distribution type are sampled in the outer loop and samples of the input random variable are generated in the inner loop. The double-loop sampling procedure is computationally expensive; therefore, a single-loop sampling procedure using an auxiliary variable based on the probability integral transform is used. An auxiliary variable U_X is defined to represent the aleatory uncertainty in a random variable and represented using the probability integral transform as

$$u_X = F_X(x|\Theta = \theta, d_X = d_X^*) = \int_{-\infty}^{x} f_X(w|\Theta = \theta, d_X = d_X^*)\, dw \tag{3.5}$$

where Θ, d_X represent the uncertain distribution parameters and distribution type; their realizations are represented by θ and d_X^*, w is a dummy variable for integration, $F_X(x)$ and $f_X(x)$ represent the CDF and PDF of X respectively. Thus, for a realization of auxiliary variable $U_X = u_X$, distribution parameters θ and distribution type d_X^* generated from their corresponding PDFs, one realization of the input variable can be obtained as $x = F_X^{-1}(u_X | \Theta = \theta, d_X = d_X^*)$ using the inverse CDF method. Following this procedure, several realizations of the input can be obtained.

If a non-parametric approach is used for the representation of statistical uncertainty (i.e. no distribution type or parameters), its CDF is constructed using numerical techniques which is then used for generating samples using inverse CDF technique. If an input variable is not associated with any statistical uncertainty (i.e. only known distribution type and distribution parameters), then the conventional inverse CDF technique [31] can be used for generating samples. To perform sampling of correlated variables, the correlated variables are first transformed into an uncorrelated space using orthogonal transformation or Cholesky decomposition. Samples are then generated individually in the uncorrelated space and transformed into the correlated space. Please refer to Chapter 9 in [31] for more details.

Uncertain model parameters ($\boldsymbol{\Psi}$): Since we are adopting the modular Bayesian approach, the obtained posterior distributions of the model parameters are conditioned on the MLE of hyper parameters, and correlations are not calculated between the model parameters and the model discrepancy. Correlations would be calculated when the fully Bayesian approach of Kennedy and O'Hagan [28] is implemented. The uncertain model parameters ($\boldsymbol{\Psi}$) can be treated similar to the inputs; therefore the sampling techniques presented above for the inputs can also be used for sampling the uncertain model parameters.

Model discrepancy ($\boldsymbol{U_{MD}}$): The auxiliary variable approach to explicitly represent the uncertainty in the model discrepancy has been discussed above. Since the auxiliary variables follow a uniform distribution between 0 and 1, their sampling can be carried out through random uniform sampling between 0 and 1.

From the model calibration analysis, we have data on the inputs, model parameters and simulation model output that was used for calibration. If we have k samples, then k random samples of the model discrepancy ($\boldsymbol{U_{MD}}$) are generated and these data can be used as initial training points to construct the limit state surrogate. Then, more training points are adaptively added by maximizing the EFF until a convergence criterion is satisfied. In this work, convergence is assumed to be achieved when the maximum EFF is less than a threshold value (EFF$^{\max}$ < EFF$*$).

3.2.2 Uncertainty quantification in reliability analysis

In Stage 1, several Monte Carlo samples of the inputs ($\boldsymbol{X}, \boldsymbol{\Psi}, \boldsymbol{U_{MD}}$) were generated to carry out the maximization of EFF for adaptive selection of training points for limit state surrogate construction. Since these samples are generated from their corresponding PDFs, these samples can also be used to carry out reliability analysis. Using these samples, the failure probability can be calculated as

$$p_f = \sum_{j=1}^{n_{MCS}} I(\hat{g}_{ext}(\boldsymbol{x}^{(j)}, \boldsymbol{u}_{\boldsymbol{MD}}^{(j)}, \boldsymbol{\psi}^{(j)} \leq 0)/n_{MCS} \tag{3.6}$$

where $\boldsymbol{x}^{(j)}, \boldsymbol{u}_{\boldsymbol{MD}}^{(j)}, \boldsymbol{\psi}^{(j)}$ are the j$^{\text{th}}$ sample of $\boldsymbol{X}, \boldsymbol{U_{MD}}$ and $\boldsymbol{\Psi}$ respectively. $\hat{\boldsymbol{g}}_{\boldsymbol{ext}}(.)$ represents the surrogate used to approximate the true limit state $g_{ext}(.)$ and $I(.)$ is the failure indicator function.

A. Inclusion of surrogate uncertainty in the reliability estimate

Since a GP surrogate is used, the prediction at any input is a Gaussian distribution with parameters dependent on the input. In most cases, only the mean predictions $\mu_{\hat{g}_{ext}}(\boldsymbol{x}^{(j)}, \boldsymbol{u}_{\boldsymbol{MD}}^{(j)}, \boldsymbol{\psi}^{(j)})$ are used to estimate $\hat{g}_{ext}(\boldsymbol{x}^{(j)}, \boldsymbol{u}_{\boldsymbol{MD}}^{(j)}, \boldsymbol{\psi}^{(j)})$. The failure probability is therefore estimated as

$$\hat{p}_f^s = \sum_{j=1}^{n_{MCS}} I(\mu_{\hat{g}_{ext}}(\boldsymbol{x}^{(j)}, \boldsymbol{u}_{\boldsymbol{MD}}^{(j)}, \boldsymbol{\psi}^{(j)} \leq 0)/n_{MCS} \quad (3.7)$$

where $\hat{p}_f^s$ is the estimate of the failure probability obtained by using the mean predictions. If our purpose is only to estimate the expected failure probability, we may ignore the correlation between the uncertain responses [81]. However, the focus of this work is to quantify the uncertainty in the reliability estimate. Therefore, the uncertainty due the surrogate prediction is also considered to quantify the overall uncertainty in reliability estimate. When the accuracy of the surrogate is high (i.e. the uncertainty of prediction is low), the above treatment of using the mean predictions works well, and results in a single value of the reliability estimate. If the accuracy of the model prediction is low, it becomes necessary to also include the prediction uncertainty for reliability estimation. In order to quantify the effects of surrogate uncertainty on reliability analysis, an uncertainty quantification problem is therefore formulated as shown in Fig. 3.2.

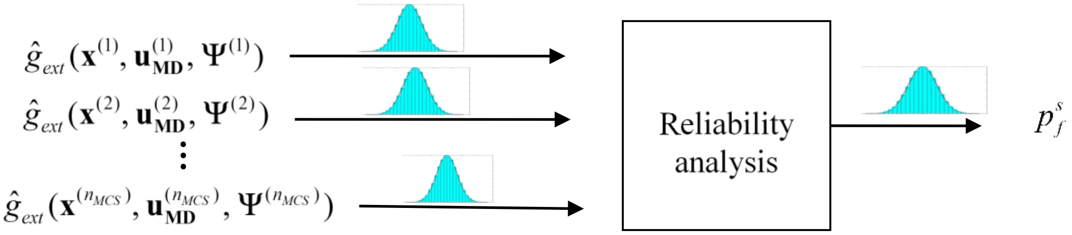

$\hat{g}_{ext}(\mathbf{x}^{(1)}, \mathbf{u}_{\mathbf{MD}}^{(1)}, \Psi^{(1)})$
$\hat{g}_{ext}(\mathbf{x}^{(2)}, \mathbf{u}_{\mathbf{MD}}^{(2)}, \Psi^{(2)})$
$\vdots$
$\hat{g}_{ext}(\mathbf{x}^{(n_{MCS})}, \mathbf{u}_{\mathbf{MD}}^{(n_{MCS})}, \Psi^{(n_{MCS})})$
Reliability analysis
p_f^s

Fig. 3.2. Effects of surrogate uncertainty on reliability analysis

(a) Correlation analysis of surrogate predictions

For any input $\boldsymbol{\phi}^{(j)} = [\boldsymbol{x}^{(j)}, \boldsymbol{u}_{\boldsymbol{MD}}^{(j)}, \boldsymbol{\psi}^{(j)}]$, the prediction from the surrogate follows a Gaussian distribution given by $\hat{g}_{ext}(\boldsymbol{\phi}^{(j)}) \sim \boldsymbol{N}(\mu_{\hat{g}_{ext}}(\boldsymbol{\phi}^{(j)}), \sigma_{\hat{g}_{ext}}(\boldsymbol{\phi}^{(j)}))$. Also, $\hat{g}_{ext}(\boldsymbol{\phi}^{(j)}), j = 1,2,3 \ldots n_{MCS}$, are correlated due to the covariance function assumed in a GP.

As indicated in Fig. 3.2, the uncertainty of the system failure probability estimate (the unconditional failure probability estimate given in Eq. (3.7)) due to surrogate uncertainty can be quantified by propagating the uncertainty in $\hat{g}_{ext}\left(\boldsymbol{\phi}^{(j)}\right), j = 1,2,3 \ldots n_{MCS}$, through Eq. (3.7). Since $\hat{g}_{ext}\left(\boldsymbol{\phi}^{(j)}\right), j = 1,2,3 \ldots n_{MCS}$, are n_{MCS} correlated random variables, their correlations are analyzed based on which a sampling-based method is developed for uncertainty quantification.

In MCS-based reliability analysis, n_{MCS} is usually large. Performing the correlation analysis for the n_{MCS} random variables is computationally expensive. In order to reduce the number of random variables $\hat{g}_{ext}\left(\boldsymbol{\phi}^{(j)}\right), j = 1,2,3 \ldots n_{MCS}$ are partitioned into two groups based on the probability that they may result in the error of failure probability estimate. The first group includes responses $\hat{g}_{ext}\left(\boldsymbol{\phi}^{(j)}\right), j = 1,2,3 \ldots n_{g_1}$, for which the probability of making an error in the sign of $I(\hat{g}_{ext}\left(\boldsymbol{\phi}^{(j)}\right))$ is very low (i.e., 0.001). Therefore, the mean predictions $\mu_{\hat{g}_{ext}}\left(\boldsymbol{\phi}^{(j)}\right)$ are used to substitute for $\hat{g}_{ext}\left(\boldsymbol{\phi}^{(j)}\right)$. The remaining responses in, $\hat{g}_{ext}\left(\boldsymbol{\phi}^{(j)}\right), j = 1,2,3 \ldots n_{MCS}$ form the second group, which are treated as random variables. The system failure probability given in Eq. (3.7) then becomes [82]

$$p_f^s = \sum_{j=1}^{n_{g1}} I(\mu_{\hat{g}_{ext}}(\boldsymbol{\phi}^{(j)}) \leq 0)/n_{MCS} + \sum_{j=1}^{n_{g2}} I(\mu_{\hat{g}_{ext}}(\boldsymbol{\phi}^{(j)}) \leq 0)/n_{MCS} \tag{3.8}$$

where n_{g1} and n_{g2} are the number of samples in the first and second groups respectively. The partition of $\hat{g}_{ext}\left(\boldsymbol{\phi}^{(j)}\right), j = 1,2,3 \ldots n_{MCS}$ is achieved based on the following function [39]

$$U_{AK}\left(\boldsymbol{\phi}^{(j)}\right) = \frac{|\mu_{\hat{g}_{ext}}(\boldsymbol{\phi}^{(j)})|}{\sigma_{\hat{g}_{ext}}(\boldsymbol{\phi}^{(j)})} \tag{3.9}$$

$U_{AK}\left(\boldsymbol{\phi}^{(j)}\right)$ represents the coefficient of variation of the model prediction, which can be used to estimate the probability of making an error in the sign of $\hat{g}_{ext}\left(\boldsymbol{\phi}^{(j)}\right)$. The first group of responses correspond to $U_{AK}\left(\boldsymbol{\phi}^{(j)}\right) \geq 3.1$ and the rest of the responses fall into the second group. Defining the training points in the current GP model as $\boldsymbol{\phi}^{\boldsymbol{s}}$ and $g_{ext}(\boldsymbol{\phi}^{s})$, the covariance matrix of $\hat{g}_{ext}(\boldsymbol{\phi}^{(\boldsymbol{k})})$, $k = 1,2,3 \ldots n_{g2}$ conditioned on the training points is given by

$$\boldsymbol{\Sigma}_{\boldsymbol{p}|\boldsymbol{t}} = \boldsymbol{\Sigma}_{\boldsymbol{pp}} - \boldsymbol{\Sigma}_{\boldsymbol{pt}} \boldsymbol{\Sigma}_{\boldsymbol{tt}}^{-\boldsymbol{1}} \boldsymbol{\Sigma}_{\boldsymbol{pt}}^{\boldsymbol{T}} \tag{3.10}$$

where $\boldsymbol{\Sigma}_{\boldsymbol{pp}}$, $\boldsymbol{\Sigma}_{\boldsymbol{pt}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{tt}}$ are the covariance matrixes between $\hat{g}_{ext}(\boldsymbol{\phi}^{(j)})$ and $\hat{g}_{ext}(\boldsymbol{\phi}^{(j)})$, $\hat{g}_{ext}(\boldsymbol{\phi}^{(j)})$ and $g_{ext}(\boldsymbol{\phi}^{s})$, $g_{ext}(\boldsymbol{\phi}^{s})$ and $g_{ext}(\boldsymbol{\phi}^{s})$ respectively. Based on the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{p}|\boldsymbol{t}}$, the conditional correlation matrix $\boldsymbol{\rho}_{\boldsymbol{p}|\boldsymbol{t}}$ of $\hat{g}_{ext}(\boldsymbol{\phi}^{(\boldsymbol{k})})$, $k = 1,2, \ldots n_{g2}$ is equal to $\boldsymbol{\rho}_{\boldsymbol{ij}}$, which represents the correlation between $\hat{g}_{ext}(\boldsymbol{\phi}^{(i)})$, and $\hat{g}_{ext}(\boldsymbol{\phi}^{(j)})$, $i, j = 1,2, \ldots n_{g2}$, conditioned on current training points.

(b) Propagation of surrogate prediction uncertainty

After obtaining the correlation matrix, the sampling-based method can be used to propagate the surrogate prediction uncertainty of $\hat{g}_{ext}(\boldsymbol{\phi}^{(j)})$ to the uncertainty in p_f^s. To do this, samples of $\hat{g}_{ext}(\boldsymbol{\phi}^{(\boldsymbol{k})})$, $k = 1,2, \ldots n_{g2}$ are generated using the following expression [82]:

$$\hat{g}_{ext}(\boldsymbol{\phi}^{(k)}) = \mu_{\hat{g}_{ext}}(\boldsymbol{\phi}^{(k)}) + \sigma_{\hat{g}_{ext}}(\boldsymbol{\phi}^{(k)}) \sum_{j=1}^{n_{g2}} \zeta_j \, \boldsymbol{\phi}_j^T \rho_{:i} / \sqrt{\eta_j} \tag{3.11}$$

Where $\zeta_j, j = 1,2,3 \ldots n_{g2}$ are independent standard Gaussian variables; and η_i and $\boldsymbol{\phi}_{\boldsymbol{i}}^{\boldsymbol{T}}$ are the eigenvalues and eigenvectors of $\boldsymbol{\rho}_{\boldsymbol{p}|\boldsymbol{t}}$ and $\boldsymbol{\rho}_{:\boldsymbol{i}} = \left[\boldsymbol{\rho}_{\boldsymbol{i1}}, \ \boldsymbol{\rho}_{\boldsymbol{i2}}, \ \boldsymbol{\rho}_{\boldsymbol{i3}} \ldots . \boldsymbol{\rho}_{\boldsymbol{in}_{\boldsymbol{g2}}}\right]^{\boldsymbol{T}}$. Since the surrogate prediction at each input is a random variable, N_{sim} samples are generated for each $\hat{g}_{ext}(\boldsymbol{x}^{(k)})$, $k = 1,2 \ldots n_{g2}$, resulting in the following sampling matrix:

$$g_{N_{sim} \times n_{g2}} = \{g_{ext}(i, j)\}, \forall \, i = 1,2 \ldots N_{sim}; j = 1,2,3 \ldots n_{g2} \tag{3.12}$$

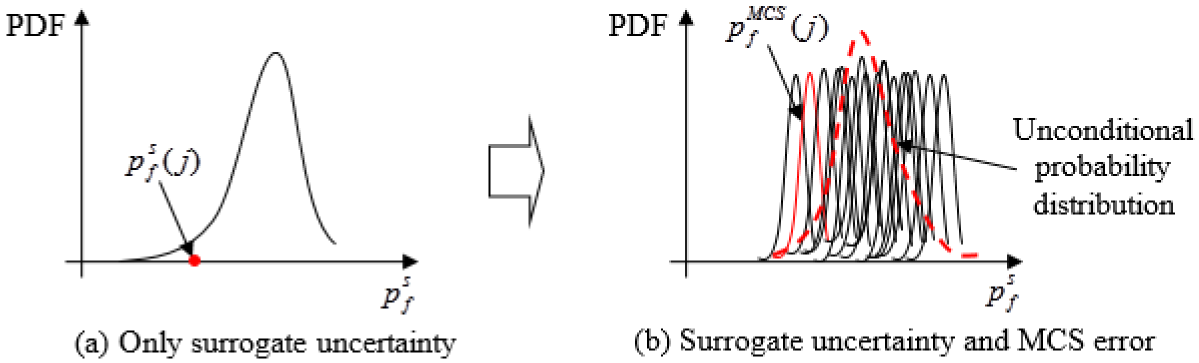
Using the samples of $\hat{g}_{ext}(\boldsymbol{x}^{(\boldsymbol{k})})$, $k = 1,2 \ldots n_{g2}$ and Eq. (3.11) samples of $\boldsymbol{p}_{\boldsymbol{f}}^{\boldsymbol{s}}$ are obtained as

$$p_f^s(j) = \frac{\left\{\sum_{i=1}^{n_{g1}} I\left(\mu_{g_{ext}}\left(\boldsymbol{\phi}^{(i)}\right)\right) + \sum_{i=1}^{n_{g2}} I\left(g_{ext}(i,j)\right)\right\}}{n_{MCS}}, j = 1,2,3 \ldots N_{sim} \tag{3.13}$$

where $p_f^s(j)$ is the j^{th} sample of p_f^s due to the surrogate prediction uncertainty. Using the samples of $p_f^s(j), j = 1,2,3 \ldots N_{sim}$, a PDF can be constructed for the system failure probability p_f^s. This distribution represents the uncertainty in p_f^s due to surrogate uncertainty.

B. Inclusion of MCS error in reliability estimate

Using Eq. (3.13), several samples of $\boldsymbol{p_f^s}$ are obtained through correlated sampling of the model predictions at several inputs. As discussed in Sec. 2.2.3, there exists an uncertainty in the estimation of each failure probability sample, $p_f^s(j), j = 1,2,3 \ldots N_{sim}$ due to the limited number of Monte Carlo samples (referred to here as MCS error), which results in each $p_f^s(j)$ being a random variable. To avoid any confusion in the notation, the failure probability when MCS error is also considered is denoted as $p_f^{MCS}(j)$. Following the discussion on MCS error in Section 2.2.3, we can construct the PDF of $p_f^{MCS}(j)$ by estimating its quantiles using $p_f^s(j)$, number of samples n_{MCS} and degree of accuracy $1-\gamma$ as $\left[p_f^s(j) + \frac{z_{\frac{\gamma}{2}}^2}{2 \times n_{MCS}} \pm \sqrt{\frac{p_f^s(j)\,(1-p_f^s(j))}{N_s} + \frac{z_{\frac{\gamma}{2}}^2}{4 \times n_{MCS}^2}}\right] \times \frac{1}{1+\frac{z_{\frac{\gamma}{2}}^2}{n_{MCS}}}$, where $z_{\frac{\gamma}{2}}$ represents the $1-\frac{\gamma}{2}$ quantile of a standard normal distribution. After including surrogate uncertainty and MCS error, p_f^s is represented by a family of PDFs (Fig. 3.3 (b)). After the inclusion of surrogate uncertainty, the failure probability is represented using a PDF as shown in Fig. 3.3 (a). $\boldsymbol{p_f^s}$ is a sample from this PDF (shown as a red dot in Fig 3.3 (a)). When MCS error is also considered, $\boldsymbol{p_f^s}$ is not a single value but a PDF denoted as $\boldsymbol{p_f^{MCS}}$ (shown as a continuous red curve in Fig 3.3 (b)). Similarly, each sample from the PDF in Fig 3.3 (a) corresponds to a different PDF in Fig 3.3 (b). Thus, the failure probability is represented as a family of PDFs in the presence of surrogate uncertainty and MCS error. This family of PDFs can then be integrated to an unconditional PDF (bold broken red curve in Fig. 3.3(b)) using the auxiliary variable approach described in Section 3.1. The family of PDFs for the failure probability estimate (Fig. 3.3(b)) can be treated similar to the family of PDFs for an input variable and uncertain distribution parameters. Note that in the case of an input variable, the auxiliary variable represents its aleatory uncertainty whereas in the case of the failure probability estimate, it represents the epistemic uncertainty due to limited Monte Carlo samples. The single loop sampling approach used for sampling the input variables can also be used for sampling the failure probability estimates.



(a) Only surrogate uncertainty

(b) Surrogate uncertainty and MCS error

Fig. 3.3. Uncertainty in failure probability due to surrogate uncertainty and MCS error

Assume that there are N_{sim} PDFs of the failure probability (i.e., N_{sim} samples of p_f^s), N_{sim} samples of the auxiliary variable U_{MCS} are generated in the interval [0,1]. Here, the auxiliary variable is used to represent the contribution of epistemic uncertainty of MCS error to the overall uncertainty in the reliability estimate. The reliability estimate without the inclusion of surrogate uncertainty and MCS errors is a point-value (single-value). However, when surrogate uncertainty and MCS errors are included, the reliability estimate is represented through a family of PDFs as shown in Fig. 3.3(b). As discussed in Section 2.2.3, the reliability estimate is represented through a PDF with some distribution parameters. Here, these parameters are calculated after incorporating aleatory uncertainty, statistical uncertainty in input variables, uncertain model parameters, model discrepancy and surrogate uncertainty. Therefore, for each realization of distribution parameters, the reliability estimate follows a PDF.

For one realization of the auxiliary variable and p_f^s, one realization of the failure probability estimate from its unconditional PDF is obtained. For a given value of p_f^s, there exists an associated PDF due to MCS error which is numerically estimated after obtaining the quantiles as stated above. Denoting the generated samples as $u_{MCS}^{(1)}$, $u_{MCS}^{(2)} \dots u_{MCS}^{(N_{sim})}$, the samples of unconditional failure probability estimate can be obtained as

$$p_f^{UN}(j) = F_{MCS}^{-1}\left(u_{MCS}^{(j)}\middle|p_f^s(j)\right), j = 1,2,3..N_{sim} \tag{3.14}$$

where $\boldsymbol{F_{MCS}^{-1}}(.)$ is the inverse CDF of the MCS error. Based on the generated samples of $\boldsymbol{p_f^{UN}(j)}$, the unconditional PDF of the failure probability can be estimated. This PDF represents the uncertainty in the failure probability estimate due to both surrogate prediction uncertainty and MCS error.

3.3. Example 1: Mechanical beam

Consider a short cantilever beam subjected to a point load at its free end as shown in Fig. 3.4. Two limit states – maximum deflection and maximum stress are considered for the failure of the beam. The goal in this problem is to compute the reliability with respect to each of the individual limit states and the system reliability, considering both the limit states.

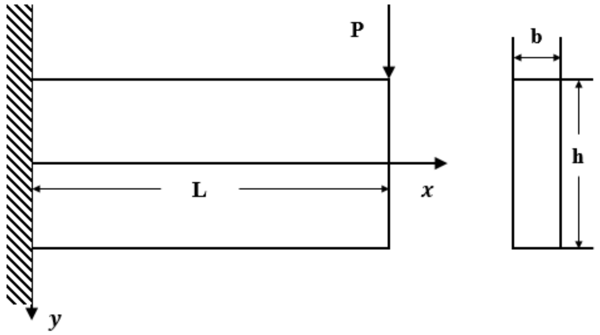


Fig. 3.4. A cantilever beam with point-load at the free end

Assume that the free end deflection of the beam due to the point load is modeled (according to the Euler-Bernoulli beam theory) as $d(P) = \frac{PL^3}{3EI}$. Here, E and I represent the Young's modulus and moment of inertia respectively. Since the beam is short, there exists a discrepancy between the experimental observations and predicted deflections computed (because the Euler-Bernoulli model is not accurate for short beams). This model discrepancy is calibrated using the KOH framework. A total of 30 points were generated through Latin Hypercube Sampling (LHS) across the domain of inputs, at which both simulation and experimental data are available. Among them 25 points were used for calibration (model parameters and model discrepancy using the KOH framework) and 5 points were used for validation. The maximum COV (coefficient of variation) among the testing points is 0.1134. Therefore, the deflection of the beam, after accounting for model discrepancy $\delta(P)$, is given as

$$d(P) = \frac{PL^3}{3EI} + \delta(P) \tag{3.15}$$

The expression for the computation of maximum stress is given as

$$s(P) = \frac{PLh}{2I} \tag{3.16}$$

The load P is assumed to be aleatory with uncertain distribution type and distribution parameters. E and L are aleatory following Gaussian distributions with known distribution parameters. The cross-section parameters (b, h) are also assumed to be aleatory with known parameters due to geometric variations in the manufacturing process. Table 3.1 shows the random variables and their statistics used in this example, and Table 3.2 gives two candidate distribution types of P and their corresponding probabilities.

The two limit state functions corresponding to deflection and stress are given as

$$g_d(P) = d_0 - d(P) \tag{3.17}$$

$$g_s(P) = s_0 - s(P) \tag{3.18}$$

where $g_d(P) < 0$ and $g_s(P) < 0$ indicate failure, and s_0, d_0 are the limiting values of stress and deflection. In this example, the threshold values for stress and deflection are assumed deterministic (d_0 = 0.01 m, s_0 = 500 MPa). Let the models in Eqs. (3.17, 3.18) be known as simulation models for deflection and stress respectively.

Table 3.1. Example 1. Variables and their statistics

Parameters		Distribution	Mean	Standard deviation
P	$\mu_P (\times 10^5\ N)$	Normal	35	0.3
	$\sigma_P (\times 10^5\ N)$	Lognormal	4	0.1
$E (\times 10^9\ N/m^2)$		Normal	210	10
$L(m)$		Normal	2	0.01
$b(m)$		Normal	0.18	0.005
$h(m)$		Normal	0.75	0.005

Table 3.2. Distribution types and their probabilities for Load *P*

Distribution type	Normal	Type 1 EVD
Probability	0.2	0.8

The system fails when either $g_d(P) < 0$ or $g_s(P) < 0$, and the system failure probability is given by

$$p_f^s = \Pr(g_d(P) < 0 \ \cup \ g_s(P) < 0) \tag{3.19}$$

(a) Component reliability analysis

The reliability analysis with respect to each of the individual limit states is first performed using the proposed method. Since the deflection limit state is associated with model discrepancy, the reliability analysis is performed with and without discrepancy, in order to investigate the effectiveness of the proposed method in handling model discrepancy during surrogate construction.

Tables 3.3 and 3.4 show the reliability analysis results, along with the number of function evaluations (NOF), without considering the surrogate uncertainty and MCS error, in order to compare with the simulation model estimate. In this example, the threshold EFF value for selection of training points in surrogate construction is assumed to be 0.002. The results show that the proposed method can comprehensively estimate the component reliability in the presence of various sources of epistemic uncertainty. The results illustrate that the reliability estimate can be improved by considering the model discrepancy (Table 3.3).

Table 3.3. Example 1. Component failure probability results (Deflection)

	p_f	NOF	ε (%)
Simulation model	0.0249	1x10^6	-
Limit state surrogate without model discrepancy	0.0057	25+0	77.11
Limit state surrogate with model discrepancy	0.0248	25+2	0.4

Note: The NOF include the initial number of samples from calibration (25) and the number of added new samples.

Fig. 3.5 shows the comparison between the simulation model limit state, and the limit-state from surrogate with and without consideration of model discrepancy. Since it is not possible to show the limit state contours with all the random variables, Fig. 3.5 shows the contours between length and load on the X and Y axes respectively. The other random variables E, b and h are conditioned at their mean values. It shows that the surrogate constructed taking model discrepancy into consideration is closer to the true limit state than the surrogate without considering the model discrepancy.

Fig. 3.6 provide the failure probability distribution with respect to each limit state after considering both the surrogate uncertainty and MCS error. The figures illustrate that the proposed method can effectively quantify the uncertainty in the reliability analysis results.

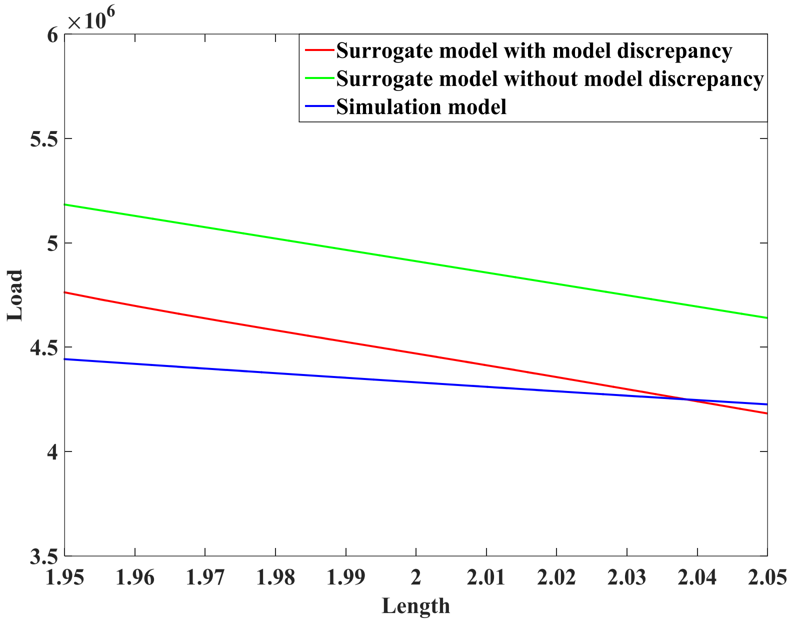


Fig. 3.5. Comparison of limit states between the simulation model and surrogate, with and without considering model discrepancy

Table 3.4. Example 1. Component failure probability results (Stress)

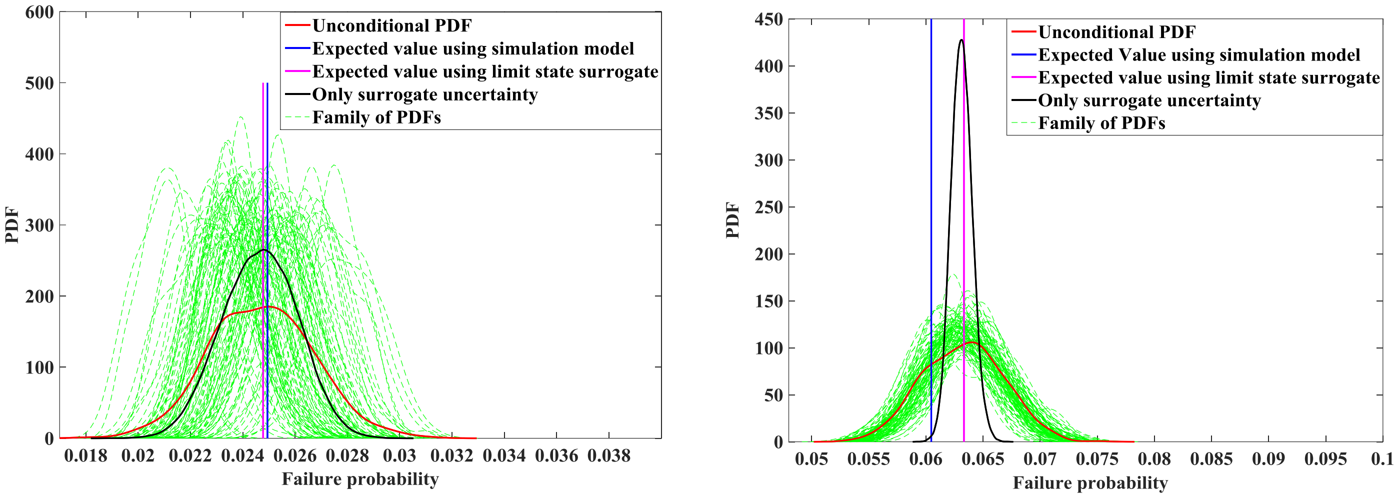
	p_f	NOF	ε (%)
Simulation model	0.0605	1×10^6	-
Limit state surrogate model	0.0633	32	4.63

(b) System reliability analysis

A surrogate is constructed considering both the limit states for system reliability analysis using the method discussed in Section 3.2. Fig. 3.7 gives the PDF of failure probability after considering both the deflection and stress limit states. It shows that the proposed method can effectively quantify the uncertainty in the system failure probability estimate.

Table 3.5. Example 1. Failure probability estimates using MCS and proposed method

	p_f^s	NOF	ε (%)
Simulation model	0.0625	1×10^6	-
Limit state surrogate model	0.0648	276	3.68



(a) (b)

Fig. 3.6. Failure probability with respect to (a) deflection, and (b) stress

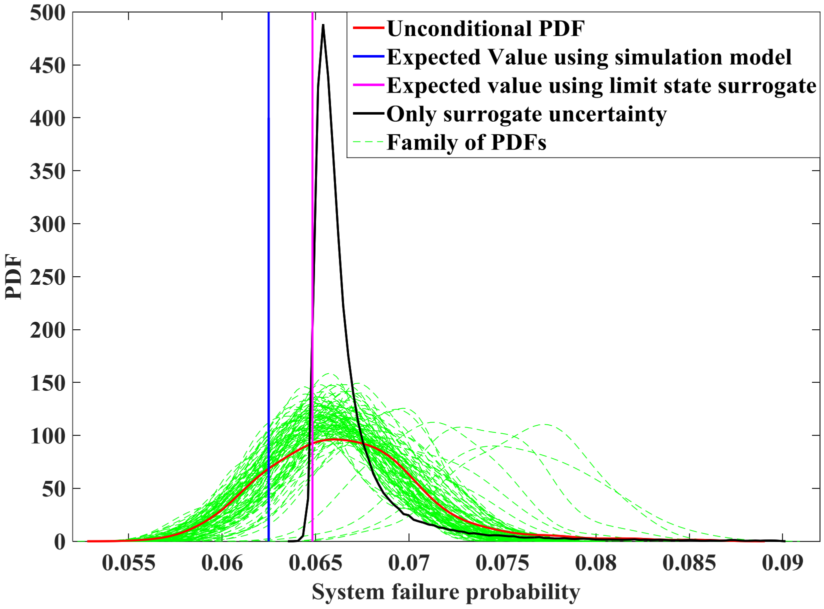


Fig. 3.7. System failure probability with respect to both deflection and stress limit states

3.4. Example 2: A two-bar system

Example 1 demonstrated the proposed method for component reliability analysis and reliability analysis of a series system. In this example, reliability analysis of a parallel system is demonstrated. Consider the two-bar system shown in Fig. 3.8. The system consists of two bars supporting a panel on which a load P is applied at its center.

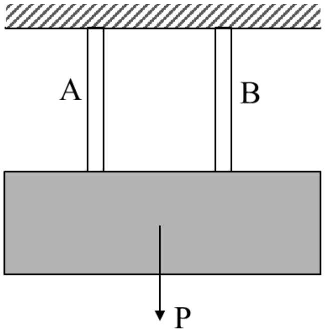


Fig. 3.8. A two-bar system

The two bars are assumed to be made of different materials with different failure stress characteristics. The Young's moduli of the materials (E_A, E_B) are assumed to be aleatory variables with known distribution types and distribution parameters. For illustration, the bars are assumed to be of equal length (L= 1 m) and the area of cross-section of the two bars are assumed to be deterministic (A_A= 0.04 m^2, A_B= 0.0625 m^2). The forces in the bars are given by

$$F_i = P \times \frac{E_i A_i}{E_A A_A + E_B A_B}, i = A, B \tag{3.20}$$

Let σ_A^o and σ_B^o represent the failure stresses of the bars; the limit state functions are given by

$$g_i(X) = A_i \sigma_i^o - P \times \frac{E_i A_i}{E_A A_A + E_B A_B}, i = A, B \tag{3.21}$$

The applied load is assumed to aleatory, but with uncertain distribution type and uncertain distribution parameters. The failure stress is assumed to be deterministic but not known precisely. Some point and interval data are assumed to be available from material testing. Table 3.6 provides the list of variables used in this example and their statistics. Using the available data, non-parametric distributions are constructed for both the threshold stresses using the likelihood approach discussed in Section 2.1, and spline-based interpolation is employed for the PDF modeling. The failure stress of a material is not calibrated. We assume that data about failure stress is directly available from material testing and we construct a non-parametric distribution to represent the uncertainty. After constructing the PDF, it is treated as any other input random variable for reliability analysis.

Table 3.6. Example 2. Variables and their statistics

Variable		Distribution	Mean	Standard deviation
Load P	$\mu_P (\times 10^6\ N)$	Normal	19	0.3
	$\sigma_P\ (\times 10^6\ N)$	Lognormal	1.6	0.1
Young's Modulus $E_A (\times 10^9\ Pa)$		Normal	210	20
Young's Modulus $E_B (\times 10^9\ Pa)$		Normal	180	15
Failure stress of bar A, σ_{A}^{0} $(\times 10^7$ Pa)		Point data – [24.7, 24.95,25.1,25.3] Interval data – [(24.6, 24.62), (24.8, 24.84), (25.1, 25.15)]		
Failure stress of bar B, σ_{B}^{0} $(\times 10^7$ Pa)		Point data – [21.8, 21.92,22.08,22.15] Interval data – [(21.85, 21.88), (22.06, 22.08), (22.13, 22.16)]		

For illustration purposes, the same candidate distribution types for load $\boldsymbol{P}$ and their probabilities are used as in Table 3.2. The system described here is a parallel system and therefore failure occurs when both the bars fail i.e. $g_A(X) < 0$ and $g_B(X) < 0$. The system failure probability is given by

$$p_f^s = \Pr(g_A(X) < 0\ \cap\ g_B(X) < 0) \tag{3.22}$$

Table 3.6 gives the system failure probability analysis results after consideration of statistical uncertainty, model discrepancy and uncertain model parameters, and without considering surrogate uncertainty and MCS error. Similar to the previous example, the threshold EFF value for selection of training points in surrogate construction is assumed to be 0.002. Fig. 3.9 shows the system failure probability estimates without considering surrogate uncertainty and MCS error, considering only surrogate uncertainty and considering surrogate both uncertainty and MCS error. It can be seen that the surrogate uncertainty and MCS error result in larger uncertainty in the reliability estimate compared to considering only surrogate uncertainty. In order to reduce the uncertainty in the estimate, the surrogate needs to be further refined by adding more training points and the number of Monte Carlo samples need to be increased to reduce the MCS error.

Table 3.7. System Failure probability estimates using simulation model and limit state surrogate

	p_f^s	NOF	Error (%)
Simulation model (Euler model + model discrepancy)	0.0048	6×10^5	-
Limit state surrogate model	0.0049	302	2.08

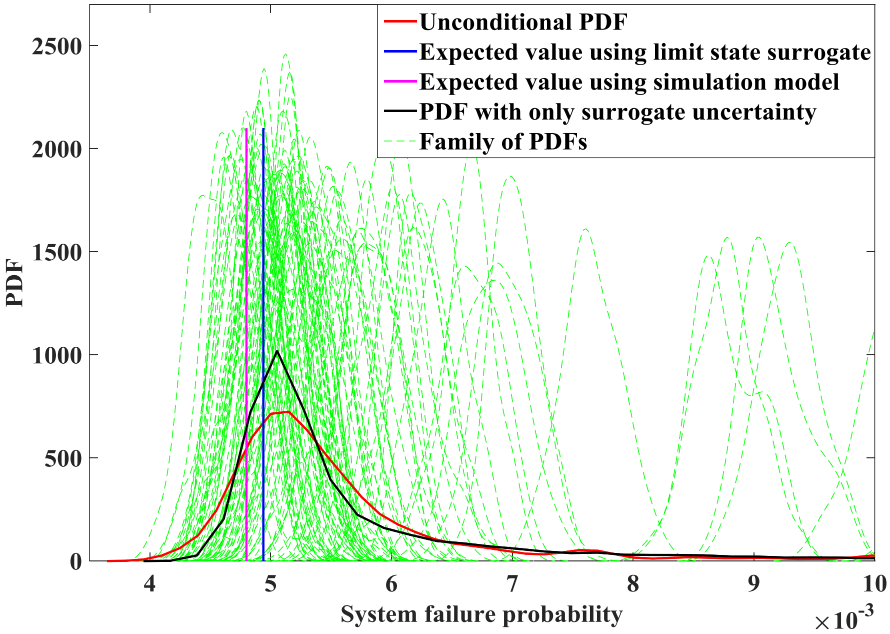


Fig. 3.9. System failure probability estimate using simulation model and limit state surrogate

3.5. Summary

This chapter proposed a unified framework to quantify the uncertainty in the reliability estimate due to the incorporation of different types of epistemic uncertainty. Epistemic uncertainty due to data (statistical uncertainty) and model (model parameter uncertainty, model discrepancy, surrogate uncertainty, Monte Carlo sampling error) are considered. A parametric approach is proposed for the quantification of input variables with statistical uncertainty. Non-parametric distributions are used to quantify uncertain model parameters with statistical uncertainty. First, model calibration is carried out using the KOH framework, whose output include GP surrogates for simulation model and model discrepancy. The general purpose (global) GP surrogate of simulation model is further refined by adding more training points close to the limit state (local refinement) to obtain the limit state surrogate for reliability analysis; this is referred to as a hybrid approach (local refinement of the general purpose GP surrogate). An auxiliary variable is used to represent the model discrepancy, which enables the construction of a limit state surrogate that includes the model discrepancy. The inputs for the limit state surrogate are the original input variables along with the uncertain model parameters and the model discrepancy. The selection of training points is carried out using a learning function called the Expected Feasibility Function (EFF). The GP model is then used to carry out reliability analysis using Monte Carlo sampling. A single-loop sampling approach using an auxiliary variable is used for the sampling of input variables with statistical uncertainty and samples of the inputs are generated and passed through the surrogate to obtain the reliability estimate. Note that this results in a single value of the failure probability.

The model prediction using a GP model is a Gaussian distribution with the parameters dependent on the input. In addition, the model predictions at different inputs are correlated due to the covariance function in the GP model. To account for the surrogate uncertainty (i.e., the variability in the prediction), correlated sampling of model predictions at several inputs is carried out and used for reliability analysis. Inclusion of surrogate uncertainty results in a probability distribution (PDF) for the failure probability. When MCS error is incorporated, each sample in the failure probability distribution is represented by a PDF, which results in a family of PDFs for the failure probability. Two examples demonstrated the effectiveness of the proposed method. The proposed method is able to not only address heterogeneous sources of epistemic uncertainty during system reliability analysis, but also provide the uncertainty associated with the reliability estimate.

This chapter considered performance assessment of single-component systems; the next chapter considers performance assessment of multi-level systems, which could be considered as a collection of multiple components.

CHAPTER 4

PERFORMANCE ASSESSMENT IN MULTI-LEVEL SYSTEMS

4.1. Introduction

Multi-level systems represent systems composed of several components (subsystems) organized in a hierarchical manner. An example of a multi-level system is a manufacturing process shown in Fig. 1.2. In such systems, estimation of a system-level QoI requires aggregation of uncertainty from several subsystems at multiple levels. The focus in this chapter is to develop a methodology for performance assessment of multi-level systems using Hierarchical Bayesian Networks [9], which are an extension to Bayesian networks. A brief introduction to a BN and an HBN is provided in Section 2.6. HBNs offers several advantages for modeling, analysis and visualization of multi-level systems. Learning of an HBN in its entirety can be computationally expensive as the system consists of several subsystems and therefore, may have a large number of variables. When learning algorithms such as score-based methods in Section 2.6.2, are used to learn the entire HBN, they may not provide accurate dependence relationships as they output locally optimum models. Therefore, in its chapter, we propose a segmented learning framework, where the learning of a large HBN is divided into several small subsystem-level BNs; this makes the learning more efficient.

As mentioned in Section 2.6.1, the BNs for several subsystems can be constructed using physics-based models or learnt from available data using BN learning algorithms or by their combination (hybrid approach). Section 1.1 detailed the issues relating to learning in hybrid Bayesian networks such conditional linear Gaussian assumption [5] and discretization of continuous variables. In this work, an improvement to the existing algorithm is presented which eliminates discretization challenges and Gaussian assumption in learning BNs.

One advantage of using a BN is that it encodes the expert domain knowledge into the model. Experts, in general, may have a wealth of knowledge on a subsystem that they have experience with and therefore can support in building a BN associated with that unit process. A HBN approach allows the integration of expert knowledge from multiple domains in modeling the multi-level. HBNs allow modeling at different resolutions depending upon the analysis requirements. As a result, HBNs provide better visualization of the models.

In a multi-level system, data for calibration might be available at multiple levels. As the complexity of the multi-level system increases, the dimension of the HBN also increases, in terms of the number of calibration parameters. Estimating all the unknown parameters using the HBN with the entire data together becomes computationally expensive. Therefore, a segmented approach for model calibration is proposed here for estimating the unknown parameters at multiple levels. A segmented approach for model calibration has been studied earlier for multi-physics problems [83, 84]. In this work, the segmented approach is expanded to multi-level systems that include a mixture of discrete and continuous variables, in the presence of mathematical models, expert knowledge and observation data in different segments of the manufacturing network.

The complexity of the model increases with the number of variables/parameters used in the model. Therefore, dimension reduction techniques have been studied to reduce the complexity of the model by eliminating the variables that do not significantly influence the model prediction. Nannapaneni et al [85] demonstrated a variance-based global sensitivity analysis approach for dimension reduction in a single-component BN. In this work, a multi-level sensitivity analysis approach is developed for hierarchical systems, thus enabling dimension reduction and improving the scalability and affordability of the analysis. The proposed methodologies are demonstrated for an injection-modeling process.

4.2. Proposed methodology

HBNs, as stated in Section 2.6.3, are extensions of BNs for modeling hierarchical systems, i.e. a HBN is a BN where each node may represent a lower level BN. Another interpretation of a HBN is that it is a fusion of BNs at different levels of hierarchy. The BNs at different levels can be individually learnt from either available physics-based models or data. An improved method for learning hybrid Bayesian network is presented first. After learning the BNs at different levels in the hierarchy, composition of the BNs to form the HBN is pursued. Then methods for model updating and dimension reduction are discussed.

4.2.1 Learning Bayesian networks

As stated in Section 2.6.2, the learning of a BN consists of structure learning (topology) and parameter learning (conditional probability distributions). The proposed refinement belongs to the parameter-learning task, in the estimation of conditional probability distributions.

The topology of a Bayesian network with discrete and continuous variables can be learnt by first discretizing the continuous variables using Gaussian mixture models (GMM) [86] using the algorithm presented in [7], which uses a score-based method for learning. During the learning process, every continuous node is represented as a two-component GMM, where the statistics (mean and variance) of the two normal distributions are estimated from the data. As a pre-processing step to the learning procedure, data is clustered into two groups using the k-means clustering algorithm [87]. For data in each group, a Gaussian distribution is fit by estimating the mean and variance of the data. The learning algorithm is also able to use any prior knowledge about the ordering of the nodes. In addition, any available information on a possible set of parent nodes for a child node can also be used. Using such information from domain experts makes the learning of the BN faster and more accurate.

The output of the learning procedure provides the topology of the BN, conditional probability tables for the discrete nodes and conditional probability distributions for the continuous nodes. The conditional probability tables of the discrete nodes are exact since no approximations are made in their representation. The conditional probability distributions of the continuous nodes are not exact but approximate due to their representation as two-component GMMs. The proposed refinement is in estimation of the conditional probability distributions of continuous nodes, in particular, the following three cases – (1) continuous node, with all continuous parent nodes, (2) continuous node, with parent nodes, which are a collection of discrete and continuous nodes, and (3) Continuous nodes with all discrete parent nodes.

One interpretation of a conditional probability distribution is that it is a stochastic model connecting the parent nodes to the child node. Therefore, the proposed method is to fit parametric or non-parametric models using the available data and topology; these models can then be used as conditional probability distributions.

Consider case 1 – continuous child node X_c with continuous parent nodes $Pa(X_c)$. Due to the possibility of many parametric and non-parametric models, the following models are tried and the best one is chosen – (1) Gaussian conditional probability, where the mean is a linear function of parent nodes and with constant but unknown variance, $f(X_c|Pa(X_c)) \sim N\left(f^1\left(Pa(X_c)\right), \sigma^2\right)$ where f^1 represents a linear function and σ^2 represents the variance; (2) Gaussian conditional probability, where the mean is a quadratic function of parent nodes and with constant variance, i.e., $f(X_c|Pa(X_c)) \sim N\left(f^2\left(Pa(X_c)\right), \sigma^2\right)$; and (3) a Gaussian Process model, i.e., $f(X_c|Pa(X_c)) \sim GP(Pa(X_c))$. The parameters of the above models are estimated from the data. The first two models are parametric models with variance independent of the parent nodes whereas the third model is non-parametric and the variance is dependent on the values of the parent nodes. The three models are in increasing order of complexity (and computational expense). The selection of the best-fit model can be based on quantitative model selection techniques such as residual mean squared error (RMSE), Akaike information criterion (AIC), and Bayesian Information criterion (BIC) [88].

For case 2, where the parent node set consists of both discrete and continuous nodes, one of the above three parametric and non-parametric models can be fit to every combination of the discrete nodes, i.e., for every combination of the parent discrete nodes, a parametric or a non-parametric model is constructed for the continuous child node conditioned on the continuous parent nodes. Let $Pa(X_c) = Pa(X_c)^D + Pa(X_c)^C$, where $Pa(X_c)^D, Pa(X_c)^C$ represent the sets of discrete parent nodes and continuous parent nodes. Then the conditional probability distribution can be represented as

$$f(X_c|Pa(X_c)) \sim \sum_{\nu} \delta(Pa(X_c)^D - \nu) M_\nu (Pa(X_c)^C) \tag{4.1}$$

where $\delta(.)$ represents the Dirac delta function, ν represents a possible combination of discrete parent nodes $Pa(X_c)^D$, and M_ν represents any of the three parametric or non-parametric models for X_c dependent on the continuous parent nodes $(Pa(X_c)^C)$.

For case 3, when the continuous child node is dependent on only discrete parent nodes, the child node assumes a different probability distribution for every combination of the discrete parent nodes. Flexible parametric distributions families such as Johnson, beta etc. [89] or non-parametric distributions can be fit to the available data [24], for every combination of the discrete parent nodes. The key idea of this learning algorithm is to eliminate the approximations made in existing learning algorithms in representation of the continuous variables.

4.2.2 Construction of hierarchical Bayesian networks

Here, construction of the HBN corresponding to a multi-level system is discussed. In addition, an automated procedure for the construction of HBN is described.

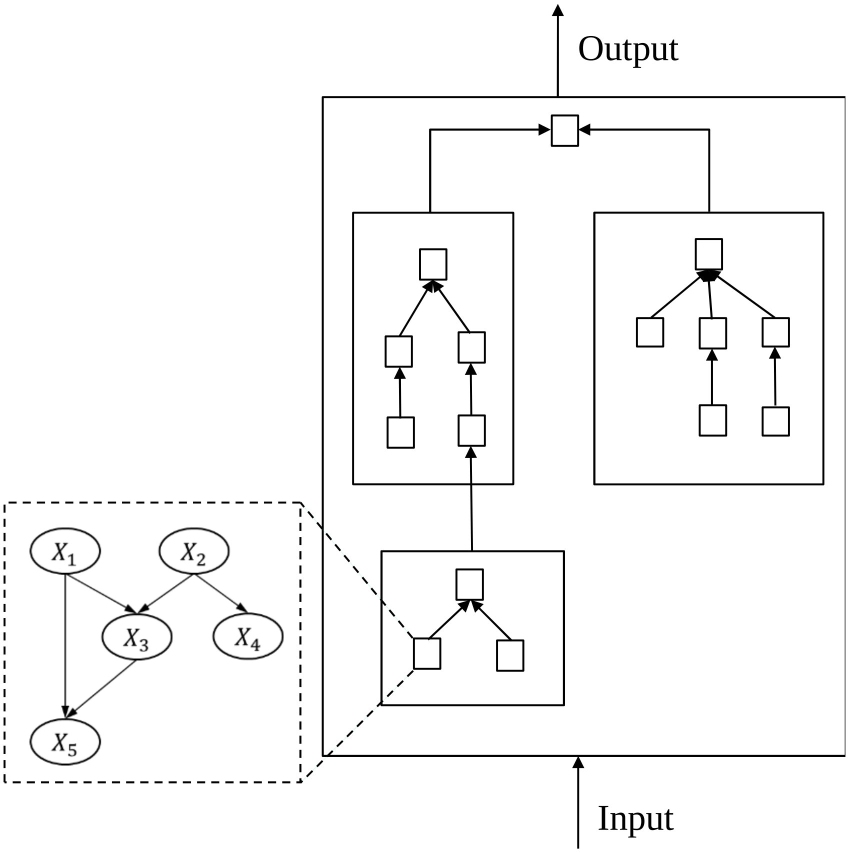


Fig. 4.1. Hierarchy in a multi-level process

BNs at multiple levels can be constructed either using physics-based models or available data using BN learning algorithms (as shown in Fig. 4.1). In cases when physics-based models are unavailable, the BNs are constructed using available data following the procedure in Section 4.2.1, which are usually specific to that process. As stated in Section 4.1, the topology of the multi-level system can be used to connect the BNs for individual subsystems, to obtain a preliminary HBN. In some cases, when multiple subsystems are composed, unknown dependences may be observed. (For example, consider two simple systems – a heating system and a pipe to transport a flammable liquid. The two systems when separated by a large distance may function properly but when the heating system is brought close to the pipe, the pipe gets heated, thereby heating up the flammable liquid and eventually causing fire).

The BN learning procedure is used to quantify the unknown dependences across several subsystems. The HBN constructed by composing the individual BNs following the topology of multi-level system is used as a prior network for the BN learning algorithm. When learning the unknown dependencies, the dependencies within each subsystem are fixed and any dependencies across subsystems are learnt. It is assumed that all the dependencies within a subsystem are characterized when learning the individual BNs. Using the operational data for the entire multi-level system; the unknown dependences across individual subsystems are learnt. In summary, the HBN for a multi-level system can be constructed in three steps – (1) learning BNs for individual subsystems, (2) constructing an initial HBN following the topology of the multi-level system, and (3) learning the unknown dependences across several subsystems using the BN learning algorithms.

Automation: BNs built for individual subsystems can be stored in a model library in an exchangeable format such as JavaScript Object Notation (JSON) [90], and can be imported from the model library wherever that subsystem is used. In the case of BN learning, the outputs (nodes and their conditional probability distributions) are stored in the JSON format. Thus, all the JSON representations of BNs at different levels can be fused together to obtain a JSON representation of the entire system or process. An algorithm can then be developed to automate the conversion of the JSON representations of all the variables into a corresponding Bayesian network. In this work, a Python script is created to convert the JSON representation into a BN, using the PyMC module [91] in Python, and the constructed BN is then used for further UQ analysis. The unknown dependences depend on the topology of the network. Therefore, after automated construction the HBN following the network topology, the unknown dependences are quantified. Note that only the BNs for the individual subsystems or components (constructed through physics-models or learning algorithms) may be stored in the JSON format and *not* the entire HBN, because the unknown dependences are configuration-dependent.

4.2.3 Multi-level segmented model calibration

Typically, the data for model calibration may be available at multiple levels in a system. Using the entire available data to estimate all unknown parameters becomes unnecessarily expensive (in terms of computation). A more suitable method for model calibration is by adopting a multi-level segmented approach where data in each lower-level BN is used to obtain the posterior distributions of the nodes in that BN; these posterior distributions are used as prior distributions for re-calibration of the nodes that go into higher-level BNs using the corresponding data. The remaining nodes in the lower-level BN can also be re-calibrated if necessary, based on passing the information down the hierarchy. This multi-level calibration approach is explained using the HBN shown in Fig. 2.4.

Let the calibration parameters include $S_{11}, S_{12}, S_{21}, S_{22}, S_{23}$. Since S_{11}, S_{12} and S_{21}, S_{22}, S_{23} correspond to parameters of subsystems at different levels, by using the independence assumption, their joint prior distribution $\Pi(S_{11}, S_{12}, S_{21}, S_{22}, S_{23})$ can be decomposed into a product of $\Pi(S_{11}, S_{12})$ and $\Pi(S_{21}, S_{22}, S_{23})$. For convenience, let $S_1^p = \{S_{11}, S_{12}\}$ and $S_2^p = \{S_{21}, S_{22}, S_{23}\}$. Their corresponding joint prior and posterior distributions are denoted by $\Pi\left(S_1^p, S_2^p\right) = \Pi\left(S_1^p\right) \Pi(S_2^p)$ and $\Pi(S_1^p, S_2^p | S_1^*, S^*)$ respectively. Let $S_1 = S_1^*$ and $S = S^*$ represent the data available on S_1 and S. Two cases are presented below – (1) S_1^*, S^* are independent, and (2) S_1^*, S^* have a one-to-one correspondence. Let us consider the first case. The posterior distributions $\Pi(S_1^p, S_2^p | S_1^*, S^*)$ can be obtained as

$$\begin{aligned}
\Pi\left(S_1^p, S_2^p \middle| S_1^*, S^*\right) &\propto L\left(S_1^*, S^* \middle| S_1^p, S_2^p\right)\Pi(S_1^p, S_2^p) \\
&\propto L\left(S_1^* \middle| S_1^p\right) L\left(S^* \middle| S_1^*, S_1^p, S_2^p\right) \Pi\left(S_1^p\right)\Pi(S_2^p) \\
&\propto L\left(S_1^* \middle| S_1^p\right) L\left(S^* \middle| S_1^p, S_2^p\right)\Pi\left(S_1^p\right)\Pi(S_2^p) \\
&\propto \; L\left(S_1^* \middle| S_1^p\right) \Pi\left(S_1^p\right) L\left(S^* \middle| S_1^p, S_2^p\right)\Pi\left(S_2^p\right)
\end{aligned} \tag{4.2}$$

In the last step of the above expression, the first two terms provide the posterior distributions of S_1^p using the data on S_1, i.e., updating the Level 1 BN corresponding to S_1. The posterior distributions of S_1^p are then used as prior distributions for model calibration in the HBN using the data on S. The last two terms provide the posterior distributions of S_1^p and S_2^p using the data on S. $L\left(S^*\middle|S_1^*, S_1^p, S_2^p\right)$ can be simplified to $L\left(S^*\middle|\, S_1^p, S_2^p\right)$ since S_1^*, S^* are assumed to be independent. In short, the data in lower-level BNs are used to obtain posterior distributions, which are later used as prior distributions for re-calibration using the data in the higher-level BN.

In the second case, the data on S_1 and $S$are assumed to have a one-to-one correspondence, (i.e., for a $S_1 = S_1^*$ there exists an associated $S = S^*$). Here, the posterior distributions can be obtained as

$$\begin{aligned} \Pi\left(S_1^p, S_2^p\middle|S_1^*, S^*\right) &\propto L\left(S_1^*, S^*\middle|S_1^p, S_2^p\right)\Pi(S_1^p, S_2^p) \\ &\propto L\left(S_1^*\middle|S_1^p\right)L\left(S^*\middle|S_1^*, S_1^p, S_2^p\right)\, \Pi\left(S_1^p\right)\Pi(S_2^p) \\ &\propto L\left(S_1^*\middle|S_1^p\right)L\left(S^*\middle|S_1^*, S_2^p\right)\Pi\left(S_1^p\right)\Pi(S_2^p) \\ &\propto\ L\left(S_1^*\middle|S_1^p\right)\Pi\left(S_1^p\right)L\left(S^*\middle|S_1^*, S_2^p\right)\Pi\left(S_2^p\right) \end{aligned} \tag{4.3}$$

From the HBN topology, it can be observed that S is independent of S_1^p when S_1 is known. Therefore, $L\left(S^*\middle|S_1^*, S_1^p, S_2^p\right)$ can be simplified to $L\left(S^*\middle|S_1^*, S_2^p\right)$. In the last expression, similar to the previous case, the first two terms provide the posterior distributions of S_1^p using S_1^* whereas the last two terms can be used to estimate S_2^p using S_1^*, S^*. In this case, the final posterior distributions of S_1^p, S_2^p can be obtained in one shot separately as opposed to the earlier case, where the final posterior of S_1^p is obtained in two steps – (1) calibrate using S_1^* only, and (2) re-calibrate with both S_2^p using S^*, using the posterior from step (1). The above two cases can be extended to calibrate HBNs with multiple levels.

After model calibration, the posterior distributions of all other variables in the HBN can be obtained through forward uncertainty propagation where the samples from posterior distributions of estimated parameters are propagated through the HBN using Monte Carlo sampling (MCS).

4.2.4 Multi-level sensitivity analysis for dimension reduction

Dimension reduction techniques are classified into two types – filter approach [92] and wrapper approach [93]. In the filter approach, all input variables are ranked according to a ranking criterion and the most significant variables can be selected. The number of variables selected depends on the analysis requirements and accuracy. In the wrapper method, a subset of variables is selected from all possible subsets of variables that best describe the output quantity of interest. An optimization search technique is generally used to obtain the best subset of variables. The proposed dimension reduction technique is based on variance-based global sensitivity analysis and falls under the filter approach. Fig. 4.2 describes the dimension reduction methodology using variance-based global sensitivity analysis.

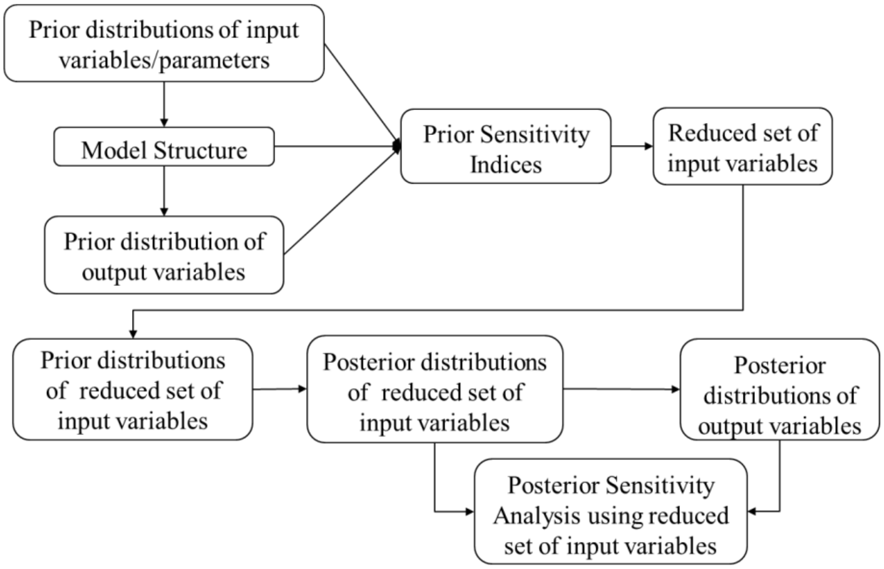

Prior distributions of input variables/parameters
Model Structure
Prior distribution of output variables
Prior Sensitivity Indices
Reduced set of input variables
Prior distributions of reduced set of input variables
Posterior distributions of reduced set of input variables
Posterior distributions of output variables
Posterior Sensitivity Analysis using reduced set of input variables

Fig. 4.2. Dimension reduction using global sensitivity analysis

As stated in Section 4.1, the global sensitivity analysis approach for dimension reduction has been illustrated in [85]; the same approach can also be used for hierarchical systems. However, as the number of levels increases in the network, the sensitivity analysis can become computationally expensive due to repeated evaluation of the entire network. Here, we develop an efficient dimension reduction approach where a segmented approach is taken, i.e., sensitivity analysis is carried out at multiple individual levels of the network as against the entire network.

Using the prior distributions of the root nodes (nodes with no parent nodes), the prior distributions of all the nodes in the HBN can be obtained using Monte Carlo sampling. The overall uncertainty in a system-level variable is due to the accumulation of several sources of uncertainty occurring at different lower levels, starting from the root variables. Thus, the uncertainty accumulation follows a bottom-to-top approach. The key idea in the proposed method is top-to-bottom decomposition of uncertainty at the system-level until the uncertainty contribution of a variable is below the threshold value.

As stated in Section 2.6.3, a HBN is a BN where some nodes may represent lower-level BNs. Therefore, the uncertainty contribution of the system-level node to the overall process performance includes the contributions of the uncertain variables in the lower-level BN that the node represents. Therefore, if the sensitivity index of the node is less than the threshold value then the sensitivity indices of each of the variables in the lower level, when calculated, are lower than the sensitivity index of the higher level node. As a result, the entire lower level BN can be collapsed and the node can be assumed deterministic at its most probable value. If the sensitivity indices of some of the nodes at any level are greater than the threshold value, then the nodes are replaced by their corresponding lower-level BNs and sensitivity indices are again re-computed.

The HBN in Fig. 2.6 is again used for illustration of the proposed approach. Using the prior distributions of root nodes ($S_{11}, S_{12}, S_{21}, S_{22}, S_{23}$), the prior distributions of higher-level variables can be obtained using some assumed conditional probability distributions. Using the prior distributions, sensitivity analysis is first performed in the Level 2 BN. For a given value of S_1 and S_2, the prediction of S is a PDF due to the conditional probability distribution $(S|S_1, S_2)$; this represents a stochastic relation between (S_1, S_2) and S. GSA requires a deterministic relation between the inputs and the output; therefore, the stochastic relationship is converted to a deterministic relationship through explicit representation of the stochasticity using an auxiliary variable. The auxiliary variable approach was briefly discussed in Section 2.5.

In this work, we extend the auxiliary variable approach to represent the stochasticity due to a CPD. Note that the stochasticity due to the CPD may represent aleatory uncertainty (such as process imperfections) or epistemic uncertainty (lack of knowledge about a process). For illustration, an auxiliary variable (u_S) can be used to represent the stochasticity due to the CPD of $S|S_1, S_2$ as

$$u_S = F_S(s|S_1 = S_1^*, S_2 = S_2^*) = \int_{-\infty}^{s} f_S(w|S_1^*, S_2^*)\ dw \tag{4.5}$$

where S_1^*, S_2^* are realizations of S_1, S_2 respectively. A deterministic relationship can thus be constructed between S_1, S_2 and u_S to S as given by Eq. (4.6), which therefore facilitates global sensitivity analysis:

$$S = F_S^{-1}(u_S = u_S^*|S_1 = S_1^*, S_2 = S_2^*) \tag{4.6}$$

where F_S^{-1} represents the inverse CDF of the CPD and u_S^* is a realization of u_S. Let the threshold value of sensitivity index of a variable for it to be considered insensitive be equal to 0.1. Assume the first-order sensitivity indices of S_1 and S_2 to S be equal to 0.7 and 0.08 respectively. The uncertainty contribution of S_2 to S includes the contribution from S_{21}, S_{22} and S_{23}. Therefore, the sensitivity indices of S_{21}, S_{22} and S_{23} each would be smaller than 0.08. Hence, the entire Level 1 BN corresponding to S_2 can be eliminated and all the corresponding variables can be assumed deterministic at their most probable values or their mean values. The most probable value of an input corresponds to that input with the highest PDF value. In the above illustration, the dimension reduction with respect to a random variables (S_1, S_2) is demonstrated.

In some cases, it may also be possible that the sensitivity index of an auxiliary variable representing the stochasticity due to a CPD such as u_S is less than the threshold value. Low sensitivity index of such an auxiliary variable signifies that the stochasticity due to a CPD does not cause substantial uncertainty in the model prediction. Hence, the auxiliary variable can be assumed deterministic at its mean value (u_S = 0.5). Here, the mean value is chosen as opposed to the most probable value since every value is equally probable for the uniformly distributed auxiliary variable. Assuming a deterministic value for the auxiliary variable results in the CPD to be replaced by a deterministic function given as

$$S = F_S^{-1}(u_S = 0.5|S_1 = S_1^*, S_2 = S_2^*) \tag{4.7}$$

After performing sensitivity analysis and dimension reduction in the Level 2 BN, we then advance to the next lower level, i.e., Level 1 BN. Sensitivity analysis can now be performed between S_{11}, S_{12} and S. From S_{11}, S_{12} to S, there exist four CPDs of S_{13}, S_{14}, S_1 and S. The stochasticity due to the CPDs of S_{13}, S_{14}, S_1 and S is explicitly represented using auxiliary variables $u_{S_{13}}, u_{S_{14}}, u_{S_1}$ and , u_S respectively as

$$S_{13} = F_{S_{13}}^{-1}(u_{S_{13}}|S_{11} = S_{11}^*, S_{12} = S_{12}^*) \tag{4.8}$$

$$S_{14} = F_{S_{14}}^{-1}(u_{S_{14}}|S_{11} = S_{11}^*, S_{12} = S_{12}^*) \tag{4.9}$$

$$S_1 = F_{S_1}^{-1}(u_{S_1}|S_{13} = S_{13}^*, S_{14} = S_{14}^*) \tag{4.10}$$

$$S = F_S^{-1}(u_S|S_1 = S_1^*, S_2 = S_2^M) \tag{4.11}$$

where ψ^* represents a realizations of ψ ($\psi = S_{11}, S_{12}, S_{13}, S_{14}, u_{S_{13}}, u_{S_{14}}, u_{S_1}, u_S$) and S_2^M represents the most probable value of S_2 since it is assumed deterministic from sensitivity analysis of the Level 2 BN. Thus, the constructed deterministic relationship from S_{11}, S_{12} S can be used to carry out sensitivity analysis. If the sensitivity indices of variables are less than a threshold value, then they can be assumed deterministic at their most probable values or their mean values.

4.3. Illustrative example: Injection molding

In this section, the proposed methodologies are demonstrated to evaluate the overall energy consumption of an injection molding process. The injection molding process is composed of three stages – (1) Melting of the polymer, (2) Injection of the polymer into the mold, and (3) Cooling of the polymer. Physics-based mathematical models are available for the estimation of energy consumption of each of the three stages and for overall energy consumption. Using the physics-based models, Bayesian networks can be constructed in each of the three stages, along with the overall energy consumption to estimate the uncertainty in overall energy consumption. Thus, the BNs corresponding to each of the three stages form the lower-level BN and the overall energy consumption forms the higher-level BN, thus forming a two-level HBN. To demonstrate the learning of Bayesian networks, one of the physics-based models (energy consumption for melting process) is assumed unavailable and it is learnt from available data. A synthetic dataset is generated by adding Gaussian errors to the corresponding mathematical model and used in the learning process. For melting and injection stages, overall energy consumption, the physics-based models are used. The segmented approach for model calibration is demonstrated using another synthetic dataset and later the proposed multi-level sensitivity analysis is demonstrated for dimension reduction. The physics-based models for energy consumption in the injection molding process are described below.

Melting process

This is the first stage of the injection molding process where the polymer, which initially is in the solid state, is converted into the liquid state. The power consumption in melting the polymer is given as

$$P_{melt} = 0.5 \times \rho \times Q \times C_P \times \left(T_{inj} - T_{pol}\right) + 0.5 \times \rho \times Q \times H_f \tag{4.12}$$

where P_{melt} refers to power consumption in melting process, ρ, Q, C_p, H_f refer to the density, flow rate, heat capacity and heat of fusion of the polymer respectively. T_{inj} and T_{pol} represent the injection temperature and temperature of polymer respectively. If V_{part} represents the volume of a part, then the volume of a shot (V_{shot}) considering the shrinkage (ϵ), buffer (Δ), number of cavities (n) is given as

$$V_{shot} = V_{part} \times (1 + \frac{\epsilon}{100} + \frac{\Delta}{100}) \times n \tag{4.13}$$

Using the power consumption for melting and volume of a shot, the energy consumption for melting process (E_{melt}) is given as

$$E_{melt} = \frac{P_{melt} \times V_{shot}}{Q} \tag{4.14}$$

Note that this model will only be used to generate a synthetic dataset to demonstrate BN learning. It will not be used for BN construction. In reality, such data comes from observation of the actual process operational data.

Injection process

This is the second stage of the injection molding process where the molten polymer is injected into the mold. The energy consumed in the injection process (E_{inj}) is given as

$$E_{inj} = p_{inj} \times V_{part} \tag{4.15}$$

where p_{inj} refers to the injection pressure.

Cooling process

The third stage of the injection molding process where the molten polymer is cooled to form the final product. The energy consumption in cooling process (E_{cool}) is given as

$$E_{cool} = \frac{\rho \times V_{part} \times \left[C_p \times \left(T_{inj} - T_{ej}\right)\right]}{COP} \tag{4.16}$$

where T_{ej}, COP represent the ejection temperature and coefficient of performance of the cooling equipment respectively.

Overall energy consumption

Given the energy consumption for each of the three stages, the expression for overall energy consumption of a part (E_{part}) is given as

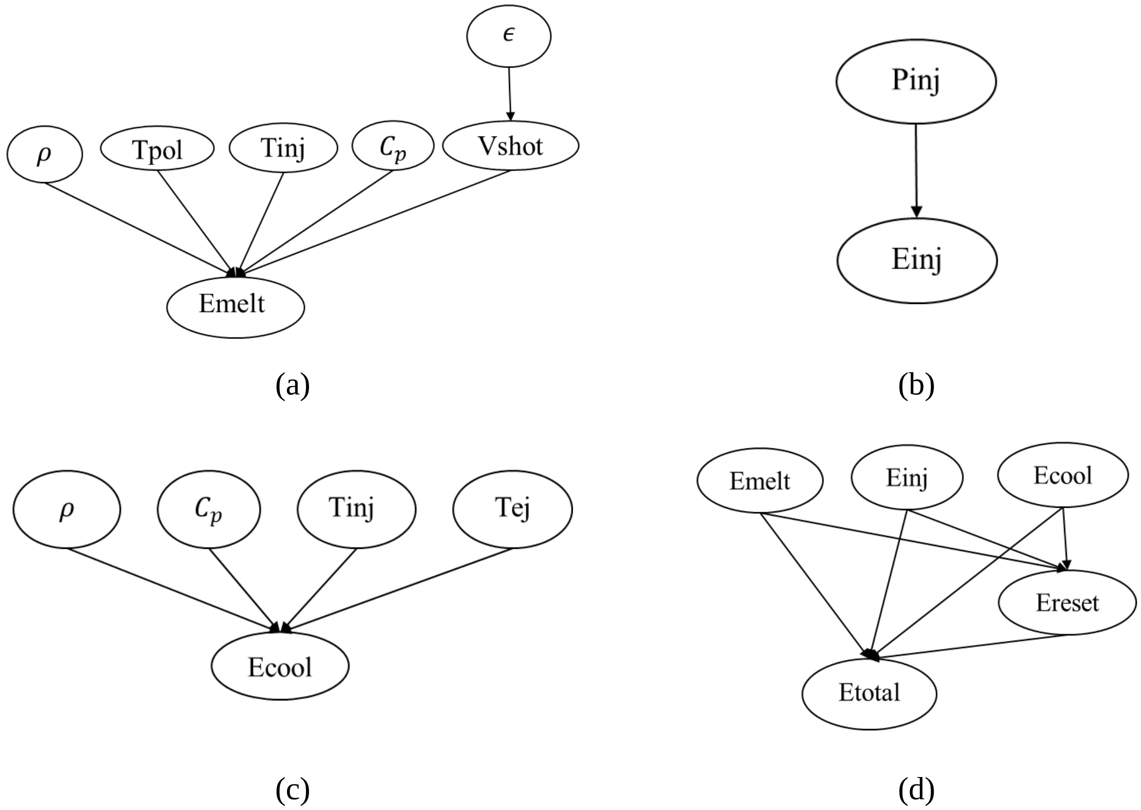
$$E_{part} = \frac{1}{n} \times \left[\left(\frac{0.75 \times E_{melt} + E_{inj}}{\eta_{inj}} + \frac{E_{reset}}{\eta_{reset}} + \frac{E_{cool}}{\eta_{cool}} + \frac{0.25 \times E_{melt}}{\eta_{heater}} \right) \right. \\ \left. \times \frac{n \times (1 + \epsilon + \Delta)}{\eta_{machine}} + P_b \times t_{cycle} \right] \tag{4.17}$$

where η_{inj}, η_{reset}, η_{cool}, η_{heater}, $\eta_{machine}$ refer to the efficiencies of injection, reset, cooling, heating and machine power respectively, t_{cycle} refers to the total cycle time, P_b refers to the power required for basic energy consumption when the machine is in stand-by mode, E_{reset} refers to the energy required for resetting the process and is given as

$$E_{reset} = 0.25(E_{inj} + E_{cool} + E_{melt}) \tag{4.18}$$

For simplicity, power consumption when machine is in stand-by model (P_b) is not considered because it depends on the type of machines used in the process. The Bayesian networks corresponding to the energy consumption in three stages and for the overall process, built using the physics-based models are provided in Fig. 4.3. For a given injection molding process, the volume of a part (V_{part}) is a constant and all the efficiency terms (η_{inj}, η_{reset}, η_{cool}, η_{heater}, $\eta_{machine}$) and COP are assumed to be known constants (equal to 0.7); therefore these variables do not appear in the Bayesian network representations in Fig. 4.3. All the physics-based models are assumed to be available in model libraries in JSON format as stated in Section 4.2.2.

BN Learning for the melting process: The BN corresponding to the melting process is learnt from data using a synthetic dataset that is generated from the physics-based models along with assumed Gaussian errors. The Gaussian errors are assumed with zero mean and a standard deviation of $1^{o}C$, $1.5 \times 10^{-6}\ m^3/s$, $0.00002\ m^3$ and $10\ kg/m^3$ for temperature, flow rate, volume and density measurements respectively. The parameters for the synthetic dataset are provided in Table 4.1.



(a) (b)

(c) (d)

Fig. 4.3. Bayesian Network representations of physics-based models for (a) Melting process, (b) Injection process and (c) Cooling process (d) Overall energy consumption

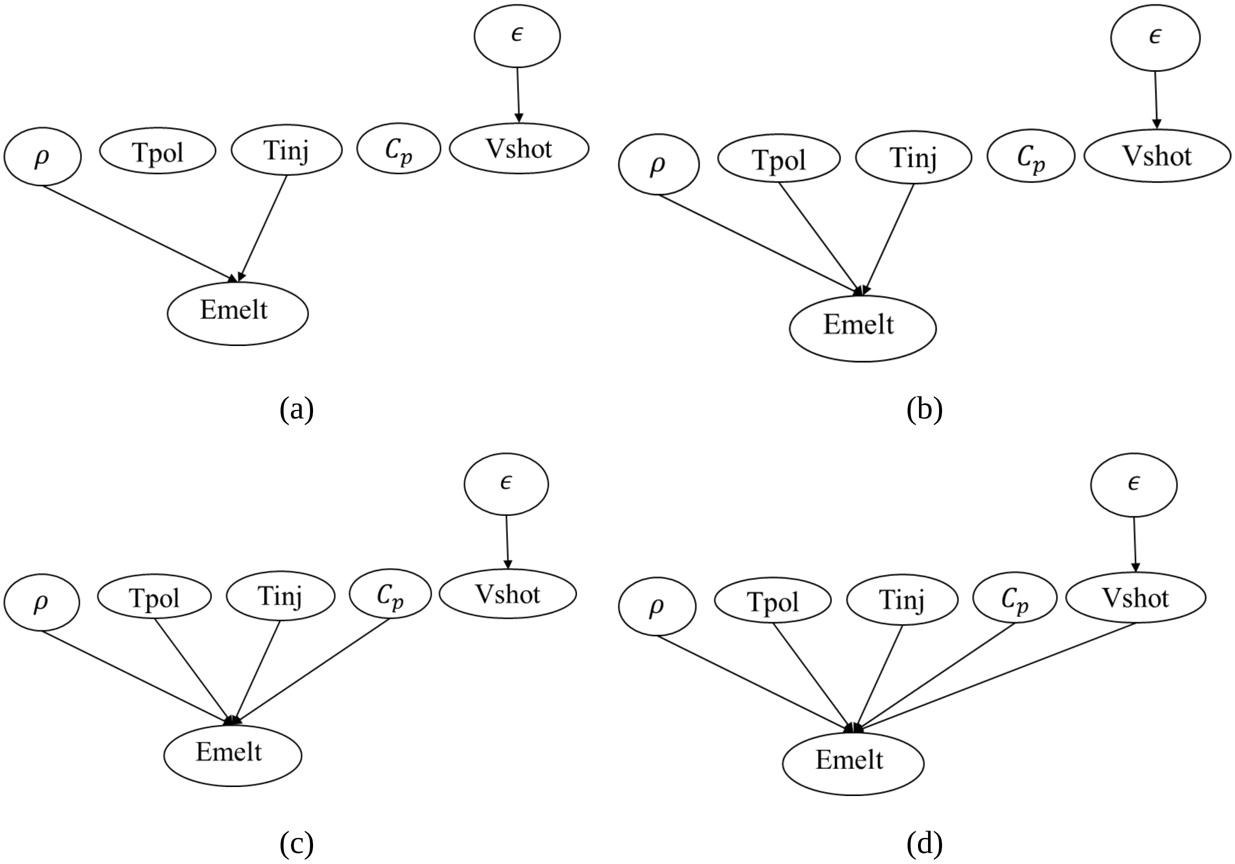
The synthetic dataset consists of 20,000 samples and using all the samples for BN learning can be computationally expensive. Therefore, subsets of different sizes (N = 100, 500, 1000, 2500, 5000, 7500, 10000) are obtained from the synthetic dataset and used for learning. To verify the learnt BN at any given dataset size, the learning procedure is carried out with two higher dataset sizes. If the learnt BN remains the same, then that BN is assumed to represent the true underlying BN. In this example, the BN learning converges within 5,000 samples and the learnt BN is further verified with 7,500 and 10,000 samples. The topology of the Bayesian network is learnt using the algorithm in [5] where a continuous variable is divided into a two-component Gaussian mixture model. The learned Bayesian network structures using different dataset sizes are provided in Fig. 4.4. From the learning results, it can be seen that a large amount of data is required to establish a relationship (an arc in the BN) Energy for melting (E_{melt}) vs Heat capacity (C_p) and Volume of shot (V_{shot}) which indicates that their influence on E_{melt} is weak. To verify this conclusion, sensitivity analysis is carried out and the results are provided in Table 4.2.

Table 4.1.Synthetic dataset for learning BN of melting process

Parameter	**Value**
Shrinkage (ϵ)	Uniform(0.018,0.021)
Volume of a part (m^3)	0.002048
Buffer (Δ)	0.01
Polymer temperature (T_{pol})	Normal(50,2)
Injection temperature (T_{inj})	Uniform(205,220)
Density (ρ) (kg/m^3)	Uniform(960,990)
Heat of fusion (H_f) (kJ/kg)	240
Heat capacity (C_p) ($J/(kg\ K)$)	Uniform(2250,2290)

Table 4.2. Sensitivity analysis results for the melting process

Parameter	**First order**	**Total effects**
Shrinkage (ϵ) / V_{shot}	0.000516	0.000644
Polymer temperature T_{pol}	0.148	0.0159
Injection temperature (T_{inj})	0.767	0.777
Heat capacity (C_p)	0.0083	0.0087
Density of polymer (ρ)	0.067	0.071



(a) (b)

(c) (d)

Fig. 4.4. Learnt Bayesian network structures with multiple dataset sizes (a) N = 100, (b) N = 500, (c) N = 1000 and (d) N = 5000

Since V_{shot} is dependent only on shrinkage (ϵ), the sensitivity index corresponding to shrinkage is the same as the sensitivity index corresponding to V_{shot}. From the results, it could be seen that the sensitivity indices to corresponding to C_p and V_{shot} are the lowest, thereby validating the above assumption.

The next step after learning the topology of the Bayesian network is to estimate the conditional probability distributions between (1) V_{shot} and ϵ, (2) E_{melt} and $\rho, T_{pol}, T_{inj}, C_p, V_{shot}$. For fitting several models to represent the CPDs, an error threshold of 1% is assumed. The percent error is defined as the percent ratio of the root mean square (RMS) error to the mean of the model prediction. If the percent error of a model is greater than the threshold, a higher complex model is constructed. A Gaussian linear CPD is first constructed which yielded a percent error of 0.9, which is less than the assumed threshold. The estimated CPD associated with V_{shot} is given as

$$V_{shot} = N(0.002089 * \epsilon \ + 0.002067, 0.000392\,) \tag{4.12}$$

For the conditional probability distribution (CPD) associated with E_{melt}, a Gaussian linear CPD was fit which resulted a percent error of 0.03. Therefore, a Gaussian linear CPD is used to represent the CPD associated with E_{melt} as

$$\begin{aligned} E_{melt} = N(2332.3 \times T_{inj} - 2331.6 \times T_{pol} + 641.43 \times \rho + 167.08 \ \times C_p + 2.96 \\ \times 10^8 \times V_{shot} - 1383730.2, 219.69) \end{aligned} \tag{4.13}$$

For estimating the above CPDs, 3500 random samples were used for fitting and another 1500 samples were used for estimating the RMS error (validation). Thus, the Bayesian network corresponding to melting process is learnt from data. A JSON representation of the constructed BN is then created which is later used for the automated construction of HBN for the overall injection molding process (Fig. 4.5).

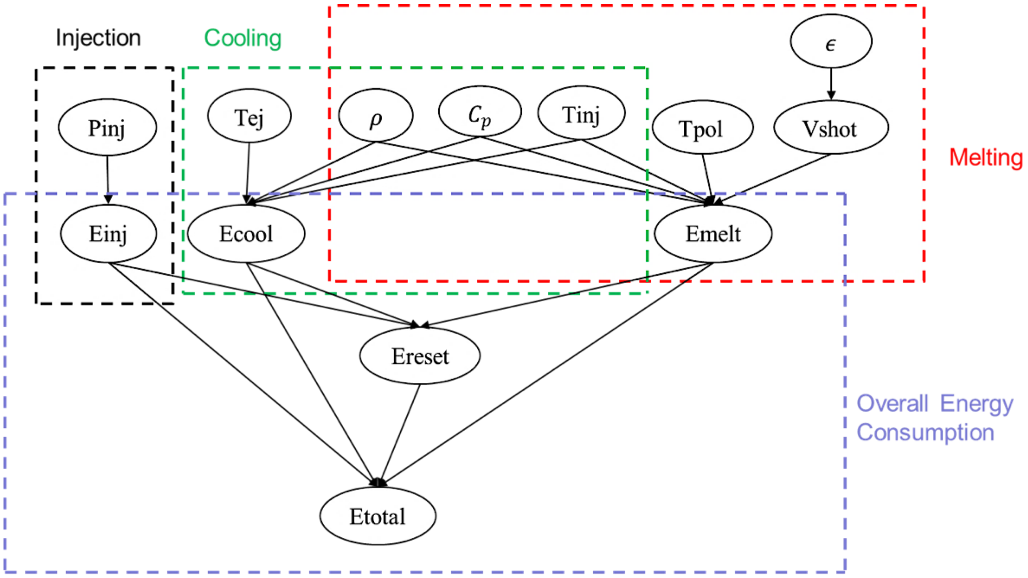


Fig. 4.5. Hierarchical Bayesian network for the injection molding process

Dimension reduction: The parameters to be estimated are the density of the polymer (ρ), injection temperature (T_{inj}), ejection temperature (T_{ej}), polymer temperature (T_{pol}) and injection pressure (P_{inj}). The prior distributions and the true values of the calibration parameters are provided in Table 4.3. Using the prior distributions of calibration parameters, prior distributions of all the variables in the HBN are obtained using Monte Carlo sampling (MCS).

Table 4.3. Prior distributions and true values of calibration parameters

Parameter	**True value**	**Prior distribution**
Density of polymer (ρ) (kg/m^3)	985	Uniform(970,990)
Injection temperature (T_{inj}) (^oC)	215	Uniform(205,220)
Ejection temperature (T_{ej}) (^oC)	55	Uniform(45,60)
Polymer temperature (T_{pol}) (^oC)	49	Normal(50,2)
Injection pressure (P_{inj}) (MPa)	93	Uniform(88,95)

Using the prior distributions, the multi-level sensitivity analysis is carried out for dimension reduction. First, sensitivity analysis in the higher-level BN is carried out and the sensitivity indices are provided in Table 4.4. Note that the sum of first-order sensitivity indices is less than 1; this can be attributed to the presence of a strong correlation of 0.84 between E_{cool} and E_{melt}. The strong correlation between them can be due to a shared set of input variables (T_{inj}, ρ and C_p).

Table 4.4. Sensitivity analysis results in the higher level BN of the injection molding process

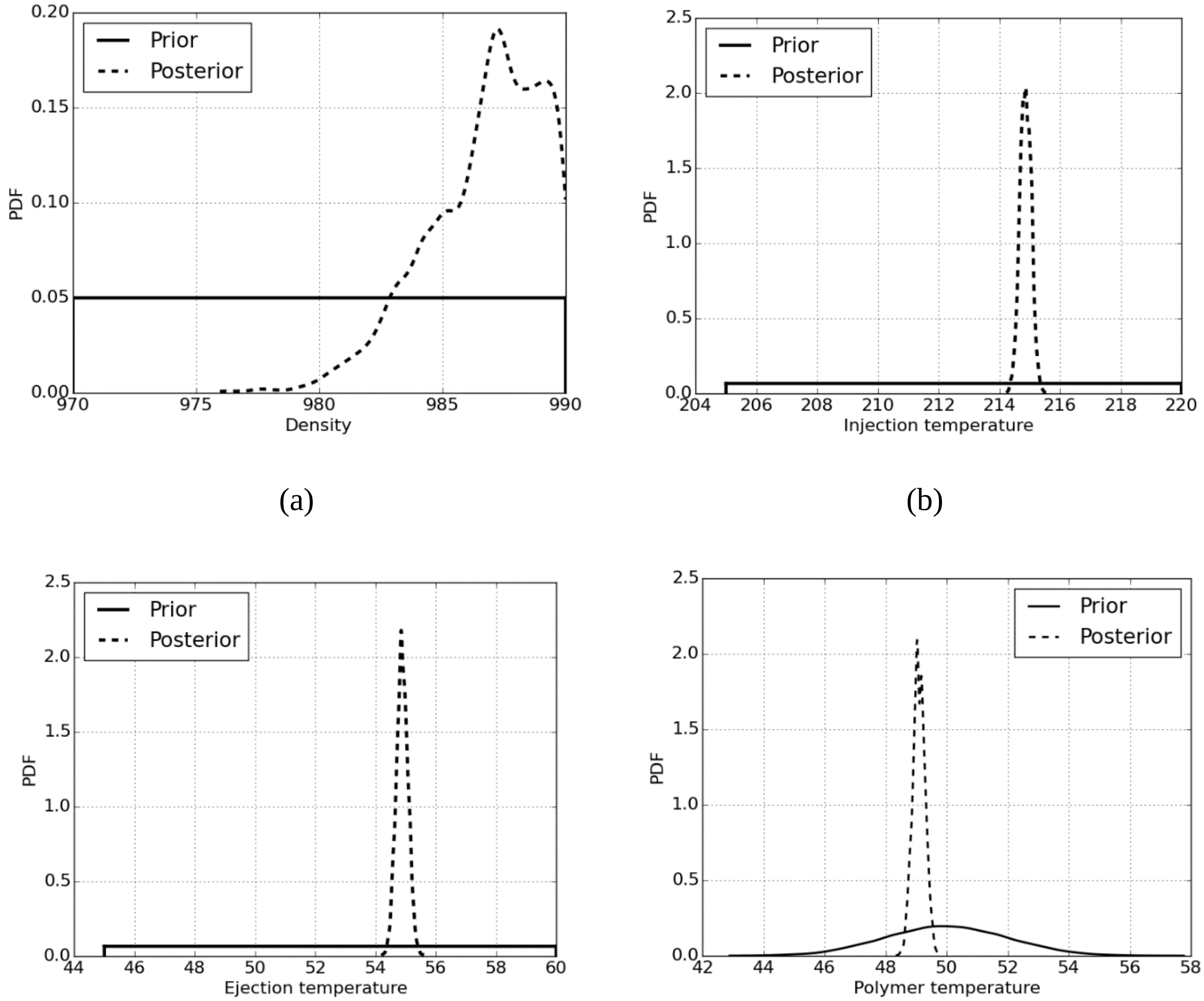
Parameter	**First-order sensitivity index**
Energy consumption in melting process (E_{melt})	0.06
Energy consumption in injection process (E_{inj})	0.007
Energy consumption in cooling process (E_{cool})	0.68

After carrying out sensitivity analysis in the higher-level BN, the retained variables (E_{melt}, E_{cool}) are replaced by their corresponding lower-level BNs and sensitivity analysis is again carried out between T_{inj}, T_{ej}, T_{pol} and ρ to the overall energy consumption (E_{total}); the sensitivity results are shown in Table 4.5. Since the sensitivity indices of all the variables are not less than the threshold value (0.01), all the variables can be retained for model calibration.

Table 4.5. Sensitivity analysis results in the HBN of the injection molding process

Parameter	First-order sensitivity index
Injection temperature (T_{inj})	0.61
Ejection temperature (T_{ej})	0.33
Polymer temperature (T_{pol})	0.01
Density of the polymer (ρ)	0.039

Model calibration: To estimate the unknown parameters, observation data is assumed to be available on T_{inj}, T_{ej}, T_{pol} and E_{total}. The observation data is synthetically generated using the true values of calibration parameters (Table 4.3) and by adding Gaussian observation errors. The observation errors associated with temperature and energy are assumed to be Gaussian distributions with zero mean and standard deviations of 2^oC and $100\ KJ$ respectively. The multi-level segmented approach (Section 4.2.4) is used to carry out model calibration. The data in the lower-level BN (T_{inj}, T_{ej}, T_{pol}) is used to obtain the posterior distributions of the calibration parameters using Markov Chain Monte Carlo sampling (Metropolis-Hastings algorithm), which are then used as prior distributions for calibration using the data in the higher-level BN (E_{total}). The prior and posterior distributions of the parameters after the multi-level model calibration approach are provided in Fig. 4.6.



(a) (b)

(c) (d)

Fig. 4.6. Prior and posterior distributions of (a) density of polymer (b) injection temperature (c) ejection temperature (d) polymer temperature

4.4. Summary

This chapter proposed a systematic methodology for modeling multi-level systems using hierarchical Bayesian networks, where some of the nodes may represent lower-level BNs. BNs at different levels can either be constructed using physics-based models or from available data using BN learning algorithms. An improved BN learning algorithm is presented for learning a hybrid BN (with both discrete and continuous variables), where several parametric and non-parametric models are proposed to obtain the best fitting conditional probability distributions. The BNs constructed from physics-based or data-driven models are stored in an exchangeable data format (such as JSON). Then all the JSON representations of BNs at different levels are fused together which are then used to create a Bayesian network in an automated manner. In this work, a Python script is created and PyMC package is used for BN construction.

Data for model calibration can be available at multiple levels; therefore, a multi-level segmented approach for model calibration is used where data from lower-level BNs is used to obtain posterior distributions of unknown distributions which are later used as prior distributions for calibration using data in the higher-level BN. To address to issue of scalability in large systems, a multi-level sensitivity analysis approach is developed for dimension reduction. First, sensitivity analysis is performed for the top-level BN and if the sensitivity index of any node is less than an assumed threshold value, the entire lower-level BN associated with that node in the top-level BN is discarded. The most influential nodes in the top-level BN are then replaced by their lower-level BNs and sensitivity analysis is again carried out. This procedure is continued all the way to the variables in the lowest-level BN.

The proposed methodologies are demonstrated using an injection molding process. A two-level HBN is considered for modeling the energy consumption. The three stages of injection molding BNs (melting of the polymer, injection into the mold, cooling to form the part) forming lower level and the energy consumption for the overall process forming the higher-level BN.

The dissertation thus far considered performance assessment of time-independent systems; the next chapter considers performance assessment of coupled and time-varying systems.

CHAPTER 5

PERFORMANCE ASSESSMENT IN TIME-VARYING SYSTEMS

5.1. Introduction

Engineering systems can sometimes be associated two-directional interactions between several subsystems. As discussed in Section 1.1, the two-directional interactions can happen simultaneously or occur with a time lag depending on the resolution of time considered in the analysis. This chapter considers systems where interactions between subsystems occur with time lag and Chapter 6 considers coupled systems where interactions occur simultaneously. The partitioned approach [94] is commonly used for modeling systems with time lag interactions. In a partitioned approach [94], the several subsystems are evaluated in a time sequence, the sequence is defined by the dependence between the interactions between the subsystems. Partitioned approaches for model-based performance assessment of systems consisting of only physical subsystems have been studied earlier; this chapter extends such approaches for systems with both computational and physical subsystems. An example of a system with computational and physical subsystems is a feedback control system where the computational subsystem controls a physical subsystem by implementing an appropriate actuation. In this dissertation, we refer to such software-controlled feedback control systems as cyber-physical systems or smart systems.

The problem of performance assessment under uncertainty can be considered as a generalization of reliability evaluation, which computes the probability that a pre-defined performance function crosses a design threshold. Current studies on CPS reliability have primarily considered data-driven approaches [95–98] where failure rates for individual components are assumed to be available. Failure rates are generally obtained by repeated experimental or simulation-based evaluation of components until failure. As opposed to the existing approaches, this work considers a probabilistic model-based approach, which is particularly useful when experimentation or simulation under uncertainty becomes prohibitively expensive or not feasible. As discussed in Section 1.1, we consider a CPS as a system with four subsystems: (1) physical, (2) sensors, (3) actuation and (4) computational. The coupling between the individual subsystems is shown in Fig. 1.4.

As discussed above, we consider a partitioned approach [94] to analyze the interactions in a smart system, considering it as a feed-forward system over time. In addition to interactions between several subsystems, there exists interaction between several computational nodes when a distributed computational subsystem is used rather than a single node. We model the smart system using a two-level Dynamic Bayesian Network (DBN), a higher-level DBN to model the interactions between the subsystems and a lower-level DBN to model the interactions between the computational nodes.

A DBN model is used here for performance evaluation as opposed to performing Monte Carlo analysis on the system model (such as a Simulink model) for the following reasons: (1) In addition to prediction over time, a DBN naturally allows for performing Bayesian inference when new sensor data is available, and (2) Prediction from a smart system model (Simulink) is generally deterministic for a given input whereas the prediction from a DBN is stochastic after aggregating several uncertainty sources. DBNs have been previously used for uncertainty modeling and performance evaluation of mechanical [55], industrial [56] and aerospace systems [99]. In this work, we use its capabilities to model a smart system for performance assessment.

Two examples – a smart indoor heating system and a smart manufacturing process are used to illustrate the proposed performance assessment techniques for a cyber-physical system. The smart indoor heating system considers a distributed computational system whereas the smart manufacturing process considers a single computational node. The smart manufacturing process considers online estimation of unknown model parameters using real-time data while the smart indoor heating system considered offline parameter estimation. The control variable in the indoor heating system is discrete in nature while a continuous control variable is considered in the smart manufacturing process.

5.2. Uncertainty sources in a smart system

In this section, we discuss the several uncertainty sources that may affect the performance of a cyber-physical system.

5.2.1 Computational subsystem

We assume that a software application that runs on a computational subsystem is designed, tested and validated to perform within a set of design input ranges. We do not consider the random coding/latent errors, which are traditionally used to assess software performance [100]. Aside from software coding errors, we consider the following three uncertainty sources.

Software Inputs: The inputs for the software application are obtained from the sensors collecting data about the physical subsystem and may be from the environment. When the sensor inputs are outside the design ranges (due to faulty sensors or large environmental variability), the software application may not provide correct outputs.

Hardware resources: A software application requires hardware resources (such as memory) to perform computation. In cases when hardware resources are unavailable, the computation cannot be completed resulting in faulty outputs.

Communication uncertainty: There are three types of communication in a smart system: (1) sensors to a computational subsystem, (2) between computational nodes, and (3) computational subsystem to an actuation subsystem. Faulty computational outputs and faulty control actions may arise due to unsuccessful communication. Robust communication protocols can be implemented to avoid any communication uncertainty but such robust protocols may not be always feasible due to high design costs.

5.2.2. Actuation and physical subsystems

Uncertainty sources during the modeling of these subsystems are similar to those discussed in Section 2.1 and include model inputs, model parameters and model errors. With time, an actuation or a physical subsystem may degrade; this degradation needs to be estimated and included in the models. If the degradation cannot be observed directly, it needs to be inferred through indirect methods, which may lead to uncertainty in its estimation.

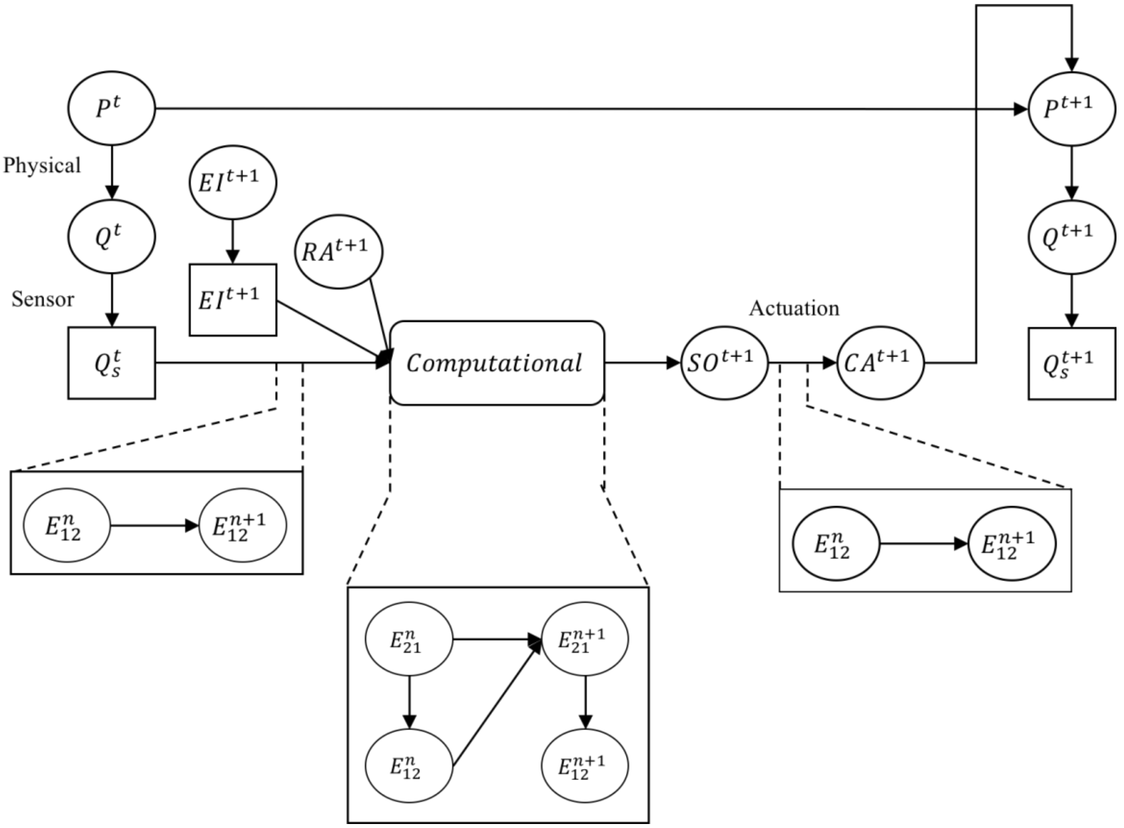


Fig. 5.1. A two-level DBN of a conceptual smart system

5.3. Multi-level DBN construction

We first discuss the construction of a DBN and then consider performance evaluation. Fig. 5.1 shows a representative DBN model; the description of variables is presented in Table 5.1. In Fig. 5.1, the rectangle with rounded corners named `Computational' does not represent a DBN node, but represents a lower-level DBN depending on the interaction pattern as detailed here. Distributed computing systems are often arranged into a sequence of components communicating over well-defined interaction protocols [101]. Such rigid interaction semantics help analyze the behavior of distributed system and also enables developing online fault detection and diagnosis mechanisms [102].

Table 5.1. Parameters in the DBN model (Fig. 5.1)

Parameter	Description
P^t, P^{t+1}	State variable at time $t = t, t+1$
Q^t, Q^{t+1}	Observation variables at time $t = t, t+1$
Q_s^t, Q_s^{t+1}	Sensor measurements of observation variables at time $t = t+1$
EI_s^{t+1}	Sensor measurements of environmental inputs at time $t = t+1$
RA^{t+1}	Resource availability at time $t = t+1$
SO^{t+1}	Computational subsystem output at time $t = t+1$
CA^{t+1}	Control action at time $t = t+1$

We focus on developing conditional relationships for a distributed computing subsystem as the conditional relationships for physical and actuation subsystems can be derived from physics models or data [53]. The conditional relationships for complex computational subsystems are studied by breaking them down into two basic interaction patterns: (1) 2-node asynchronous, and (2) 2-node synchronous [101], discussed below.

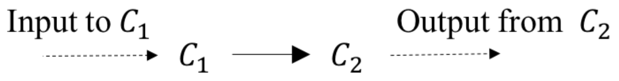


Fig. 5.2. 2-node asynchronous interaction pattern

2-node asynchronous interaction pattern: We first consider a 2-node asynchronous interaction pattern with two computational nodes C_1 and C_2 as shown in Fig. 5.2, where inputs to the computational system are input to C_1. C_1 performs some computation, communicates the results to C_2, which performs further computation and outputs the results. Let E_1 and E_2 denote the events that the C_1 and C_2 successfully perform their analysis. And let E_{12} correspond to the event that the data from C_1 is successfully transmitted to C_2. In general, data is transmitted across a network in packets. Let $E_{12} = 0,1$ correspond to the two states that a given packet reaches and does not reach the destination (here C_2). We define a smaller time-scale where a time step (n) corresponds to the time it takes to send a packet of data. Let E_{12}^n corresponds to the event that a packet reaches the destination at time step$n$. We make a Markov assumption that the state of the event E_{12}^n depends on the state of the event at the previous time step, E_{12}^{n-1}. In other words, if a packet cannot be transmitted at time $n-1$ due to network interruptions, then it is assumed to affect the probability packet transmission at time n. Therefore, the DBN corresponding to a 2-node asynchronous interaction pattern can be represented as shown in Fig. 5.3.

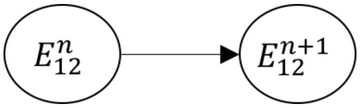


Fig. 5.3. DBN for a 2-node asynchronous pattern

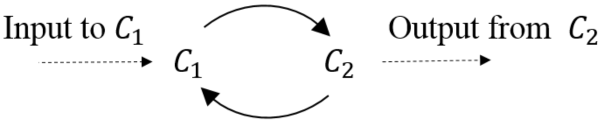


Fig. 5.4. 2-node synchronous interaction pattern

2-node synchronous interaction pattern: A synchronous interaction pattern is characterized by a sequence of *request* and *reply* messages as shown in Fig. 5.4, where C_2 requests for data and C_1 replies accordingly. Similar to the previous asynchronous case, we define a smaller time-scale where a time step contains a set of request and reply messages (represented using a rectangle in Fig. 5.5). We decompose the request-reply communication into a series of request and reply messages similar to the partitioned approach between the computational and physical subsystems.

$$\boxed{C_2 \xrightarrow{\text{request}} C_1 \xrightarrow{\text{reply}} C_2} \xrightarrow{\text{request}} C_1 \xrightarrow{\text{reply}} C_2$$

Fig. 5.5. Decoupling request and reply messages in a 2-node synchronous interaction pattern

In addition to E_1, E_2 and E_{12} defined above, we further define E_{21} as the event representing the successful transmission of data packet from C_2 to C_1. Thus, at every time step n one event each of E_{21} and E_{12} occur. Since both these events happen at the same time step, their states are assumed to be dependent one each other. If $E_{21} = 1$ (failure), then the probability of $E_{12} = 1$ will be higher when compared to when $E_{21} = 0$. In addition to the dependence at the same time step, there exists a dependence between the states of E_{21} events at successive time steps, as mentioned in the case of asynchronous case. The DBN for this 2-node synchronous case is provided in Fig. 5.6. When E_{21}^{n} is successful, then $E_{21}^{n+1}$is dependent on E_{12}^{n} and if E_{21}^{n} is not successful (failed request message and this implies no reply message), then E_{21}^{n+1} is assumed to be dependent on E_{21}^{n}.

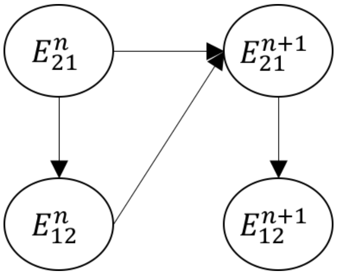


Fig. 5.6. DBN for a 2-node synchronous interaction pattern

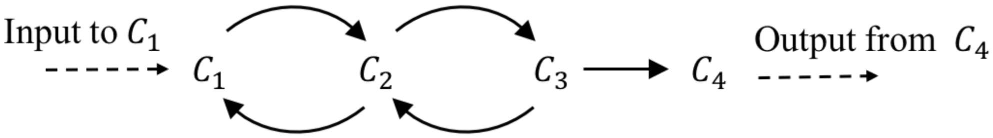


Fig. 5.7. 4-node complex interaction pattern

Extending to complex interaction patterns: If complex interaction patterns are present, then they are broken into sets of 2-node asynchronous and synchronous interaction patterns. Consider a 4-node interaction pattern shown in Fig 5.7. In this pattern, the inputs to the computational subsystem are communicated to C_1 which performs some computation and communicates its output to C_2. From C_2 information is communicated to C_3 and then C_4 which outputs the result from the computational subsystem. The interactions between C_1 and C_2, and C_2 and C_3 are synchronous whereas interaction between C_3 and C_4 is asynchronous in nature. Therefore, the DBN in Fig. 5.6 can be used to model the synchronous interactions between $C_1$and C_2, and C_2 and C_3, and the DBN in Fig. 5.5 to model the interaction between C_3 and C_4. It should be noted that the communication between the nodes are sequential in nature, i.e., communication between C_2 and C_3 occurs after the communication between C_1 and C_2. Therefore, their DBNs can also be represented sequentially. In some cases, it might be possible that the outcome of one interaction may influence the outcome of the following interaction; this influence may be quantified through a conditional dependence relationship across the DBNs.

5.4. Performance evaluation

Modeling sensor uncertainty: The sensor uncertainty ϵ_s is typically modeled using a Gaussian distribution with zero mean; since positive and negative errors occur with equal probability [103]. The relationship of the measurement variable (Q_s^t) conditioned on the unknown value of the observation variable (Q^t) can be represented as

$$Q_s^t | Q^t = Q^t + \epsilon_s \tag{5.1}$$

The sensor then reads a value from the probability distribution of Q_t^s and sends the data to the computational subsystem.

Simulating asynchronous interaction pattern: Consider the 2-node interaction pattern as shown in Fig. 5.2, where C_1 receives the input, processes the information and outputs to C_2. Let p data packets be transferred from C_1 to C_2 and if r packets get successfully transmitted, then it is assumed that all the information can be reconstructed at C_2. In the lower-level DBN, one data packet is assumed to be transmitted in each time step. Since there are p packets, n, which represents time in the lower-level DBN goes from $n = 1$ to $n = p$. The events corresponding to each data packet transmission are represented as E_{12}^n. The joint probability of all the events corresponding to transmission of p data packets is equal to $P(E_{12}^1, E_{12}^2 \dots E_{12}^p)$. This joint probability can then be decomposed into a product of marginal and conditional probabilities defined as

$$P\left(E_{12}^1, E_{12}^2 \dots E_{12}^p\right) = P(E_{12}^1) \times P(E_{12}^2 | E_{12}^1) \times \dots . P(E_{12}^p | E_{12}^1, E_{12}^2 \dots E_{12}^{p-1}) \tag{5.2}$$

Using the Markov assumption as mentioned in Section 5.3, Eq. (5.2) can be simplified as

$$P\left(E_{12}^1, E_{12}^2 \dots E_{12}^p\right) = P(E_{12}^1) \times P(E_{12}^2 | E_{12}^1) \times \dots . P(E_{12}^p | E_{12}^{p-1}) \tag{5.3}$$

Let R^1 represent the probability of a successful data packet transfer at a lower time-scale time step $n = 1$. The conditional dependence relationship for successful data transmission are represented as shown in Table 5.2. In Table 5.2, R_{ij} represents the probability of data transmission event in the current time step j conditioned on the data transmission event in the previous time step i $(i, j = 0{,}1)$. 0 and 1 represent success and failure of an event respectively. The parameters in Table 1 can be estimated through an aggregation of historical data, simulations and expert knowledge regarding the system.

Table 5.2. Conditional Probabilities of successful data transfer between two consecutive time steps

	Successful at time step $n+1$	Unsuccessful at time step $n+1$
Successful at time step n	R_{00}	R_{01}
Unsuccessful at time step n	R_{10}	R_{11}

For illustration, consider a case when $r = 2$ and $p = 3$. Across three time steps, there exist 8 combinations with two outcomes in each time step (data packet delivered or not delivered). If two successful transmissions are required, then four combinations result in successful data transmission. The set of successful combinations and their probabilities are given in Table 5.3. Since $p = 3$, there are three elements in each combination. The overall success probability can be calculated as the sum of all the individual probabilities. Using the overall probability, a binary random sample can be drawn to simulate the data transmission in a 2-node asynchronous interaction pattern.

Table 5.3. 2-node asynchronous interaction pattern: Successful combinations and their probabilities

Combination	**Probability**
[0,0,0]	$R^1 \times R_{00} \times R_{00}$
[0,0,1]	$R^1 \times R_{00} \times R_{01}$
[0,1,0]	$R^1 \times R_{01} \times R_{10}$
[1,0,0]	$(1 - R^1) \times R_{10} \times R_{00}$

Simulating synchronous interaction pattern: As opposed to the asynchronous system, we assume we require r successful request-reply pairs in a synchronous system since a *reply* does not occur unless there is a *request* and *reply* does not always occur for every *request*. The joint probability of p *request-reply* pairs, assuming one occurs at each lower-scale time step can be computed using Eqs. 2 and 3.

Let R_2 represent the probability of successful *request* message at lower time-scale time step $n = 1$. Let R_{12} represent the probability of successful reply message when the request message is successful. Therefore, $R_2 \times R_{12}$ refers to the reliability of a request-reply pair at any time step $n = 1$. For illustration, assume the same conditional relationships between two requests across two successive time steps as provided in Table 5.2. Given the dependence relationships across time steps, the probability of r successful pairs out of p can be computed. For $p = 3$ and $r = 2$, the set of successful combinations and their probabilities are provided in Table 5.4. Therefore, the overall success probability can be calculated as the sum of all the individual probabilities.

Table 5.4. 2-node synchronous interaction pattern: Successful combinations and their probabilities

Combination	**Probability**
[0,0,0]	$(R_2 \times R_{12}) \times (R_{00} \times R_{12}) \times (R_{00} \times R_{12})$
[0,0,1]	$(R_2 \times R_{12}) \times (R_{00} \times R_{12}) \times (1 - R_{00} \times R_{12})$
[0,1,0]	$(R_2 \times R_{12}) \times (R_{01} \times (R_{10} \times R_{12}) + \big(R_{00} \times (1 - R_{12})\big) \times (R_{10} \times R_{12})$
[1,0,0]	$(R_2 \times (1 - R_{12}) \times (R_{10} \times R_{12}) + (1 - R_2) \times (R_{10} \times R_{12})) \times (R_{00} \times R_{12})$

As mentioned in Section 5.3, a failure in a message transmission can be due to a failure in either *request* or *reply* message transmission. Therefore, the last two combinations in Table 5.4, have two terms representing the cases of failures in request and reply message transmissions. After the completion of computational analysis, the computational output is communicated to the actuation system; this communication can be simulated using an asynchronous or a synchronous system as described above. If the actuation system cannot receive the data, an assumption that the control action in the previous time step is continued in the current time step is made. In order to obtain a correct output from the computational subsystem, all the individual nodes should have access to resources in addition to successful data transmission between them. Simulation of data transmission between basic interaction patterns has been discussed above. We next discuss the simulation of resource availability.

Simulating resource availability: In the DBN model for a generic smart system shown in Fig. 5.1, if the data is successfully transmitted from sensors, the computational subsystem estimates the state variables P^t through Bayesian Inference (typically using particle-filtering [67]) and calculates the necessary control action for the next time step. To perform the analyses, the computational nodes should have the necessary resources. Let there be N computational nodes and $E_{i,k},\ i = 1,2,3...N$ represent the events corresponding to their resource availability. The joint probability can be defined as

$$P\left(E_{1,k}, E_{2,k} \ldots . E_{N,k}\right) = P\left(E_{1,k}\right) \times P\left(E_{2,k} \middle| E_{1,k}\right) \times \ldots .. P(E_{N,k}|E_{1.k}, E_{2,k} \ldots E_{N-1,k}) \tag{5.4}$$

In this discussion, we consider two hardware resources: power and memory. We assume that the power is supplied through a battery and each node is assumed to have an associated battery. Under this assumption, the resource availability of one node is independent of the resource availability of another node. Thus, Eq. (5.4) can be simplified as

$$P\left(E_{1,k}, E_{2,k} \ldots . E_{N,k}\right) = P\left(E_{1,k}\right) \times P(E_{2,k}) \times \ldots .. P(E_{N,k}) \tag{5.5}$$

Let S_r refers to the probability of having necessary resources at each computational node. Therefore, the probability that all the events, $E_{i,k},\ i = 1,2,3...N$, are successful (assuming the same resource probability) is equal to S_r^N. A binomial random sample with a success probability of S_r^N is drawn to simulate the resource availability. It may be possible that different nodes will have different resource probabilities and dependent on each other. An example of such a case is when a group of nodes has a common power supply unit. In cases of dependent resource availability, Eq. (5.4) cannot be simplified to obtain Eq. (5.5), and the dependence needs to be modeled using either expert knowledge or simulations. Simulation of resource availability and data transmission in the computational subsystem provides the posterior distributions of P^t and the control action at time $t + 1$, which are then used to estimate the prior distributions of P^{t+1}. Sensor measurements at time $t+1$ are used to estimate the posterior distributions of P^{t+1} and control action at time $t+2$. This process is repeated until a pre-defined analysis time for performance evaluation.

5.5. Example 1: Smart indoor heating system

We consider here a smart indoor heating system in a commercial building that can control the heating vents in different rooms independently, enabling room-by-room temperature control similar to [104]. For illustration, we consider a building with four rooms as shown in Fig. 5.8.

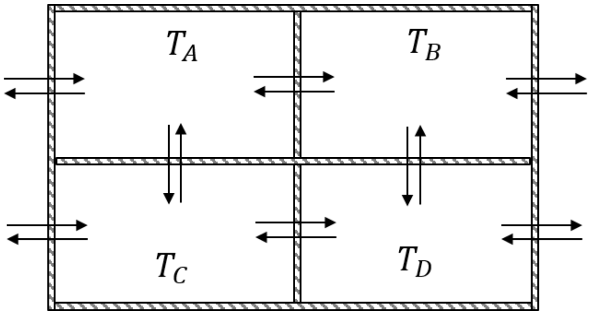


Fig. 5.8. Smart building showing heat flow across rooms and outside environment

Table 5.5. Comfort levels of occupants

Parameter	Lower bound	Mode	Upper bound
C_A	67	70	73
C_B	65	68	70
C_C	68	72	74
C_D	69	73	75

Every occupant in a room is assumed to have a temperature comfort level, defined as the temperature range at which an occupant is comfortable in. These comfort levels are quantified using triangular distributions; their parameters are shown in Table 5.5. In Table 5.5, $\boldsymbol{C_A}$, $\boldsymbol{C_B}$, $\boldsymbol{C_C}$ and $\boldsymbol{C_D}$ represent the occupant comfort levels in each of the rooms. The table should be read as follows. The occupants in room A are comfortable between temperatures 67 F and 73 F while their maximum comfort temperature is 70 F. Based on the current temperature, each occupant accumulates “comfort credits" defined in Eq. (5.6).

$$C_{cred} = \int_{t_1}^{t_2} \int_{T_1}^{T_2} \beta Pr(T,t) dT\, dt \tag{5.6}$$

In Eq. (5.6), $T_1$and T_2 represent the room temperatures when time $t = t_1$ and $t = t_2$ respectively. β is a comfort credit factor and $Pr(T,t)$ represents the probability density function of comfort level evaluated at the current room temperature at a given time t. A comfort credit can be regarded as a numerical measure that denotes the comfort level of an occupant.

Maintaining the rooms at the occupant comfort levels requires energy. To achieve energy conservation, a sustainability baseline temperature is set. If a room temperature is greater than a set baseline temperature, the occupant is penalized with negative “energy credits" and if the temperature is less than the baseline temperature, the occupant is awarded positive “energy credits". The energy credits are defined as

$$E_{cred} = \int_{t_1}^{t_2} \int_{0}^{T_B - T} \alpha dT\, dt \tag{5.7}$$

In Eq. (5.7), T_B , T and α represent the baseline temperature, current room temperature and energy credit factor respectively. The goal of this problem is to design a controller that maximizes the combination of comfort and energy credits such that a minimum comfort level is attained for all the occupants simultaneously. Due to the presence of uncertainty sources, the computed energy and comfort credits are not deterministic but stochastic. Therefore, we maximize the expected value of the sum of energy and comfort credits, formulated as

$$Max \sum_{i=A,B,C,D} E[E_{cred}^{i} + C_{cred}^{i}] \tag{5.8}$$

subject to $\Pr(C_{cred}^{A} > 0\ \cap C_{cred}^{B} > 0 \cap C_{cred}^{C} > 0 \cap C_{cred}^{D} > 0) > \gamma$

In the above formulation, H_A, H_B, H_C and H_D are boolean variables representing if the heating system in a room is turned on or off. γ represents a probability threshold that all room temperatures are in their occupant comfort ranges. $E[.]$ represents the expectation operator. E_{cred}^{i} and C_{cred}^{i}, $i = \{A, B, C, D\}$ represent the energy and comfort credits accumulated by the occupants in each of the rooms.

Table 5.6. Components with their costs and uncertainties

Component	**Performance**	**Cost**
Type 1 sensor	0.1	40
Type 2 sensor	0.15	25
Type 3 sensor	0.2	15
Type 1 network	0.95	200
Type 2 network	0.97	300
Type 3 network	0.99	400

Let three types of temperature sensors and three types of wireless network systems such as Bluetooth, 2.4 GHz Wi-Fi and 5 GHz Wi-Fi networks be available for design purposes. The uncertainty in a component performance and its costs are given in Table 5.6. These values are chosen arbitrarily for illustration purposes and can be replaced with actual values from a manufacturer. The affordable budget to this design is assumed to be 450 units. In Table 5.6, performance for a sensor refers to the standard deviation of the measurement error in F while that for a network refers to its reliability, i.e., probability that a data packet is successfully transmitted.

The uncertainty sources with respect to the computational subsystem include: (1) Sensor uncertainty in the room temperature measurements, (2) Environmental variability in outside temperature, (3) Communication uncertainty between the temperature sensors, computational subsystem and the heating system, and (4) Resource availability as the computational subsystem is assumed to perform other operations such as lightening control and security systems. Uncertainty in the physical subsystem include the uncertainty in the estimation of thermal conductivity, which is used for calculating a suitable control action. The goal is to find the best design combination of sensors and a network that satisfies the budget constraints and maximizes a performance evaluation metric, defined as the probability that the controller cannot find a control action to maintain the occupant comfort requirements.

Table 5.7. Design Configurations and their costs

	Type 1 network	Type 2 network	Type 3 network
Type 1 sensor	**400**	500	600
Type 2 sensor	**325**	**425**	525
Type 3 sensor	**275**	**375**	**475**

For each design, five temperature sensors are used (one for each room and one for outside temperature) and the overall cost for each design is given in Table 5.7. Due to budget constraints, only 5 out of 9 configurations from Table 5.7 are feasible; these configurations are indicated in bold. For each sensor type, the most reliable of all possible network types is identified and considered for analysis. This corresponds to the first two design options along the diagonal in Table 5.7. The conditional dependence relationships for communications across two lower time-scale time steps (as discussed in Section 5.3) are provided in Table 5.8. In Table 5.8, X equals 0.95, 0.97 and 0.99 when networks of Types 1,2 and 3 are used respectively (Table 5.6).

Table 5.8. Communication Probability

	Successful at time step $n+1$	Unsuccessful at time step $n+1$
Successful at time step n	X	$1-X$
Unsuccessful at time step n	0.9	0.1

Problem parameters: Each room has dimensions $9m \times 9m \times 5m$ and one window with dimensions $1m \times 1m$. The thermal conductivity of the window is assumed to be 0.3 $W/F.m$ while that of the wall is not known precisely and need to be calibrated from data. The thickness of wall and window insulations are assumed to be 0.8 m and 0.04 m respectively. The density and heat capacity of air are assumed to be equal to 1.225 kg/m^3 and 558.55 $J/kg.F$. The temperature and the amount of hot air blown from the heater are 110 F and 106 kg/hr respectively. The values of T_B, α and β are assumed as 70 F, 50 and 100 respectively. The outside temperature data are obtained from SML2010 Data Set [105], which can be downloaded from the UCI Machine Learning Repository [106]. We used the weather temperature data from this dataset as the outside temperature data for our analysis.

Computational subsystem: The computational subsystem is assumed to consist of two nodes, C_1 and C_2. C_1 receives the sensor data and estimates the indoor temperatures through Bayesian Inference; these estimates are then transmitted to C_2 which calculates the control action for the next time step. The probability of resource availability for both nodes is assumed as 0.9. Two-node asynchronous interactions are assumed between the temperature sensors and the computational subsystem, and between the computational and actuation subsystems, and a two-node synchronous interaction between the computational nodes.

DBN model: Fig. 5.9 presents the DBN for the smart heating system. It should be noted that there exists no state variable P_t but only an observation variable Q_t, which are the actual temperatures in each room and the outside temperature (T_A, T_B, T_C, T_D and T_{out}). The corresponding measurements of these variables are represented as $T_{A,s}, T_{B,s}, T_{C,s}, T_{D,s}$ and T_{out} respectively. k refers to the thermal conductivity coefficient of the wall. R_1 and R_2 refer to the resource availability variables corresponding to computational nodes C_1 and C_2 respectively. H_A, H_B, H_C and H_D refer to the control actions in each of the rooms. The thermal conductivity and outside temperature affect the temperatures in all rooms, which needs to be represented by arrows from k and T_{out} to every room temperature. For better visualization, all room temperatures are grouped in a 'dotted' rectangle. An arrow to the 'dotted' rectangle should be interpreted as an arrow going to every room temperature. Fig. 5.9 shows a higher-level DBN; the lower-level DBNs can be constructed as detailed in Section 5.3.

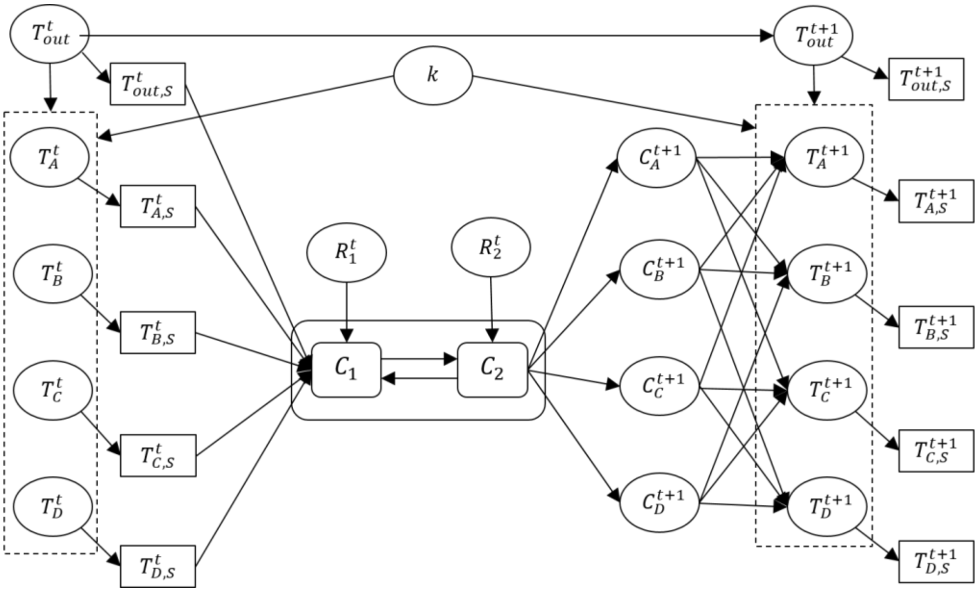
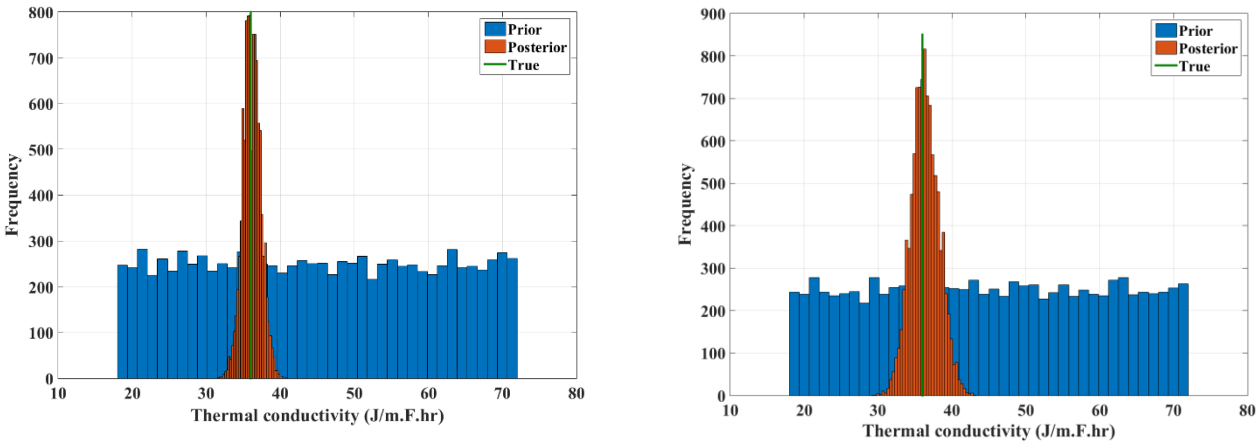


Fig. 5.9. DBN model for smart indoor heating system

Model calibration: Each design results in a different calibration of thermal conductivity due to the difference in sensor uncertainty. The prior and posterior distributions for both the design options are presented in Fig. 5.10. From the plots, it can be seen that the posterior variance in Fig. 5.10 (b) is higher due to higher sensor uncertainty.

Performance analysis: We ran the smart indoor heating system for one day (with time steps of 15 min). We used the outside temperature corresponding to March 25 in the SML2010 Dataset. The reason for choosing this date is explained later in this section. We start with available sensor measurements at 12 am and the analysis continues until 12 am to next day. For each type of communication, two messages are transmitted and even if one reaches successfully, the data is assumed to be successfully transmitted. For simplicity, the conditional relationships for communication uncertainty within and across two lower time-scale time steps in the synchronous interaction pattern between the computational nodes are assumed to be the same as given in Table 5.8. Eqs. (5.6, 5.7) are evaluated between every two successive sensor measurements (15 min); therefore, $t_2 = t_1 + 15$ and $t_1 = 0$.



(a) (b)

Fig. 5.10. Calibration results with sensor uncertainty of (a) 0.1 F and (b) 0.15 F

Using the sensor measurements, Bayesian inference is performed using particle filtering to estimate the outside temperature and room temperatures. For predicting the outside temperature in the next time step, we used the weather data from the previous 10 days to construct a Gaussian distribution to represent the change of temperature between the current time step and next time step; this Gaussian distribution is used for temperature forecasting in the next time step. We used data from March 15 to March 24 for temperature forecasting on March 25, 2012. Using the forecast outside temperature, the control action that optimizes the sum of energy and comfort credits is identified, and using this, the prior distributions of the room temperatures in the next time step are obtained. The same process is repeated 250 times to obtain the failure probability. The practical reasoning for this is as follows. If a particular design alternative is installed at 250 homes, what is the probability that the occupants be uncomfortable at any time during the analysis? The analysis is carried out using both the design alternatives and the success probabilities are 0.99 and 0.93 respectively. Therefore, the better design option is the one with the sensor uncertainty of 0.1 F (Type 1 sensor) and 0.95 network reliability (Type 1 network).

Note that the above problem formulation does not adapt over time to new conditions or changes in system properties. Thermal conductivity is a model parameter that varies with time due to the wear of insulation. Therefore, the system needs to re-calibrate the thermal conductivity on the go with the temperature sensor data for further decision-making. This learning process is not considered in this work as the time of analysis (1 day) is small compared to the time scale of the wear of insulation (typically in the order of several months).

5.6. Example 2: Smart manufacturing process (Turning)

The proposed methodology is demonstrated for real-time quality monitoring and control of a smart turning process. We first provide a brief background to the turning process, then consider a conceptual smart turning process and use it to illustrate the proposed performance evaluation methodology.

5.6.1. Introduction to Turning process

Turning is a manufacturing operation that falls under the category of machining operations, which involve transforming a raw material into desired shapes through controlled material removal processes. Turning involves removal of material from the surface of a rotating cylindrical part while moving linearly in the axis of rotation and against a cutting tool; this operation is generally carried out on a lathe (manual or computer-controlled). This turning operation results in the removal of material along an axis perpendicular to the axis of rotation of the part, i.e., along the radial direction when considering the cross section of the cylindrical part. The material of the cutting tool is typically harder than the material of the part. Several types of turning operations exist such as straight turning, taper turning, contour turning, facing and grooving. In this example, we consider only straight turning, where the final diameter of is the same all along with part. Refer to [107] for more details regarding turning operations.

The inputs for this example include: (1) Feed rate (f), which refers to the speed at which the cutting tool is fed onto the cylindrical part, (2) Cutting speed (V), which is the relative speed between a part and the cutting tool along the axis of the part, and (3) Depth of cut (d), which is the distance travelled by the cutting tool along the radial direction of the part. If the cutting tool is initially in contact with the part, then the distance travelled by the cutting tool equals the amount of material removed of the part along with radial direction. Let the initial and target final diameters of the part be represented as D_o and D_f respectively. Therefore, the depth of the cut, initial and target diameters are related as shown in Eq. (5.9) assuming the cutting tool is in contact with the part.

$$d = \frac{D_o - D_f}{2} \tag{5.9}$$

Over several turning operations, the tip of the cutting tool that comes in contact to the part wears out; this type of wear is known as flank wear (w). The presence of flank wear results in a final diameter greater than the target diameter. Therefore, the position of the cutting tool needs to be adjusted in order to achieve the desired target diameter. This adjustment to the position of cutting tool is termed as tool wear compensation (δ). Therefore, the amount of tool wear compensation increases with the amount of cutting time, i.e., the time spent in the turning process. Consider n finished parts produced by a turning process, and let their tool wear compensations be denoted as $\delta_i, i = 1,2, \dots n$. Therefore, the revised depth of cut after considering tool wear compensation is given as

$$d_i = \frac{D_o - D_f}{2} + \delta_i \tag{5.10}$$

where d_i refers to the depth of the cut of the i^{th} part. The flank wear cannot be observed directly but empirical expressions are available in [1] given as

$$w_i = k_w V^{\alpha_w} f^{\beta_w} d_i^{\gamma_w} (t_{w,i} + t)^{\sigma_w} \tag{5.11}$$

where k_w, α_w, β_w and σ_w are model parameters that needs to be estimated using experimental data. t and d_i refer to the cutting time spent on part i and depth of cut of i^{th} part, and w_i refers to the tool wear on the i^{th} part at spending time t on it. $t_{w,i}$ refers to the time that could have been spent on the i^{th} part to achieve the same tool wear that is achieved after processing $(i - 1)$ parts. Let W_i refer to the total wear after processing i parts. Similarly, if W_{i-1} refers to the tool wear after $(i - 1)$ parts, then $t_{w,i}$ can be calculated as

$$t_{w,i} = \left(\frac{1}{k_w} V^{-\alpha_w} f^{-\beta_w} d_i^{-\gamma_w} W_{i-1} \right)^{\frac{1}{w}} \tag{5.12}$$

Let $\overline{D}_i = D_o - d_i$ represent the mean diameter after the turning operation. If L denotes the length of the part, then the total cutting time for the i^{th} part can be calculated as [107]

$$t_{c,i} = \frac{\Pi(D_o - d_i)L}{Vf} \tag{5.13}$$

Using Eqs. 5.11, 5.12 and 5.13, the total wear after processing i parts can be evaluated as

$$W_i = k_w V^{\alpha_w} f^{\beta_w} d_i^{\gamma_w} (t_{w,i} + t_{c,i})^{\sigma_w} \tag{5.14}$$

The presence of tool wear causes a drift in the final diameter of the part; this drift (Δ_i for part i) with a clearance angle (sometimes referred to as relief angle) θ can be estimated as

$$\Delta_i = 2 \times (w_i - W_{i-1})\tan(\theta), \text{ for } 0 \leq t \leq t_{c,i} \tag{5.15}$$

The clearance angle is the angle made by the surface of the flank tool with finished surface of the part [108]. The drift calculated using Eq. (7) represents the additional drift that occurs in the current part i when compared to the previous part $i - 1$. To achieve the target diameter, tool wear compensation (δ_i) is initiated to counteract the effect of the increased part diameter due to the drift (Δ_i). The final diameter after considering tool wear compensation and drift can be computed as

$$D = D_f - 2 \times \delta + \mathit{\Delta} \tag{5.16}$$

Therefore, the quality losses due to the deviation from the target diameter can be quantified as [107]

$$Q_W = \int_{T_{i-1}}^{T_i} (2 \times \delta_i - \Delta_i)^2 dt \tag{5.17}$$

In Eq. (5.17), Q_W represent the quality losses due to deviations from target diameter. l_w represent the loss factors corresponding to Q_W respectively. T_i represents the cumulative machining time over i parts.

5.6.2. Smart Turning process

In most existing implementations of the turning process, the output diameter of the i^{th} part is measured externally and the tool wear compensation is implemented accordingly for the $(i+1)^{th}$ part using the deviation in the diameter for the i^{th} part. This implementation works well for (1) parts with shorter lengths, where the variation in diameters at the head and tail portions of the part is not significant, and (2) longer parts for which the tolerance is greater than the deviations observed in the diameter measurements. However, such a strategy may not be suitable for manufacturing parts with ultra-high precision requirements, i.e., the variation in the diameters at the head and tail portions of the part is greater than the tolerance required. In such cases, two options are possible to achieve the desired measurements: (1) use appropriate cutting tool and relief angle such that the deviations due to tool wear are less than the required tolerance, and (2) make real-time changes to the tool wear compensation as the part is being manufactured. Traditionally, the first option is implemented for manufacturing ultra-high precision parts, but in this work, we consider the second option, where real-time tool wear compensation is implemented. In addition, we look at the various uncertainty sources affecting the tool wear compensation and impose the appropriate compensation accordingly.

With the background of the turning process in Section 5.6.1, we consider a smart turning process. In a smart turning process, the diameter of the part is measured while the part is being manufactured; these diameter measurements are transmitted to a computational system, which then computes the necessary tool wear compensation (is equivalent to a control action in control theory literature) and implements it while the part is in being processed. Several in-process dimensional measurement techniques are available to measure the output diameter of a part. Different classes of techniques exist such as mechanical techniques, optical techniques, pneumatic techniques, and ultrasonic techniques. Brief descriptions of all the above techniques are available in [109]. In this work, we assume a scanning laser beam technique (an optical technique), which emits a high-speed laser beam onto the part, and diameter measurements are made based on the shadow produced due to the obstruction of the laser beam [110].

Uncertainty sources: Several sources of uncertainty may exist at several stages in the above-described smart turning process. The physical subsystem here refers to the part that is being manufactured, sensors are associated with the diameter measurements, the computational subsystem computes the tool wear compensation, and the actuation subsystem represents the mechanical system used to implement the tool wear compensation.

The uncertainty in the physical subsystem refers to the uncertainty in the model parameters of the empirical models discussed in Section 5.6.1. Uncertainty in the input to the physical system may refer to the actual clearance angle that is implemented, which may be slightly different from the intended angle. Sensor uncertainty refers to the uncertainty in the diameter measurements from the scanning laser beam technique. Similarly, the uncertainty in the computational subsystem may be due to the unavailability of necessary resources to perform the computation, uncertainty in the communication between the sensors and the computational subsystem, and between the computational and actuation systems. In this example, we do not consider a distributed system but a single computational node. Thus, there exists no issue of communication uncertainty between the computational nodes. The illustrative example in Section 5.5 considered a distributed computational subsystem. Uncertainty in the actuation subsystem may be due to the variation in the actual tool wear compensation implemented, which could be slightly different from the computed tool wear compensation.

5.6.3. Performance analysis

Problem Parameters: The specifications and dimensions of the parts are provided in Table 5.9. The process parameters such as the feed rate and cutting speed are assumed to be $60\,m/min$ and

0.065 mm/rev respectively. The initial tool wear compensation is assumed to be equal to 0.009 mm. The values of model parameters for several empirical models discussed in Section 5.6.1 are obtained from [107], and provided in Table 5.10. In addition, the uncertainty in the actual tool wear compensation that is implemented is quantified using a Gaussian distribution, where the mean is the intended tool wear compensation and a standard deviation of 0.0005 mm. The uncertainty in the sensor measurements is also represented using a Gaussian distribution with a standard deviation of 0.0025 mm. The uncertainty in the clearance angle is also represented using a Gaussian distribution with a mean and a standard deviation of 15 and 0.5 degrees.

Table 5.9. Specifications of the parts from the cyber-physical turning process

Parameter	Value
Initial diameter (D_o)	100 mm
Final diameter (D_f)	98 mm
Length of part (L)	100 mm
Lower bound of target diameter ($D_{f,L}$)	97.98 mm
Upper bound of target diameter ($D_{f,U}$)	98.02 mm

In this example, Gaussian distributions are used to represent several uncertainties for illustration purposes and can be replaced by their actual probability distributions, if available. The variation in model parameters, clearance angle and tool wear compensation represents aleatory uncertainty, i.e., the values of these parameters vary across several parts. However, these values are constant for a particular part but these values are unknown; therefore, they represent epistemic uncertainty sources when a single part is considered. Thus, it should be noted that the same parameters can be either aleatory, when a population of parts are considered or epistemic, when a single part is considered. When multiple parts are processed, their associated parameter values need to be calibrated using the diameter sensor measurements. The probability distributions in Table 5.10 are considered as prior distributions for the calibration process.

Table 5.10. Parameters in the tool wear empirical model

Parameter	**Value**
k_w	$N(8.2961 \times 10^{-5}, 8.2961 \times 10^{-7}$
α_w	$N(2.747, 0.02747)$
β_w	$N(1.473,0.01473)$
γ_w	$N(1.261, 0.01261)$
σ_w	$N(0.43, 0.0043)$

For the computational subsystem, asynchronous interaction is assumed between the sensors and the computational subsystem, and between the computational and actuation subsystems. The sensor data are assumed to be sent in three data packets and two packets are required for successful communication. Similarly, one of two data packets is necessary for a successful communication between the computational and actuation systems. The reliability (probability that a data packet is successfully transmitted) of the first packet is assumed to be 0.95. The reliability of the following data packet is obtained using the conditional probability table, given in Table 5.11. The probability that the necessary computational resources are available is assumed as 0.95. Therefore, a successful total wear compensation analysis requires successful communication between sensors and computational subsystem, computational resources and successful communication between computational and actuation subsystem.

Table 5.11. Conditional probability table of a data packet transmission

	Successful at time step $n+1$	Unsuccessful at time step $n+1$
Successful at time step n	0.95	0.05
Unsuccessful at time step n	0.9	0.1

The length of the time step at which sensor measurements of the output diameter of a part are made is equal to 0.25 second. The goal of the computational system is to estimate the tool wear compensation using the sensor measurements of the output diameters of a part at different time steps. Estimation of the tool wear compensation for the next segment of the part is by minimizing the loss function, given in Eq. (5.18) and subject to the tolerance requirements given in Table 5.9. Due to the presence of uncertainty in the model parameters, the clearance angle, tool wear compensation and sensor uncertainty, the loss and the constraint functions are not deterministic but stochastic in nature. Therefore, the optimization formulation for tool wear compensation after considering several uncertainty sources is given as

$$\begin{gathered} Min\ E\left[\int_0^T \left(\delta_i - \frac{\Delta_i}{2}\right)^2 dt\right] \\ \text{such that} \\ \Pr\left(D > D_{f,L} \cap D < D_{f,U}\right) \geq 0.95 \end{gathered} \tag{5.18}$$

In Eq. (5.18), T represents the time of analysis. The overall goal of this example is to quantify the uncertainty in the output diameter along the length of the part after implementing the smart turning process.

Dimension Reduction: The parameters to be calibrated using the diameter sensor measurements include the model parameters in Table 5.12 ($k_w, \alpha_w, \beta_w, \gamma_w$ and σ_w), clearance angle (θ) and tool wear compensation (δ).Variance-based global sensitivity analysis is then performed to reduce the number of parameters to be calibrated. The first-order indices are used for dimension reduction; if the first-order index of a parameter is less than 0.1, then that parameter is removed from calibration process and is assumed deterministic at its modal value (i.e., the value with the highest probability). Using the prior distributions of the parameters, the output parameter is estimated after every time step, i.e., 0.25 second. The stratified sampling algorithm proposed by Li and Mahadevan [49] is used to compute the first-order sensitivity indices. The sensitivity indices of several parameters are given in Table 5.12.

Table 5.12. Sensitivity indices for dimension reduction

Parameter	**Sensitivity index**
k_w	0.0075
α_w	0.8585
β_w	0.1201
γ_w	0.0045
σ_w	0.0076
θ	0.0072
δ	0.0022

From the results of sensitivity analysis, only α_w and β_w are considered for calibration and the remaining parameters are fixed at their modal values, i.e., the values of $k_w, \gamma_w, \sigma_w, \theta$ and δ are fixed at their mean values (for Gaussian distributions, the modal value and the mean are equal). It should be noted that the actual values of these parameters could be different from the values that were fixed at.

DBN Model: The DBN model for the smart turning process is shown in Fig. 5.11. In Fig. 5.11, the superscript k refers to the time step of analysis. Since α_w and β_w are deterministic but unknown parameters, they do not vary with time, i.e., $\alpha_w^k = \alpha_w^{k+1}$ and $\beta_w^k = \beta_w^{k+1}$. d^k, w^k, Δ^k refer to the depth of cut, tool wear and drift at the k^{th} time step. D^k and D_S^k refer to the output diameter and its sensor measurement respectively. R_1^k refers to the availability of computational resources at k^{th} time step while C_1 refers to the computational subsystem and not a physical variable. The wear at the current time step is affected by the wear in the previous time step through the parameter $t_{w,i}$, as shown in Eq. (5.11). The drift in the current time step in affected by the wear in the previous time step as shown in Eq. (5.15). The sensor measurements, D_S^i are transmitted to the computational subsystem, which under the availability of necessary resources computes the tool wear compensation in the following time step (δ^{k+1}).

Estimation of model parameters and tool wear compensation: Parameter estimation is carried out in each time step using the sensor measurement. We used the particle filtering technique and used to Sequential-Importance Resampling (SIR) algorithm [67] with 1000 samples. A brief introduction to the SIR algorithm is provided in Section 2.7.2. The posterior samples are then used to estimate the tool wear compensation in the next time step by solving the optimization problem detailed in Eq. (5.18). This process is repeated for all the time steps until a part is completely processed. The prior and posterior distributions of α_w and β_w, along with their true values, for a sample part are given in Fig. 5.12. The insensitive parameters ($k_w, \gamma_w, \sigma_w, \theta$ and δ) are fixed at their modal values; however, it should be noted that the true values may be different than the values they are fixed at. The true values of these parameters are given in Table 5.13. The true values of all parameters are denoted using a superscript 'T', as shown in Table 5.13.

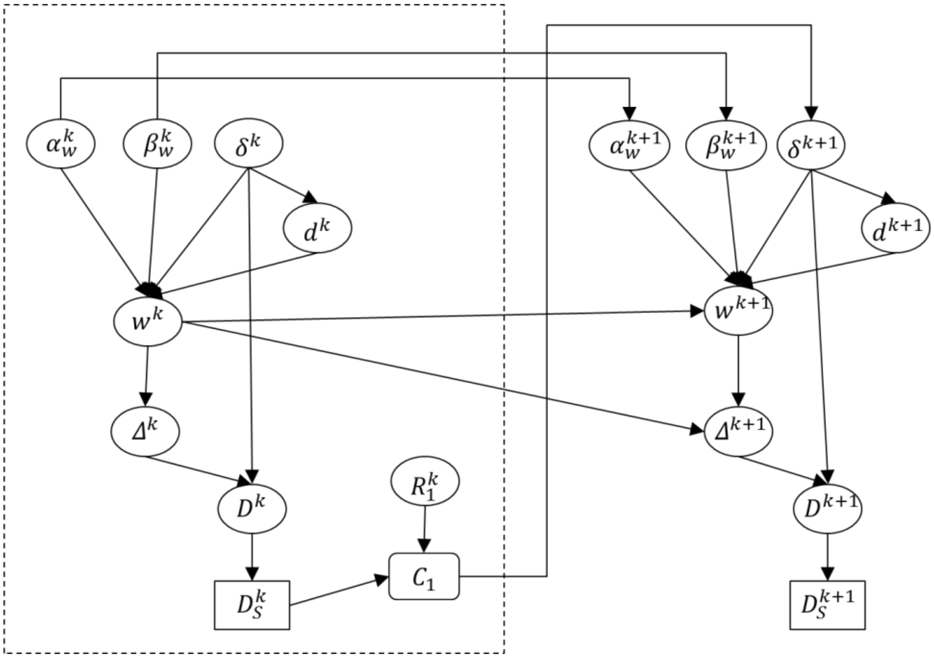
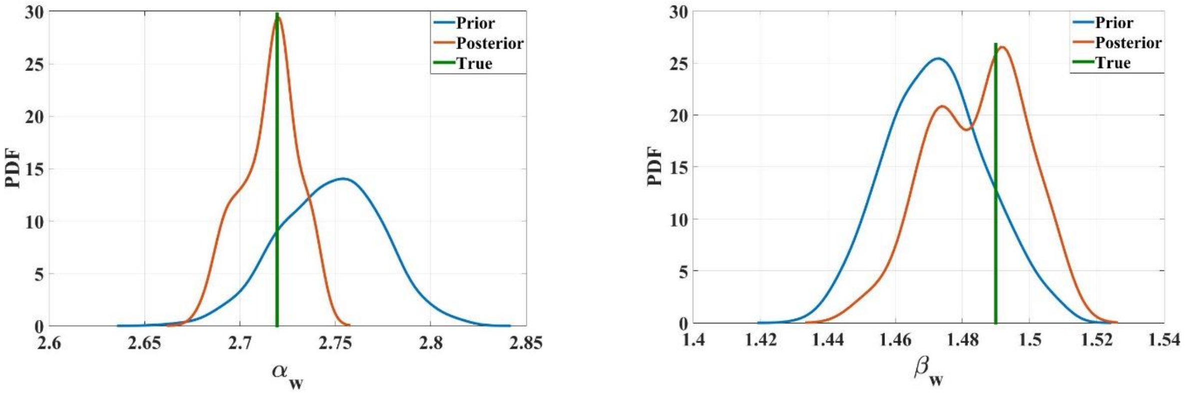


Fig. 5.11. DBN model for the smart turning process



(a) (b)

Fig. 5.12. Prior and Posterior distributions of calibration parameters

Table 5.13. Underlying true values of parameters

Parameter	**Sensitivity index**
k_w^T	8.2676e-5
α_w^T	2.7195
β_w^T	1.4899
γ_w^T	1.2627
σ_w^T	0.4247
θ^T	15.0427
δ^T(initial)	0.0093

During real-time control, an appropriate tool wear compensation is implemented to reduce the losses due to deviations from the target diameter. However, the actual tool wear compensation at each time step may be different due to the presence of associated uncertainty in tool wear compensation. Within each time step, a particular length of the part is processed; this length can be estimated using Eq. (5.13). As mentioned above, it is assumed that the length of each time step is 0.25 second. However, it may be possible that the part in its final time step may be processed for less than 0.25 second, depending on the length of the part that needs to be processed. Let K represent the total number of time steps required to process a part. Let $L_j, j = 1,2,3 \dots K-1$ represent the lengths of segments of a part processed in each of first $K-1$ time steps. Therefore, the length of the part to be processed in the last time step can be computed as $L_K = L - \sum_{j=1}^{K-1} L_j$. If the tool wear compensation in the final time step is δ_K, then the time spent in processing a length of L_K can be computed using Eq. (5.13). Let t_K represent the length of last time step, then the total processing time for a part can be computed as $T = (K-1) * \epsilon_t + t_K$. ϵ_t refers to the length of time step, which is assumed to be equal to 0.25 second in this illustration example. Thus, real-time quality control of turning process is affected by several uncertainty sources.

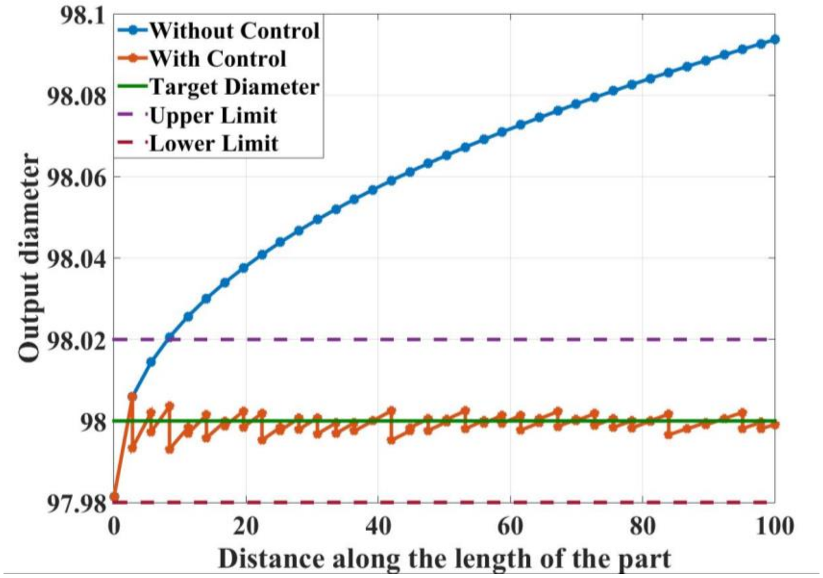


Fig. 5.13. Comparison of output diameter profiles with and without real-time control

Fig. 5.13 provides a comparison of the output diameter profiles in two cases, with and without the real-time control assuming the same initial conditions. In the case of no control, the tool wear compensation remains the same as at the initial value; however, in the real-time control, the tool wear compensation changes at each time step with the diameter sensor measurements.

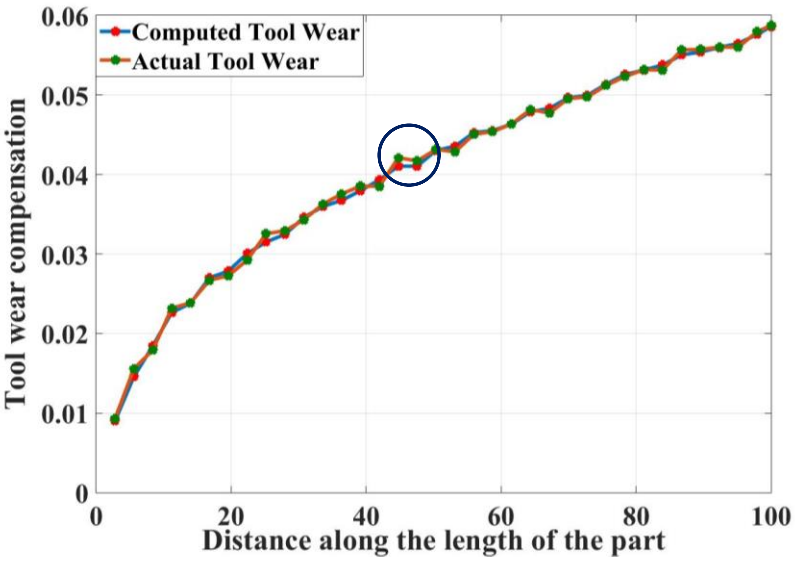


Fig. 5.14. Comparison of the computed and the actual tool wear compensation

Fig. 5.14 shows the change in the tool wear compensation at each time step. In Fig. 5.14, the computed tool wear compensation and the actual tool wear compensation implemented are shown. The difference between the two plots in Fig. 5.14 is due to the presence of uncertainty in the physical actuation of tool wear compensation. The following observations can be made from Fig. 5.14.

1. Computed tool wear compensation increases with every time step. However, the actual tool wear implemented in a time step can be lower than that implemented in the previous time step if the tool wear uncertainty is more than the increase in the computed tool wear compensation.
2. In the region identified in Fig. 5.14, the computed time wear compensation remained the same in two successive time steps. This is because of the lack of necessary computational resources to compute the tool wear compensation, as shown in Fig. 5.15.

In Fig. 5.15, ‘0’ and ‘1’ represents success and failure of action, which could be either calibration of parameters using sensor data or implementation of tool wear compensation. Calibration require successful communication between sensors and computational subsystem, and availability of computational resources. However, tool wear compensation requires successful communication between computational and actuation subsystems in addition to requirements for successful calibration. In Fig. 5.15, implementation of appropriate tool wear compensation has been unsuccessful at one instant, whereas calibration has been successful at all time instant. Therefore, unsuccessful implementation of tool wear compensation can be attributed to communication loss between computational and actuation subsystems.

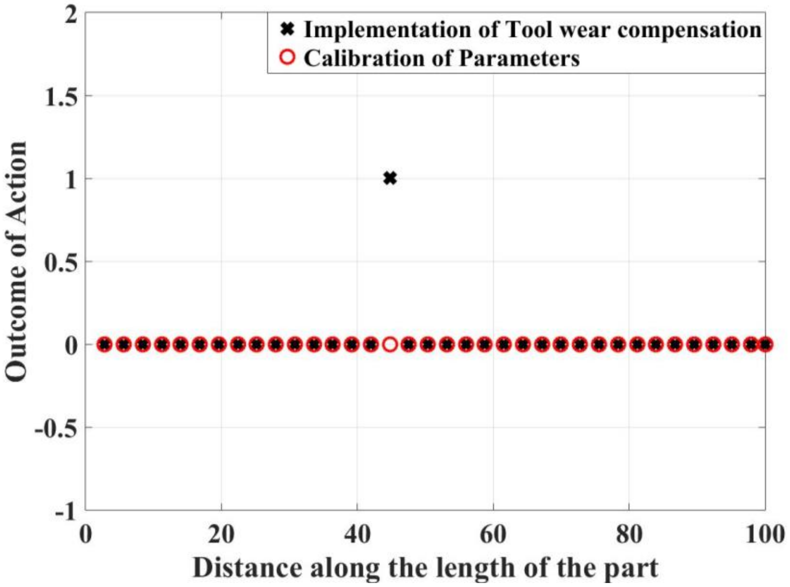


Fig. 5.15. Outcome of tool wear compensation and calibration analyses due to availability of computational resources and successful communication between several subsystems

To quantify the effects of aleatory uncertainty sources in the communication between computational and actuation systems, and sensors and computational system, resource availability for the computational system, uncertainty in the clearance angle, uncertainty in the model parameters to estimate the tool wear, and uncertainty in the physical implementation of tool wear, the above-described steps are repeated several times. This repetition simulates the processing of multiple parts using the turning operation and thus the aleatory variability across multiple parts. Fig. 5.16 represents the diameter profiles of several parts, after considering both aleatory and epistemic uncertainty sources. It should be noted that the epistemic uncertainty is considered when performing the real-time control of each individual part. After obtaining the diameter profiles, the uncertainty in the diameter along the length of the part can obtained by constructing a probability distribution using the data points at the required point of interest along the length of the part.

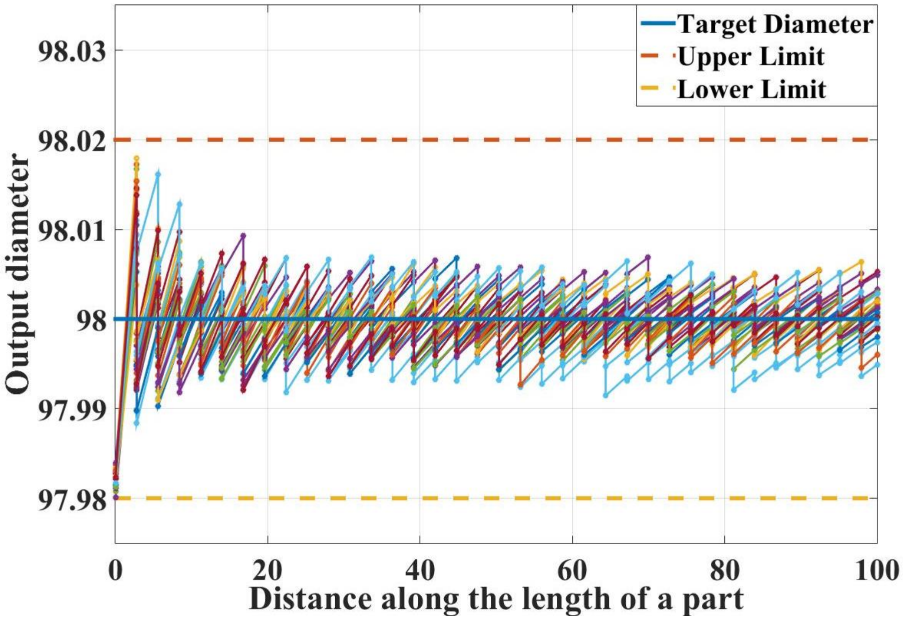


Fig. 5.16. Diameter profiles of parts considering both aleatory and epistemic uncertainty in the smart turning process

5.7 Summary

This chapter developed a model-based framework for uncertainty quantification and aggregation in a smart system for its performance assessment to enable design-time decision-making. As discussed in Section 5.1, earlier studies have primarily adopted data-driven approaches such as reliability block diagrams using failure rates of components that are experimentally obtained or expensive simulation models such as Simulink for performance evaluation of smart system. Data-driven approaches may not be always feasible in complex systems and running Simulink models multiple times when considering uncertainty sources can be computationally expensive. In addition, a simulink model does not facilitate real-time inference when new data is available. For the above reasons, we developed a Bayesian network-based approach for performance evaluation of smart systems under uncertainty.

Uncertainty in the physical subsystem may be due to uncertain model parameters and model inadequacy. Uncertainty in the computational subsystem is due to the uncertainty in the availability of hardware resources and in the network communication. A smart system containing physical, sensing, computational and actuation subsystems is analyzed as a feed-forward system in time and modeled using a Dynamic Bayesian network. The computational subsystem may have communication between multiple nodes; this interaction is also modeled using a DBN, resulting in a two-level DBN for modeling a smart system. Physics models and/or data are used to establish the conditional dependence relationships for the physical and actuation subsystems. DBNs corresponding to basic interaction patterns such as 2-node asynchronous and synchronous are detailed. Complex interaction patterns can be broken down into these basic patterns for performance evaluation. The proposed methods are demonstrated for two examples: (1) the design of a smart indoor heating system that enables room-by-room temperature control, and (2) real-time control of a smart turning process.

The first example is a design-time analysis problem where an appropriate design configuration (sensor and network types) has to be selected that maximizes the reliability under budget constraints. In this problem, reliability is defined as the probability that temperature in various rooms are according to their pre-set limits. The design-time decision-making requires the implementation of the proposed performance evaluation for multiple design configurations and choosing the best configuration among them. The second example considered real-time control (real-time decision-making, as opposed to design-time decision-making in the first example) of a smart turning process. Also, the first example considered a distributed computational system whereas the second example considered a single-node computational system. The presence of a distributed computational system results in an additional uncertainty source, i.e., the communication between the individual nodes.

The proposed methodology is generic and can be extended to more complex interaction patterns between and within the subsystems. An example with such properties of complex computational interaction pattern is a smart grid, which has a collection of sensors (smart meters), computational, actuation (electricity transfer mechanism such as lines) and physical subsystems (houses). The inputs (power generation sources such as fossil fuels, wind and solar) are associated with variability due to environmental conditions. In the case of a smart grid, the user requirements change with time every day; these user requirements can be considered similar to occupant comfort levels in the smart indoor heating system. A smart grid consists of several utility units or substations, and each utility unit can be modeled as a computational node. These units collect sensor data from a region and communicate with each other to find an optimum way to meet the electricity demands. The proposed methodology is able to handle such systems; however, the computational effort will grow with the size of the system. The strategies for sensitivity analysis, dimension reduction, and joint distribution approximation may be employed in scaling up the methodology to large systems.

After considering systems where interactions between several subsystems occur with a time lag, the next chapter considers systems where interactions occur simultaneously such as multi-physics systems.

CHAPTER 6

PERFORMANCE ASSESSMENT IN COUPLED MULTI-PHYSICS SYSTEMS

6.1. Introduction

Systems with coupled interactions between multiple physics disciplines are often encountered in engineering applications such as hypersonic aircraft (fluid, structural and thermal interactions) [111], and long-span bridge (fluid and structural interaction) [112]. In such coupled systems, the interactions between several subsystems occur simultaneously as opposed to the software-controlled feedback control systems discussed in Chapter 5, where interactions between several subsystems occur with a time lag. As discussed in Section 1.1, models are often employed for each individual subsystem (may represent a different physics in a multi-physics system) and are run iteratively until the results from individual models are compatible; this approach is commonly referred to as fixed point iteration [113]. This results in a high computational expense as the individual models are run multiple times to reach compatibility. This computational expense is further exacerbated in the presence of uncertainty, as the multi-physics system evaluation needs to be repeated a large number of times at the same design input (thus simulating multiple realizations of the uncertain variables), in order to compute the probability distribution of the output.

As discussed in Section 1.1, two approaches have been developed in the literature to counter the large computational expense due to the use of physics-based models for performance assessment of multi-physics systems under uncertainty. The first approach reduces the large computational expense by replacing the expensive disciplinary computational models with inexpensive surrogate models. A variety of surrogate modeling techniques have been studied, such as Polynomial Chaos models, Kriging (Gaussian Process) models and Neural Networks [10, 114]. These surrogates are termed as deterministic (or algebraic) surrogates as they provide point-output values, corresponding to a realization of design input and uncertain variables. It should be noted that Gaussian Process surrogate results in a stochastic output; the stochasticity merely represents the uncertainty in the prediction; however, this stochastic prediction corresponds to a realization of design input and uncertain variables.

The second approach to reduce the computational effort is to decouple the disciplinary analyses. Du and Chen [11] proposed the collaborative reliability analysis framework, where the multidisciplinary compatibility are treated as optimization constraints when performing reliability analysis in the RBDO framework. Mahadevan and Smith [115] proposed a decoupled framework, where the coupling variables between several individual disciplinary analyses are estimated through First-Order Second Moment (FOSM) [31] approximations and later used in reliability analysis. Sankararaman and Mahadevan [116] developed a decoupled likelihood-based approach (LAMDA) to obtain the distributions of the coupling variables by using the First-Order Reliability Methods (FORM) [31].

The existing methods for multidisciplinary analysis (MDA) and optimization under uncertainty have several shortcomings. When surrogates are used to replace the individual disciplinary computational models, the accuracy of those surrogates is of concern since it affects the accuracy of the optimization results. Note that repeated runs of the surrogate model at the same design values with multiple realizations of the uncertainty variables are required in order to compute the reliability or robustness constraints (or probabilistic objectives). The decoupled approaches have used first order methods (such as FOSM and FORM) to decouple the bi-directional coupling between individual disciplines, and the coupled analysis is treated as a uni-directional analysis [116]. After decoupling, the severed coupling variable is treated as an additional input to the individual disciplinary analyses. A drawback of this approach is that it loses the statistical relationship between the design input and the severed coupling variables, i.e., the distribution of the severed coupling variable remains the same irrespective of the value of the design input variables. In reality, the distribution of the coupling variable changes with the values of design input variables.

Thus, there exists a need for the development of an approach that is computationally efficient yet overcomes the approximations introduced by first-order estimates of coupling variable distributions, and is scalable to coupled problems with multiple disciplines (more than two). This chapter also uses a surrogate-based strategy but a key distinction with the existing approaches is the use of a probabilistic (i.e., distribution) surrogate as opposed to an algebraic surrogate of the model output. An algebraic surrogate (also referred to as deterministic surrogate in Liang [117]) provides a point output value for a given set of inputs also opposed to a probabilistic surrogate (such as a Bayesian network), which provides a distribution output for a given set of inputs. In other words, the probabilistic surrogate is constructed in the probability space whereas the deterministic surrogate is constructed in the variable space.

It should be noted that response surrogate models (also referred to as algebraic) such as Kriging also provide distributions for output predictions but such distributions arise due to sparse or noisy data [117]. In the presence of sufficient and precise data, the uncertainty in the output prediction decreases. When developing algebraic surrogates, the inputs for such surrogates are the design variables and uncertain variables. To obtain the uncertainty in the output prediction at a given input, the algebraic surrogate needs to be evaluated multiple times at the same design input for multiple realizations of uncertain variables. In contrast, a single evaluation of the probabilistic surrogate provides the entire output distribution considering all the uncertain variables at a given value of the design variable [117].

In a generic Bayesian network, prediction (i.e., forward propagation) and inference (i.e., backward or inverse propagation) do not have analytical solutions; instead, sampling-based techniques such as Markov Chain Monte Carlo (MCMC) methods need to be used, which are computationally expensive and not exact (see Chapter 2). This chapter considers three types of multivariate distribution models (that have analytical solutions for the forward and inverse problems) as approximations to the general Bayesian network. The three models are: (1) Multivariate Gaussian (MG), (2) Gaussian Copula (GC), and (3) Gaussian Mixture Model (GMM).

Copulas have been previously used in reliability analysis and RBDO for correlated input random variables [117]. Liang and Mahadevan [118] approximated the Bayesian network with the Gaussian copula and used it for single objective reliability-based MDO and multi-objective RBDO [119]. A mixture model represents the joint distribution of multiple variables through a weighted combination of individual multivariate distributions. Mixture models are typically used to model multi-modal distributions. When the individual components are modeled using Gaussian distributions, then the mixture model is known as a Gaussian Mixture Model. A GMM can be considered as an extension of MG for modeling multi-modal joint distributions. Mixture models have been studied in reliability analysis [120], traffic flow forecasting [121], process monitoring [122], uncertainty quantification in dynamic systems [123] and probabilistic community discovery [124]. In this chapter, we use these probabilistic surrogates with analytical inference for efficient performance assessment in coupled multi-physics systems.

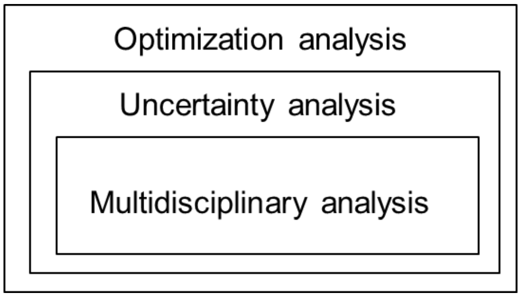


Fig. 6.1. Nested three-loop analysis for multidisciplinary design under uncertainty

The results from performance assessment are typically used in decision-making process either at design-time (such as design optimization) or at run-time (such as control). In the chapter, we illustrate the use of developed performance assessment framework for design optimization of multi-physics under uncertainty; however, these techniques can also be used for run-time decision-making. Optimization under uncertainty adds a third layer of iterative computation in order to converge to the optimum solution. Overall, the design of multidisciplinary systems under uncertainty can be regarded as a nested three-loop process as shown in Fig. 6.1, with the convergence analysis in the innermost loop, uncertainty analysis in the middle loop, and optimization analysis in the outermost loop. Design optimization under uncertainty has been pursued primarily in three directions: (1) Reliability-based design optimization (RBDO) [125], (2) Robustness-based design optimization (RDO) [126] and (3) Reliability-based Robust Design Optimization (RBRDO) [127]. RBDO generally considers the design of systems with reliability requirements and RDO considers robustness requirements, whereas RBRDO considers both reliability and robustness requirements. The reliability and robustness computations are performed in the middle loop of Fig. 6.1, as part of the uncertainty analysis. In this chapter, we exploit the analytical solution of MG, GC and GMM models for efficient performance assessment and design optimization of coupled multi-physics systems.

6.2. Generation of training data for a probabilistic surrogate

In this section, the one-pass analysis approach proposed by Liang and Mahadevan [118] to generate training data for a multi-disciplinary system is briefly discussed. Then, we define a two-input and a two-output system to illustrate the probabilistic modeling.

6.2.1. One-pass multidisciplinary analysis

Consider a two-discipline coupled multi-physics system as shown in Fig. 6.2. Let $\boldsymbol{X_D}$, $\boldsymbol{X_U}$ and $\boldsymbol{X_{D,U}}$ represent the set of deterministic design variables, uncertain but non-design variables and design variables with uncertainty respectively. Let $\boldsymbol{g_1}$ and $\boldsymbol{g_2}$ represent the outputs of the coupled systems, which are propagated through A_3 to obtain the system outputs $\boldsymbol{f}$ and $\boldsymbol{c}$. The coupling variables between A_1 and A_2 are represented as $\boldsymbol{u_{12}}$ and $\boldsymbol{u_{21}}$ respectively.

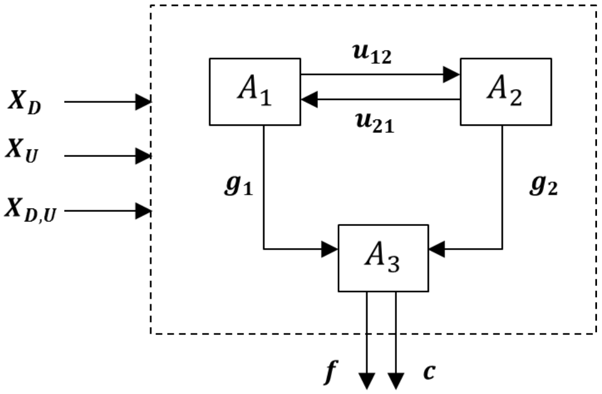


Fig. 6.2. A conceptual two-discipline coupled system

Given the physics models of the two individual disciplinary models, we perform a one-pass analysis to generate necessary training data, which is later used to build a probabilistic surrogate. One-pass analysis refers to one passage of all the individual disciplinary models. To reach compatibility among individual disciplines, the one-pass analysis need to be carried out multiple times with the outputs of previous one-pass analysis as the inputs for the following one-pass analysis. A possible path for a one-pass analysis is shown in Fig. 6.3.

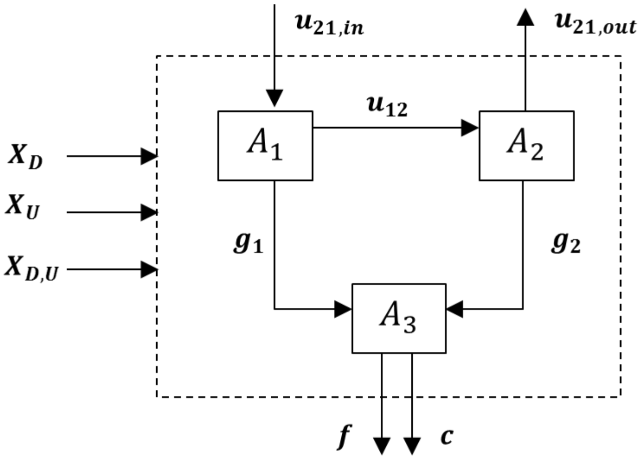


Fig. 6.3. One-pass analysis path for a two-discipline coupled system

In Fig. 6.3, $\boldsymbol{u}_{21,in}$ and $\boldsymbol{u}_{21,out}$ are the same set of coupling variables but before and after carrying out a one-pass analysis. One-pass analysis refers to one passage through the individual disciplines, here A_1, A_2 and A_3. Compatibility is assumed to be achieved when $\boldsymbol{u}_{21,in}$ = $\boldsymbol{u}_{21,out}$. To achieve compatibility, a few iterations of the coupled analysis are necessary in traditional MDA. However, instead of iterations to convergence, we build a Bayesian network using one-pass data, and then impose the compatibility condition on the Bayesian network. This saves tremendous computational expense.

Using the one-pass training data, Liang and Mahadevan [118] constructed a Gaussian copula, which was used for two-disciplinary analysis under uncertainty. In this chapter, we extend this idea to three-discipline and four-discipline multidisciplinary systems.

6.2.2. Probabilistic modeling

Section 2.6 presented a brief introduction to a Bayesian network, which is used as a probabilistic model to represent the joint representation over a set of random variables. Fig. 6.4(a) shows a conceptual example, which is later used in the discussion of probabilistic modeling. The model M has two inputs X_1 and X_2; and two outputs Y_1 and Y_2. Based on the model in Fig. 6.4(a), a Bayesian network can be constructed as shown in Fig. 6.4(b). The nodes represent random variables, and edges represent the probabilistic dependence relations between them.

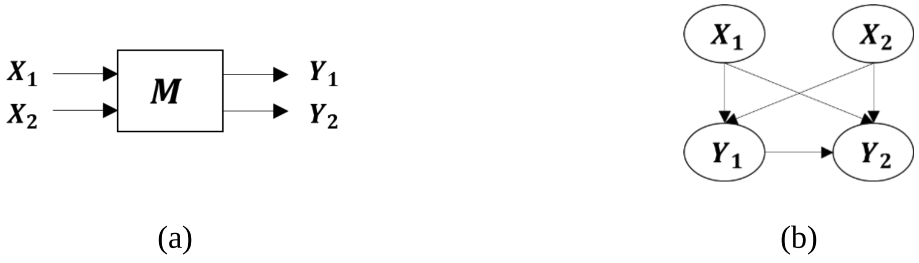


Fig. 6.4. A Bayesian network surrogate shown in (b) of a model with two inputs and two outputs shown in (a)

The Bayesian network can be used for model prediction $f(Y_1, Y_2| X_1 = x_1, X_2 = x_2)$, i.e., estimation of outputs Y_1, Y_2 for given values of inputs X_1, X_2 or model inference $f(X_1, X_2| Y_1 = y_1, Y_2 = y_2)$, i.e., inferring the possible values of the inputs that may have resulted in given values of outputs. The above joint conditional PDFs (for prediction or inference) are often evaluated using expensive sampling strategies such as Markov Chain Monte Carlo (MCMC) methods, for inference, and Monte Carlo analysis, for model prediction.

6.3. Performance assessment of multi-physics systems

The proposed methodology for performance assessment of a coupled multidisciplinary system is carried out in three steps: (1) Generation of the training points, (2) Construction of a probabilistic surrogate, and (3) Evaluation of system quantities of interest at inter-disciplinary compatibility.

6.3.1. Generation of training points

In this section, we extend the one-pass analysis approach developed by Liang and Mahadevan [118] to higher-order coupled multidisciplinary systems. Performing one-pass analysis requires the generation of a feasible one-pass analysis path; this path determines the sequence in which several disciplinary analyses need to be carried out to generate training points. We describe the one-pass analysis sequences for conceptual three-discipline and four-discipline coupled systems, both with bi-directional coupling, and later for a three-discipline system with bi-directional and uni-directional coupling between disciplines.

A. Conceptual three-discipline coupled system

Consider a conceptual coupled three-discipline system as shown in Fig. 6.5, where A_1, A_2 and A_3 represent the three-coupled disciplines and A_4 represent the fourth discipline, which does not have coupling with any of the other three disciplines. Let u_{ij} represent the coupling variables between any two disciplines, which are output from the i^{th} discipline and input to the j^{th} discipline. Let $g_m, m = 1,2,3$ represent the outputs from each of the three disciplines which are input to A_4, the outputs of which are the values of QoIs. The primary goal to carry out the one-pass analysis is to generate training points, which are later used to construct a probabilistic surrogate.

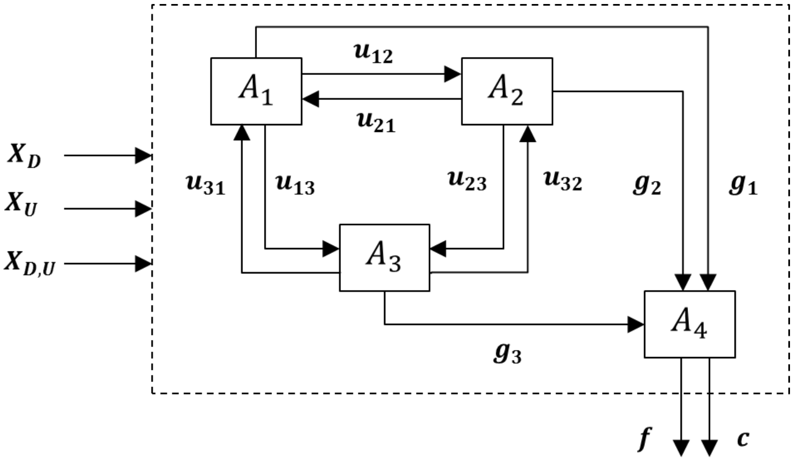


Fig. 6.5. Conceptual three-discipline coupled system

The one-pass analysis path should be developed in such a way that individual disciplines do not form a closed loop. Since there are three sets of coupling $\{A_1, A_2\}$, $\{A_2, A_3\}$ and $\{A_1, A_3\}$, we can choose the one-pass path between the three disciplines in one of several ways, such that the path does not form a closed loop. First, let us consider the coupling between A_1 and A_2. Between the two directional links, we can choose either $\boldsymbol{u_{21}}$ or $\boldsymbol{u_{12}}$ as an input for one-pass analysis. For every selected coupling variable ($\boldsymbol{u_{12}}$ or $\boldsymbol{u_{21}}$), we create two sets of the same variable, here, $\boldsymbol{u_{21,in}}$and $\boldsymbol{u_{21,out}}$ assuming $\boldsymbol{u_{21}}$ as an input, which represent the values of $\boldsymbol{u_{21}}$ before and after the one-pass analysis. The 'in' variables are inputs to a disciplinary analysis while 'out' variables refer to outputs from a disciplinary analysis. Here, $\boldsymbol{u_{21,in}}$ and $\boldsymbol{u_{21,out}}$ are inputs to A_1 and outputs from A_2 respectively. After selecting one of the coupling variables between A_1 and A_2 as an input, let us consider one of the two remaining sets – $\{A_1, A_3\}$. Again, the coupling variables in either of the directions can be chosen. Let us choose $\boldsymbol{u_{31}}$ resulting in $\boldsymbol{u_{31,in}}$ and

$\boldsymbol{u}_{\mathbf{31},out}$. In order to not form a loop, we need consider $\boldsymbol{u}_{\mathbf{23}}$ as the final input for one-pass analysis among coupling variables, resulting in $\boldsymbol{u}_{\mathbf{23},in}$ and $\boldsymbol{u}_{\mathbf{23},out}$. Thus, a possible one-pass analysis path for the three-discipline coupled system is given in Fig. 6.6.

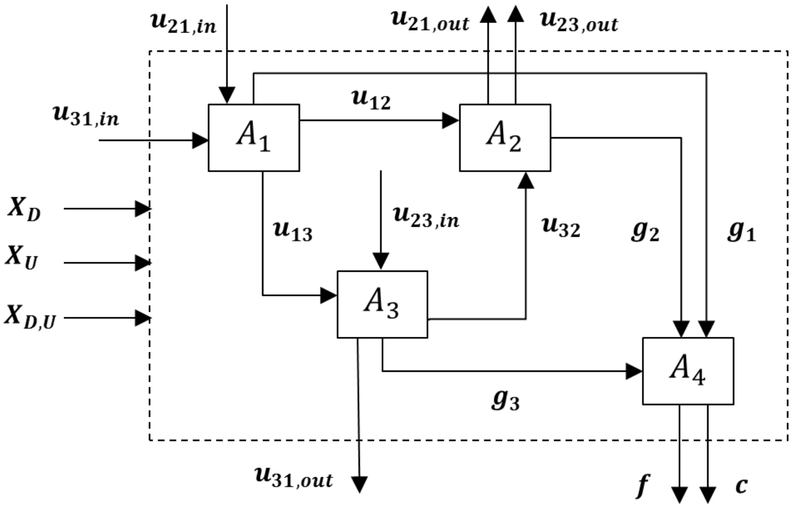


Fig. 6.6. One-pass analysis path for a three-discipline coupled system

The inputs for the one-pass analysis are the original inputs ($\boldsymbol{X}_{\boldsymbol{D}}, \boldsymbol{X}_{\boldsymbol{U}}$ and $\boldsymbol{X}_{\boldsymbol{D},\boldsymbol{U}}$) and all the ‘in’ coupling variables, i.e., $\boldsymbol{u}_{\mathbf{21},in}$, $\boldsymbol{u}_{\mathbf{23},in}$ and $\boldsymbol{u}_{\mathbf{31},in}$. The ranges of the design variables and distributions of the uncertain variables are available at the beginning of analysis. The input ranges for the coupling variables can be obtained from three sources: (1) expert opinion, (2) design of experiments, and (3) a few realizations of the coupled analysis. In the presence of available subject experts, they can provide plausible input ranges for the coupling variables. The multidisciplinary analysis can be carried out (until convergence) at several inputs values, obtained from the Design of Experiments techniques such as Latin Hypercube sampling. The values of coupling variables after convergence can be used to obtain their ranges.

After obtaining the ranges of the coupling variables, one-pass analysis is performed at several realizations of design and uncertain variables, and the ‘in’ coupling variables. We start the one-pass analysis with A_1 where the values of necessary design and uncertain variables, $\boldsymbol{u}_{\mathbf{21},in}$ and $\boldsymbol{u}_{\mathbf{31},in}$ are available. Next, we perform A_3 (and not A_2, as it requires values of u_{32} from A_3) using the values of $\boldsymbol{u_{13}}$ from A_1 and $\boldsymbol{u_{23,in}}$. Later, we perform A_2 and then A_4 in that order. Next, the values of inputs and the outputs from all the individual disciplines are used as training points for the construction of a probabilistic surrogate. If the samples of uncertain variables ($\boldsymbol{X_U}$) are considered in surrogate construction, the resulting surrogate would have been an algebraic surrogate. In a probabilistic surrogate, the samples of uncertain variables ($\boldsymbol{X_U}$) are not considered, and therefore, the output prediction is a distribution considering all values of the uncertain variables.

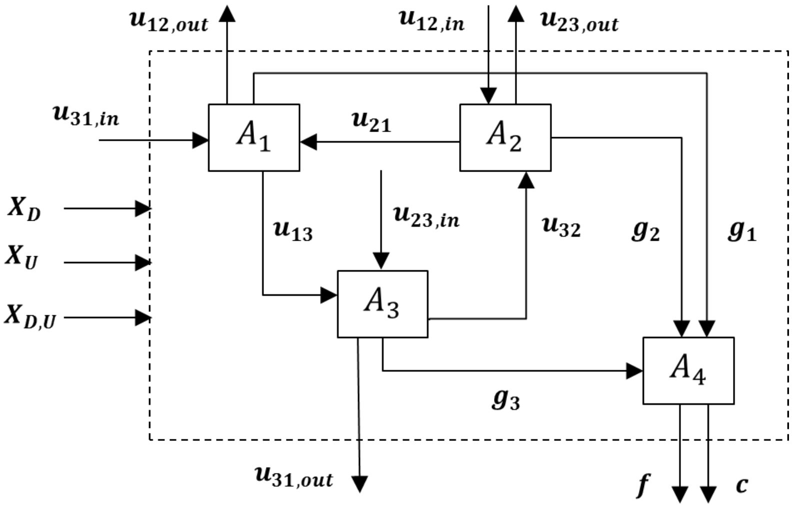


Fig. 6.7. Incorrect one-pass analysis path for a three-discipline coupled system

Earlier in this section, it was mentioned that the one-pass analysis path (sequence of individual disciplines) should be selected in such a way that the disciplines do not form a closed loop. For illustration, Fig. 6.7 shows a one-pass analysis path that forms a closed loop; this path does not enable the generation of training points for the reasons below. For one-pass analysis, we require a discipline whose inputs are all known (such as A_1 in Fig. 6.6). The one-pass analysis starts with this discipline and all the other disciplinary analysis are performed in succession, as discussed earlier in this section. For the one-pass analysis path in Fig. 6.7, there is not a single discipline, for which all the inputs are known. $\boldsymbol{u_{21}}$ is required for A_1, which is an output of A_2; A_2 requires $\boldsymbol{u_{32}}$ from A_3 and A_3 requires $\boldsymbol{u_{13}}$ from A_1. Thus, one-pass analysis path that forms a closed loop is not feasible for generating training points.

B. Conceptual four-discipline coupled system

After detailing the one-path analysis path for a three-discipline coupled system, we now demonstrate it for a four-discipline coupled system. An example of a four-discipline multidisciplinary system can be found in Culler and McNamara [128]. We show one of the several paths for one-pass analysis that is feasible for generating training points. Fig. 6.8 shows a four-discipline coupled multidisciplinary system, and Fig. 6.9 shows a possible one-pass analysis path.

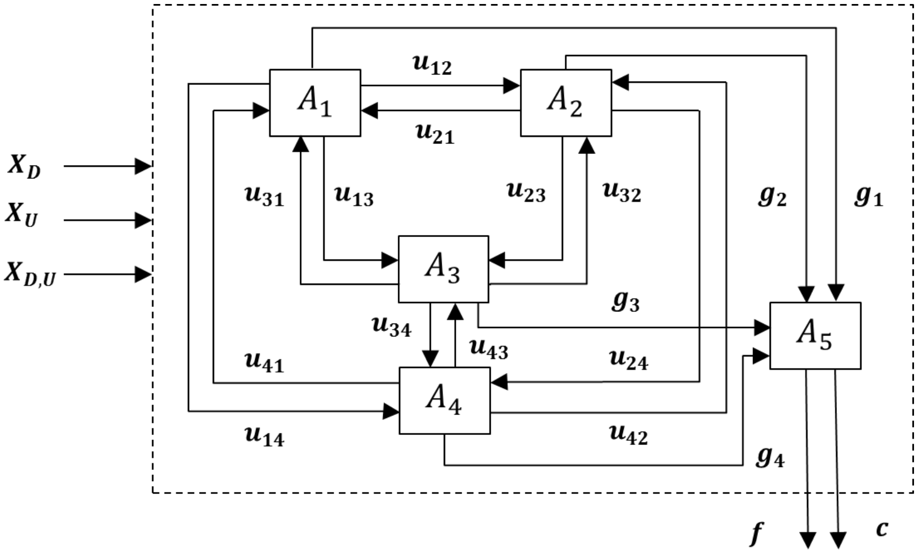


Fig. 6.8. Conceptual four-discipline coupled system

In the one-pass analysis path shown in Fig. 6.9, the analysis starts with A_1 with the design and uncertain variable inputs, and associated 'in' coupling variables ($u_{21,in}$, $u_{31,in}$ and $u_{41,in}$). A_2 is performed after A_1 using values of u_{12} (obtained from A_1), $u_{32,in}$ and $u_{42,in}$. A_3 follows A_2 using values of u_{13} (from A_1), u_{23} (from A_2) and $u_{43,in}$. A_4 is analyzed later as it requires coupling variable values from all the three other disciplines - u_{14} (from A_1), u_{24} (from A_2) and u_{34} (from A_3). After discussing the one-pass analysis paths for three-discipline and four-discipline coupled multidisciplinary system, we discuss below a three-discipline system with both uni-directional and bi-directional coupling.

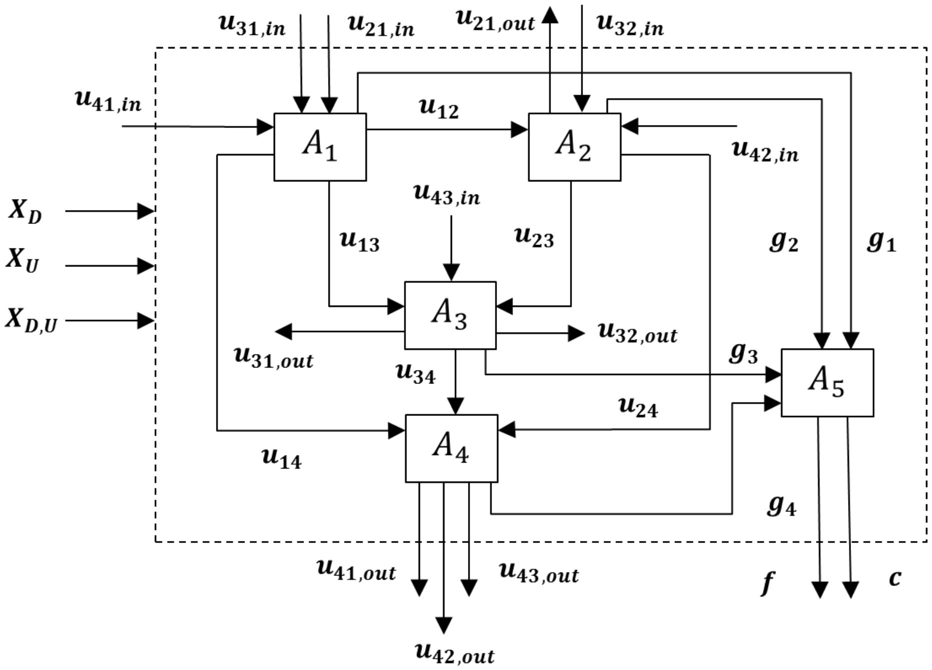


Fig. 6.9. One-pass analysis path for a four-discipline coupled system

C. Three-discipline coupled system with one-directional coupling between two disciplines

The discussion thus far considered one-pass analysis paths for three-discipline and four-discipline coupled systems with bi-directional coupling between individual disciplines. However, in some cases, it might be possible to have a one-directional coupling between some of the individual disciplines as shown in Fig. 6.10. In Fig. 6.10, one-directional coupling exists between A_1 and A_2 whereas bi-directional couplings exist between A_1 and A_3, and A_2 and A_3. Two possible one-pass analysis paths are presented in Fig. 6.11 and Fig. 6.12 respectively. The difference between the two paths is that one path results in an acyclic loop of individual disciplines whereas the other path does not form a loop. Both paths are feasible as they facilitate the one-pass analysis. The choice among the possible one-pass analysis paths depends on the analyst. It should be remembered that the ultimate goal of identifying a feasible one-pass analysis path is to generate training points for the construction of the probabilistic surrogate.

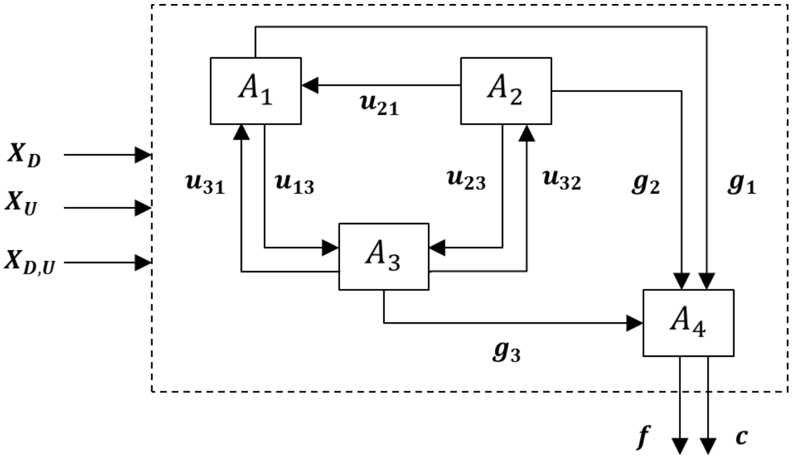


Fig. 6.10. Three-discipline coupled system with one-directional coupling between two disciplines

In Fig. 6.11, one-directional interactions exist between any two of A_1, A_2 and A_3; however, in Fig. 6.12, there exists no interaction between A_1 and A_2. In Fig. 6.11, the sequence of analyses is A_2, A_1, A_3 and A_4 in that order. However, in Fig. 6.12, the sequence of analyses is A_1, A_3, A_2 and A_4 respectively. The different sequence of analyses in Fig. 6.11 and Fig. 6.12 is due to the different in one-pass paths.

6.3.2. Construction of a probabilistic surrogate

Using the generated training points, we construct a probabilistic surrogate and use it for further multidisciplinary analysis. As discussed in Section 6.1, there exist two options for constructing a probabilistic surrogate - (1) A generic Bayesian network, where all the dependencies are accurately quantified, but the inference is approximate through computationally expensive MCMC methods, and (2) An approximate Bayesian network, where dependencies are not accurately quantified (due to simplifying joint probability assumptions), but which has fast, analytical inference. This chapter considers the second option of building an approximate surrogate with analytical inference. Fig. 6.13 shows the dependency relationships between several variables; the relationships between variables are obtained from the available physics-based models. Note that as discussed in Section 2.6, the dependence relationships between variables can be obtained from physics-based models, available data or their combination.

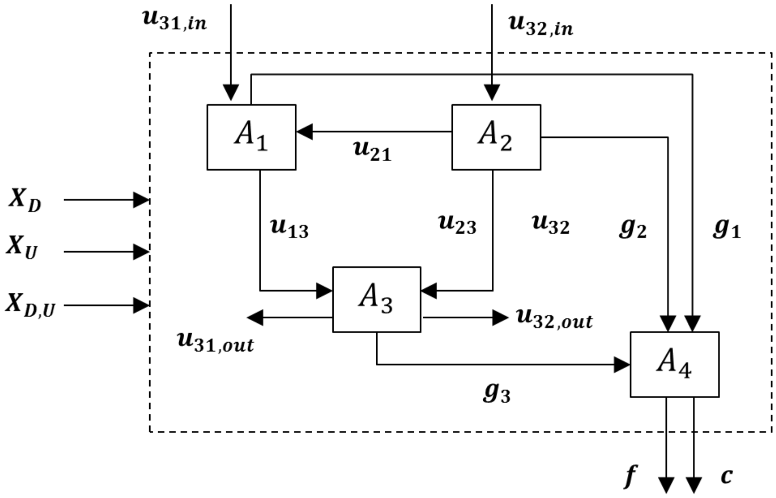


Fig. 6.11. One-pass analysis path of a three-discipline coupled system with one-directional coupling between two disciplines, resulting in one-directional coupling between all disciplines

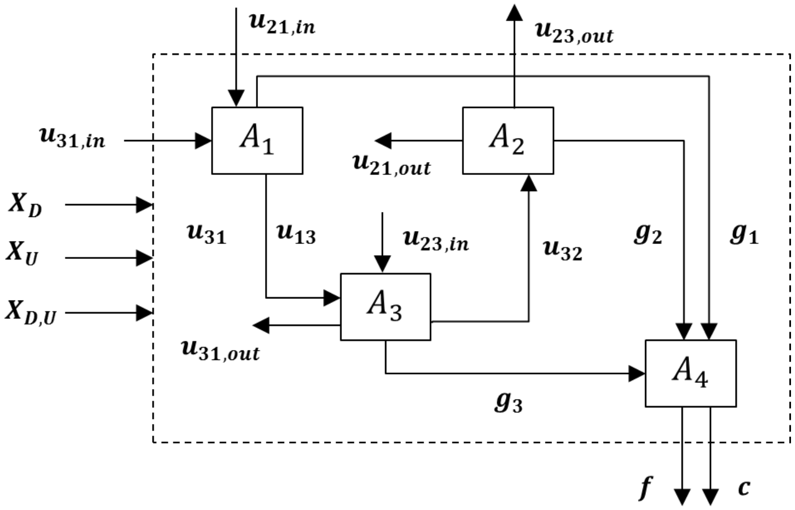


Fig. 6.12. One-pass analysis path of a three-discipline coupled system with one-directional coupling between two disciplines, resulting in no interaction between two disciplines

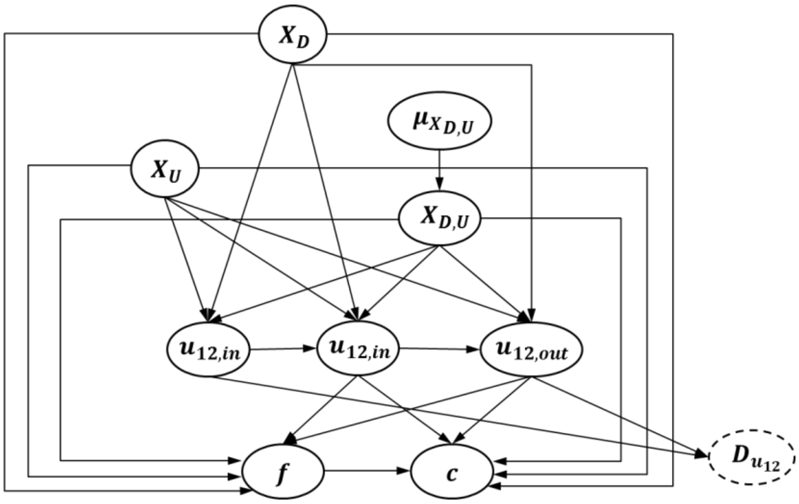


Fig. 6.13. Bayesian network for a conceptual multi-physics system

In Fig. 6.13 $\boldsymbol{D_{u_{12}}}$ represents the difference between the coupling variables $(\boldsymbol{u_{12}})$ in successive iterations ($\boldsymbol{u_{12,in}}$ and $\boldsymbol{u_{12,out}}$). Note that the values of $\boldsymbol{u_{12,out}}$ in one iteration becomes the values of $\boldsymbol{u_{12,in}}$ in the following iteration. $\boldsymbol{D_{u_{12}}}$ is not a stochastic variable, but a deterministic variable as it is the difference between the same set of coupling variables in successive iterations ($\boldsymbol{D_{u_{12}}} = \boldsymbol{u_{12,out}} - \boldsymbol{u_{12,in}}$). To achieve multidisciplinary compatibility, we require the difference variables to be zero ($\boldsymbol{D_{u_{12}}} = 0$). As $\boldsymbol{D_{u_{12}}}$ are deterministic nodes, the arcs between $\boldsymbol{u_{12,out}}$ and $\boldsymbol{D_{u_{12}}}$ are reversed (commonly referred to as arc reversal process [129]) such that $\boldsymbol{D_{u_{12}}}$ and $\boldsymbol{u_{12,out}}$ become stochastic and deterministic nodes respectively. To reduce the complexity of the Bayesian network, $\boldsymbol{u_{12,out}}$ variables are discarded as they can be estimated from $\boldsymbol{u_{12,in}}$ and $\boldsymbol{D_{u_{12}}}$. The transformed Bayesian network surrogate is shown in Fig. 6.14. Upon constructing the topology of the Bayesian network, we then proceed to quantify the conditional dependence relationships between them. The conditional relationships can be regarded as stochastic models connecting several random variables in a Bayesian network, as discussed in Chapter 4.

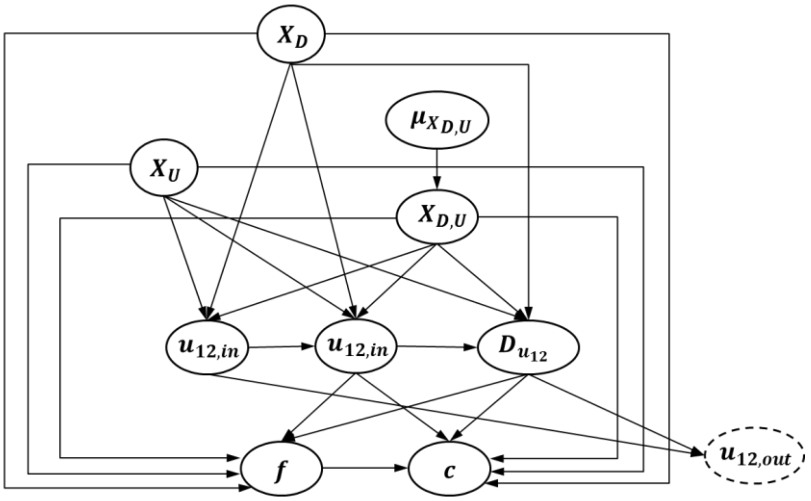


Fig. 6.14. Transformed Bayesian network of multidisciplinary system (after arc reversal)

In this work, we discuss three approximate probabilistic surrogates with analytical inference, namely (1) Multivariate Gaussian, (2) Gaussian Copula, and (3) Gaussian Mixture Model. All the three models are trained using the training data from Section 6.3.2. Similar to the Bayesian network, we consider $\boldsymbol{D}_{\boldsymbol{u}_{\mathbf{12}}}$ and ignore $\boldsymbol{u}_{\mathbf{12},\boldsymbol{out}}$ in training the three surrogates. A brief introduction to the three models and their application to multidisciplinary analysis are discussed below.

A. Multivariate Gaussian distribution

The multivariate Gaussian distribution to represent the joint distribution of p random variables $\boldsymbol{X} = \{X_1 \ldots X_p\}$ is given as

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = N_{\boldsymbol{X}}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{p}{2}}\,|\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})) \tag{6.1}$$

where $\boldsymbol{\mu} = E[\boldsymbol{X}]$ is a p-dimensional vector representing the expectation values of all the p random variables and $\boldsymbol{\Sigma} = E[(\boldsymbol{X}-\boldsymbol{\mu})(\boldsymbol{X}-\boldsymbol{\mu})^{\boldsymbol{T}}]$ is a $p \times p$ covariance matrix. Here. $E[.]$ refers to the expectation operator. As discussed in Section 6.1, an MG assumes that every random variable follows a univariate

Gaussian distribution. Similarly, the joint distribution or a conditional distribution of a subset of p random variables also follows a multivariate Gaussian. In Eq. (6.1), $|\mathbf{\Sigma}|$ represents the determinant of the covariance matrix.

B. Gaussian Copula

Consider p random variables $X_1 \dots X_p$, with continuous CDFs $F_1(x_1) \dots F_p(x_p)$, then a copula function can be defined as shown in Eq. (6.3).

$$C(u_1 \dots u_p) = P[F_1(x_1) \leq u_1, \dots, F_p(x_p) \leq u_p] \tag{6.2}$$

In Eq. (6.2), $P[.]$ represents the probability density function, $C(.)$ is the copula function and $C(u_1 \dots u_n)$ represents the joint CDF of $X_1 \dots X_n$. Several types of copula functions are available to model various dependence relationships between random variables such as Gaussian, Gumbel, Clayton and Independence copulas [130]. Among them, only Gaussian copula has analytical inference whereas other copulas require sampling-based inference. A Gaussian copula represents the joint CDF of all marginal CDFs using a multivariate Gaussian distribution. The dependence between the variables in Fig. 6.4(b), when a Gaussian copula is used can be represented as shown in Eq. (6.3) [130].

$$C_G(u_{x_1}, u_{x_2}, u_{y_1}, u_{y_2}) = \Phi_G(\Phi^{-1}(u_{x_1}), \Phi^{-1}(u_{x_2}), \Phi^{-1}(u_{y_1}), \Phi^{-1}(u_{y_2})) \tag{6.3}$$

In Eq. (6.3), $C_G(u_{x_1}, u_{x_2}, u_{y_1}, u_{y_2})$ represents the Gaussian copula used to model the joint CDF of x_1, x_2, y_1, y_2; $\Phi^{-1}(.)$ represents the inverse CDF of a standard Gaussian and $\Phi_G(.)$ represents the joint CDF of x_1, x_2, y_1, y_2. Using the joint CDF, the joint PDF can be obtained as shown in Eq. (6.4), where R represents the correlation matrix and $c_G(u_{x_1}, u_{x_2}, u_{y_1}, u_{y_2})$ represents the joint PDF.

$$c_G\left(u_{x_1}, u_{x_2}, u_{y_1}, u_{y_2}\right) = \frac{1}{\sqrt{\det(R)}} \exp\left(-\frac{1}{2} \begin{pmatrix} \Phi^{-1}\left(u_{x_1}\right) \\ \Phi^{-1}\left(u_{x_2}\right) \\ \Phi^{-1}\left(u_{y_1}\right) \\ \Phi^{-1}\left(u_{y_2}\right) \end{pmatrix}^{\mathrm{T}} \cdot \left(R^{-1} - I\right) \cdot \begin{pmatrix} \Phi^{-1}\left(u_{x_1}\right) \\ \Phi^{-1}\left(u_{x_2}\right) \\ \Phi^{-1}\left(u_{y_1}\right) \\ \Phi^{-1}\left(u_{y_2}\right) \end{pmatrix} \right) \tag{6.4}$$

The multivariate Gaussian distribution in Eq. (6.4) can be used to generate a large number of samples of correlated normal random variables. In this case, samples of 4 variables from this joint normal distribution are generated and denoted as X_1', X_2', Y_1', Y_2'. For each sample of the variables, compute the CDF with respect to the marginal distributions of standard normal distribution, and denote the CDF values as $u_{X_1}, u_{X_2}, u_{Y_1}$ and u_{Y_2}. Samples of X_1, X_2, Y_1 and Y_2 are then obtained by taking the inverse CDFs of $u_{X_1}, u_{X_2}, u_{Y_1}$ and u_{Y_2} with respect to their marginal distributions.

C. Gaussian mixture model

Let $\boldsymbol{X} = \{X_1, X_2 \dots X_p\}$ represent p random variables whose joint probability, $f_{\boldsymbol{X}}(\boldsymbol{x})$ is represented using an N-component GMM given as [86]

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = \sum_{i=1}^{N} w_i \times N_{\boldsymbol{X},i}(\boldsymbol{x}|\boldsymbol{\mu_i}, \boldsymbol{\Sigma_i}) \tag{6.5}$$

where $N_{\boldsymbol{X},i}(\boldsymbol{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ and w_i represent the i^{th} Gaussian Mixture component and its corresponding weight respectively. It should be noted that the sum of the weights of all Gaussian components is equal to 1. The first step in approximating a joint probability distribution through a GMM is the selection of the number of components N. The optimal number of components is typically obtained by maximizing a model selection score measure such as Akaike Information Criterion (AIC) [131], Bayesian Information Criterion (BIC) [132] and Mutual Information (MI) [52]. For illustration, this work used the BIC score which is the defined as

$$BIC = \ln(L) - 0.5 \times k \times \ln(M) \tag{6.6}$$

where L refers to the likelihood of observing the data given a GMM, k and M refer to the number of parameters to be estimated in the GMM and the amount of available data respectively. The parameters in a GMM to be estimated include the weights (w_i), the expectation and variance matrices of all Gaussian Mixture components. The estimation of parameters in a GMM is typically through the Expectation-Maximization (EM) algorithm [133]. The EM algorithm estimates the parameters in two steps: the E step and the M step. The E step estimates the weights of the Gaussian components conditioned on the Gaussian component parameters (mean and variance). The M step estimates the Gaussian component parameters conditioned on the weights. The E and M steps are repeated until convergence is achieved in weights and Gaussian component parameters.

D. Model characteristics

Here, we discuss the underlying assumptions in each of the three approximate models mentioned above. The Multivariate Gaussian assumes that all the individual random variables follow a Gaussian distribution, both individually and jointly, and can be used to model a uni-modal joint distribution. A less restrictive model compared to Multivariate Gaussian is the Gaussian Copula, which does not assume that the individual random variables follow Gaussian distributions. The Gaussian copula can model non-Gaussian marginal distributions and only a uni-modal joint distribution. A Gaussian mixture is the least restrictive of the three models, as it does not assume that the individual variables follow Gaussian distributions and can be used to model multi-modal joint distributions. By changing the number of Gaussian components in the GMM, any order of modality can be represented; however, it should be noted that a multivariate Gaussian distribution is used in modeling each of the individual GMM components.

6.3.3. Forward propagation and inference

We first discuss below the generalized updating procedure for a GC and GMM, and then then discuss their applicability to multidisciplinary analysis. Since an MG can be considered a special case of a GMM, its updating is not particularly discussed, as the updating procedure for a GMM can be adapted for updating in MG. One-pass analysis results in one-way interactions between several disciplines. Here, forward propagation refers to the estimation of downstream variables (such as system QoI) for a given realization of the input variables. Inference refers to the estimation of upstream variables when a downstream variable is fixed a given value. In the case of a multi-physics system, the difference variable is the downstream variable and is fixed at zero (multidisciplinary compatibility). Here, we perform forward propagation and inference together (termed as updating) by conditionalizing the input variable at some chosen values and the difference variables $\boldsymbol{D_u}$ at zero, and obtain the distributions of the remaining variables. Let $\boldsymbol{x_D^*}$ and $\boldsymbol{\mu_{D,U}^*}$ refer to the design values of $\boldsymbol{X_D}$ and $\boldsymbol{\mu_{X_{D,U}}}$ at which the unobserved variables need to be evaluated.

A. Updating in a Gaussian copula

First, we illustrate the procedure of obtaining the conditional PDF $f(Y_1, Y_2 |\, X_1 = x_1, X_2 = x_2)$ for the example in Fig. 6.4(b) and then extend to the Bayesian network in Fig. 6.14. Conditionally sampling with the Gaussian copula assumption is easy to implement since the joint PDF in Eq. (6.5) can be converted to a conditional PDF analytically. For example, the conditional samples of Y_1 and Y_2 need to be generated given $X_1 = x_1, X_2 = x_2$. The equivalent normals corresponding to $X_1 = x_1, X_2 = x_2$ are first calculated as $x_1' = \Phi^{-1}\left(F_{X_1}(x_1)\right)$, $x_2' = \Phi^{-1}\left(F_{X_2}(x_2)\right)$. Let μ be the mean of X_1, X_2, Y_1 and Y_2 in the equivalent normal space; therefore, μ is a zero vector with 4 entries, and $\boldsymbol{R}$ is the covariance matrix given as

$$\boldsymbol{R} = \begin{array}{c} \begin{array}{cc} \overbrace{\qquad\qquad}^{\boldsymbol{\Sigma_3}} & \overbrace{\qquad\qquad}^{\boldsymbol{\Sigma_1}} \end{array} \\ \left[\begin{array}{cc:cc} 1 & \rho_{x_1x_2} & \rho_{x_1y_1} & \rho_{x_1y_2} \\ & 1 & \rho_{x_2y_1} & \rho_{x_2y_2} \\ \hdashline & & 1 & \rho_{y_1y_2} \\ & & & 1 \end{array}\right] \end{array} \begin{array}{l} \\ \\ \left.\begin{array}{c} \\ \\ \end{array}\right\} \boldsymbol{\Sigma_2} \end{array} \tag{6.7}$$

Then the conditional joint distribution of Y_1' and Y_2' given $X_1' = x_1', X_2' = x_2'$ is denoted as: $f(Y_1', Y_2' | X_1' = x_1', X_2' = x_2') \sim N(\widetilde{\boldsymbol{\mu}}, \widetilde{\boldsymbol{\Sigma}})$ where the conditioned mean vector $\widetilde{\boldsymbol{\mu}}$ and covariance matrix $\widetilde{\boldsymbol{\Sigma}}$ are given as

$$\tilde{\mu} = \Sigma_1 \Sigma_3^{-1} * \begin{bmatrix} x_1' \\ x_2' \end{bmatrix} \tag{6.8}$$

$$\tilde{\Sigma} = \Sigma_2 - \Sigma_1 \Sigma_3^{-1} \Sigma_1^{\mathrm{T}} \tag{6.9}$$

Samples Y_1' and Y_2' are jointly generated from a multivariate normal distribution, of which the mean and covariance matrix are calculated as in Eqs. (6.8, 6.9). The CDF values of each Y_1' and Y_2' sample with respect to the standard normal distribution (u_{Y_1}, u_{Y_2}) are computed, and the inverse CDF is taken to obtained the conditional samples of Y_1 and Y_2.

$$c_G\left(u_{X_D}, u_{\mu_{X_{D,U}}}, \; u_{X_{D,U}}, u_{u_{12}}, u_{u_{21}}, u_{D_u}, u_f, u_c, u_{X_U}\right) = \frac{1}{\sqrt{\det(R_M)}} \exp\left(-\frac{1}{2} \begin{pmatrix} \Phi^{-1}(u_{X_D}) \\ \Phi^{-1}\left(u_{\mu_{X_{D,U}}}\right) \\ \Phi^{-1}(u_{X_{D,U}}) \\ \Phi^{-1}(u_{u_{12}}) \\ \Phi^{-1}(u_{u_{21}}) \\ \Phi^{-1}(u_{D_u}) \\ \Phi^{-1}(u_f) \\ \Phi^{-1}(u_c) \\ \Phi^{-1}(u_{X_U}) \end{pmatrix}^T \times (R_M^{-1} - I) \times \begin{pmatrix} \Phi^{-1}(u_{X_D}) \\ \Phi^{-1}\left(u_{\mu_{X_{D,U}}}\right) \\ \Phi^{-1}(u_{X_{D,U}}) \\ \Phi^{-1}(u_{u_{12}}) \\ \Phi^{-1}(u_{u_{21}}) \\ \Phi^{-1}(u_{D_u}) \\ \Phi^{-1}(u_f) \\ \Phi^{-1}(u_c) \\ \Phi^{-1}(u_{X_U}) \end{pmatrix} \right) \tag{6.10}$$

The Gaussian Copula model between several variables of one-pass analysis, as seen from Fig. 6.14 is given in Eq. (6.11), in which R_M refers to the covariance matrix. Using the above GC and observed data, i.e., $\boldsymbol{x_D} = \boldsymbol{x_D^*}$, $\boldsymbol{\mu_{D,U}} = \boldsymbol{\mu_{D,U}^*}$ and $\boldsymbol{D_u} = \boldsymbol{0}$ (multidisciplinary compatibility), the updating can be performed analytically using expressions in Eqs. (6.9, 6.10).

B. Updating in a Gaussian mixture Model

In a GMM, let data ($\boldsymbol{x}_{obs}^D$) be available on a subset of variables, $\boldsymbol{X}_{obs} \subset \boldsymbol{X}$, and $\boldsymbol{X}_{unobs}$ represent the set of variables to be inferred. The posterior distributions of the unobserved variables can be obtained as

$$f(\boldsymbol{x}_{unobs}|\boldsymbol{X}_{obs} = \boldsymbol{x}_{obs}^D) = \frac{f\left(\boldsymbol{x}_{unobs}, \boldsymbol{x}_{obs}^{\boldsymbol{D}}\right)}{f(\boldsymbol{x}_{obs} = \boldsymbol{x}_{obs}^D)}$$

$$= \frac{\sum_{i=1}^N w_i \times N_{\boldsymbol{X},i}(\boldsymbol{x}_{unobs}, \boldsymbol{x}_{obs} = \boldsymbol{x}_{obs}^D|\mu_i, \Sigma_i)}{\int \sum_{i=1}^N w_i \times N_{\boldsymbol{X},i}(\boldsymbol{x}_{unobs}, \boldsymbol{x}_{obs} = \boldsymbol{x}_{obs}^D|\mu_i, \Sigma_i)\ d\boldsymbol{x}_{unobs}}$$

$$= \frac{\sum_{i=1}^N w_i \times N_{\boldsymbol{X},i}(\boldsymbol{x}_{unobs}|\ \boldsymbol{x}_{obs} = \boldsymbol{x}_{obs}^D) \times N_{X,i}(\boldsymbol{x}_{obs} = \boldsymbol{x}_{obs}^D)}{\int \sum_{i=1}^N w_i \times N_{\boldsymbol{X},i}(\boldsymbol{x}_{unobs}, \boldsymbol{x}_{obs} = \boldsymbol{x}_{obs}^D|\mu_i, \Sigma_i)\ d\boldsymbol{x}_{unobs}} \quad (6.11)$$

$$= \sum_{i=1}^N \gamma_i \times N_{\boldsymbol{X},i}(\boldsymbol{x}_{unobs}|\ \boldsymbol{x}_{obs} = \boldsymbol{x}_{obs}^D)$$

In Eq. (6.11), w_i, $N_{X,i}(\boldsymbol{x}_{obs} = \boldsymbol{x}_{obs}^D)$ and the denominator $\int \sum_{i=1}^N w_i \times N_{\boldsymbol{X},i}(\boldsymbol{x}_{unobs}, \boldsymbol{x}_{obs} = \boldsymbol{x}_{obs}^D|\mu_i, \Sigma_i)\ d\boldsymbol{x}_{unobs}$ are all constants, and therefore Eq. (6.12) can be simplified to Eq. (6.13) as

$$f(\boldsymbol{x}_{unobs}|\boldsymbol{x}_{obs} = \boldsymbol{x}_{obs}^D) = \sum_{i=1}^N \gamma_i \times N_{\boldsymbol{X},i}(\boldsymbol{x}_{unobs}|\ \boldsymbol{x}_{obs} = \boldsymbol{x}_{obs}^D) \quad (6.12)$$

where γ_i is a new constant combining all the three constants described above. In Eq. (6.12), $N_{\boldsymbol{X},i}(\boldsymbol{x}_{unobs}|\ \boldsymbol{x}_{obs} = \boldsymbol{x}_{obs}^D)$ represents the posterior distribution of each Gaussian mixture component. As the posterior distribution of each component also represents a Gaussian distribution, the overall posterior distribution, $f(\boldsymbol{x}_{unobs}|\boldsymbol{x}_{obs} = \boldsymbol{x}_{obs}^D)$, represents another GMM. The posterior distributions associated with each of the Gaussian components, can be obtained using Eqs. (6.9, 6.10). Thus, inference can be carried out in a GMM in an analytical manner. We now implement the inference procedure for the Bayesian network shown in Fig. 6.14.

Let $\boldsymbol{X} = \{\boldsymbol{X_D},\ \boldsymbol{X_U},\ \boldsymbol{\mu_{X_{D,U}}}, \boldsymbol{X_{D,U}}, \boldsymbol{u_{12,in}}, \boldsymbol{u_{21}}, \boldsymbol{D_{u_{12}}}, \boldsymbol{f}, \boldsymbol{c}\}$, then their joint probability distribution $f_{\boldsymbol{X}}(\boldsymbol{x})$ when represented as a GMM is given in Eq. (6.6) and the inference can be carried out using Eq. (6.13). In this case, $\boldsymbol{x_{obs}} = \{\boldsymbol{X_D}, \boldsymbol{\mu_{D,U}}, \boldsymbol{D_u}\}$ and $\boldsymbol{x^D_{obs}} = \{\boldsymbol{x^*_D}, \boldsymbol{\mu^*_{D,U}}, \boldsymbol{0}\}$ and $\boldsymbol{x_{unobs}} = \{\boldsymbol{X_U}, \boldsymbol{X_{D,U}}, \boldsymbol{u_{12,in}}, \boldsymbol{u_{21}}, \boldsymbol{f}, \boldsymbol{c}\}$. Using the joint posterior distributions, the marginal distribution of the system QoIs can be obtained by integrating over all the other variables. Following the probabilistic surrogate approach, the compatibility and the uncertainty analyses can be performed simultaneously. Thus, the traditional double-loop approach where uncertainty analysis is carried out in the outer loop and compatibility analysis in the inner loop is collapsed into a single-loop approach. As discussed in Section 6.1, the results from performance assessment can be extended for design optimization of multi-physics systems under uncertainty. In the next section, we briefly discuss multidisciplinary optimization under uncertainty and ways to extend the developed performance assessment approach for design optimization.

6.4. Design optimization of multi-physics systems

The generic optimization formulation (RBDO, RDO and RBRDO) for the design of multi-physics systems under uncertainty can be given as

$$\begin{gathered} Min\,\{H[f_i(\boldsymbol{X_D}, \boldsymbol{X_U}, \boldsymbol{X_{D,U}})]\} \quad i = 1,2,\dots k \\ s.t \quad \begin{gathered} Pr\big(c_j(\boldsymbol{X_D}, \boldsymbol{X_U}, \boldsymbol{X_{D,U}}, \boldsymbol{u_{12}}, \boldsymbol{u_{21}}) < 0\big) > \boldsymbol{\gamma_j} \quad j = 1,2,\dots m \\ h_l(\boldsymbol{X_D}, \boldsymbol{X_U}, \boldsymbol{X_{D,U}}, \boldsymbol{u_{12}}, \boldsymbol{u_{21}}) = 0 \quad l = 1,2,\dots n \\ \boldsymbol{lb_d} \le \boldsymbol{X_D} \le \boldsymbol{ub_d} \\ \Pr\big(\boldsymbol{X_{D,U}} \ge \boldsymbol{lb_{X_{D,U}}}\big) \ge \boldsymbol{p_{lb}} \end{gathered} \end{gathered} \tag{6.13}$$

$$\Pr\left(\boldsymbol{X_{D,U}} \leq \boldsymbol{ub_{X_{D,U}}}\right) \geq \boldsymbol{p_{ub}}$$

In the above formulation, $f_i(\ i = 1\ to\ k), c_j(j = 1\ to\ m)$ and $h_l(l = 1\ to\ n)$ represent the sets of objective functions, constraint functions and multidisciplinary compatibility conditions respectively. $c_j < 0$ represents the safe region and γ_j represents the reliability threshold for the j^{th} constraint. Note that for design variables associated with uncertainty ($\boldsymbol{X_{D,U}}$), we optimize their mean values ($\boldsymbol{\mu_{X_{D,U}}}$). For deterministic design variables ($\boldsymbol{X_D}$), we have strict lower and upper bounds ($\boldsymbol{lb_d}$ and $\boldsymbol{ub_d}$). Similarly, design variables with associated uncertainty ($\boldsymbol{X_{D,U}}$) have probabilistic bounds given as the last two constraints. $\boldsymbol{p_{lb}}$ and $\boldsymbol{p_{ub}}$ represent the probability thresholds for $\boldsymbol{X_{D,U}}$ that they are in between their lower and upper bounds ($\boldsymbol{lb_{X_{D,U}}}$ and $\boldsymbol{ub_{X_{D,U}}}$). $H[.]$ refers to an operator, which can be either an expectation operator or an operator to compute standard deviation, depending on the type of optimization. For RBDO, $H[.]$ is an expectation operator, a standard deviation operator for RDO and both for RBRDO, i.e., for some objective functions, $H[.]$ acts an expectation operator and for others, as standard deviation operator. As RDO does not consider any reliability constraints, γ_j in the above formulation can be assumed to be equal to 1. As mentioned in Section 6.1, the multi-objective optimization is converted to a single optimization problem through a weighted-sum of the individual objective functions. Therefore, the new single objective function is defined as

$$F = \sum \beta_i \frac{H[f_i]}{f_{i,th}}, \ \sum \beta_i = 1 \tag{6.14}$$

In Eq. (6.14), $f_i, i = 1,2 \dots k$ refer to k competing objective functions. $f_{i,th}$ correspond to the normalization values of these objective functions and β_i refer to the weights associated with each of the objective functions. By changing the weight coefficients (β_i) several single optimization functions can be generated, and for each of these single objective functions, multidisciplinary optimization is carried out and an optimum design point is obtained. The results from these optimization evaluations can be visualized through the creation of a Pareto surface, which can be used for further decision-making such as obtaining the best design point among all the available design points considering trade-offs between individual objective functions. The number of weight coefficient combinations depends on the desired accuracy of the Pareto surface. Typically, the design of experiments techniques such as Latin Hypercube sampling can be used to determine several combinations of weight coefficients.

Using different types of probabilistic surrogates, the joint posterior distributions of several variables at a given realization of input variables after imposing the compatibility condition can be obtained analytically, as described in Section 6.3.3. From the joint posterior distribution, the marginal posterior distributions of the objectives and constraints can be obtained for further analysis such as reliability calculations for the constraints. Thus, reliability-based multidisciplinary and multi-objective optimization is carried out using a Bayesian network surrogate and weighted-sum approach for the Pareto surface. The design optimization process, which traditionally is a nested three-loop process as shown in Fig. 6.1 is converted to a double-loop analysis where optimization analysis is carried out in the outer loop and uncertainty and convergence analysis in the loop simultaneously using the procedure laid down in Section 6.3. The proposed methodology is demonstrated below for an airfoil design problem with two design objectives and a reliability constraint.

6.5. Illustration example: Fluid-Structure interaction on an aircraft wing

We demonstrate the proposed methods for performance assessment and design optimization of multi-physics systems using an aircraft wing with a NACA 0012 airfoil.

6.5.1 Performance assessment of an aircraft wing

An aircraft wing can be regarding as a multi-physics system due to the fluid-structure interaction occurring on the surface of the aircraft wing. The fluid-structure interaction can be analyzed by creating two separate individual disciplinary models (structural analysis and fluid dynamics) and solving them iteratively until their results are compatible. The individual disciplines along with the coupling variables between them are illustrated in Fig. 6.15. The fluid and structural meshes for the analyses are shown in Fig. 6.16.

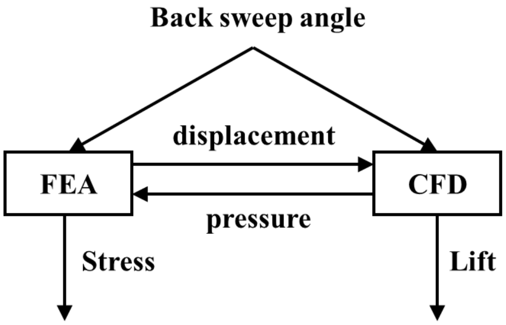
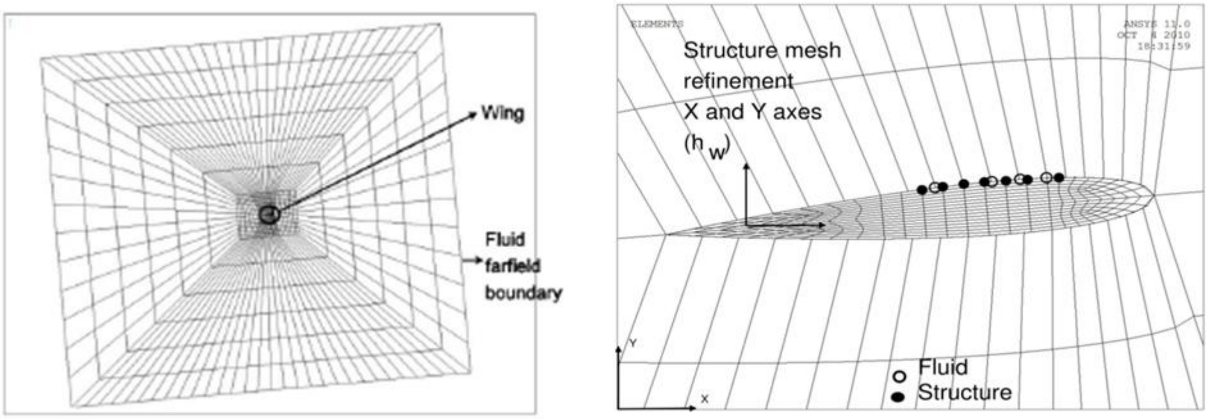


Fig. 6.15. Coupling in aircraft wing analysis



(a)

(b)

Fig. 6.16. Aircraft wing modeling in ANSYS (a) Overall view, and (b) Structural and Fluid mesh nodes

In this example, the system QoIs are considered as the maximum stress experienced at the wing surface and the generated lift. The input variable is the backsweep angle. The backsweep angle is assumed to be associated with aleatory uncertainty represented through a Gaussian distribution with parameters 0 and 0.04 respectively. Since, the variability in the backsweep angle is known, we consider the mean of the backsweep angle as the design variable. As mentioned in Section 6.1, the traditional approach to compute the distributions of maximum stress and lift is through a double-loop analysis, where the compatibility analysis between CFD and FEA is solved in the inner loop while the uncertainty analysis, i.e, multiple realizations of the aleatory uncertainty associated with the backsweep angle are sampled in the outer loop; this could be computationally expensive. Therefore, we build the three probabilistic surrogates discussed in Section 6.3.3 (MG, GC and GMM) and obtain the distributions of QoIs by performing compatibility and uncertainty analyses simultaneously.

The training points for the construction of a probabilistic surrogate are obtained through ANSYS fluid-structure interaction simulations. The probability distribution of the mean of the backsweep angle (μ_{bw}) is assumed to be a uniform distribution between 0 and 0.5. One-pass analysis is performed to generate training points later used to construct several probabilistic surrogates. We discarded the first two iterations and considered the third iteration data on nodal pressures since the nodal pressures in the first two iterations varied drastically. 200 training points with different realizations of backsweep angle and nodal pressures are used to generate training points on lift and maximum stress. This problem has a large number of coupling variables; the displacement and pressure variables at each node on the boundary on the wing where the fluid-structure interaction occurs are the coupling variables. In this analysis, we have 258 coupling variables. Due to the large number of coupling variables, the Principal Component Analysis (PCA) is performed to reduce the number of coupling variables from 258 to 6. The first 6 PCs are chosen as they explain 95% variance in the data. Please refer to Liang and Mahadevan [134] regarding the use of PCA for reducing model complexity in multidisciplinary analysis.

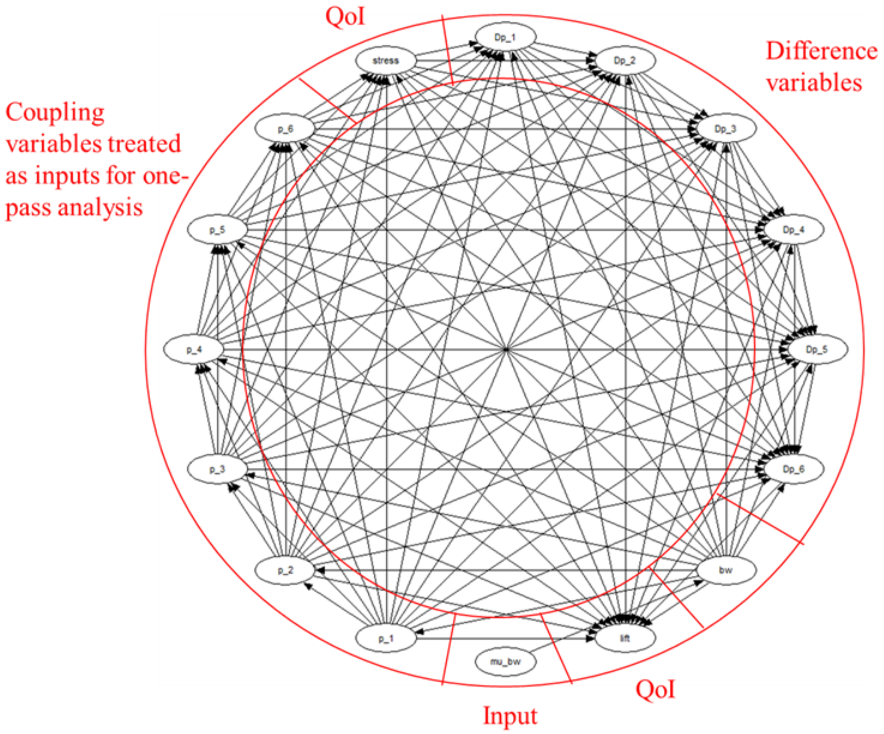


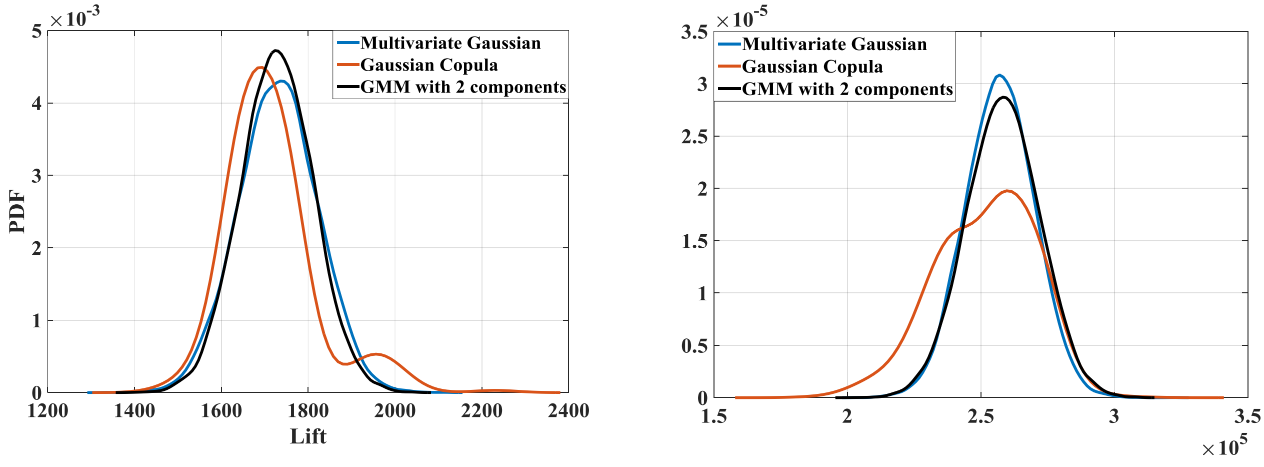
Fig. 6.17. Graphical representation of the dependence between variables

In total, we have 16 variables in the Bayesian network surrogate – mean of backsweep angle, backsweep angle (after considering aleatory uncertainty), 6 'in' nodal pressures, 6 difference values of nodal pressures, 1 maximum stress and 1 lift variable; the graphical representation of dependence between these variables is shown in Fig. 6.17. The graphical model is drawn using the Uninet software package [135]. The intermediate variable in Fig. 6.17 refers to variables that are not coupling variables, design variables, objectives or constraints. In the problem, the backsweep angle after considering aleatory uncertainty is treated as the intermediate variable. In this problem, the intermediate coupling variables (i.e., coupling variables that are not considered as inputs in the one-pass analysis) refer to the nodal displacements. The nodal displacements affect the maximum stress on the wing. The three probabilistic surrogates discussed earlier in Section 6.3.2 are then fit for 16 variables and with 200 samples. In the case of a GMM, a two-component model is used as it has the lowest BIC score as seen from Table 6.1.

Table 6.1. Variation of BIC score with Gaussian Mixture Components

Number of GMM components	BIC score
1	36946.3
2	36675.6
3	37129.3
4	37294.9

After constructing the three surrogates, we use them for performance assessment by conditionalizing them at a given value of the mean of backsweep angle and imposing the multidisciplinary compatibility. For illustration, the distributions of system QoI (lift and maximum stress) when μ_{bw} is fixed at 0.4 are given in Fig. 6.18.



(a)

(b)

Fig. 6.18. Prediction of QoIs (lift and maximum stress) using the three probabilistic surrogates (MG, GC and GMM)

6.5.2 Design optimization of an aircraft wing airfoil

After performance assessment, we consider the three probabilistic surrogates for design optimization. We consider two competing objective functions to perform Reliability-based Robust Design Optimization (RBRDO): (1) Maximize the expected value of lift, and (2) Minimize the standard deviation of lift. The design is performed under a reliability constraint associated with the maximum stress and the backsweep angle is the design variable. Overall, the mathematical formulation of the design can be written as

$$\begin{gathered} \text{Max}\, E[L(\mu_{bw})] \;\; \& \;\; \text{Min}\, Std[L(\mu_{bw})] \\ s.t \; \Pr(s > 3 \times 10^5) \leq 0.001 \\ 0 \leq \mu_{bw} \leq 0.5 \end{gathered} \tag{6.15}$$

In Eq. (6.15), $E[.]$ and $Std[.]$ represent the expectation and standard deviation operators respectively. L, μ_{bw} and s represent the lift, mean of backsweep angle and maximum stress respectively. To facilitate the Pareto surface construction, the multi-objective optimization is converted to a single-objective optimization using the weighted-sum approach. The new objective function is defined as

$$\text{Min} \;\; \beta * \frac{Std[L]}{S_{bl}} - (1-\beta) * \frac{E[L]}{E_{bl}} \tag{6.16}$$

In Eq. (6.16), β represents the weight factor for combining the two objective functions. E_{bl} and S_{bl} represent the normalization factors since the two objectives are in different order of magnitude; these values are assumed to be 1666.7 and 113 respectively. 51 values of β equally spaced between 0 and 1, and including them, are considered for the construction of the Pareto surface. Using the surrogates, multidisciplinary optimization is performed by conditionalizing the 6 ‘difference’ variables at zero and the design variable (mean of backsweep angle) at a different value in each iteration of the optimization analysis. For different values of β, optimization is carried out and Pareto surfaces using all the three surrogates are constructed. The global optimizer DIRECT [136] is used to carry out each single objective optimization at a given value of β. For comparison of Pareto surfaces, the optimum design points obtained from optimization analysis using the three types of surrogates are evaluated using the GMM and plotted against each other in Fig. 6.19. In Fig. 6.19, the points in upper right corner correspond to $\beta = 0$ (maximization of the expected value of lift) and correspondingly, the lower left corner correspond to $\beta = 1$, i.e., minimization of the standard deviation of lift. It can be seen that all the three surrogates provide similar solutions for the optimization of expected value of lift, whereas the GMM provides lower values of standard deviation, when compared to the other two surrogates. From this plot, it can be concluded that a GMM is better suited to carry out design optimization with variance requirements such as Robust Design Optimization (RDO) and Reliability-based Robust Design Optimization (RBRDO).

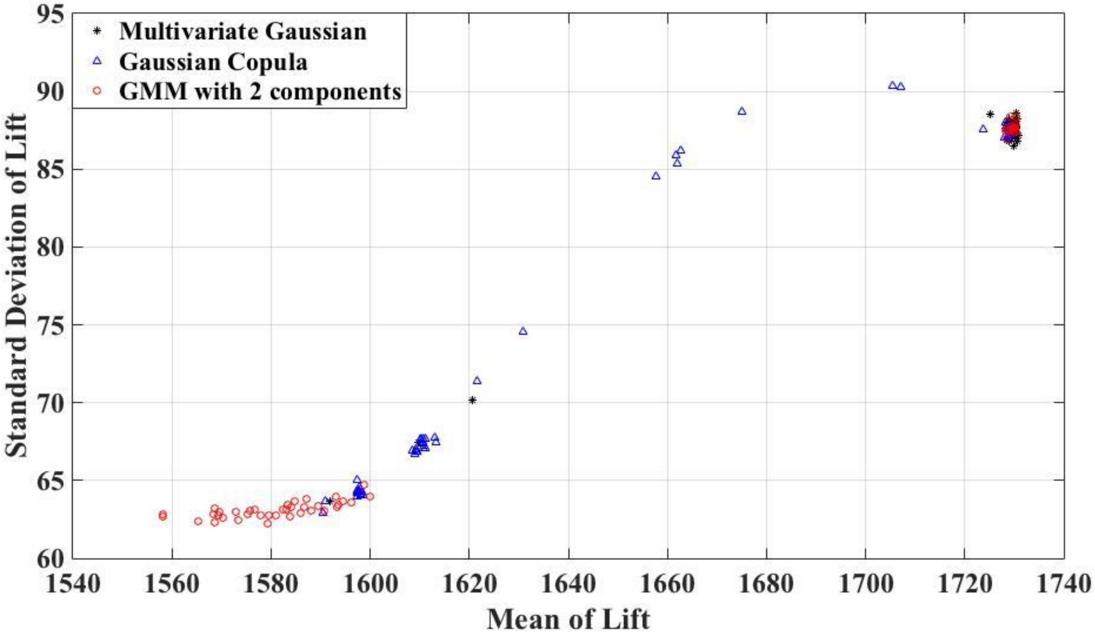


Fig. 6.19. Comparison of Pareto Surfaces obtained using Multivariate Gaussian, Gaussian Copula and a Gaussian Mixture Model

6.6 Summary

This chapter developed a probabilistic surrogate-based approach to perform efficient multidisciplinary analysis and optimization under uncertainty. Multidisciplinary analysis is computationally expensive, as it requires iterative analysis of individual disciplinary models until convergence to inter-disciplinary compatibility. Optimization under uncertainty involves a nested three-loop process where the multidisciplinary analysis is performed in the inner loop, uncertainty analysis in the middle loop and optimization in the outer loop. For computational efficiency, the physics-based computational models are typically replaced by computationally inexpensive surrogate models. Deterministic surrogates (in the variable space), which provide a point-valued output for a given set of inputs are traditionally considered; these surrogates are run at several realizations of uncertain variables and at a given set of design variables, to perform optimization analysis under uncertainty. In contrast, we considered a probabilistic surrogate (in the probability space), one run of which provides a distribution prediction considering the uncertain variables. Two options exist for building a probabilistic surrogate: (1) build an accurate Bayesian network, but which has approximate and computational expensive sampling-based inference through Markov Chain Monte Carlo methods (MCMC), or (2) build an approximate surrogate but with analytical inference (fast and exact) such as a Multivariate Gaussian, Gaussian Copula and Gaussian Mixture Model; the latter option is investigated in this chapter.

Multivariate Gaussian is the most restrictive model as it assumes a uni-modal joint normal distribution along with Gaussian marginal distributions. Gaussian Copula also assumes a uni-modal joint distribution but can handle any distribution type for the individual variables. The Gaussian Mixture Model does not make any assumptions regarding the distribution types of individual variables or the modality of the joint distribution. The training points for building the surrogate are obtained from a one-pass analysis through the individual disciplines in a multidisciplinary system. Performing a successful one-pass analysis requires careful organization of the analysis sequence, as shown for three- and four-discipline systems. The probabilistic surrogate is then trained, and used to infer the distributions of coupling variables, the objective and the constraint functions by imposing the multidisciplinary compatibility condition.

The proposed methodologies are demonstrated for performance assessment and reliability-based and robust design optimization of a simplified two-disciplinary aircraft wing airfoil design. From the example, it was observed the design solutions obtained the Gaussian Mixture Model are significantly better when compared against the other two probabilistic surrogates (Multivariate Gaussian and Gaussian Copula).

It is important to note that by using the probabilistic surrogate, the three-loop optimization analysis is now collapsed into a two-loop analysis as the uncertainty analysis and multidisciplinary analysis are performed simultaneously in the inner loop, which is also very fast due to the availability of the analytical solution. This feature helps to scale up multidisciplinary optimization under uncertainty to large systems with multiple coupled disciplines and a large number of coupling variables between individual disciplines. In this chapter, we considered multi-physics systems; however, the developed techniques are applicable to systems with coupled interactions between multiple subsystems following the same physics such as a composite beam, as discussed in Section 1.1.

Chapters 3 to 6 discussed solution methodologies for various issues discussed in Section 1.1 regarding performance assessment in single-component systems, multi-level systems, coupled and time-varying and multi-physics systems. The next chapter presents concluding remarks and discusses some possible extensions of this dissertation as future work.

CHAPTER 7

CONCLUSION

Model-based methods are increasingly becoming popular in designing and analyzing complex engineering systems for which real-world testing can be prohibitively expensive. Engineering systems, when operated under real-world scenarios are impacted by several sources of uncertainty. Identification and incorporation of all such uncertainty sources is necessary for a comprehensive performance assessment framework. The performance assessments can later be used for both design-time and run-time decision-making processes. This dissertation developed efficient Bayesian probabilistic methods for model-based performance assessment of single-component systems, systems with one-directional interactions between subsystems, coupled systems with time-varying interactions, and coupled systems with simultaneously interactions (such as multi-physics systems) under different sources of aleatory and epistemic (statistical and model) uncertainty sources. A summary of the intellectual contributions made through this dissertation are detailed below.

7.1. Summary of contributions

This dissertation made the following contributions to advance the state of the art in performance assessment of engineering systems.

1. A unified framework connecting the model calibration analysis to the reliability analysis, and quantifying the uncertainty in the reliability estimate after the inclusion of different types of aleatory and epistemic uncertainty (statistical, model and sampling errors).
2. Methods for learning Bayesian networks considering qualitative and quantitative data, and extension of the approach to the construction of hierarchical Bayesian networks.

3. Efficient methods for calibration and dimension reduction in hierarchical Bayesian networks through segmented calibration approaches and multi-level variance-based sensitivity analysis.
4. A dynamic Bayesian network approach for the performance assessment of systems where the interactions between subsystems occur with a time lag. In particular, we considered cyber-physical control systems, where the coupled interactions occur at two levels – between the individual subsystems at the higher-level and between several computing nodes in the lower-level; the performance assessment is therefore performed using a two-level DBN.
5. Efficient performance assessment approaches of multi-physics systems analysis through a probabilistic surrogate, which helps collapse the computational intensive double loop uncertainty and multidisciplinary compatibility analysis into a single loop analysis, where uncertainty and compatibility analysis are performed simultaneously.

7.2 Future work

Based on the research reported in this dissertation, the following issues may be investigated as part of future research.

1. This work primarily considered quantitative data in performance assessment analysis. Other forms of information such as system model data, linguistic and image data, streaming data need to be considered. In addition, unstructured data also needs to be considered.
2. The first step for model-based analysis is the development of models. As discussed in Section 1.1, models can be constructed from available physics, data or their combination. This work considered constructing models from "small" data sets (datasets that can be stored on a stand-alone workstation). Future work should consider model construction in the presence of big data. Two approaches may be investigated – (1) Dimension and/or data reduction, and (2) Analyzing the entire data using big data frameworks such as Apache Spark and Hadoop.

3. The constructed models can then be used for design-time and run-time decision-making. When these models are used for run-time decision-making (such as control), fast Bayesian inference techniques need to be implemented in order to update the models using the sensor data and implement appropriate control action. The Bayesian inference can be sped up by implementing fast and approximate algorithms (such as Kalman filter), perform parallelized MCMC or other inference algorithms in a distributed framework (such as Apache Spark). Kalman filter involves strong assumptions regarding the systems and the variables (as described in Section 2.6); therefore, less restrictive methods such as parallelized inference techniques need to be investigated.
4. This work considered CPS with no human involvement in its functioning. Future work should consider Human-in-the-loop CPS, also referred to as H-CPS. Analysis of H-CPS requires consideration of additional uncertainty sources associated with human such as human reliability and modeling human interactions to the CPS, which may be stochastic in nature, as every person reacts differently when presented with same scenarios.
5. In this dissertation, we considered modeling of systems using with either Bayesian networks (with sampling-based inference), or analytical probabilistic models such as Multivariate Gaussian, Gaussian Copula and Gaussian Mixture Model (each with analytical inference). For efficient computation and scalability to large-scale systems, a hybrid approach for modeling need to be investigated, where some subsystems are modeled with analytical probabilistic models (wherever appropriate) and other subsystems with Bayesian networks.

REFERENCES

[1] G. Dahll, “Combining disparate sources of information in the safety assessment of software-based systems,” *Nucl. Eng. Des.*, vol. 195, pp. 307–319, 2000.

[2] N. Friedman, M. Linial, I. Nachman, and D. Pe’er, “Using Bayesian networks to analyze expression data,” *J. Comput. Biol.*, vol. 3, no. 4, pp. 601–620, 2000.

[3] X. Jiang and G. Cooper, “A Bayesian spatio-temporal method for disease outbreak detection,” *J. Am. Med. Informatics Assoc.*, vol. 17, no. 4, pp. 462–471, 2010.

[4] M. Bensi and A. Der Kiureghian, “Seismic hazard modeling by Bayesian network and application to a high-speed rail system,” in *Proceedings of International Symposium on Reliability Engineering and Risk Management*, 2010.

[5] S. Bøttcher, “Learning Bayesian networks with mixed variables,” Department of Mathematical Science, Aalborg University, 2004.

[6] N. Friedman and M. Goldszmidt, “Discretizing continuous attributes while learning Bayesian networks,” in *Proceedings of the ICML*, 1996, pp. 157–165.

[7] B. Wilczyński and N. Dojer, “BNFinder: exact and efficient method for learning Bayesian networks,” *Bioinformatics*, vol. 25, no. 2, 2008.

[8] V. Romero, R. Rumí, and A. Salmerón, “Learning hybrid Bayesian networks using mixtures of truncated exponentials,” *Int. J. Approx. Reason.*, vol. 42, no. 1, pp. 54–68, May 2006.

[9] E. Gyftodimos and P. Flach, “Hierarchical bayesian networks: A probabilistic reasoning model for structured domains,” in *Proceedings of the ICML-2002 Workshop on Development Representations*, 2002, pp. 23–30.

[10] R. Jin, X. Du, and W. Chen, “The use of metamodeling techniques for optimization under uncertainty,” *Struct. Multidiscip. Optim.*, vol. 25, no. 2, pp. 99–116, 2003.

[11] X. Du and W. Chen, “Collaborative reliability analysis under the framework of multidisciplinary systems design,” *Optim. Eng.*, vol. 6, pp. 63–84, 2005.

[12] A. Der Kiureghian and O. Ditlevsen, “Aleatory or epistemic? Does it matter?,” *Struct. Saf.*, vol. 31, no. 2, pp. 105–112, Mar. 2009.

[13] S. Nannapaneni and S. Mahadevan, “Reliability analysis under epistemic uncertainty,” *Reliab. Eng. Syst. Saf.*, vol. 155, pp. 9–20, Nov. 2016.

[14] S. Nannapaneni, Z. Hu, and S. Mahadevan, “Uncertainty quantification in reliability estimation with limit state surrogates,” *Struct. Multidiscip. Optim.*, vol. 54, no. 6, pp. 1509–1526, Dec. 2016.

[15] S. Rangavajhala, C. Liang, and S. Mahadevan, “Design Optimization Under Aleatory and Epistemic Uncertainties,” in *Proceedings of the 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2012.

[16] S. S. Rao and L. Cao, “Optimum Design of Mechanical Systems Involving Interval Parameters,” *J. Mech. Des.*, vol. 124, no. 3, pp. 465–472, 2002.

[17] Y. Ben-Haim, “Convex models of uncertainty: Applications and implications,” *Erkenntnis*, vol. 41, pp. 139–156, 1994.

[18] A. A. Alola, M. Tunay, and V. Alola, “Analysis of Possibility Theory for Reasoning under

Uncertainty," *Int. J. Stat. Probab.*, vol. 2, no. 2, pp. 12–23, 2013.

[19] H. R. Bae, R. V. Grandhi, and R. A. Canfield, "Epistemic uncertainty quantification techniques including evidence theory for large-scale structures," *Comput. Struct.*, vol. 82, pp. 1101–1112, 2004.

[20] K. Zaman, S. Rangavajhala, M. P. McDonald, and S. Mahadevan, "A probabilistic approach for representation of interval uncertainty," *Reliab. Eng. Syst. Saf.*, vol. 96, no. 1, pp. 117–130, 2011.

[21] M. Beer, S. Ferson, and V. Kreinovich, "Imprecise probabilities in engineering analyses," *Mech. Syst. Signal Process.*, vol. 37, pp. 4–29, 2013.

[22] S. Ferson, V. Kreinovich, L. R. Ginzburg, D. S. Myers, and K. Sentz, "Constructing Probability Boxes and Dempster-Shafer Structures," in *Technical Report, Sandia National Laboratories*, 2003.

[23] B. Efron and G. Gong, "A Leisurely Look at the Bootstrap, the Jackknife, and Cross-Validation," *Am. Stat.*, vol. 37, p. 36, 1983.

[24] S. Sankararaman and S. Mahadevan, "Likelihood-based representation of epistemic uncertainty due to sparse point data and/or interval data," *Reliab. Eng. Syst. Saf.*, vol. 96, no. 7, pp. 814–824, Jul. 2011.

[25] S. Sankararaman and S. Mahadevan, "Distribution type uncertainty due to sparse and imprecise data," *Mech. Syst. Signal Process.*, vol. 37, no. 1–2, pp. 182–198, May 2013.

[26] J. A. Hoeting, D. Madigan, A. E. Raftery, and C. T. Volinsky, "Bayesian Model Averaging: A Tutorial," *Stat. Sci.*, vol. 14, no. 4, pp. 382–417, 1999.

[27] Y. Ling and S. Mahadevan, "Quantitative model validation techniques: New insights," *Reliab. Eng. Syst. Saf.*, vol. 111, pp. 217–231, 2013.

[28] M. C. Kennedy and A. O'Hagan, "Bayesian calibration of computer models," *J. R. Stat. Soc. Ser. B (Statistical Methodol.*, vol. 63, no. 3, pp. 425–464, Aug. 2001.

[29] P. D. Arendt, D. W. Apley, and W. Chen, "Quantification of Model Uncertainty: Calibration, Model Discrepancy, and Identifiability," *J. Mech. Des.*, vol. 134, no. 10, p. 100908, 2012.

[30] P. Wang, C. Hu, and B. D. Youn, "A Generalized Complementary Intersection Method (GCIM) for System Reliability Analysis," *J. Mech. Des.*, vol. 133, no. 7, p. 71003, 2011.

[31] A. Haldar and S. Mahadevan, *Probability, Reliability and Statistical Methods in Engineering Design*. New York, NY: John Wiley & Sons Inc, 2000.

[32] B. Liang and S. Mahadevan, "Error and uncertainty quantification and sensitivity analysis in mechanics computational models," *Int. J. Uncertain. Quantif.*, vol. 1, no. 2, pp. 147–161, 2011.

[33] T. Emura and Y.-S. Lin, "A Comparison of Normal Approximation Rules for Attribute Control Charts," *Qual. Reliab. Eng. Int.*, vol. 31, no. 3, pp. 411–418, Apr. 2015.

[34] A. Agresti and B. A. Coull, "Approximate is better than 'exact' for interval estimation of binomial proportions," *Am. Stat.*, vol. 52, no. 2, pp. 119–126, 1998.

[35] J. Sacks, W. Welch, T. Mitchell, and H. Wynn, "Design and analysis of computer experiments," *Stat. Sci.*, pp. 409–423, 1989.

[36] M. Seeger, "Gaussian processes for machine learning.," *Int. J. Neural Syst.*, vol. 14, no. 2, pp. 69–106, Apr. 2004.

[37] B. J. Bichon, M. S. Eldred, L. P. Swiler, S. Mahadevan, and J. M. McFarland, "Efficient Global Reliability Analysis for Nonlinear Implicit Performance Functions," *AIAA J.*, vol. 46, no. 10, pp. 2459–2468, Oct. 2008.

[38] V. Dubourg and B. Sudret, “Meta-model-based importance sampling for reliability sensitivity analysis,” *Struct. Saf.*, vol. 49, pp. 27–36, 2014.

[39] B. Echard, N. Gayton, and M. Lemaire, “AK-MCS: An active learning reliability method combining Kriging and Monte Carlo Simulation,” *Struct. Saf.*, vol. 33, no. 2, pp. 145–154, Mar. 2011.

[40] D. Jones, M. Schonlau, and W. Welch, “Efficient global optimization of expensive black-box functions,” *J. Glob. Optim.*, 1998.

[41] B. J. Bichon, J. M. McFarland, and S. Mahadevan, “Efficient surrogate models for reliability analysis of systems with multiple failure modes,” *Reliab. Eng. Syst. Saf.*, vol. 96, no. 10, pp. 1386–1395, Oct. 2011.

[42] C. Li, “Sensitivity Analysis and Uncertainty Integration for System Diagnosis and Prognosis,” Vanderbilt University, 2016.

[43] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, and J. Cariboni, *Global sensitivity analysis: the primer*. John Wiley & Sons, 2008.

[44] A. Saltelli and S. Tarantola, “On the relative importance of input factors in mathematical models: safety assessment for nuclear waste disposal,” *J. Am. Stat. Assoc.*, vol. 97, no. 459, pp. 702–709, 2002.

[45] I. . Sobol′, “Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates,” *Math. Comput. Simul.*, vol. 55, no. 1–3, pp. 271–280, Feb. 2001.

[46] A. Saltelli, S. Tarantola, and K. Chan, “A quantitative model-independent method for global sensitivity analysis of model output,” *Technometrics*, vol. 41, no. 1, pp. 39–56, 1999.

[47] S. Tarantola, D. Gatelli, and T. Mara, “Random balance designs for the estimation of first order global sensitivity indices,” *Reliab. Eng. Syst. Saf.*, vol. 91, no. 6, pp. 717–727, 2006.

[48] D. Sparkman, J. Garza, H. Millwater Jr, and B. Smarslok, “Importance Sampling-based Post-Processing Method for Global Sensitivity Analysis,” in *Proceedings of the 18th AIAA Non-Deterministic Approaches Conference*, 2016.

[49] C. Li and S. Mahadevan, “An efficient modularized sample-based method to estimate the first-order Sobol index,” *Reliab. Eng. Syst. Saf.*, vol. 153, pp. 110–121, 2016.

[50] S. Sankararaman and S. Mahadevan, “Separating the contributions of variability and parameter uncertainty in probability distributions,” *Reliab. Eng. Syst. Saf.*, vol. 112, pp. 187–199, 2013.

[51] C. Li and S. Mahadevan, “Relative contributions of aleatory and epistemic uncertainty sources in time series prediction,” *Int. J. Fatigue*, vol. 82, pp. 474–486, 2016.

[52] M. Scutari, “Learning Bayesian Networks with the bnlearn R Package,” *J. Stat. Softw.*, vol. 35, no. 3, pp. 1–22, 2010.

[53] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT Press, 2009.

[54] K. P. Murphy, “Dynamic bayesian networks: representation, inference and learning,” University of California, Berkeley, 2002.

[55] D. Straub, “Stochastic modeling of deterioration processes through dynamic Bayesian networks,” *J. Eng. Mech.*, vol. 135, no. 10, pp. 1089–1099, 2009.

[56] K. Medjaher, J.-Y. Moya, and N. Zerhouni, “Failure prognostic by using dynamic Bayesian

networks," *IFAC Proc. Vol.*, vol. 42, no. 5, pp. 257–262, 2009.

[57] M. Jha, "Dynamic Bayesian network for predicting the likelihood of a terrorist attack at critical transportation infrastructure facilities," *J. Infrastruct. Syst.*, vol. 15, no. 1, pp. 31–39, 2009.

[58] C. Li, S. Mahadevan, Y. Ling, S. Choze, and L. Wang, "Dynamic Bayesian Network for Aircraft Wing Health Monitoring Digital Twin," *AIAA J.*, vol. 55, no. 3, pp. 1–12, 2017.

[59] I. Simonsson and P. Mostad, "Exact Inference on Conditional Linear Γ-Gaussian Bayesian Networks," in *Conference on Probabilistic Graphical Models*, 2016, pp. 474–486.

[60] A. Darwiche, "A Differential Approach to Inference in Bayesian Networks," *J. ACM*, vol. 50, no. 3, pp. 280–305, 2003.

[61] S. Chib and E. Greenberg, "Understanding the metropolis-hastings algorithm," *Am. Stat.*, vol. 49, no. 4, pp. 327–335, 1995.

[62] W. Gilks and P. Wild, "Adaptive rejection sampling for Gibbs sampling," *Appl. Stat.*, pp. 337–348, 1992.

[63] R. Neal, "Slice sampling," *Ann. Stat.*, pp. 705–741, 2003.

[64] R. Olfati-Saber, "Distributed Kalman filtering for sensor networks," in *46th International Conference on Decision and Control*, 2007, pp. 5492–5498.

[65] R. Dhaouadi, N. Mohan, and L. Norum, "Design and implementation of an extended Kalman filter for the state estimation of a permanent magnet synchronous motor," *IEEE Trans. Power Electron.*, vol. 6, no. 3, pp. 491–497, 1991.

[66] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proc. IEEE*, vol. 92, no. 3, pp. 401–422, 2004.

[67] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. signal Process.*, vol. 50, no. 2, pp. 174–188, 2002.

[68] X. Du, "System reliability analysis with saddlepoint approximation," *Struct. Multidiscip. Optim.*, vol. 42, no. 2, pp. 193–208, Aug. 2010.

[69] S. Mahadevan, R. Zhang, and N. Smith, "Bayesian networks for system reliability reassessment," *Struct. Saf.*, vol. 23, no. 3, pp. 231–251, 2001.

[70] J. Song and A. Der Kiureghian, "Bounds on System Reliability by Linear Programming," *J. Eng. Mech.*, vol. 129, no. 6, pp. 627–636, Jun. 2003.

[71] Z. Wang and P. Wang, "A double-loop adaptive sampling approach for sensitivity-free dynamic reliability analysis," *Reliab. Eng. Syst. Saf.*, vol. 142, pp. 346–356, Oct. 2015.

[72] Z. Hu and X. Du, "Mixed Efficient Global Optimization for Time-Dependent Reliability Analysis," *J. Mech. Des.*, vol. 137, no. 5, p. 51401, May 2015.

[73] B. Youn and K. Choi, "A new response surface methodology for reliability-based design optimization," *Comput. Struct.*, 2004.

[74] L. Faravelli, "Response-Surface Approach for Reliability Analysis," *J. Eng. Mech.*, vol. 115, no. 12, pp. 2763–2781, Dec. 1989.

[75] S.-K. Choi, R. V. Grandhi, and R. A. Canfield, "Structural reliability under non-Gaussian stochastic behavior," *Comput. Struct.*, vol. 82, no. 13–14, pp. 1113–1121, May 2004.

[76] M. Papadrakakis and V. Papadopoulos, "Structural reliability analyis of elastic-plastic structures

using neural networks and Monte Carlo simulation," *Comput. methods*, 1996.

[77] V. Dubourg, B. Sudret, and F. Deheeger, "Metamodel-based importance sampling for structural reliability analysis," *Probabilistic Eng. Mech.*, 2013.

[78] I. Kaymaz, "Application of kriging method to structural reliability problems," *Struct. Saf.*, vol. 27, no. 2, pp. 133–151, Apr. 2005.

[79] H. Song, K. K. Choi, I. Lee, L. Zhao, and D. Lamb, "Adaptive virtual support vector machine for reliability analysis of high-dimensional problems," *Struct. Multidiscip. Optim.*, vol. 47, no. 4, pp. 479–491, Apr. 2013.

[80] Z. Hu, S. Mahadevan, and X. Du, "Uncertainty Quantification in Time-Dependent Reliability Analysis," in *ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2015.

[81] Z. Jiang, W. Chen, Y. Fu, and R.-J. Yang, "Reliability-Based Design Optimization with Model Bias and Data Uncertainty," *SAE Int. J. Mater. Manuf.*, vol. 6, no. 3, pp. 2013-01–1384, Apr. 2013.

[82] Z. Hu and S. Mahadevan, "Global sensitivity analysis-enhanced surrogate (GSAS) modeling for reliability analysis," *Struct. Multidiscip. Optim.*, vol. 53, no. 3, pp. 501–521, 2016.

[83] Y. Ling, "Uncertainty Quantification in Time-Dependent Reliability Analysis," Vanderbilt University, 2013.

[84] E. DeCarlo, B. Smarslok, and S. Mahadevan, "Segmented Bayesian Calibration of Multidisciplinary Models," *AIAA J.*, pp. 3727–3741, 2016.

[85] S. Nannapaneni, S. Mahadevan, and S. Rachuri, "Performance evaluation of a manufacturing process under uncertainty using Bayesian networks," *J. Clean. Prod.*, vol. 113, pp. 947–959, Feb. 2016.

[86] D. Reynolds, "Gaussian mixture models," in *Encyclopedia of biometrics*, Springer US, 2015, pp. 659–663.

[87] J. Hartigan and M. Wong, "Algorithm AS 136: A k-means clustering algorithm," *J. R. Stat. Soc. Ser. C (*, vol. 28, no. 1, pp. 100–108, 1979.

[88] R. Neapolitan, *Learning Bayesian Networks*. Upper Saddle River, NJ, USA: Pearson Prentice Hall, 2004.

[89] K. Zaman, M. McDonald, and S. Mahadevan, "Probabilistic Framework for Uncertainty Propagation With Both Probabilistic and Interval Variables," *J. Mech. Des.*, vol. 133, no. 2, p. 21010, 2011.

[90] P. Bourhis, J. Reutter, F. Suárez, and D. Vrgoč, "JSON: data model, query languages and schema specification," in *Proceedings of the 36th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, 2017, pp. 123–135.

[91] A. Patil, D. Huard, and C. Fonnesbeck, "PyMC: Bayesian stochastic modelling in Python," *J. Stat. Softw.*, vol. 35, no. 4, 2010.

[92] Y. Saeys, I. Inza, and P. Larrañaga, "A review of feature selection techniques in bioinformatics," *bioinformatics*, vol. 23, no. 19, pp. 2507–2517, 2007.

[93] R. Kohavi and G. John, "Wrappers for feature subset selection," *Artif. Intell.*, vol. 97, no. 1, pp. 273–324, 1997.

[94] C. A. Felippa, K. C. Park, and C. Farhat, "Partitioned analysis of coupled mechanical systems,"

Comput. Methods Appl. Mech. Eng., vol. 190, no. 24, pp. 3247–3270, 2001.

[95] X. Sun, N. Huang, B. Wang, and J. Zhou, "Reliability of cyber physical systems assessment of the aircraft fuel management system," in *4th IEEE Annual International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, 2014, pp. 424–428.

[96] L. Wu and G. Kaiser, "FARE: A framework for benchmarking reliability of cyber-physical systems," in *IEEE Conference on Systems, Applications and Technology Conference (LISAT)*, 2013, pp. 1–6.

[97] Z. Li and R. Kang, "Strategy for reliability testing and evaluation of cyber physical systems," in *IEEE International Conference on Industrial Engineering and Engineering Management*, 2015, pp. 1001–1006.

[98] S. Nannapaneni, S. Mahadevan, S. Pradhan, and A. Dubey, "Towards Reliability-Based Decision Making in Cyber-Physical Systems," in *2016 IEEE International Conference on Smart Computing (SMARTCOMP)*, 2016, pp. 1–6.

[99] C. Li, S. Mahadevan, Y. Ling, L. Wang, and S. Choze, "A dynamic Bayesian network approach for digital twin," in *19th AIAA Non-Deterministic Approaches Conference*, 2017, p. 1566.

[100] A. Verma, S. Ajit, and D. Karanki, "Software Reliability," in *Reliability and Safety Engineering*, London: Springer, 2016, pp. 183–217.

[101] A. Dubey, G. Karsai, and N. Mahadevan, "A component model for hard real-time systems: CCM with ARINC-653," *Softw. Pract. Exp.*, vol. 41, no. 12, pp. 1517–1550, 2011.

[102] N. Mahadevan, A. Dubey, and G. Karsai, "Application of software health management techniques," in *Proceedings of the 6th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, 2011, pp. 1–10.

[103] N. C. Barford, "Experimental measurements: precision, error and truth," *Chichester Wiley, 1985, 2nd ed.*, 1985.

[104] "Keen Home." [Online]. Available: https://keenhome.io.

[105] F. Zamora-Martínez, P. Romeu, P. Botella-Rocamora, and J. Pardo, "On-line learning of indoor temperature forecasting models towards energy efficiency," *Energy Build.*, vol. 83, pp. 162–172, 2014.

[106] "SML2010 Data Set." [Online]. Available: https://archive.ics.uci.edu/ml/datasets/SML2010.

[107] T. F. Abdelmaguid and T. M. El-hossainy, "Optimal Cutting Parameters for Turning Operations with Costs of Quality and Tool Wear Compensation," in *Proceedings of the 2012 International Conference on Industrial Engineering and Operations Management Istanbul, Turkey, July 3 – 6*, 2012, vol. 6, pp. 924–932.

[108] M. El Baradie, "The effect of varying the workpiece diameter on the cutting tool clearance angle in tool-life testing," *Wear*, vol. 1, no. 2, pp. 201–205, 1996.

[109] T. Yandayan and M. Burdekin, "In-process dimensional measurement and control of workpiece accuracy," *Int. J. Mach. Tools*, vol. 37, no. 10, pp. 1423–1439, 1997.

[110] S. Binks, "The development of a laser-operated scanning rod gauge," *Meas. Control*, vol. 4, no. 4, pp. 49–53, 1971.

[111] R. Löhner, C. Yang, J. Cebral, J. Baum, H. Luo, D. Pelessone, and C. Charman, "Fluid-structure-thermal interaction using a loose coupling algorithm and adaptive unstructured grids," in *Proceedings of the 29th AIAA Fluid Dynamics Conference*, 1998.

[112] G. Morgenthal and A. McRobie, “A comparative study of numerical methods for fluid structure interaction analysis in long-span bridge design,” *Wind Struct.*, vol. 5, no. 2, pp. 101–114, 2002.

[113] I. Kroo, S. Altus, R. Braun, P. Gage, and I. Sobieski, “Multidisciplinary optimization methods for aircraft preliminary design,” in *Proceedings of the 5th Symposium on Multidisciplinary Analysis and Optimization*, 1994.

[114] P. Koch, B. Wujek, O. Golovidov, and T. Simpson, “Facilitating probabilistic multidisciplinary design optimization using kriging approximation models,” in *Proceedings of the Ninth AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 2002.

[115] S. Mahadevan and N. Smith, “Efficient first-order reliability analysis of multidisciplinary systems,” *Int. J. Reliab. Saf.*, vol. 1, pp. 137–154, 2006.

[116] S. Sankararaman and S. Mahadevan, “Likelihood-Based Approach to Multidisciplinary Analysis Under Uncertainty,” *J. Mech. Des.*, vol. 134, no. 3, p. 31008, 2012.

[117] C. Liang, “Multidisciplinary Analysis and Optimization under Uncertainty,” Vanderbilt University, 2016.

[118] C. Liang and S. Mahadevan, “Multidisciplinary Optimization under Uncertainty Using Bayesian Network,” *SAE Int. J. Mater. Manuf.*, vol. 9, pp. 419–429, 2016.

[119] C. Liang and S. Mahadevan, “Reliability-Based Multi-Objective Optimization Under Uncertainty,” in *Proceedings of the 16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2015.

[120] M. Zuo, R. Jiang, and R. Yam, “Approaches for reliability modeling of continuous-state devices,” *IEEE Trans. Reliab.*, vol. 48, no. 1, pp. 9–18, 1999.

[121] S. Sun, C. Zhang, and G. Yu, “A Bayesian network approach to traffic flow forecasting,” *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 1, pp. 124–132, 2006.

[122] J. Yu and S. Qin, “Multimode process monitoring with Bayesian inference-based finite Gaussian mixture models,” *AIChE J.*, vol. 54, no. 7, pp. 1811–1829, 2008.

[123] G. Terejanu, P. Singla, T. Singh, and P. Scott, “Uncertainty propagation for nonlinear dynamic systems using Gaussian mixture models,” *J. Guid. Control. Dyn.*, vol. 31, no. 6, pp. 1623–1633, 2008.

[124] H. Zhang, C. Giles, H. Foley, and J. Yen, “Probabilistic community discovery using hierarchical latent gaussian mixture model,” *AAAI*, vol. 7, pp. 663–668, 2007.

[125] Y. Noh, K. Choi, and L. Du, “Reliability-based design optimization of problems with correlated input variables using a Gaussian Copula,” *Struct. Multidiscip. Optim.*, vol. 38, pp. 1–16, 2009.

[126] K. Zaman, M. McDonald, S. Mahadevan, and L. Green, “Robustness-based design optimization under data uncertainty,” *Struct. Multidiscip. Optim.*, vol. 44, no. 2, pp. 183–197, Feb. 2011.

[127] B. Youn and Z. Xi, “Reliability-based robust design optimization using the eigenvector dimension reduction (EDR) method,” *Struct. Multidiscip. Optim.*, vol. 37, no. 5, pp. 475–492, 2009.

[128] A. Culler and J. McNamara, “Impact of fluid-thermal-structural coupling on response prediction of hypersonic skin panels,” *AIAA J.*, vol. 49, no. 11, pp. 2393–2406, 2011.

[129] R. Shachter, “Evaluating influence diagrams,” *Oper. Res.*, vol. 34, no. 6, pp. 871–882, 1986.

[130] H. Joe, *Dependence modeling with copulas*. CRC Press, 2014.

[131] W. Pan, “Akaike’s information criterion in generalized estimating equations,” *Biometrics*, vol. 57,

no. 1, pp. 120–125, 2001.

[132] D. Weakliem, “A critique of the Bayesian information criterion for model selection,” *Sociol. Methods Res.*, vol. 27, no. 3, pp. 359–397, 1999.

[133] J. Bilmes, “A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models,” *Int. Comput. Sci. Inst.*, vol. 4, no. 510, p. 126, 1998.

[134] C. Liang and S. Mahadevan, “Stochastic Multidisciplinary Analysis with High Dimensional Coupling,” *AIAA J.*, vol. 54, no. 4, pp. 1209–1219, 2016.

[135] “Uninet package from Lightwist software.” [Online]. Available: http://www.lighttwist.net/wp/uninet. [Accessed: 04-Aug-2017].

[136] D. Finkel, “Global optimization with the DIRECT algorithm,” North Carolina State University, 2005.