

Classification Algorithms
based on
Generalized Polynomial Chaos

by

Yuncheng Du

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Chemical Engineering

Waterloo, Ontario, Canada, 2016

© Yuncheng Du 2016

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Classification is one of the most important tasks in process system engineering. Since most of the classification algorithms are generally based on mathematical models, they inseparably involve the quantification and propagation of model uncertainty onto the variables used for classification. Such uncertainty may originate from either a lack of knowledge of the underlying process or from the intrinsic time varying phenomena such as unmeasured disturbances and noise. Often, model uncertainty has been modeled in a probabilistic way and Monte Carlo (MC) type sampling methods have been the method of choice for quantifying the effects of uncertainty. However, MC methods may be computationally prohibitive especially for nonlinear complex systems and systems involving many variables.

Alternatively, stochastic spectral methods such as the generalized polynomial chaos (gPC) expansion have emerged as a promising technique that can be used for uncertainty quantification and propagation. Such methods can approximate the stochastic variables by a truncated gPC series where the coefficients of these series can be calculated by Galerkin projection with the mathematical models describing the process. Following these steps, the gPC expansion based methods can converge much faster to a solution than MC type sampling based methods.

Using the gPC based uncertainty quantification and propagation method, this current project focuses on the following three problems: (i) fault detection and diagnosis (FDD) in the presence of stochastic faults entering the system; (ii) simultaneous optimal tuning of a FDD algorithm and a feedback controller to enhance the detectability of faults while mitigating the closed loop process variability; (iii) classification of apoptotic cells versus normal cells using morphological features identified from a stochastic image segmentation algorithm in combination with machine learning techniques. The algorithms developed in this work are shown to be highly efficient in terms of computational time, improved fault diagnosis and accurate classification of apoptotic versus normal cells.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to Professor Hector M. Budman and Professor Thomas A. Duever. I truly appreciate their continuous support, encouragement, understanding, and their selfless dedication to both my personal and academic development. Without their help, this work would not have been possible.

I also would like to thank the members of my Ph.D. examining committee: Professor Sirish L. Shah, Professor Fakhri O. Karray, Professor Ali Elkamel and Professor Luis Ricardez-Sandoval, for devoting the time to reading my thesis and for providing valuable suggestions.

I also thank everyone in my research group for all the memories we have in the last four years, and also, to all my friends for the wonderful moments.

Dedication

To you as a reader.

To everyone for coming into my life and giving me the memories.

Table of Contents

AUTHOR'S DECLARATION	ii
Abstract.....	iii
Acknowledgements	iv
Dedication.....	v
Table of Contents.....	vi
List of Figures.....	xi
List of Tables	xiii
Chapter 1 Introduction.....	1
1.1 Background	1
1.2 Objectives.....	2
1.3 Contributions.....	3
Chapter 2 Theoretical Background and Literature Review	5
2.1 Spectral Representation of Stochastic Process	5
2.1.1 Quantification of Uncertainty	5
2.1.2 Generalized Polynomial Chaos Expansion	6
2.1.3 Uncertainty Propagation	9
2.2 Fault Detection and Diagnosis	11
2.2.1 Model based Analytical Methods.....	12
2.2.2 Data Driven based Empirical Methods	13
2.2.3 Hybrid Algorithms	15
2.2.4 Interaction between Control and Fault Diagnosis	15
2.2.5 Estimation based on Sequential Monte Carlo Methods	16
2.3 Classification of Cells States.....	17
2.3.1 Microscopic Image Acquisition	17
2.3.2 Image Segmentation Algorithms	18
2.4 Conclusion.....	20
Chapter 3 Fault Detection and Diagnosis with Parametric Uncertainty	21
3.1 Overview	21
3.2 Introduction	21
3.3 Background and Methodology	23
3.3.1 Generalized Polynomial Chaos Expansion	23
3.3.2 Stochastic Finite Difference Method: Response Representation	24
3.4 Case Study: Two-dimensional Heat Conduction Problem	26
3.4.1 Uncertainty Propagation and Model Calibration.....	28

3.4.2 Model Calibration and Optimization	30
3.4.3 FDD Algorithms	31
3.5 Results and Discussion.....	34
3.5.1 Case Study I: Individual Fault Case.....	34
3.5.2 Case Study II: Simultaneous Two Faults Case	39
3.5.3 Computational Efficiency	42
3.6 Conclusion	46
Chapter 4 Fault Diagnosis for Nonlinear Dynamic Processes	47
4.1 Overview.....	47
4.2 Introduction.....	47
4.3 Generalized Polynomial Chaos	49
4.4 Fault Detection and Diagnosis Methodology	51
4.4.1 Unknown Input Fault Detection and Classification Problem Formulation	51
4.4.2 Level-1 Algorithm	52
4.4.3 Level-2 Algorithm	54
4.4.4 Summary of Level-1 and Level-2 Algorithm.....	57
4.4.5 Sensitivity Analysis based Sensor Selection.....	57
4.5 Example: Reactor-Separator Process	58
4.6 Results and Discussion.....	60
4.6.1 Model Formulation for the Reactor-Separator Process.....	60
4.6.2 Sensor Selection based on Sensitivity Analysis.....	61
4.6.3 Level-1 Algorithm with PDF Profiles.....	62
4.6.4 FDD with Level-2 Algorithm using Maximum Likelihood.....	65
4.6.5 FDD with Level-2 Algorithm using Bayesian Inference	66
4.6.6 FDD Using Solely Level-2 Algorithm.....	67
4.6.7 Application of the Two-level Algorithm to Detect An Incipient Fault	69
4.6.8 Comparison Studies to Particle Filter based Fault Detection.....	70
4.7 Conclusion	71
4.8 Supplementary Material	71
Chapter 5 Integration of Fault Diagnosis and Process Control.....	74
5.1 Overview.....	74
5.2 Introduction.....	74
5.3 Theoretical Background	76
5.4 Optimal Tuning of Controller Supervised by a FDD Algorithm.....	78
5.4.1 Fault Detection and Diagnosis Algorithm	79

5.4.2 Integration of Control and FDD Algorithm	82
5.5 Case Study.....	84
5.6 Results and Discussion.....	86
5.6.1 Formulation of gPC model.....	86
5.6.2 Stability Constraints for the gPC model.....	87
5.6.3 Calibration of the PDF Profiles for Inner Optimization.....	87
5.6.4 Case Study 1: Tuning of a Gain-Scheduled Controller.....	88
5.6.5 Case Study 2: Tuning of a Gain-Scheduled Controller and Set-point.....	90
5.6.6 Case Study 3: Tuning of a Global Controller.....	90
5.6.7 Case Study 4: Investigation of Tuning Weights.....	91
5.6.8 Comparison of Fault Identification Results	93
5.6.9 Discussion of Computational Efficiency.....	96
5.7 Conclusions	97
Chapter 6 Classification of Apoptotic and Normal Cells	98
6.1 Overview.....	98
6.2 Introduction.....	98
6.3 Segmentation of Stochastic Images.....	100
6.3.1 Fluorescence Imaging	100
6.3.2 Active Contours without Edges	101
6.3.3 Approximation of Intensity with gPC	103
6.3.4 Stochastic Level Set Function based Segmentation	104
6.4 Automated Classification of Cells.....	108
6.4.1 Feature Extraction.....	108
6.4.2 SVM based Differentiation	109
6.5 Results and Discussion.....	109
6.5.1 Model Calibration	109
6.5.2 Image Segmentation with One Random Variable.....	110
6.5.3 Morphological Features	112
6.5.4 Differentiation of Cells States.....	114
6.6 Conclusion.....	115
Chapter 7 Concluding Remarks and Future Work.....	116
7.1 Overview.....	116
7.2 Concluding Remarks.....	117
7.3 Future Work	118
7.3.1 Arbitrary Uncertainty Quantification and Propagation	118

7.3.2	Integration of Plant Design, Control and Fault Diagnosis	119
7.3.3	Image Segmentation and Classification	119
Appendix A	Comparison of Stochastic Fault Diagnosis Algorithms	120
A.1	Overview	120
A.2	Introduction	120
A.3	Problem Formulation and Theoretical Background	122
A.3.1	Formulation of Unknown Stochastic Faults	122
A.3.2	Generalized Polynomial Chaos Expansion	123
A.3.3	Gaussian Process	125
A.4	Fault Detection and Diagnosis Algorithms	126
A.4.1	FDD using First Principle Models	126
A.4.1.1	FDD Algorithm	126
A.4.1.2	Sensitivity Analysis based Sensor Selection for FDD	127
A.4.1.3	Quantitative Analysis of Fault Detectability	128
A.4.2	FDD using Empirical Models	129
A.4.2.1	FDD Algorithm	129
A.4.2.2	Model Calibration through Minimal Adjustments	129
A.5	Case Study	131
A.6	Results and Discussion	132
A.6.1	Model Formulation with gPC approximations	132
A.6.2	Sensitivity Analysis with gPC model	134
A.6.3	Model Calibration with the gPC model	134
A.6.4	Detectability Analysis with gPC model	136
A.6.5	Minimum Adjustment of GP model	136
A.6.6	Summary of Fault Detection Rate	137
A.6.7	Combination of the gPC model with GP model	138
A.6.8	Discussion and Comparison	139
A.7	Conclusion	140
Appendix B	Reactivity Ratio Estimation in Copolymerization	141
B.1	Overview	141
B.2	Introduction	141
B.3	Background and Methodology Formulation	143
B.3.1	Stochastic Inverse Problem	143
B.3.2	Generalized Polynomial Chaos Expansion	144
B.3.3	Gram-Schmidt Orghogonalization	146

B.3.4 Maximum Likelihood Estimation with gPC model.....	147
B.3.5 Parameter Estimation Algorithms	148
B.4 Case Studies: Reactivity Ratio Estimations in Copolymerization	151
B.5 Results and Discussion	153
B.5.1 Reactivity Ratio Estimation.....	153
B.5.1 Parameter Estimation with Gram-Schmidt Polynomial Chaos.....	155
B.5.3 Joint Confidence Regions for Parameter Estimation	157
B.5.4 Uncertainty Quantification of Measured Variables	161
B.5.5 Computational Efficiency.....	162
B.6 Conclusion	163
Appendix C Quantitative Analysis of Normal and Apoptotic Cells	164
C.1 Overview	164
C.2 Introduction	164
C.3 Fluorescence Imaging.....	165
C.4 Image Processing Methodology	166
C.4.1 Image Segmentation.....	166
C.4.2 Coarse Segmentation using a combination of a Range Filter and a Marching Square Algorithm ...	168
C.4.3 Feature Extraction	170
C.4.4 SVM based Classification	171
C.5 Results and Discussion	172
C.5.1 Coarse Segmentation Results	172
C.5.2 Comparison of Image Segmentation Results.....	173
C.5.3 Comparison of Computational Time Required for Segmentation	174
C.5.4 Feature Extraction	174
C.5.5 Differentiation Results using a SVM Classifier	175
C.6 Conclusion	176
Bibliography	177

List of Figures

Figure 2.1 General scheme of fault detection and diagnosis	11
Figure 3.1 Faults profiles.....	27
Figure 3.2 The grid points for two-dimensional heat conduction problem.....	29
Figure 3.3 Flowchart to formulate the adaptive optimization model.....	31
Figure 3.4 FDD algorithm by using the PDF profiles of measured variables.....	31
Figure 3.5 Joint Confidence Region (JCR) array	32
Figure 3.6 Sketch of JCR based FDD algorithm	33
Figure 3.7 Mean and variance distribution over two-dimensional domain	34
Figure 3.8 Sensors placement for model optimization (top-left part of the square domain)	35
Figure 3.9 PDF profiles of six classes at grid point 8 by gPC model ($Q = -100$).....	36
Figure 3.10 Fault detection rate for single fault with gPC model.....	37
Figure 3.11 Fault detection rate for single fault by gPC model with 10 replicates.....	38
Figure 3.12 Fault detection rate for single fault with different weights.....	38
Figure 3.13 Mean and variance distribution over two-dimensional domain	39
Figure 3.14 Sensor placements for Case II (one stochastic boundary).....	40
Figure 3.15 JCRs for two measurements at sensor 1 and 3 with a 99% confidence interval.....	41
Figure 3.16 Comparisons of expected value (a) and variance (b) between gPC and MC	42
Figure 3.17 Comparison of model calibration results between gPC and MC (single fault)	43
Figure 3.18 PDF profiles of six classes at grid 8 by MC ($Q = -100$, 10,000 samples).....	44
Figure 3.19 Comparison of result at each grid point between gPC and MC ($Q = -100$)	44
Figure 4.1 Fault profile representing an intermittent stochastic input fault and resulting measured variable	51
Figure 4.2 Visual interpretation of FDD with the level-1 algorithm	54
Figure 4.3 Two reactors in series with separator and recycle unit.....	59
Figure 4.4 Comparisons of the gPC model and MC simulations using controlled variable T_1	61
Figure 4.5 Multi-level pseudo random sequence.....	62
Figure 4.6 The PDF profiles of the measured variable (Q_i) at 3 operating modes.....	64
Figure 4.7 Illustration of Bayesian inference estimation based fault detection	66
Figure 4.8 Illustration of Maximum likelihood based fault estimator	68
Figure 5.1 Fault profile representing an intermittent stochastic input fault and resulting measured variable	79
Figure 5.2 The PDF profiles of measured variables	81
Figure 5.3 The CSTR with a concentration control loop and typical industrial stochastic faults	85
Figure 5.4 Simulation results of the gPC model, MC simulations and deterministic nonlinear model	86
Figure 5.5 Multi-level pseudo random sequence.....	88
Figure 5.6 The PDF profiles of five operating modes on inlet concentration.....	89

Figure 5.7 Illustration of the effect of weights on the control performance	92
Figure 5.8 Illustration of maximum likelihood estimation based fault detection	95
Figure 6.1 Fluorescent photomicrograph of CHO cells stained with AO and EB	100
Figure 6.2 Visual interpretation of stochastic images	103
Figure 6.3 Stochastic segmentation algorithm	108
Figure 6.4 Sketch of the morphological feature along the boundary	109
Figure 6.5 Segmentation results and PDF of pixel intensities defining boundary	110
Figure 6.6 Visual illustration of pixels intensities in the background	111
Figure 6.7 Segmentation results with deterministic and stochastic level set algorithms	112
Figure 6.8 Histograms of curvature for apoptotic and normal cells	113
Figure 6.9 Visual illustration of normal and apoptotic testing cells	114

List of Tables

Table 2.1 Correspondence of Wiener-Askey polynomial and random input.....	9
Table 3.1 Comparison of acceptance rate for six sensor placement structures.....	35
Table 3.2 Summary of model calibration results (noise variance $\sigma^2=0.1$)	36
Table 3.3 Summary of model calibration results (noise variance $\sigma^2=0.1$)	40
Table 3.4 Summary of results for fault detection rate for two simultaneous faults (noise variance $\sigma^2=0.1$).....	41
Table 3.5 Type I and Type II analysis for training set (gPC)	45
Table 3.6 Type I and Type II analysis for training set (MC)	46
Table 4.1 Parameter declaration for the Reactor-Separator process	60
Table 4.2 Sensitivity analysis of reactor 1	62
Table 4.3 Sensitivity analysis of reactor 2.....	62
Table 4.4 Sensitivity analysis of separator	62
Table 4.5 Model calibration result for the <i>level-1 algorithm</i>	63
Table 5.1 Parameter declaration and setting used for CSTR	85
Table 5.2 Comparison of the <i>inner level optimization</i> strategies (noise 1%)	88
Table 5.3 Summary of the results for the <i>outer level optimization</i> without tuning set point	89
Table 5.4 Summary of the results for the outer level optimization with tuning set point.....	90
Table 5.5 Summary of the <i>FIR</i> using transient measurements	93
Table 5.6 Summary of inner level optimization with Latin hypercube sampling.....	96
Table 6.1 Examples of feature vector (apoptosis)	113
Table 6.2 Examples of feature vector (normal)	113
Table 6.3 Summary of differentiation rate	115

Chapter 1

Introduction

1.1 Background

The quantitative analysis of phenomena occurring in many engineering applications is generally based on mathematical models. Such models can provide a representation of a real system by using a number of hypotheses, approximations and parameters. The system of interest cannot be exactly characterized in practice since models are never exact. Model uncertainties may originate from: (i) a lack of knowledge about the underlying process, (ii) the intrinsic time varying nature of model parameters; and (iii) the inaccurate measurements due to random noise. Thus uncertainties are generally related to both errors in the assumed model structures as well as inaccuracies in the estimated model parameters. Three main tasks are involved in the use of models with uncertainties, (a) the quantification of these uncertainties from data, (b) the propagation of the uncertainties through the mathematical model onto variables of interest, and (c) the characterization of the models' outputs resulting from the propagation of the uncertainty.

Probabilistic analysis such as Monte Carlo (MC) simulations is the most popular method for propagating uncertainties and characterizing models' outputs for uncertain models. For this approach, uncertainty can be quantified by drawing a large number of samples and running the model with each of these samples. However, approaches such as MC simulations are computationally prohibitive especially for complex systems. Moreover, the uncertainty propagation results may be questionable when the available information does not provide a strong basis/support for a particular probability assumption. To improve the computational efficiency and the accuracy of the uncertainty propagation step, the generalized polynomial chaos (gPC) in this work which leads to significant reduction in computational time. Then, using a gPC approach, it was possible to treat in this thesis a variety of problems that would be otherwise computationally prohibitive when approached with MC methods.

Abnormal events defined as faults such as sensor/actuator failures usually occur in chemical processes, which can affect the process reliability and lead to economic losses. Different fault detection and diagnosis (FDD) approaches can be used to diagnose and isolate faults, prevent them from propagating, and improve the reliability and efficiency of the supervisory control. The main restrictive factor of an efficient model-based FDD algorithm is the model uncertainty. The step of quantifying the effect of uncertainty onto the variables used for isolation or diagnosis is typically omitted, leading to a loss of the performance of the FDD algorithm. Moreover, faults often may occur intermittently, i.e., systems may switch between non-faulty to faulty operating conditions in a random fashion. Such intermittent occurrences are difficult to diagnose and further complicate the proper detection of faults. In terms of application, fault diagnosis that explicitly considers the dynamic transients has not been extensively addressed in the literature. FDD algorithms that are based on steady state analysis may result in high false alarm rate or mis-detection of faults, when they use data collected during dynamic transients. In the current

work, the gPC method is combined either with the Maximum Likelihood or with Bayesian Inference to recursively estimate faults of a stochastic nature, while taking the uncertainty and dynamic transients into account.

In practice, most of the available FDD systems are implemented at a supervisory hierarchical level above the closed-loop control system and use measurements that are also used for feedback control. While there is a large body of methods for FDD, the problem of integrating process control and fault diagnosis algorithm has not been addressed as much in particular in the presence of stochastic faults. The key challenge for such integration is that these two activities have competing objectives. For example, if the measured quantities are perfectly controlled, they will not a sufficient amount of variability required for detection of faults. Thus, there is a trade-off between the closed loop control performance and the fault detectability. The optimal trade-off between these two activities has been addressed in the present project by a bi-level optimization problem that is accounting for the uncertainty and dynamic transients.

Automated cell detection and characterization is important in many problems such as cancer research, stem cell research and wound healing. Studying in vitro cellular behavior via living-cell imaging and high throughput screening involves a great amount of imaging data. Accurate and fast quantitative analysis of these images is useful for the evaluations of experimental outcomes and cells' culture protocols. However, these images usually have varying image qualities, and the manual quantification and analysis of these data is time consuming and prone to errors. Motivated by this, the current work proposed new image processing tools to segment cells from the background in a computationally efficient way. The main idea behind automated image segmentation is to detect the boundary of cells and separate the cells from the background. However, any measurement error due to the noise or uncertainty in the pixels' intensities may result in significant variations in the results of segmentation. To address this problem, a stochastic image segmentation algorithm is developed to account for the uncertainty in a given image.

1.2 Objectives

In this current project, the following objectives were investigated:

- i- The development of new fault detection and diagnosis (FDD) algorithm to identify and diagnose stochastic intermittent fault/s and evaluate the detectability of faults with statistical analysis methods.
- ii- The development of recursive FDD algorithms to improve accuracy of fault diagnosis accounting for dynamic transients and uncertainties.
- iii- The investigation of the trade-off between fault detectability and closed loop control performance.
- iv- The development of efficient algorithms to distinguish apoptotic versus normal cells using identified morphological features of cells in combination with machine learning techniques.

1.3 Contributions

To summarize, the contributions of this current work are (i) the use of generalized polynomial chaos (gPC) expansions for efficient uncertainty quantification and propagation, and (2) their application to a wide array of engineering problems including fault detection and diagnosis (FDD), integration of FDD and feedback control, and efficient image segmentation. The contributions in each of chapter of this work can be summarized as:

- i- Chapter 2 provides an up-to-date literature review that covers the main aspects of this work, i.e., gPC based uncertainty propagation, FDD, integration of fault detection and control, as well as image segmentation.
- ii- Chapter 3 presents a computationally efficient FDD algorithm and its application to a two-dimensional heat conduction problem. The proposed method is specifically targeted to detect the average of input faults consisting of stochastic perturbations around mean values that change intermittently. The detectability of faults is assessed by calculating Type I and Type II error. This method is shown to be significantly better in terms of computational efficiency and accuracy as compared to Monte Carlo simulations.
- iii- Chapter 4 develops FDD algorithms to identify fault/s of a stochastic nature with dynamic transients by combining gPC approximation with nonlinear models of the process and by using either the Maximum Likelihood or the Bayesian Inference based estimators. Optimal selection of sensors is addressed based on sensitivity analysis of the gPC model. This method is shown to be more computationally efficient than an equivalent Particle Filter and less sensitive to the user selected tuning parameters as compared to Particle Filter (PF).
- iv- Chapter 5 investigates the problem of the optimal simultaneous tuning of a FDD algorithm and a controller in the presence of stochastic time varying faults. This method is successful in achieving a trade-off between fault detectability and closed loop control performance, and is advantageous in terms of computational efficiency as well as fast fault detection.
- v- Chapter 6 presents an efficient gPC model based image segmentation algorithm for fast segmentation of fluorescence microscopy images of Chinese Hamster Ovary (CHO) cells. An automated support vector machine (SVM) classifier is formulated to distinguish apoptotic versus cells based on morphological features identified with the segmentation algorithm. The combination of developed morphological feature extraction method and the trained SVM classifier is shown to be more efficient in terms of differentiation accuracy.
- vi- Chapter 7 concludes with detailed recommendations for future work on the following topic: (i) arbitrary uncertainty quantification and propagation; (ii) integration of plant design, control and fault diagnosis; (iii) Image Segmentation and Classification.

Most of the findings in the current work have been presented in referred journal papers and conferences' proceeding as below:

Referred Publications

1. **Y. Du**, T. A. Duever, H. Budman, "Fault detection and diagnosis with parametric uncertainty using generalized polynomial chaos", **Computers and Chemical Engineering**, vol. 76, p. 63~75, 2015.
2. **Y. Du**, H. Budman, T. A. Duever, "Integration of fault diagnosis and control based on a trade-off between fault detectability and closed-loop performance", **Journal of Process Control**, vol. 38, p. 42~53, 2016.
3. **Y. Du**, T. A. Duever, H. Budman, "Generalized polynomial chaos based fault detection and classification for nonlinear dynamic processes", **Industrial & Engineering Chemistry Research**, in press.
4. **Y. Du**, H. Budman, T. A. Duever, "Classification of normal and apoptotic cells from fluorescence microscopy images using generalized polynomial chaos and level set functions", **Microscopy and Microanalysis**, 2nd revision.
5. **Y. Du**, H. Budman, T. A. Duever, "Parameter estimation for an inverse nonlinear stochastic problem: reactivity ratio studies in copolymerization", **Computers and Chemical Engineering**, submitted.
6. **Y. Du**, T. A. Duever, H. Budman, "Comparison of stochastic fault detection and diagnosis algorithms for nonlinear chemical processes", **Chemometrics and Intelligent Laboratory Systems**, ready to submit.
7. **Y. Du**, H. Budman, T. A. Duever, "Segmentation and quantitative analysis of normal and apoptotic cells from fluorescence microscopy images", the 11th International Federation of Automatic Control (IFAC) Symposium on Dynamics and Control of Process Systems, including Biosystems (DYCOPS-CAB), June 6~8, 2016, Trondheim, Norway.
8. **Y. Du**, T. A. Duever, H. Budman, "Stochastic fault diagnosis using generalized polynomial chaos and maximum likelihood", the International Symposium on Advanced Control of Chemical Processes (ADCHEM), June 7~10, 2015, Whistler, British Columbia, Canada.
9. **Y. Du**, T. A. Duever, H. Budman, "Integration of fault diagnosis and control by finding a trade-off between observability of stochastic faults and economics", the 19th World Congress of the International Federation of Automatic Control (IFAC), August 24~29, 2014, Cape Town, South Africa.

Chapter 2

Theoretical Background and Literature Review

Fault diagnosis in chemical processes and classification of cells' states of bioengineering are two typical examples of classification problems in engineering. For fault diagnosis, the classification methods are used to predict whether the process is operated at faulty or non-faulty operating condition. In the context of classification of cells' states, the goal is to assess the in-vitro status of cells, e.g., healthy cells versus cells undergoing programmed cell death or apoptosis.

This chapter provides a brief literature review on the fault detection and diagnosis (FDD), and on cell imaging techniques. Section 2.1 discusses the general uncertainty quantification and propagation method used in this work. This is followed by reviews on fault detection and diagnosis methods, and on the interaction between process control and fault detection. Understanding this interaction is essential for achieving an optimal trade-off of fault detection and control, since in industrial practice both algorithms are operated simultaneously. The review on segmentation of images is given in Section 2.3 followed by a summary of the literature review in Section 2.4.

2.1 Spectral Representation of Stochastic Process

There has been a good amount of research on the numerical solution of large scale engineering problems in the presence of uncertainty (Stefanou G. , 2009). Such uncertainties may originate from either intrinsic time varying phenomena or may result from the use of stochastic noisy data for model calibration. Then, uncertainty model parameters can be used to describe the model uncertainty. Different techniques have been proposed to take the uncertainties into account from the very beginning of the problem definition and analysis (Xiu & Karniadakis, 2003). Uncertainties may be associated with uncertain boundary or initial conditions and/or geometric discrepancies between model and process. A common approach to describe uncertainty is by assuming that the uncertain parameters are stochastic quantities. However, the treatment of these uncertainties as stochastic with a specific probability distribution is not simple due to lack of relevant experimental data to calibrate this distribution. Stochastic processes can be roughly categorized into two main groups based on their probability distribution, i.e., Gaussian and non-Gaussian. The simulations of Gaussian and non-Gaussian stochastic processes are different and a review of available methods for both representations is presented in the following two subsections.

2.1.1 Quantification of Uncertainty

Although most of the uncertainties in engineering problems may be represented as non-Gaussian, the Gaussian assumption is usually made to keep the analysis simple (Spanos & Zeldin, 1998). Current available methods for simulation of Gaussian processes are divided into two categories, i.e., the spectral representation method (Shinozuka & Deodatis, 1996) and the Karhunen-Loeve (K-L) expansion (Ghanem & Spanos, 1991).

Both approaches are based on the representation of a stochastic process $f(x)$ as a summation of particular predefined functions with respect to specific random variables as follows:

$$f(x) = \sum_{n=0}^N C_n \Phi_n(x) \quad (2.1)$$

The spectral representation approach is based on expanding $f(x)$ as a sum of trigonometric functions with random phase angles ($\Phi_n(x)$ in Eq. 2.1) and amplitudes (C_n in Eq. 2.1). The simplest version of this type of representation which is widely adopted in most applications is given as a function of one random phase angle. The coefficients of the description given in Eq. 2.1 are deterministic and depend on the prescribed power spectrum of the stochastic field (Stefanou G. , 2009). Spectral representation algorithms have been employed in various kinds of Gaussian stochastic process, such as multivariate, multidimensional, and non-homogeneous problem (Liang, Chaudhuri, & Shinozuka, 2007; Spanosa, Tezcanb, & Tratskasc, 2005), and have been successfully implemented in the framework of Monte Carlo (MC) simulations for solving problems with the stochastic finite element method (Lagaros & Papadopoulos, 2006).

The K-L expansion is a special case of an orthogonal series expansion, in which the orthogonal functions are chosen as the eigenfunctions of a Fredholm integral equation. In a K-L expansion, the first term in Eq. 2.1 ($n = 0$) is the expectation of the random variable, and it is identical to 0 in most applications. In addition, $\Phi_n(x)$ is defined as the multiplication of eigenvalues by their corresponding eigenfunctions of a set of uncorrelated random variables, where the eigenvalues and eigenfunctions are calculated from the covariance function. This expansion is particularly suitable for the representation of strongly correlated random variables where only a few terms in Eq. 2.1 suffice to capture the majority of the information contained in the data used for calibration (Stefanou G. , 2009). However, there are drawbacks for the K-L expansion, which limits its application (Xiu D. , 2010). The first challenge is solving the Fredholm integral equation, since the analytical solution for this kind of integral equation is only available for simple geometries and special forms of the autocovariance function. Furthermore, the covariance function of the stochastic system is generally unknown, and the computation of eigenvalues and corresponding eigenfunctions from the autocovariance function is strongly influenced by the K-L expansion (Phoon, Huang, & Quek, 2002; Schwab & Todor, 2006). In order to overcome those shortcomings, polynomial chaos expansion (PCE) and generalized polynomial chaos (gPC) expansion were proposed.

2.1.2 Generalized Polynomial Chaos Expansion

The problem of modeling non-Gaussian uncertainty has gained considerable attention since uncertain model components often exhibit non-Gaussian probabilistic characteristics. The polynomial chaos expansion (PCE) is an alternative method to generate sample functions of non-Gaussian, non-stationary stochastic process that employs the Hermite polynomial as an orthogonal basis function of random variables. However, the Hermite polynomial has difficulties in approximating probabilities for non-Gaussian uncertainties. Subsequently, the

generalized polynomial chaos (gPC) method was proposed (Xiu & Karniadakis, 2002). Different kinds of orthogonal polynomials can be selected as basis function depending on the probability distribution function (PDF) of the random variables to be described by the expansion so as to obtain optimal convergence and to maintain orthogonality.

A random process $X(\theta)$, viewed as a function of a random event θ is expressed as:

$$\begin{aligned} X(\theta) &= a_0 H_0 + \sum_{i_1=1}^{\infty} a_{i_1} H_1(\xi_{i_1}(\theta)) + \sum_{i_1=1}^{\infty} \sum_{i_2=1}^{i_1} a_{i_1 i_2} H_2(\xi_{i_1}(\theta) \xi_{i_2}(\theta)) \\ &\quad + \sum_{i_1=1}^{\infty} \sum_{i_2=1}^{i_1} \sum_{i_3=1}^{i_2} a_{i_1 i_2 i_3} H_3(\xi_{i_1}(\theta), \xi_{i_2}(\theta), \xi_{i_3}(\theta)) + \dots \end{aligned} \quad (2.2)$$

where $H_n(\xi_{i_1}, \dots, \xi_{i_n})$ is the Hermite polynomial of order n in terms of the multidimensional independent standard Gaussian random variables $\xi = (\xi_{i_1}, \dots, \xi_{i_n})$ with zero mean and unit variance. This expression is the discrete version of the original Wiener polynomial chaos expansion, in which the continuous integrals are replaced by summations. The general equation of the Hermite polynomial is defined as:

$$H_n(\xi_{i_1}, \dots, \xi_{i_n}) = e^{1/2\xi^r \xi} (-1)^n \frac{\partial^n}{\partial \xi_{i_1} \dots \partial \xi_{i_n}} e^{-1/2\xi^r \xi} \quad (2.3)$$

For example, one dimensional Hermite polynomials are:

$$I_0 = 1, I_1 = \xi, I_2 = \xi^2 - 1, I_3 = \xi^3 - 3\xi \dots \quad (2.4)$$

For notational convenience, Eq. 2.2 can be rewritten as follows:

$$X(\theta) = \sum_{j=0}^{\infty} \hat{a}_j I_j(\xi) \quad (2.5)$$

There is one-to-one correspondence between the function $H_n(\xi_{i_1}, \dots, \xi_{i_n})$ and $I_j(\xi)$, as well as the coefficients $a_{i_1 \dots i_r}$ and \hat{a}_j . In Eq. 2.2, the summation is carried out according to ascending order of the Hermite polynomials.

The Hermite based chaos expansion sometimes converges very slowly or may diverge for non-Gaussian random inputs (Xiu D. , 2009). In order to deal with more general random inputs, t basis functions other than Hermite can be used. These basis functions are selected as per the Wiener-Askey scheme (Xiu & Karniadakis, 2002), which is a generalization of the original Wiener's Hermite-chaos expansion. Due to their ability to produce more compact representations, gPC's are considered in the current work. Similar to the one-dimensional Hermite polynomial, a general two-dimensional expansion of random process $X(\theta)$ is defined as:

$$X(\theta) = c_0 \psi_0 + \sum_{i_1=1}^{\infty} c_{i_1} \psi_1(\xi_{i_1}(\theta)) + \sum_{i_1=1}^{\infty} \sum_{i_2=1}^{i_1} c_{i_1 i_2} \psi_2(\xi_{i_1}(\theta) \xi_{i_2}(\theta))$$

$$+ \sum_{i_1=1}^{\infty} \sum_{i_2}^{i_1} \sum_{i_3}^{i_2} c_{i_1 i_2 i_3} \psi_3 \left(\xi_{i_1}(\theta), \xi_{i_2}(\theta), \xi_{i_3}(\theta) \right) + \dots \quad (2.6)$$

where $\psi_n(\xi_{i_1}, \dots, \xi_{i_n})$ is the gPC from the Askey-chaos scheme, and n is the order of multi-dimensional random variables $\boldsymbol{\xi} = (\xi_{i_1}, \dots, \xi_{i_n})$. The polynomials in Eq. 2.6 are not restricted to Hermite polynomials and are selected according to the Askey scheme dependent on the PDF of the random variables to be used in a particular problem. For example, Jacobi polynomials can be used for when the random variables have a Beta distribution. For notational convenience, Eq. 2.6 can be also expressed as:

$$\mathbf{X}(\theta) = \sum_{j=0}^{\infty} \hat{c}_j \Phi_j(\boldsymbol{\xi}) \quad (2.7)$$

There is one-to-one correspondence between the functions $\psi_n(\xi_{i_1}, \dots, \xi_{i_n})$ and $\Phi_j(\boldsymbol{\xi})$, as well as their coefficients \hat{c}_j and $c_{i_1 \dots i_r}$. Since each polynomial considered in the Askey scheme forms a complete basis in the Hilbert space determined by their corresponding support, it can be concluded that each type of Askey-chaos will converge to any L_2 functional in the L_2 sense in the corresponding Hilbert functional space, i.e.,

$$\langle \Phi_i \Phi_j \rangle = \langle \Phi_i^2 \rangle \delta_{ij} \quad (2.8)$$

where δ_{ij} is the Kronecker delta and $\langle \cdot, \cdot \rangle$ means the inner product in the Hilbert space of the variables.

$$\langle f(\boldsymbol{\xi}) g(\boldsymbol{\xi}) \rangle = \int f(\boldsymbol{\xi}) g(\boldsymbol{\xi}) W(\boldsymbol{\xi}) d\boldsymbol{\xi} \quad (2.9)$$

where $W(\boldsymbol{\xi})$ is the weighting function in Eq. 2.9, and is defined as:

$$W(\boldsymbol{\xi}) = \frac{1}{\sqrt{2\pi^n}} e^{-1/2 \boldsymbol{\xi}^T \boldsymbol{\xi}} \quad (2.10)$$

where n is the dimension of random variables $\boldsymbol{\xi}$. The key difference between gPC and many other possible expansions is that the polynomials are orthogonal with respect to the weighting function $W(\boldsymbol{\xi})$. The correspondence between the type of Wiener-Askey polynomial chaos and the uncertain inputs of continuous chaos is given in Table 2.1 (Xiu D. , 2009). It is worthwhile mentioning that uniformly distributed random variables correspond to a special case of the Jacobi polynomial with parameter $\alpha=\beta=0^*$, and this case is separately shown in table 2.1. The support is defined as the set of points where the PDF of particular polynomial is not zero-valued. Specifically, the support is defined by two parameters for the Beta as well as the Uniform distribution, a and b , which are their minimum and maximum values.

*The weighting function of a uniform distribution in $(-1, 1)$ is $W(\zeta) = 1/2$, and the first few Legendre orthogonal polynomials are:

$$u_0(\zeta) = 1, u_1(\zeta) = \zeta, u_2(\zeta) = (3/2)\zeta^2 - (1/2), \dots$$

The weighting function of a beta distribution in $(-1, 1)$ is $W(\zeta) = (1-\zeta)^\alpha (1+\zeta)^\beta$, ($\alpha, \beta > 0$), and the first few Jacobi orthogonal polynomials are:

$$b_0(\zeta) = 1, b_1(\zeta) = (1/2)[\alpha - \beta + (\alpha + \beta + 2)\zeta], \dots$$

Note that the Legendre polynomial chaos becomes a special case of the Jacobi polynomial chaos with $\alpha = \beta = 0$.

Table 2.1 Correspondence of Wiener-Askey polynomial and random input

Random Input	Polynomial	Support
Gaussian	Hermite-chaos	$(-\infty, \infty)$
Gamma	Laguerre-chaos	$[0, \infty)$
Beta	Jacobi-chaos	$[a, b]$
Uniform	Legendre-chaos	$[a, b]$

2.1.3 Uncertainty Propagation

The second part in the analysis of a stochastic system consists of propagating the effect of uncertainties in the model parameters onto the system outputs. The stochastic finite difference or element method is an extension of the corresponding classical deterministic approach and has been gaining attention in the past decades to solve stochastic problems (Ghanem & Spanos, 1991). This method basically proceeds as per the following three steps: (1) the representation of the random inputs by the spectral approach; (2) the propagation of uncertainties into the stochastic system equation (first at the element and then at the global system level); and (3) the response variability calculation with respect to the stochastic inputs/parameters.

In this work, a gPC approximation is used for the first step as per the discussion in the previous subsection. Then, for step 2, the gPC's are substituted into the governing equations and subsequently, a Galerkin projection calculation is applied to compute the coefficients of the gPC expansions using their orthogonality properties. The general procedures for Galerkin projection are presented as below.

Suppose the general stochastic elliptic partial differential equations with random inputs are given as[†]:

$$\begin{aligned} \nabla \cdot [\kappa(x; \omega) \nabla u(x; \omega)] &= f(x; \omega) && \text{on } \mathcal{D} \times \Omega \\ u(x; \omega) &= g(x; \omega) && \text{on } \partial \mathcal{D} \times \Omega \end{aligned} \quad (2.11)$$

where \mathcal{D} is the spatial domain and Ω is the probability space, f , g and κ are functions on $\mathcal{D} \times \Omega$. u is the solution, f is the source term, g is the Dirichlet boundary condition, and κ is a model parameter. All of these operators are a function of the uncertainty ω , which may be introduced into the system via stochastic boundary conditions, initial conditions, material properties, etc.

In order to solve for solution u , which is a random variable, the gPC's are employed to expand the variables as follows:

$$\kappa(x; \omega) = \sum_{i=0}^P \kappa_i(x) \phi_i(\xi) \quad (2.12)$$

$$u(x; \omega) = \sum_{i=0}^P u_i(x) \phi_i(\xi) \quad (2.13)$$

[†] The application of the gPC approximation to ordinary differential equations follows the similar procedures and will be further explained in each chapter individually.

$$f(x; \omega) = \sum_{i=0}^P f_i(x) \phi_i(\xi) \quad (2.14)$$

where the infinite summation of ξ in Eq. 2.5 has been replaced by a truncated finite term summation of $\{\phi\}$ in the finite dimensions of $\xi = \{\xi_1, \dots, \xi_n\}$. The dimensionality n of ξ is determined by the random inputs. According to the gPC expansion, the random parameter ω is embedded into the polynomial basis $\phi(\xi)$ while the coefficients in the above equations, i.e., κ_i, u_i, f_i , are deterministic.

The truncated finite summation parameter P is determined by the dimensionality (n) of random inputs and the highest order (p) of the polynomials $\{\phi_i\}$, which satisfies:

$$(P + 1) = (n + p)!/n! p! \quad (2.15)$$

In order to achieve exponential convergence in the coefficients u_i , the optimum polynomial should be chosen from the Askey-chaos scheme (see Table 2.1) and the weighting function is calculated accordingly. By substituting the expansions into Eq. 2.11:

$$\nabla \cdot \left[\sum_{i=0}^P \kappa_i(x) \phi_i(\xi) \nabla \sum_{j=0}^P u_j(x) \phi_j(\xi) \right] = \sum_{i=0}^P f_i(x) \phi_i(\xi) \quad (2.16)$$

After some algebra:

$$\sum_{i=0}^P \sum_{j=0}^P [\kappa_i(x) \nabla^2 u_j(x) + \kappa_i(x) \nabla u_j(x)] \phi_i \phi_j = \sum_{i=0}^P f_i(x) \phi_i \quad (2.17)$$

The choice of ξ and $\phi(\xi)$ define the weighting function to be used. Using the concept of the inner product, a Galerkin projection of Eq. 2.17 onto each basis polynomial $\{\phi_i\}$ is then conducted. The projection ensures that the error is orthogonal to the functional space spanned by the finite dimensional basis $\{\phi_i\}$. Based on the orthogonality of $\{\phi_i\}$, the following expression can be obtained:

$$\sum_{i=0}^P \sum_{j=0}^P [\kappa_i(x) \nabla^2 u_j(x) + \kappa_i(x) \nabla u_j(x)] e_{ijk} = \sum_{i=0}^P f_k(x) \langle \phi_k^2 \rangle \quad (2.18)$$

where $e_{ijk} = \langle \phi_i \phi_j \phi_k \rangle$. Based on the orthogonality of the basis function some of these products will be vanish, and then the original stochastic partial differential equation is reduced to a system of coupled deterministic differential equations with the coefficients obtained from the truncated gPC expansion. The central differencing method is used to solve the deterministic system. Once the coefficients of the expansion are obtained, it is possible to compute statistics for the solved output with the following formulae:

$$\begin{aligned} \mathbb{E}(u) &= E \left[\sum_{i=0}^P u_i \phi_i \right] = u_0 E[\phi_0] + \sum_{i=1}^P E[\phi_k] = u_0 \\ \text{Var}(u) &= E \left[(u - E(u))^2 \right] = E \left[\left(\sum_{i=0}^P u_i \phi_i - u_0 \right)^2 \right] \end{aligned} \quad (2.19)$$

$$= E \left[\left(\sum_{i=1}^P u_i \phi_i \right)^2 \right] = \sum_{i=1}^P u_i^2 E(\phi_i^2) \quad (2.20)$$

Also, the PDF of u can be efficiently calculated by sampling from the distribution of ξ and substituting the corresponding sampled values into Eq. 2.13. It should be noted that Taylor approximations are needed for using Galerkin projection for nonlinear terms that are not of polynomial form. The polynomial chaos quadrature (PCQ) can be used to overcome this challenge when using a non-intrusive PCE method (Xiu D. , 2009). In appendix B, PCQ is used to replace the exact integration in Eq. 2.17 with respect to ξ and is applied to the estimation of reactivity ratios in copolymerization.

2.2 Fault Detection and Diagnosis

Distributed control systems have brought great benefits to the modern engineering systems, such as chemical and petrochemical industries. However, abnormal events usually occur in practice affecting their performance and resulting in economic losses (Isermann, 2005). To detect faults and improve the reliability and efficiency of supervision, fault detection and diagnosis (FDD) become essential activity.

FDD activities involve the timely detection of abnormal events, correct diagnosis of their causal origins, efficient isolation of a fault and appropriate actions to bring the process back to its normal operating state. Generally, FDD methods can be categorized into three classes: model based analytical methods, data driven based empirical methods and hybrid approaches (Frank, 1990). All of the available methodologies involve a series of steps: (1) information transformation; (2) symptoms extraction and (3) classification, and (4) cause-effect mapping according to the obtainable measurements or constructible reference indicator (signal) (Venkatasubramanian, Rengaswamy, & Yin, 2003). A general schematic depiction of FDD is given in Figure 2.1 (Gerlter, 1998).

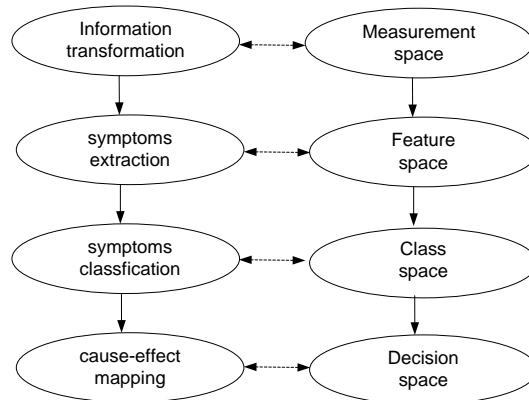


Figure 2.1 General scheme of fault detection and diagnosis

2.2.1 Model based Analytical Methods

Different mathematical models have been proposed for use in the framework of FDD. A straightforward approach to detect a potential fault in a process is to compare the process behavior with a mathematical model describing the nominal process performance, i.e., without the faults. The inconsistencies between the measurements and the ideal model predictions are employed as an indicator to describe the discrepancies between the actual behavior and the normal operation state predicted by the model (Isermann, 2005). When a fault occurs, a nonzero indicator should be obtained to reveal the relation between the observed variables and the model based predictions.

The advantage of model based FDD method is that the effects of faults and other inputs, such as disturbances and noise, can be mathematically modeled as either additive or multiplicative contributions according to the physical understanding of the process (Frank, 1990; Isermann, 2005). Therefore, the discrepancy between the nominal model and the true system can be clearly illustrated by a mathematical expression, and then the fault can be further classified easily. According to the types of measured input signals and output signals, there are three kinds of model based FDD methods: parameter estimation, state/output observer and a parity space based approach (Frank, 1990).

The parameter estimation method is based on the premise that the fault in the process can change a model parameter significantly. Thus changes in model parameters, as obtained from regression of the model with data, can be used to infer faults (Isermann, 2005). The presence of the fault can be inferred from the discrepancies between the nominal model parameter values and the estimated parameter where the nominal model parameters are associated with normal (fault free) operating conditions. Computing the differences (Eq. 2.21) between the nominal values and the estimated parameters is a straightforward way to identify the occurrence of a fault:

$$\Delta p = p - \hat{p} \quad (2.21)$$

where p and \hat{p} are the nominal value and the estimation of the physical parameter respectively. Normally, due to the disturbance/noise as well as uncertainty of modeling, the difference Δp is not identical to 0 even if there is no fault. Therefore, a threshold must be set up to indicate whether a fault has occurred or not. If the value of indicator Δp is greater than the threshold a fault is identified.

An alternative method is to use either state observers or output observers. This kind of methodology is referred to as the observer based method (Isermann R. , 2005; Venkatasubramanian, Rengaswamy, & Yin, 2003). A state observer can be applied if the faults can be modeled as a state variable, and the output observer is used if the state observer is not feasible, e.g. because of lack of observability. The observer based method is especially appropriate if the fault occurs in sensors and actuators because the latter are not part of the state space model used for state estimation. Similar to the parameter estimation approach, a relatively precise mathematical model for the plant is required. An indicator is also necessary, which is defined as the residual between the estimated state and the measured state, or the nominal output and the measurement of output from the process when the state observer is

not available. Although generally linear observers have been used, nonlinear state estimators have been also reported. For nonlinear systems, the extended Kalman filter (EKF) has also been used (Chetouania, Mouhaba, Cosmaoa, & Estela, 2002). However, the EKF can result in a suboptimal solution, since it is based on linearization of the nonlinear equations at each time interval. A class of estimators that do not require explicit linearization has been investigated recently involving particle filtering (Rawlings & Bakshi, 2006). However, this kind of approach belongs to the Markov Chain Monte Carlo based methodology, and its computational cost is very large.

In addition to the employment of observers for identifying potential faults, another promising approach is fault identification by input-output models (Isermann, 2005). Parity space based residual analysis belongs to this group. This method is based on comparing predictions from a fixed model G_m to the measured outputs from process G_p , thereby forming a residual vector with respect to the selected input u and output y :

$$r(s) = G_{My}(s)y(s) - G_{Mu}(s)u(s) \quad (2.22)$$

where $r(s)$ is the residual vector and G_{My} and G_{Mu} are transfer functions. Ideally, for a model structure error and noise free system, the residual is 0 in the absence of faults. If the fault, model structure error and noise can be mathematically modeled, the parity space based method is capable of decoupling fault from model structure error and noise. Therefore, the parity space based method exhibits certain robustness with respect to model structure error and noise.

2.2.2 Data Driven based Empirical Methods

Empirical methods are mainly based on univariate and multivariate statistical algorithms to identify the occurrence of fault (Negiz & Cinar, 1997). They are useful in real process operations since accurate mathematical mechanistic (first-principles based) models are difficult to obtain due to lack of knowledge about the process. Considering that the systems are influenced by random inputs (disturbance or noise), it is reasonable to represent the measurements as statistical time series that can be analyzed in a probabilistic framework (Venkatasubramanian & Kavuri, 2003). When the process is fault free, the observations can be represented by a probability distribution that is assigned to the normal operation. If the process works under faulty condition, the underlying distributions will deviate from the normal distribution thus revealing that the process is out of control. Accordingly, the fault is identified by detecting changes in the probability distribution of the collected data.

For the data driven method, measurements are sampled sequentially and decisions are made based on the observations up to the current time. The easiest way to make a decision regarding the occurrence or absence of a fault is to compare the values of the observations with predefined control limits. If the value is beyond the limits (or ranges) this can be interpreted as the occurrence of a fault. Obviously, an effective algorithm should be sensitive to the faults and robust to the random noise and model structure error. However, the sensitivity to process noise usually increases along with the sensitivity to actual input changes, which means that often false alarm rates tend to increase while detection ability increases.

The Shewhart control chart and the cumulative sums chart were the earliest algorithms proposed for online monitoring and fault detection. They are based on the assumption that a process subject to its natural variability will remain in a relatively steady state of statistical control where certain process and monitored variables remain close to the desired values. Therefore, abnormal events or faults can be identified as soon as they occur by monitoring deviations from the steady state of statistical control. On the other hand, since most of the chemical and petrochemical processes are characterized by strong interaction, the monitored variables are generally not independent, which limits the effectiveness of univariate control charts. Instead, multivariate statistical techniques have been proposed as a way of providing a better solution (MacGregor & Kourti, 1995).

Most of the available multivariate analysis based algorithms are based on the idea of Principal component analysis (PCA). PCA not only transforms a number of related process variables into a smaller set of uncorrelated variables, but it can also be used for control-detection in the presence of interactions among variables. Similar to PCA, partial least squares (PLS) conceptually is another kind of dimension reduction method, which is employed to reduce the dimensions of both process variables and product quality variable to make the analysis simpler. There are different versions of PCA/PLS algorithms reported in literature (Venkatasubramanian & Kavuri, 2003).

PCA is based on an orthogonal decomposition of the covariance matrix for the underlying process variables along their directions that could explain the maximum variability in the obtained data. Therefore, the advantage of using PCA is its ability to represent the original variables in a relatively lower dimension where the information can be properly explained and the major trends in the original data set can be identified. A major limitation of PCA based monitoring methods reported in the literature is that the time invariant PCA models have been used whereas most practical processes are time varying. To address this, some studies developed algorithms to update the PCA model recursively. A general scheme for recursive PCA update should include: mean, covariance, principal components including number of components to be retained, and the confidence limits for T^2 (scaled squared scores) and Q (residual) statistics. An algorithm involving recursive PCA (Li, Yue, Valle-Cervantes, & Qin, 2000) has been used for adaptive monitoring of a rapid thermal annealing process. A similar recursive PLS algorithm was employed to monitor a complex industrial process (Wang, Kruger, & Lennox, 2003).

Another variant of the PCA method is the multi-resolution or multi-scale PCA. In the latter approach wavelet analysis was combined with PCA method and has been proposed to deal with both cross-correlated and auto-correlated variables (Bakshi, 1998) as well as with robustness problems (Chen, Bandoni, & Romagnoli, 1996; Wang & Romagnoli, 2005). The combination algorithm of PCA and wavelet analysis can provide multi-resolution and multi-scale capabilities for fault detection. In particular it can reveal frequency information about the fault.

To overcome the nonlinear behavior that is typical in most chemical processes, different algorithms have been developed. A neural network based PCA model was proposed where an internal layer referred to as the bottleneck was used to reduce the model dimension (Kramer, 1991). A multi-scale nonlinear PCA was proposed using wavelet analysis (Maulud, Wang, & Romagnoli, 2006). Alternatively, a Kernel PCA method has also been

proposed as a relatively simple alternative to neural network based approaches since it requires straightforward solution of an eigenvalue problem (Lee, Yoo, Choi, Vanrolleghem, & Lee, 2004).

Compared with FDD schemes that are based on mechanistic models, multivariate statistical methods do not require an explicit mechanistic model and can handle high dimensional and correlated processes. However, they fail in predicting faults for data that is significantly different from the ones used for model calibration. Thus, hybrid methods that combine mechanistic models and multivariate statistical models were proposed to overcome this shortcoming (Gertler & Cao, 2004; Mylaraswamy & Venkatasubramanian, 1997).

2.2.3 Hybrid Algorithms

To determine the effectiveness of the available fault detection algorithm, four issues have to be addressed: (1) whether the fault is observable; (2) can the fault be distinguished from another unknown fault; (3) can the fault be detected in the presence of process and measurement noise; and (4) can the fault be distinguished from other known faults. All these questions are related to the subject of observability of a fault from available measurements or mathematical model. Since no single method is accurate enough to deal with all the requirements for a fault diagnostic system, hybrid approaches that combine mechanistic models and data driven empirical models become more attractive (Gertler & Cao, 2004). A successful implementation of such a hybrid framework has been conducted for the Amoco model IV fluid catalytic cracking unit. It was adopted by Honeywell for the development of an intelligent control system (Mylaraswamy & Venkatasubramanian, 1997).

2.2.4 Interaction between Control and Fault Diagnosis

Most of the FDD systems are implemented at the supervisory level on top of the available control system. As mentioned above, fault detection methods are based on measurements and some of these measurements are used for feedback in control loops. Thus, variations in the tuning of control loops may affect the closed loop dynamics of the controlled variables and subsequently may affect the performance of the fault detection algorithms. For example, detuning of a controller may be required to increase the variability in a controlled variable so as to improve the observability of fault. However in such case the performance of the control unit would deteriorate. Hence, there is a tradeoff between fast fault detection and acceptable performance of the control unit. A control system that is tolerant to faults is referred to as a fault tolerant control system (FTC). More precisely, FTCs are closed loop control systems that can tolerate malfunctions of the system while maintaining desirable performance (Isermann, 2006).

Although the fault tolerant control problem has been extensively studied, most of the work on FTC was carried out on either one of the two components of the systems, i.e. the FDD component and the control strategy. The issue of interaction between control and diagnostic together has not been addressed as much. Hence, most available FDD algorithms that are operated together with a controller have not been designed to achieve an

optimal trade-off between control and FDD performance. Thus, it is important to integrate FDD and control to develop flexible algorithms that satisfy both objectives (Blanke, Kinnaert, & Lunze, 2006).

Generally, interactive FTC approaches can be categorized into two classes, i.e., passive FTC and active FTC (Zhang & Jiang, 2008). For the passive FTC strategy, the controllers are fixed and predesigned to be robust against a class of predefined faults (Eterno, et al., 1985). In contrast, active FTC system can react to the potential faults by reconfiguring the control strategy to preserve stability and system properties. Thus, in active FTC, the controller has to compensate for the impacts of the possible faults either by selecting a pre-assumed control algorithm or by synthesizing a new one online (Patton, 1997). These two approaches rely highly on the real time FDD algorithm to provide timely information about the status of the system. Thus, the goal of a FTC system is to design controllers with flexible structures while maintaining stability and improving the performance, not only when all control components are performing normally, but also when faults occur.

The active FTC can be divided into four units: (1) a re-configurable controller; (2) a FDD algorithm; (3) a controller reconfiguration mechanism; and (4) a flexible reference governor (Zhang & Jiang, 2008). The issues are how to: (1) design controllers that can be reconfigured; (2) develop FDD schemes that are sensitive to faults while robust to model uncertainties, disturbances as well as noise; and (3) manipulate controllers in the event of faults to achieve desirable performance of monitored parameters. A four parameter controller setup that is a generalization of the two degrees of freedom controller was proposed to address the interaction between fault detection and control (Jacobson & Nett, 1991). The four degrees of freedom controller was reformulated into a general framework, where tools from optimal and robust control were applied (Tyler & Morari, 1994). Based on a standard fault diagnostic algorithm, simultaneous design of a controller and multivariate statistical model based fault diagnosis scheme was proposed and the economic impact of unobservable faults was discussed (Shams, Budman, & Duever, 2011). The influence of control on the fault detection problem was studied from the modeling point of view (Gertler & Cao, 2004), where the set point of the feedback control and/or the ratio coefficient to be used for ratio control was changed to improve the fault identification.

2.2.5 Estimation based on Sequential Monte Carlo Methods

Classification involves estimating unknown quantities from some given observations. When the prior knowledge about the phenomenon being modelled is available, Bayesian models can be formulated with this knowledge. The knowledge includes prior distributions for the unknown quantities and likelihood functions relating these quantities to the observations. Following this, all inference on the unknown quantities is based on the posterior distribution obtained from Bayes' theorem. In terms of implementation, the observations (data) arrive sequentially in time and we are interested in performing inference online. Therefore, it is necessary to update the posterior distribution as new data become available. Computational efficiency is an additional motivation for real-time estimation with new data (Doucet, Freitas, & Gordon, 2001).

When the data can be modelled by a linear Gaussian state space model, it is possible to derive an exact analytical expression to compute the evolving sequence of posterior distributions. This procedure is the well-known Kalman filter (Ristic, Arulampalam, & Gordon, 2004). If the data are modelled as a partially observed finite state-space Markov chain, it is also possible to derive an analytical solution, which is known as the Hidden Markov Model (HMM) filter (Elliott, Aggoun, & Moore, 2008). These two popular filters rely on various assumptions to ensure mathematical tractability. However, observations (data) collected can be very complex. For example, these data typically involve elements of non-Gaussian and nonlinearity, which may preclude analytic solution. Many schemes, such as extended Kalman filter, Gaussian sum approximation and grid-based filter, have been proposed to overcome this challenge. The first two methods cannot take all the salient statistical features into account for the process of interest, which may lead to poor estimation results. The third method, grid-based filter (Ristic, Arulampalam, & Gordon, 2004), using deterministic numerical integration methods, can provide accurate results, but are difficult to implement and computational prohibitive for high dimensional nonlinear problem.

Sequential Monte Carlo (SMC) methods are a set of simulation based methods that can provide a convenient and attractive approach to computing the posterior distributions. SMC methods are flexible and can be easily applied to complicated problem (Doucet, Freitas, & Gordon, 2001). Over the last decades, several related algorithms, such as particle filter and Monte Carlo filter, have been proposed in several research fields. Since their introduction, particle filters have been become a very popular method to solve the solution of optimal estimation problem in nonlinear and non-Gaussian scenarios. In the context of fault detection and diagnosis (FDD), the principle of particle filters is to approximate the conditional state probability distribution that can be used for fault detection by a number of particles. These particles contain samples from the state space and a set of weights that are associated with the particles. Particles can be easily generated and recursively updated using a given process model, which can be further used to describe the evolution in time of the system under analysis. Thus, particle filters algorithm can be used to estimate the probability density function of state, which can be further used to indicate the probability of the occurrence of fault.

2.3 Classification of Cells States

2.3.1 Microscopic Image Acquisition

Microscopy images of cells can be used to discriminate normal, apoptotic and necrotic cells. The morphological difference between apoptosis and necrosis was first observed by electron microscopy (Huerta, Goulet, Huerta-Yopez, & Livingston, 2007). Due to its high resolution, the electron microscopy has the capacity of detecting the specific morphological changes during early and late apoptotic cells. However, this method requires special technical training and it takes much time, which limit its application in practice.

Fluorescence microscopy can improve the observation of apoptotic bodies and also discern necrosis by staining cells with fluorescent dyes. Different fluorescent dyes such as Hoechst stains and Annexin V can be used to label

the cells to visualize nuclear and morphological changes by fluorescence microscopy. This technique has been used to differentiate and quantify apoptotic versus normal cells as well as for determining cell viability (Mercille & Massie, 1994). This method involves two nuclear-fluorescent dyes, acridine orange (AO) in combination with ethidium bromide (EB). These dyes are mixed in a fixed ratio with the cell suspension and can be analyzed by fluorescence microscopy. The microscope has a filter combination suited for detecting fluorescein. The nucleic acid selective cationic fluorescent dye AO can penetrate both viable and nonviable cells, interact with DNA and RNA by intercalation or electrostatic attraction and make the cells appear green. In contrast, EB can only diffuse into nonviable cells and as a predominant dye makes them appear red/orange. Both normal and apoptotic viable cells appear green, whereas non-viable cells appear red/orange.

Flow cytometry is a useful technique that can provide simultaneous multi-parametric analysis of the heterogeneous cell population based on light scattering and emission fluorescence, which allows identifying a homogeneous subpopulation within the total cell population. Using this method, the cells pass through a beam of laser light individually and they are distinguished and quantified according to a set of specific characteristics and phenotype data such as size, granularity or fluorescent molecular binding to the cell. The fluorescent agent may be coupled with a dye or conjugated to a MAb specific for molecules either on the cell surface or within specific intracellular components.

Gel electrophoresis is the other powerful tool to detect DNA laddering that is a hallmark of early event in apoptosis. Moreover, the enzyme-linked immunosorbent assay (ELISA) has also been applied for detection of DNA fragmentation using specific monoclonal antibody (Huerta, Goulet, Huerta-Yeppez, & Livingston, 2007). Elastic Scattering spectroscopy is an optical technique which is based on changes in light scattering properties of cells that are related to morphological changes during the progress of apoptosis (Mulvey, Curtis, Singh, & Bigio, 2007). For most methods, the detection of apoptosis *in vitro* involves fixing and staining the cells to explore morphological or biochemical characteristic of apoptotic cells. The challenge is to segment cells from these images and to develop fast and accurate algorithms to measure morphological features that are representative of apoptosis, which is the focus of this current research.

2.3.2 Image Segmentation Algorithms

A segmentation algorithm for microscopic images subdivides an image into its constituent regions or objects (cellular regions). In terms of application, these algorithms can be roughly classified into three groups, i.e., edge detection, region detection and pattern recognition.

Edge detection is based on gradients between the contour of the cellular region and the background. Filtering techniques such as the Sobel filter can be used to detect the boundary of cells (Gonzalez & Woods, 2009), and have been applied to segment cells from the images of living-cells. The more advanced techniques to detect the boundary of cells are active contour (Kass, Witkin, & Terzopoulos, 1998) and level set method (Sethian, 2002),

for which an energy function is used. This function can be minimized iteratively and can be used to evolve the geometric boundary iteratively until it identifies cells in a given image. The main difference between the active contour and the level set method is in the implementation and the way that the boundary is calculated. The active contour keeps the same number of cells for two consecutive iterations; while the level set method can merge and create cellular regions. Both methods require an initial segmentation guess to start the numerical calculation. It will be shown in Chapter 6 that the level set algorithm can be formulated as a time-varying two dimensional partial differential equation where time refers to iteration time rather than physical time.

The active contour and level set method have been applied to microscopy images. For example, these methods can be used to count and characterize myocytes (Acton, Yang, Hossack, & Wamhoff, 2009), cancer cells (Said, Karam, Berens, Lacroix, & Renault, 2007) and neutrophils (Chen, Chen, & Guan, 2009). One of the main challenges when using the active contour and the level set method is to have an optimal estimation of the initial contour, which can ensure proper convergence to the true boundary. Another issue is the computational time required to evolve the boundary because of the low contrast between the cells and the background. Computational time is critical if the objective is to identify cells in high throughput experiments.

The region detection method consists of splitting and merging regions in an image based on a homogeneity or similarity criterion. For example, the quad-tree method was used to segment histological plant cells, white blood cells and red blood cells in microscopic images (Ko, Seo, & Nam, 2007). For the region growing based segmentation, the first step is to highlight a region (seed region/pixels) manually or by automatic methods (a threshold or other techniques). The seed pixels then can be grown to their neighbor's pixels and those neighbor pixels will be included when they satisfy specific conditions. Generally, potential pixels surrounding the regions are compared to the mean value of intensities (Adams & Bischof, 1994). Nested kernels are another special case of the region detection method, which are based on the computation of a local pixel pattern around the cells centroid. The objective is to provide an indication of where the cells are for tracking purposes. For example, this method was used to segment and track different kinds of cells from microscopy images (Debeir, Van Ham, Kiss, & Decaestecker, 2005).

For the pattern recognition method, a feature described with a quantitative value can be used as a descriptor to classify pixels within an image for segmentation (Gonzalez & Woods, 2009). Both the intensities of pixels and the spatial arrangements of these values can be studied to generate numerical values or features that carry information of interest about the image. The features can be classified as intensity feature, textural/structural features and shape/morphological features (Rodenacker & Bengtsson, 2003). Then, features that are directly extracted from the raw image can be combined with machine learning algorithms to further classify the images.

2.4 Conclusion

On the basis of the above literature review, new methodologies are developed for fault diagnosis and image segmentation using the generalized polynomial chaos (gPC) framework. The main idea of the gPC methodology is to quantify the model uncertainty involved in a stochastic system described by ordinary/partial differential equation with the gPC polynomial expansions. Then, the Galerkin projection is used to propagate the uncertainty into the system and to transform the stochastic system into a new system, which can be expressed by a set of coupled deterministic equations. Since the level set method based image segmentation can be formulated as a partial differential equations (PDEs), the methods developed for fault detection involving the solution of PDEs using gPC and Galerkin projection can be readily applied to the image segmentation problem treated in this work.

Following these ideas, this work focuses on three topics: (1) the development of computationally efficient model-based fault detection and diagnosis (FDD) algorithms in the presence of uncertainties; (2) the formulation of an optimization problem to seek for the optimal trade-off between the detectability of fault and the control performance; (3) the development of accurate and fast quantitative analysis tools for living-cells from stochastic fluorescence microscopy images to evaluate the cells' culture protocols and experimental outcomes.

Chapter 3

Fault Detection and Diagnosis with Parametric Uncertainty

(Adopted from Du et al., 2015, Computer and Chemical Engineering, vol. 76, p. 63~75, 2015)

3.1 Overview

In this chapter, we present a new methodology to identify and diagnose intermittent stochastic faults occurring in a nonlinear process. A Generalized Polynomial Chaos (gPC) expansion representing the stochastic inputs is employed in combination with the nonlinear mechanistic model of the process to calculate the resulting statistical distribution of measured variables that are used for fault detection and classification. A Galerkin projection based stochastic finite difference analysis is utilized to transform the stochastic mechanistic equation into a coupled deterministic system of equations which is solved numerically to obtain the gPC expansion coefficients. To detect and recognize faults, the probability density functions (PDFs) and joint confidence regions (JCRs) of the measured variables to be used for fault detection are obtained by substituting samples from a random space into the gPC expansions. The method is applied to a two dimensional heat transfer problem with faults consisting of stochastic changes combined with step change variations in the thermal diffusivity and in a boundary condition. The proposed methodology is compared with a Monte Carlo (MC) simulations based approach to illustrate its advantages in terms of computational efficiency as well as accuracy.

3.2 Introduction

Distributed control systems have brought great benefits to the chemical and petrochemical industries. However, abnormal events defined as faults usually occur affecting closed loop performance and resulting in economic losses (Isermann R. , 2005). Fault detection and diagnosis (FDD) techniques can be applied to detect the occurrence of faults and improve the reliability and efficiency of supervisory control (Gerlter, 1998; Patton, Frank, & Clark, 2010).

Most of the available FDD algorithms involve comparing the observed behavior of the process to the corresponding model runs obtained by first-principles or empirical models (Venkatasubramanian, Rengaswamy, & Yin, 2003). If the fault is observable, the FDD system will provide fault symptom patterns, which in turn are referred back to the fault diagnosis scheme to identify the root cause of the observed abnormal behavior by a point estimate (Isermann R. , 2006). However, the main restrictive factor of a model-based FDD algorithm is the model uncertainty. The accuracy of FDD schemes can be affected by uncertainty in parameters of the model used for detection. Such uncertainty may originate from either intrinsic time varying phenomena or may result from the use of stochastic noisy data used to calibrate the model. The step of quantifying the uncertainty is typically omitted in reported FDD studies, leading to a loss of information about the influence of the uncertainties on the performance of the FDD by a point symptom pattern comparison. There are few studies that takes model

uncertainties into consideration while developing the FDD algorithm (Li & Yang, 2012; Eriksson, Frisk, & Krysander, 2013; Scott J. , Findeisen, Braatz, & Raimondo, 2013). However, these approaches are based on model linearization and tell nothing about what the probability is that a fault has occurred, due to model uncertainties. Also, since faults occurring in a process may be of a stochastic nature, the use of point estimates for FDD may not be effective.

In terms of applications, FDD schemes based on mechanistic models have been generally applied for processes described by ordinary differential equations (ODEs) (Prashant, Charles, Adiwinata, Panagiotis, & James, 2008; Chilin, Liu, Pena, Christofides, & DavisJames, 2010). Many chemical processes such as heat conduction problems, however, are modeled by partial differential equations (PDEs). The application of FDD methodologies to distributed parameter systems described by PDEs is lacking (Ghantasala & El-Farra, 2009).

The current paper addresses the limitations outlined above by proposing a FDD methodology for systems represented by a first-principles model where both parameters and faults are stochastic and by applying the methodology to a process characterized by PDEs. The faults considered in the present work are stochastic perturbations superimposed on step changes in specific input variables. The key goal of the proposed FDD algorithm is to identify the intermittent step changes in the presence of the random perturbations in the inputs, measurement noise and parametric uncertainty in the models used to detect the fault. Thus, a main feature of the proposed algorithm is quantifying the effect of stochastic changes in inputs and uncertainty in parameters and propagating these variations to the outputs to be measured for use by the FDD algorithm.

One possibility to propagate stochastic variations in inputs onto the outputs is to use Monte Carlo (MC) simulations (Spanos & Zeldin, 1998). However, MC based approaches are very computationally demanding since they require a large number of samples to get accurate results. It should be noticed that although the calculation for calibrating a FDD model are mostly done off-line, approaches such as MC are still computationally prohibitive especially for problems of large dimensions as shown later in the manuscript. Also, improving the computational efficiency of the FDD algorithm may facilitate their future application in the implementation of adaptive algorithms which may require online calculations.

Recently, uncertainty analysis using Generalized Polynomial Chaos (gPC) expansion has been studied by a few authors in different modeling, control and optimization problems (Foo, Yosibash, & Karniadakis, 2007; Nagy & Braatz, 2007; Mandur & Budman, 2014), and has been shown to be more computationally efficient compared to MC. The advantages of gPC are that they can be used to propagate complex probability distributions of input variables onto measured variables (outputs) of interest and the statistical moments of the outputs can be calculated analytically (Ghanem & Spanos, 1991).

The main objective of this paper is to investigate a novel computationally efficient FDD algorithm using probability distributions of measured variables obtained from the propagation of variability in inputs and model parameters by using the gPC expansions. To the best knowledge of the authors, while the gPC has been applied

before for modeling and control applications it has not been used before for FDD as in the current study. The proposed methodology is specifically targeted to detect the average of input faults consisting of stochastic perturbations around mean values that change intermittently. The method does not require any approximations of the model and it explicitly considers its nonlinear nature by directly substituting the gPC expansions of the dependent variables into the nonlinear equations describing the process. The use of the gPC permits the detection of faults of a stochastic nature which are common in the process industries whereas previous fault detection algorithms mostly dealt with deterministic faults, e.g., steps and ramps. It is shown in this work that the proposed gPC based algorithm can efficiently tackle the numerical difficulties involved in the treatment of stochastic faults.

To test the efficiency of the proposed approach, it is illustrated for a two-dimensional heat conduction problem described by a second order partial differential equation (PDE), where stochastic faults related to the changes in the diffusivity and a boundary condition are considered. The stochastic model is calibrated by minimizing the deviation between the first-principle based model prediction and noisy measurements. Different measurement noise levels are studied to verify the effectiveness of the proposed algorithm as well as to provide information for sensor placements.

This paper is organized as follows. Section 3.3 presents the background and the principal methodology. Section 3.4 illustrates the proposed methodology for a two-dimensional steady state heat conduction problem followed by a summary of results in Section 3.5 and conclusions in Section 3.6.

3.3 Background and Methodology

3.3.1 Generalized Polynomial Chaos Expansion

A polynomial chaos expansion represents an arbitrary random variable of interest as a function of another random variable with a prescribed distribution. Define a probability space (Ω, F, π) , where Ω is the sample space, F is the σ -algebra on Ω and π is a probability measure. Let $\{\xi_i(\theta)\}_{i=1}^{\infty}$ be a set of independent random variables from F with probability measure π , then a random process X , defined as a function of a random event θ is expressed as:

$$\begin{aligned} X(\theta) = & c_0 \psi_0 + \sum_{i_1=1}^{\infty} c_{i_1} \psi_1(\xi_{i_1}(\theta)) + \sum_{i_1=1}^{\infty} \sum_{i_2=1}^{i_1} c_{i_1 i_2} \psi_2(\xi_{i_1}(\theta), \xi_{i_2}(\theta)) \\ & + \sum_{i_1=1}^{\infty} \sum_{i_2=1}^{i_1} \sum_{i_3=1}^{i_2} c_{i_1 i_2 i_3} \psi_3(\xi_{i_1}(\theta), \xi_{i_2}(\theta), \xi_{i_3}(\theta)) + \dots \end{aligned} \quad (3.1)$$

where $\psi_n(\xi_{i_1}, \dots, \xi_{i_n}, \dots)$ is the polynomial chaos basis function from the Askey-chaos scheme (Xiu, 2009), n is the n th random variable in a multi-dimensional random variable $\xi = (\xi_{i_1}, \dots, \xi_{i_n}, \dots)$, and $c_{(\cdot)}$ are deterministic gPC expansion coefficients. For notational convenience, Eq. 3.1 is often rewritten as:

$$\mathbf{X}(\theta) = \sum_{j=0}^{\infty} \hat{c}_j \Phi_j(\xi) \quad (3.2)$$

where there is one-to-one correspondence between the functions and their coefficients for the above Eqs. 3.1 and 3.2. For computational efficiency, the gPC expansion is considered in its truncated sum form as:

$$\mathbf{X}(\theta) = \sum_{j=0}^N \hat{c}_j \Phi_j(\xi) = \sum_{j=0}^N \hat{c}_j \Phi_j(\xi_{i_1}, \dots, \xi_{i_n}) \quad (3.3)$$

where i_n is the total number of independent random variables and N is the total number of terms in the expansion determined by the dimensionality of random variables (n) and the highest order (p) of the polynomials $\{\Phi_j\}$ such as:

$$(N+1) = (n+p)!/n!p! \quad (3.4)$$

A key property of a gPC expansion is that all basis functions are orthogonal to each other with respect to the probability distribution of the independent random variable ξ , and accordingly the following applies:

$$\langle \Phi_i, \Phi_j \rangle = \int \Phi_i(\xi) \Phi_j(\xi) W(\xi) d\xi = \langle \Phi_i^2 \rangle \delta_{ij} \quad (3.5)$$

where δ_{ij} is the Kronecker delta, $\langle \cdot, \cdot \rangle$ denotes the inner product of two polynomial functions from the Askey chaos framework with respect to the random variables ξ , and $W(\xi)$ is the weighting function for a particular polynomial. For example, the Hermite polynomials are the basis functions of choice for normally distributed variables and Laguerre polynomials for Gamma random variables (Xiu, 2009). The orthogonality of the basis functions is utilized to compute the j th coefficient in Eq. (3) by a projection calculation as follows:

$$\hat{c}_j = \frac{\langle \mathbf{X}, \Phi_j \rangle}{\langle \Phi_j^2 \rangle} = \frac{\int \mathbf{X} \Phi_j W(\xi) d\xi}{\int \Phi_j^2 W(\xi) d\xi} \quad (3.6)$$

where the integrals in Eq. 3.6 can be calculated by quadrature rules.

3.3.2 Stochastic Finite Difference Method: Response Representation

Assuming a general stochastic elliptic PDE with random inputs is given as:

$$\begin{aligned} \nabla \cdot [\kappa(x; \omega) \nabla u(x; \omega)] &= f(x; \omega) && \text{on } D \times \Omega \\ u_b(x; \omega) &= g(x; \omega) && \text{on } \partial D \times \Omega \end{aligned} \quad (3.7)$$

where D is the spatial domain and Ω is the probability space, f , g and κ are functions on $D \times \Omega$, u is the solution. f is a source term, g is the Dirichlet boundary conditions and κ means a problem specific physical property. All of these operators are subjected to uncertainty ω , which may be introduced into the system via variability in boundary conditions, initial conditions or physical properties. The latter set of variables will be jointly referred heretofore as uncertain input variables.

In order to solve for the random variable u , the gPC expansions are used to represent the uncertain input variables of interest by the following stochastic descriptions:

$$\kappa(x; \omega) = \sum_{i=0}^P \kappa_i(x) \Phi_i(\xi) \quad (3.8)$$

$$u_b(x; \omega) = \sum_{i=0}^P u_{bi}(x) \Phi_i(\xi) \quad (3.9)$$

$$f(x; \omega) = \sum_{i=0}^P f_i(x) \Phi_i(\xi) \quad (3.10)$$

where a truncated finite summation of $\{\Phi_i\}$ with P terms in the finite dimensions of $\xi = \{\xi_1, \dots, \xi_n\}$ is used. The dimensionality n of ξ is determined by the number of random events. The random parameter ω is embedded into the polynomial basis $\{\Phi_i\}$, where ω is represented with a curve spanned by the set ξ , thus the expansions' coefficients in Eqs. 3.8 ~ 3.10 are deterministic. The solution of $\{u_m\}$ for every interior grid point m of the model in Eq. 3.7 is rewritten as:

$$u_m(x; \omega) = \sum_{i=0}^P u_{m,i}(x) \Phi_i(\xi) \quad (3.11)$$

where m is a grid point in the spatial domain and i is the i th gPC coefficients for this particular grid point. In order to achieve exponential convergence in the coefficients of solution $\{u_m\}$, the optimum polynomial is chosen from the Askey-chaos scheme with respect to the PDF of the random input variables, which are assumed to be a priori known and for which the weighting function is selected accordingly (Xiu, 2009). Substituting the gPC expansions given in Eqs. 3.8 ~ 3.11 into Eq. 3.7 yields:

$$\nabla \cdot \left[\sum_{i=0}^P \kappa_i(x) \Phi_i(\xi) \nabla \sum_{j=0}^P u_{m,i}(x) \Phi_j(\xi) \right] = \sum_{i=0}^P f_i(x) \Phi_i(\xi) \quad (3.12)$$

Upon re-arrangement of Eq. 3.12 as follows:

$$\sum_{i=0}^P \sum_{j=0}^P \left[\kappa_i(x) \nabla^2 u_{m,j}(x) + \kappa_i(x) \nabla u_{m,j}(x) \right] \Phi_i \Phi_j = \sum_{i=0}^P f_i(x) \Phi_i \quad (3.13)$$

Employing the inner product, a Galerkin projection of Eq. 3.13 onto each basis polynomial $\{\Phi_j\}$ is then conducted. The projection ensures that the error is orthogonal to the functional space spanned by the finite dimensional basis $\{\Phi_i\}$. Based on the orthogonality of $\{\Phi_j\}$, the following expression can be obtained:

$$\sum_{i=0}^P \sum_{j=0}^P \left[\kappa_i(x) \nabla^2 u_{m,j}(x) + \kappa_i(x) \nabla u_{m,j}(x) \right] e_{ijk} = \sum_{i=0}^P f_k(x) \langle \Phi_k^2 \rangle \quad (3.14)$$

where $e_{ijk} = \langle \Phi_i \Phi_j \Phi_k \rangle$. Thus, the original stochastic PDE is transformed into a system of coupled deterministic equations where the unknowns are the coefficients obtained by a truncated gPC expansion. Once the coefficients of the expansion are calculated from the system of equations represented by Eq. 3.14, it is possible to compute statistical moments for the solution at each grid point with the following formulae:

$$E(u_m) = E \left[\sum_{i=0}^P u_{m,i} \Phi_i \right] = u_{m,0} E[\Phi_0] + \sum_{i=1}^P E[\Phi_k] = u_{m,0} \quad (3.15)$$

$$\begin{aligned} V(u_m) &= E \left[(u_m - E(u_m))^2 \right] = E \left[\left(\sum_{i=0}^P u_{m,i} \Phi_i - u_{m,0} \right)^2 \right] \\ &= E \left[\left(\sum_{i=1}^P u_{m,i} \Phi_i \right)^2 \right] = \sum_{i=1}^P u_{m,i}^2 E(\Phi_i^2) \end{aligned} \quad (3.16)$$

Also, the PDFs of solution $\{u_m\}$ can be rapidly calculated by sampling from the distribution of ξ and substituting the corresponding samples into Eq. 3.11. The ability of explicitly calculating statistical momenta by analytical equations is the basis for the computational efficiency of gPC based approaches.

3.4 Case Study: Two-dimensional Heat Conduction Problem

The proposed methodology is applied to a two-dimensional steady state heat conduction problem with stochastic input parameters, which is described as:

$$\kappa \left(\frac{\partial^2 T(x,y)}{\partial x^2} + \frac{\partial^2 T(x,y)}{\partial y^2} \right) = Q(x,y) \quad (3.17)$$

over a rectangular domain $D = \{(x,y) | x_0 \leq x \leq x_f, y_0 \leq y \leq y_f\}$ with boundary conditions:

$$\begin{aligned} T(x_0, y) &= g_{x0}(y), T(x_f, y) = g_{xf}(y) \\ T(x, y_0) &= g_{y0}(x), T(x, y_f) = g_{yf}(x) \end{aligned}$$

Intermittent step changes on input parameters with superimposed random perturbations as shown in Fig.1 are assumed for both the internal thermal diffusivity κ and the boundary conditions g . The goal is to detect the changes in the mean values of the input variables in the presence of the random perturbations occurring around these changes. A practical application for this problem is the detection of changes in process operating conditions based on the analysis of a two dimensional temperature distribution as obtained with a thermal scanner or an array of thermocouples. For example, two dimensional scanners are extensively used in the process industries to detect hot spots or abnormal changes in heat sources, such as flames (Murat, 2012).

The faults considered in the current work consist of intermittent step changes in the diffusivity and one of the boundary conditions with superimposed stochastic variations as shown in Fig.3.1 (a), while previous fault detection algorithm mostly dealt with faults mostly consisting of steps or ramps as shown in Fig.3.1 (b). To detect faults such as the ones described in Fig.3.1 (a), it is proposed to propagate the stochasticity of the inputs onto the measured output temperatures by the gPC based approach outlined in the previous section. Then, measurements of output variables are used to detect and diagnose the changes in parameter means from the calculated PDFs (JCRs) of the measured outputs.

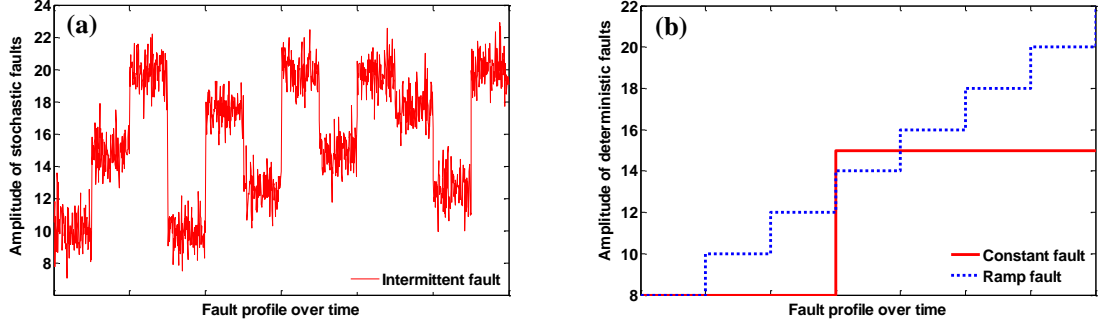


Figure 3.1 Faults profiles

- (a) Fault profile representing intermittent changes in mean values of input with superimposed random perturbations around these means (b) Fault profile representing the step or ramp type faults

In contrast with other reported model based FDD approaches, the proposed method does not rely on a single point estimate. Instead, to identify an abnormal event, probability distributions in the measured variables, i.e., a set of temperatures obtained at certain locations in this case study, are used to recognize the occurrence of a fault. Following the above, the fault detection and fault isolation steps can be performed as follows:

- i- A fault is detected by assessing the probability of a measured temperature to correspond to operation around a particular mean value of the inputs of the form shown in Fig.3.1 (a). This calculation is based on a priori calculated PDF around a specific input mean value.
- ii- To isolate a fault, PDFs (JCRs) are constructed for the measured temperatures corresponding to each of the mean values used for each of the input variables considered for the study. Isolation of a particular fault is conducted by assessing the relative probability of a measured temperature with respect to each one of the calculated PDFs (JCRs).

The key challenge for accomplishing the detection and isolation steps arises from the measurements being corrupted by noise in addition to the stochasticity of the inputs. Therefore, it is necessary to take the input stochasticity and noise into account to develop a robust FDD algorithm. While in the absence of measurement noise the output PDFs can be calculated exactly from a priori knowledge of the stochastic distributions around the means in the inputs, in a real implementation the PDFs of measured variables have to be calibrated from actual noisy process data. By calibration it is meant that in the presence of noise the means and variances of the input PDFs that explain the measured output PDFs will be different from the ones that would lead to the output PDFs if noise would not be present. Thus, the means and variances of the input PDFs have to be changed with respect to their actual values to minimize the discrepancy between the predicted and measured outputs.

Two steps are involved in the calibration of the FDD algorithm in the presence of measurement noise and stochasticity: (1) uncertainty propagation and model formulation using the Galerkin projection method described above; (2) model calibration and optimization by minimizing the sum of squared errors between model predictions

and noisy measurements. After the model is properly calibrated with noisy data, the resulting model can be tested for detection and isolation with data that were not used for model calibration. The calibration method will be further detailed in section 3.4.2 below.

3.4.1 Uncertainty Propagation and Model Calibration

3.4.1.1 Individual Fault Case (Case Study I)

The first case study assumes that the diffusivity is defined as a stochastic variable whereas the boundary conditions and external source are assumed to be known constants. The resulting stochastic PDE is used to solve for the temperature distribution over a square plate. The measured temperatures to be used for isolation and detection are assumed to be corrupted by Gaussian noise. The problem is solved by assuming a gPC representation of the diffusivity and the solution as follows:

$$\kappa(x,y;\omega) = \sum_{i=0}^P \kappa_i(x,y) \Phi_i(\zeta) \quad (3.18)$$

$$T(x,y;\omega) = \sum_{i=0}^P T_i(x,y) \Phi_i(\zeta) \quad (3.19)$$

where ω denotes a random event causing a random normally distributed change in diffusivity and the Hermite polynomials are chosen as the basis functions. For example, the diffusivity has a mean of $\bar{\kappa}$ and unit variance, i.e., $\kappa = \bar{\kappa} + \zeta$. This is a one dimensional random space problem since only one random variable is considered, i.e., $\zeta = \zeta_1$.

To apply the finite differences method, the solution domain on both horizontal and vertical directions is divided into $(N_{xy}+1)$ equal subsections resulting in $(N_{xy} \times N_{xy})$ interior grid points over the square plate. Some of the interior grid points for Eq. 3.17 for finite difference analysis are shown in Fig.3.2. Assuming the lengths of each subsection along the horizontal and vertical directions are equal:

$$\Delta x = \Delta y = (x_f - x_0)/(N_{xy} + 1) = (y_f - y_0)/(N_{xy} + 1) \quad (3.20)$$

The second derivative is approximated by a five-point central difference approximation. For every interior point (x_j, y_i) with $1 \leq m, n \leq N_{xy}$ (as shown in Fig.3.2), the finite difference equation is described as:

$$(\bar{\kappa} + \zeta_1) \left(\frac{T_{m,n+1} - 2T_{m,n} + T_{m,n-1}}{\Delta x^2} + \frac{T_{m+1,n} - 2T_{m,n} + T_{m-1,n}}{\Delta y^2} \right) = Q \quad (3.21)$$

where

$$T_{m,n} = T(x_n, y_m, \omega) = \sum_{i=0}^P T_i(x_n, y_m) \Phi_i(\zeta_1) \quad (3.22)$$

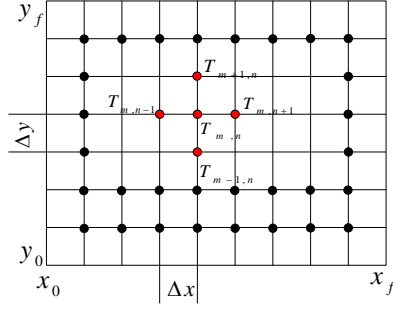


Figure 3.2 The grid points for two-dimensional heat conduction problem

Using Galerkin projection, both sides of Eq. 3.21 are multiplied by polynomials $\{\Phi_j(\zeta_i)\}$ and taking inner product results in a set of equations:

$$\sum_{i=0}^P ((T_i)_{xx} + (T_i)_{yy}) \langle \Phi_j(\bar{\kappa} + \zeta_i) \Phi_i \rangle = Q \langle \Phi_j \rangle \quad (3.23)$$

Assume Ξ is the central differencing operator matrix and vector T_i is the gPC coefficient of the solution. Substituting Eqs. 3.18 and 3.19 into Eq. 3.23 gives:

$$\sum_{i=0}^P \langle \Phi_j(\zeta_i) (\bar{\kappa} + \zeta_i) \Phi_i(\zeta_i) \rangle \Xi T_i = Q \langle \Phi_j \rangle e \quad (3.24)$$

where e is a vector with all elements equal to 1. For the coupled deterministic Eq. 3.24, all the gPC expansion coefficients are solved by quadrature rules with Eq. 3.6. The PDFs of the temperature $\{T_{(m,n)}\}$ at a particular grid point can be approximated by substituting random samples of ζ_i into the solution expansion given by Eq. 3.22.

3.4.1.2 Simultaneous Two Faults Case (Case Study II)

Simultaneous random intermittent changes in the diffusivity and boundary condition are considered as described by Fig.3.1 (a). The external heat source is assumed to be a known constant. Random changes in temperature at only one boundary ($T(x_f, y) = g_{xy}(y)$) are studied for simplicity, while at the other boundaries the temperatures are assumed to be constant. It is also assumed that the stochastic perturbations around the different means of the diffusivity and the boundary condition temperature values are independent stochastic events. Thus, a two dimensional random space is considered, i.e., $\xi = \{\zeta_1, \zeta_2\}$, where ζ_1 and ζ_2 denote the random events in κ and $g_{xy}(y)$, respectively. It is assumed that κ follows the same distribution as in case study I, and the boundary is a random variable with a mean of \bar{g} and unit variance, i.e., $\kappa = \bar{\kappa} + \zeta_1$ and $g = \bar{g} + \zeta_2$. For this case, the solution at particular grid can be described by a gPC expansion as:

$$T_{m,n} = T(x_n, y_m, \omega) = \sum_{i=0}^P \sum_{j=0}^P T_i(x_n, y_m) \Phi_i(\zeta_1, \zeta_2) \quad (3.25)$$

where ω represents the random events resulting in intermittent changes in both diffusivity and boundary condition. Substituting these inputs' gPCs into Eq. 3.17 results in:

$$\begin{aligned}
(\bar{\kappa} + \zeta_1) \sum_{i=0}^P ((T_i)_{xx} + (T_i)_{yy}) \Phi_i(\zeta_1, \zeta_2) &= Q & (3.26) \\
T(x_0, y) &= C, \quad T(x_f, y) = \bar{g} + \zeta_2 \\
T(x, y_0) &= C, \quad T(x, y_f) = C
\end{aligned}$$

where C denotes known deterministic constant boundary conditions. After approximating the second derivative by a five point central difference approximation and substituting this approximation into Eq. 3.25 into Eq. 3.26, and using Galerkin projection with respect to $\langle \cdot, \Phi_k \rangle$ on both sides of Eq. 3.26 yields:

$$\sum_{k=0}^P \sum_{i=0}^P \sum_{j=0}^P (\bar{\kappa} + \zeta_1) ((T_i)_{xx} + (T_i)_{yy}) \langle \Phi_i \Phi_j \Phi_k \rangle = Q \langle \Phi_k^2 \rangle \quad (3.27)$$

Eq. 3.27 represents a system of coupled deterministic linear equations with respect to the gPC coefficients that can be solved numerically.

3.4.2 Model Calibration and Optimization

Model calibration is done by using noisy output measurements to adjust the expectation and variance of the input PDFs. It should be noticed that the resulting calibrated input PDFs may result in input PDFs that are different from their actual values due to the noise in the measurement used for calibration. It is assumed that a limited set of output measurements is available for model calibration. Then a cost function is defined as follows:

$$\min_{\lambda} J = \sum_{i=1}^n \mu_{1,i} (\gamma_{1,i} - v_{1,i})^2 + \sum_{i=1}^n \mu_{2,i} (\gamma_{2,i} - v_{2,i})^2 \quad (3.28)$$

where $\gamma_{1,i}$ and $\gamma_{2,i}$ are the mean and variance of measured variables calculated from the gPC model by Eqs. 3.15 and 3.16, $v_{1,i}$ and $v_{2,i}$ are the mean and variance approximated by the sampled measured outputs, λ is a vector of expectations and variances for the inputs, i.e., the mean and the variance of diffusivity in Case study I or the mean and the variance of diffusivity and boundary condition for two simultaneous faults in Case study II. The variable n is the number of sensors located within the discretized square domain. The weights $\mu_{1,i}$ and $\mu_{2,i}$ in the cost function Eq. 3.28 are used to penalize the mean relative to the variance. The effect of the choice of these weights is further discussed in the case study. A detailed flowchart summarizing the steps to solve Eq. 3.28 is given in Fig.3.3. To generate the training set of measurements, Gaussian noise is added to the simulated data obtained by the PDE based deterministic model.

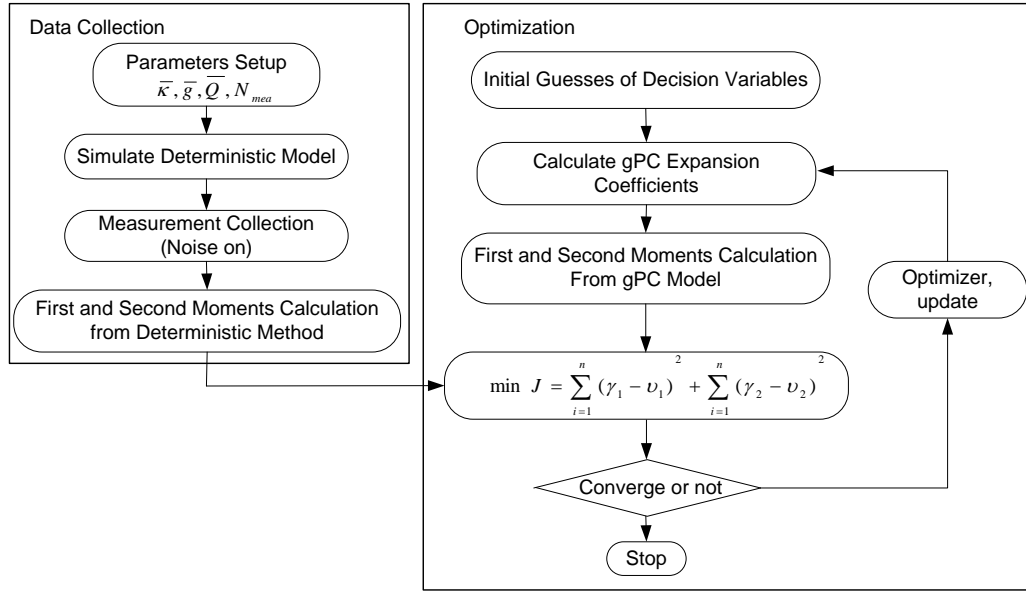


Figure 3.3 Flowchart to formulate the adaptive optimization model

3.4.3 FDD Algorithms

Following model calibration as per the procedure explained above, the PDFs for the measured variables can be constructed for each one of the mean values considered in the diffusivity in Case study I by solving the problem in Eq. 3.17. The PDFs of the measured variables (temperature at different grid points) are obtained by sampling from the distribution of the random event and substituting the samples into Eq. 3.19. Then, the mean values of inputs, i.e., diffusivity, can be inferred by assessing the probability of a measured temperature with respect to each one of the calculated PDFs as shown in Fig.3.4, where each of the different mean values of the input defines a class.

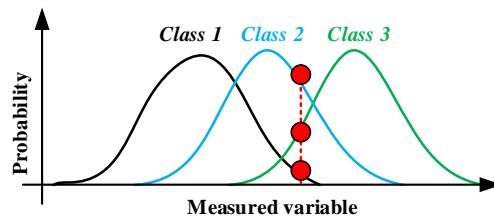


Figure 3.4 FDD algorithm by using the PDF profiles of measured variables

As seen in Fig.3.4, if one of the PDFs corresponding to a specific mean value is considered as the normal operating regime, this procedure allows to both detecting abnormal operation as well as isolating the particular input mean value for any given measured output. Since a particular value of temperature can be found within different PDFs with different probabilities due to the measurement noise and stochasticity on inputs, the value of

diffusivity is inferred as the class for which the largest probability of the measurement is obtained. For example, if three probability values are obtained for a given measurement as shown in Fig.3.4, the maximum probability value is used to determine that the system is operated around the mean value corresponding to ‘Class 2’.

In the case of simultaneous faults entering the system, a joint confidence region (JCR) of more than one sensor is used to infer the input changes. The JCRs are generated as per the following steps: i- a range of maximum temperatures’ variations is estimated for each one of the two sensors used for inferring the faults. ii- a discrete grid made of combinations of temperature values at the two sensor locations is generated based on the temperature variations estimated in (i) as shown in Fig.3.5. iii- the random variables ζ_1 and ζ_2 corresponding to the faults in diffusivity and boundary conditions, are sampled from their respective distributions and substituted into the temperature gPC expansion (Eq. 3.25) thus generating corresponding temperature values. iv- each calculated temperature value in item (iii) is assigned to the closest grid point in Fig.3.5, and after all the temperature values are assigned, the number of samples for each grid point is calculated. v- the probability at each grid point is calculated by dividing the number of samples at the grid point by the total number of samples (total number of combinations of ζ_1 and ζ_2 substituted into the gPC model), and vi- the JCR is formed by connecting the grid points corresponding to a particular probability value (e.g. 90%).

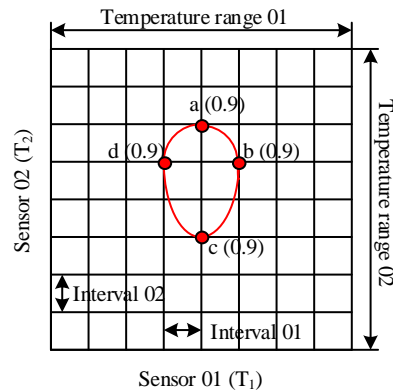


Figure 3.5 Joint Confidence Region (JCR) array

Two approaches are developed for FDD in the case of simultaneous multiple faults: (i) a JCR profiles based method where the JCRs are approximated with the gPC approach and (ii) a gPC model based minimum distance method. In Fig.3.6, the projection of two JCR profiles onto a two dimensional plane are shown, where each axis represents a measured variable (temperature in our case) for a particular pair of sensors.

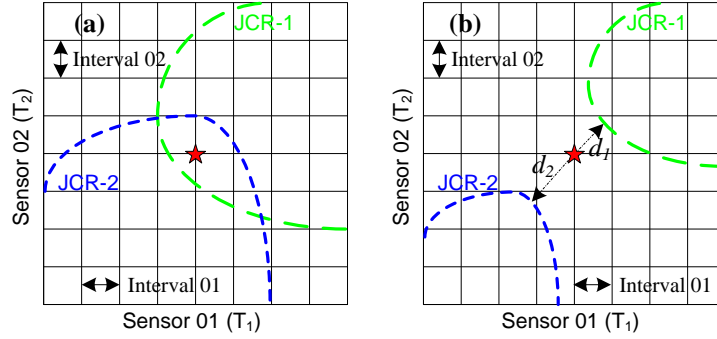


Figure 3.6 Sketch of JCR based FDD algorithm

A JCR only predicts that a set of measurements (indicated by a red star in Fig.3.6) has a particular probability of being within a particular class. If a set of measurement is found within a JCR or the overlapping regions between two JCRs (as shown in Fig.3.6 (a)), the maximum probability can be used to infer the class, i.e., means of diffusivity and boundary condition. However, due to noise, a particular set of measurements may lay outside of the JCRs for a given confidence interval (as shown in Fig.3.6 (b)) especially if there is small or no overlap between the JCRs. Therefore, a second gPC model based minimum distance approach is used for this latter case. By using a three-sigma confidence level, the bounds of random variables (ξ) corresponding to a particular confidence level are approximated first. The analytical gPC expression for each combination of diffusivity and boundary condition is used to calculate the distance between new points (measurements) and JCRs with a prescribed confidence region. The shortest distance between the measurements and the contour of a particular JCR indicates that operation within the corresponding class is the most probable. For example, in Fig.3.6 (b) ‘Class 1’ is indicated that corresponds to ‘JCR-1’ since the distance d_1 is smaller than d_2 . The Euclidean distance is used to find the closest class as follows:

$$\min_{\lambda_{\xi}} J = \sum_{i=1}^n (M_1 - gPC_{1,i})^2 + \sum_{i=1}^n (M_2 - gPC_{2,i})^2 \quad (3.29)$$

where M_1 and M_2 are the two measurement samples used for FDD, the $gPC_{1,i}$ and $gPC_{2,i}$ are the gPCs expressions given by Eq. 3.25 for two different sensor locations for a particular combination i ($i \in (1, 2, \dots, n)$) of diffusivity and boundary condition mean values (classes), n is the number of classes, and λ_{ξ} is the decision variables and is a vector of random samples (ξ) in the sample domain Ω_I that minimizes the distance with respect to each class represented by a corresponding JCR.

3.5 Results and Discussion

3.5.1 Case Study I: Individual Fault Case

Case Study I consists of detecting intermittent changes in the mean of diffusivity from a small set of temperature measurements located within the domain of the heat transfer problem. To illustrate the problem, six mean values of diffusivity κ are considered (10, 12, 14, 16, 18 and 20). Operation around each one of these mean values is referred to as a class. Thus, six classes described by the expectation of diffusivity are formulated. The random variations in diffusivity around these means are assumed to be normally distributed with zero mean and unit variance for simplicity. However, the gPC model can be easily extended to distributions other than normal. The solution domain is defined as $D = \{(x,y) / -1 \leq x \leq 1, -1 \leq y \leq 1\}$.

The ability of the proposed method to detect changes in means is evaluated with respect to different locations of the temperature sensors within the domain. For design simplicity, it is desired to keep the required number of sensors small. Fig.3.7 shows the expectation and variance contour lines calculated by the gPCs' coefficients described in Eqs. 3.15 and 3.16, where the highest order of one dimensional gPC polynomial is 3. Since the gPC coefficients over the domain are symmetrical, the domain is divided horizontally and vertically into four symmetrical regions. For simplicity, one quarter of the square domain (top-left) is studied with respect to the number and locations of the sensors to be used for FDD (as shown in Fig.3.8).

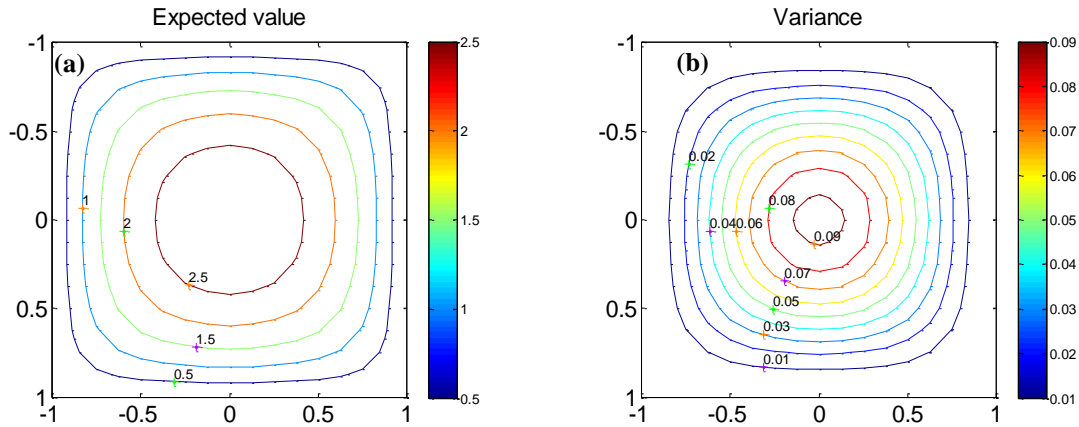


Figure 3.7 Mean and variance distribution over two-dimensional domain

(a) Expected value and (b) variance of the solution ($\bar{\kappa} = 10$, $Q = -100$, $N_{xy} = 15$)

Additionally, an assumption is made that the location of each sensor coincides with the location of an interior grid point used for finite differences analysis. Accordingly, the total number of potential locations of the sensors is a function of the number of interior grid points. If the number of interior grid point is $(N_{xy} \times N_{xy})$, for example, there are $\frac{1}{4}N_{xy}^2$ potential locations in one quarter of the domain. For simplicity, the sensors are only placed on the diagonal direction, from the top-left to the bottom-right. There are 8 potential sensors' locations along the diagonal

direction for one quarter of the domain, if the N_{xy} is 15. As shown in Fig.3.8, the sensors along the diagonal direction are numbered from 1 to 8, starting from the top-left to the bottom-right.

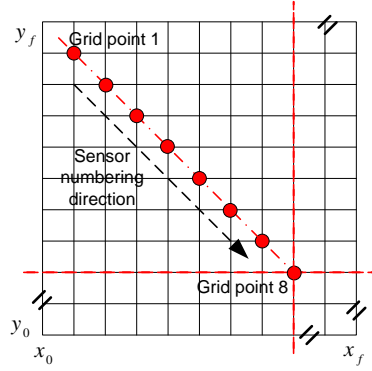


Figure 3.8 Sensors placement for model optimization (top-left part of the square domain)

To demonstrate the influence of the number and the placements of sensor on the model calibration as defined in Eq. 3.28, a criterion is defined as:

$$c_e = (\bar{\kappa}_{simu} - \bar{\kappa}_{opt})^2 \quad (3.30)$$

where $\bar{\kappa}_{simu}$ is the known expected value of diffusivity, $\bar{\kappa}_{opt}$ is the optimum estimate obtained for different sensors' numbers and locations by solving the optimization problem given by Eq. 3.28, and c_e is the discrepancy between the estimation and the known mean. If c_e is bigger than a prescribed threshold value, the corresponding sensor location is ruled out. In addition, a percentage for the acceptable estimates defined in Eq. 3.31 is used to evaluate the model calibration results, where N_{trial} is the total number of trials and n_{trial} is the number of desirable estimates satisfying Eq. 3.30.

$$A_{rate} = n_{trial}/N_{trial} \quad (3.31)$$

Based on the 8 sensors shown in Fig.3.8, six different sensor placement structures are presented to assess the influence of sensors' number as well as location on the model calibration. Table 1 shows the results of six sensor placement strategies for three different measurement noise levels.

Table 3.1 Comparison of acceptance rate for six sensor placement structures

Sensor structure	Noise level (0.1)	Noise Level (0.2)	Noise Level (0.3)
1 (grid point 1)	0.10	0.03	0.03
2 (grid point 8)	0.9333	0.7667	0.6333
3 (grid points 1~4)	0.9667	0.7333	0.5500
4 (grid points 3~6)	0.9833	0.7833	0.7667
5 (grid points 5~8)	1	0.8167	0.7833
6 (grid points 1~8)	1	0.8667	0.8500

In Table 3.1, the threshold used to evaluate Eq. 3.30 is chosen as 1 and 60 trials are studied for each combination of sensor placement strategy and different noise levels. The measurement noise is assumed to be normally distributed with zero mean and variances of 0.1, 0.2 and 0.3, respectively. As seen from Table 3.1, if only one

sensor (grid point 1 in Fig.3.8) is located close to the boundary of domain, A_{rate} is lower than 10% for the first noise level. By contrast, A_{rate} increases to 93.33% for one sensor situated at the center of the domain (grid point 8 in Fig.3.8), compared with grid point 1.

The effect of the weights $\{\mu_{1,i}\}$ and $\{\mu_{2,i}\}$ on the cost function Eq. 3.28 were investigated using sensor structure 6 defined in Table 3.1. Two sets of weights were studied, i.e., (i) equal weights ($\mu_{1,i} = \mu_{2,i} = 1$) and (ii) based on the relative magnitudes of the mean and variance (mean~10 and variance~1) unequal weights ($\mu_{1,i} = 0.1, \mu_{2,i} = 1$) were used. For the latter case, Table 3.2 shows that the weights have negligible effects on the model calibration in terms of the predicted mean and variance.

Table 3.2 Summary of model calibration results (noise variance $\sigma^2=0.1$)

$\bar{\kappa}_{simu}$	Equal weights		Unequal weights	
	$\bar{\kappa}_{opt}$	ξ_{opt}	$\bar{\kappa}_{opt}$	ξ_{opt}
10	10.1802	1.0220	9.9651	1.0132
12	11.6243	1.0163	12.0095	1.0125
14	13.6627	1.0115	14.0217	1.0174
16	15.9600	1.0083	16.0846	1.0153
18	18.0425	1.0064	18.0710	1.0326
20	20.0542	1.0051	19.9315	1.0204

Fig.3.9 shows the PDF profiles corresponding to the six different mean values in diffusivity inferred from temperature measurements acquired at grid point 8, where the horizontal axis is the solution of Eq. 3.17, and the vertical axis is the normalized probability. The rest of the aforementioned grid points in Fig.3.8 have similar PDF profiles except that the temperature ranges of the PDFs are different. In Fig.3.9, *Classes* are referred to by the mean value of diffusivity, i.e., ‘Class: 10’ means that the expectation of the diffusivity is 10.

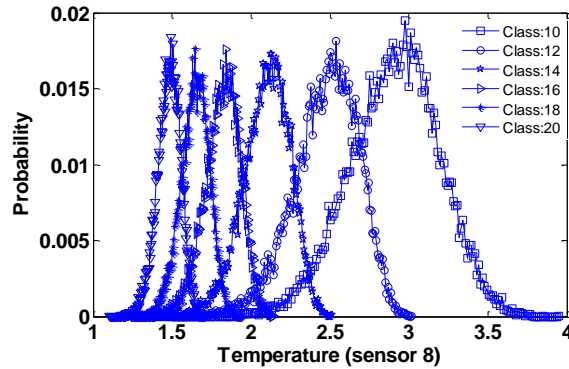


Figure 3.9 PDF profiles of six classes at grid point 8 by gPC model ($Q = -100$)

After model calibration, FDD tests are first conducted by assuming that a single sensor is used to investigate the influence of each individual sensor location on the detectability of the fault, i.e., changes in diffusivity. To demonstrate the influence of measurement noise on the observability of faults, different measurement noise levels are tested. The efficiency of the FDD algorithm is judged based on the fault detection rate defined as follows:

$$d_{rate} = \frac{1}{n} \sum_{i=1}^n d_i / D_i \quad (3.32)$$

where n is the number of classes, i.e., the number of means on diffusivity in Case study I, i is the i th class, d_i is the number of samples that have been classified correctly and D_i is the number of testing samples in i th class.

Fig.3.10 (a) shows the fault detection rate when the variance of the noise is 0.1 and 0.2, respectively. For a particular sensor, 1000 testing samples (D_i in Eq. 3.32) are used for each class resulting in 6000 samples in total, since six classes are investigated in Case study I. As shown, the detection rate, d_{rate} , is higher if the sensor is located at the center of the domain, since the temperature at the center of domain is higher than at the other grid points thus the signal to noise ratio (SNR) is larger. When the variance of the noise is 0.1, the average of fault detection rate for sensor 8 (grid point 8) is ~80%, which is ~60% higher as compared with sensor 1 (grid point 1). When the noise level is increased to 0.2, the fault detection rate at sensor 8 has been decreased to ~55%. As expected, the fault detection rate is higher with smaller measurement noise. To further assess the effect of SNR on the fault detection rate, two different values of heat duty, i.e., Q in Eq. 3.17, are considered: -100 and -1000.

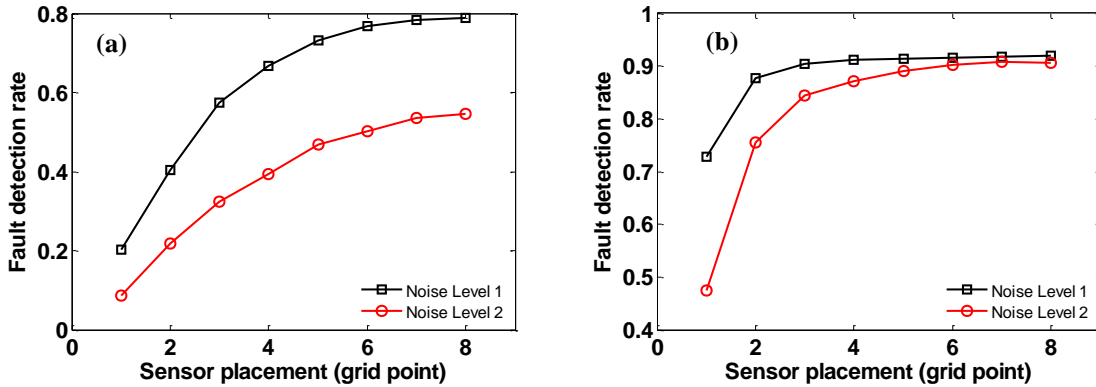


Figure 3.10 Fault detection rate for single fault with gPC model

((a): $Q = -100$, (b): $Q = -1000$)

Fig.3.10 (b) shows the fault detection rates (d_{rate}), when the heat duty Q is -1000. It can be seen by comparing (a) and (b) in Fig.3.10 that the SNR shows strong influence on the observability of the faults. For example, the average detection rate with $Q = -1000$ at sensor 1 is ~73% for the first noise level, while it is ~20% as shown in Fig.3.10 (a) for $Q = -100$. It is also observed in Fig.3.10 (b) that the fault detection rate has been decreased as the noise level increases. For example, for sensor 1, the detection rate decreased by around 24 percent point when the noise level is changed from level 1 to level 2.

To reduce the influence of the SNR on the detectability of the fault, the use of replicates of measurements was investigated. Fig.3.11 gives the results for both aforementioned heat duties where 10 replicates are used. Compared with Fig.3.10, the fault detection rate has been increased since the replicates reduce the influence of noise on the observability of the input fault. For example, the detection rate (d_{rate}) increased by 27 percent point

at sensor 8 for noise level 2 and for $Q = -100$, from 55% to 82%. Additional numerical experiments show that with 50 replicates, the fault detection rate (d_{rate}) at sensor 1 can be increased by as much as 65% for $Q = -100$.

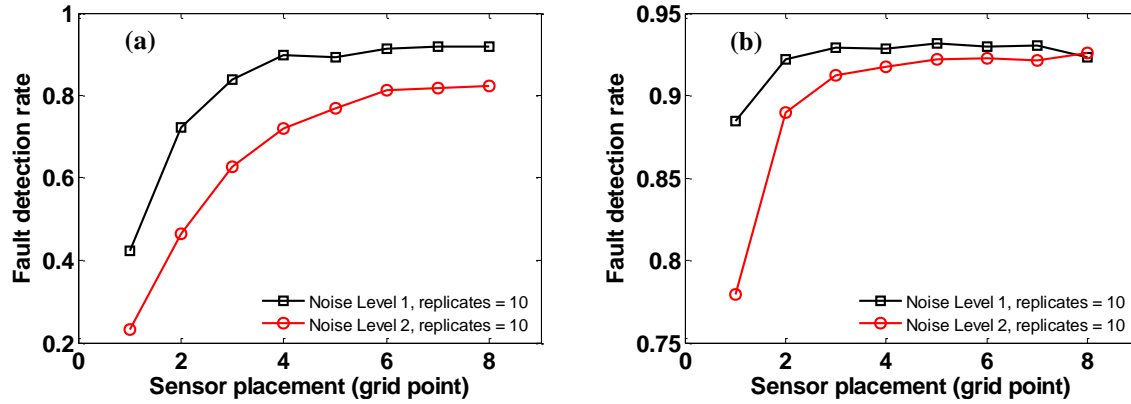


Figure 3.11 Fault detection rate for single fault by gPC model with 10 replicates

((a): $Q = -100$, (b): $Q = -1000$)

Additionally, the detection rate can be further improved by using the 8 sensors together. In this case, the detection is based on the average of the probabilities of a particular measurement calculated with respect to the PDFs calculated for each sensor. For example, for $Q = -100$, with no replicates and noise level 1 the detection rate (d_{rate}) increases to $\sim 92\%$ as compared to a maximum of $\sim 80\%$ when only sensor 8 is used (Fig.3.10 (a)).

Finally, the detection rate (d_{rate}) was investigated with the aforementioned two sets of weights $\{\mu_{1,i}\}$ and $\{\mu_{2,i}\}$ in the cost function Eq. 3.28, i.e., equal weights and unequal weights cases. Fig.3.12 shows the detection rate d_{rate} when the heat duty Q is -1000 and the measurement noise is noise level 2. As seen from Fig.3.12, d_{rate} is almost identical for the two sets of weights. For simplicity, equal weights were adopted for the rest of case studies. In principle, for a different problem, the weights could have a more significant effect on the detection rate but this can be identified from the differences in mean and variance during the model calibration step. In the current case study the differences between the two sets of weights in the calibration step were negligible as shown in Table 3.2.

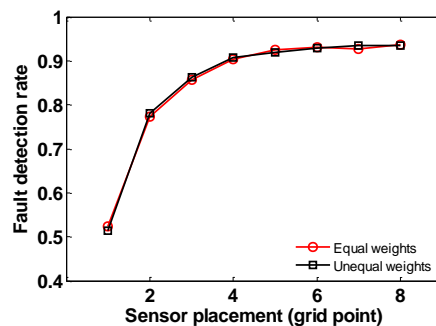


Figure 3.12 Fault detection rate for single fault with different weights

3.5.2 Case Study II: Simultaneous Two Faults Case

This case considers the detection of two simultaneous faults related to stochastic changes superimposed on step changes in diffusivity and in one of the boundary conditions. The goal is to detect the step changes in the presence of the random perturbations around the changes in mean values of diffusivity and boundary condition. To illustrate variations in these two parameters, three different mean values are considered for each: 10, 20 and 30 for the diffusivity and 5, 10 and 15 for the boundary condition. This results in a total of nine classes, i.e. a total of nine combinations of mean values of diffusivity and boundary condition. Each class is referred to by the expectation of the diffusivity and boundary condition corresponding to a particular combination of the aforementioned values. For example, 'Class: $k=10, g=5$ ' means that the expectation for the diffusivity and the boundary condition is 10 and 5, respectively. Also for simplicity, the random perturbations superimposed on the mean values of either input as shown in Fig.1 are assumed to be normally distributed with zero mean and unit variance. Fig.3.13 shows the expectation and variance contour lines for each grid point calculated by the gPC coefficients described in Eqs. 3.15 and 3.16, where the highest order of the two dimensional gPC polynomials used for the solution in the case study is 2.

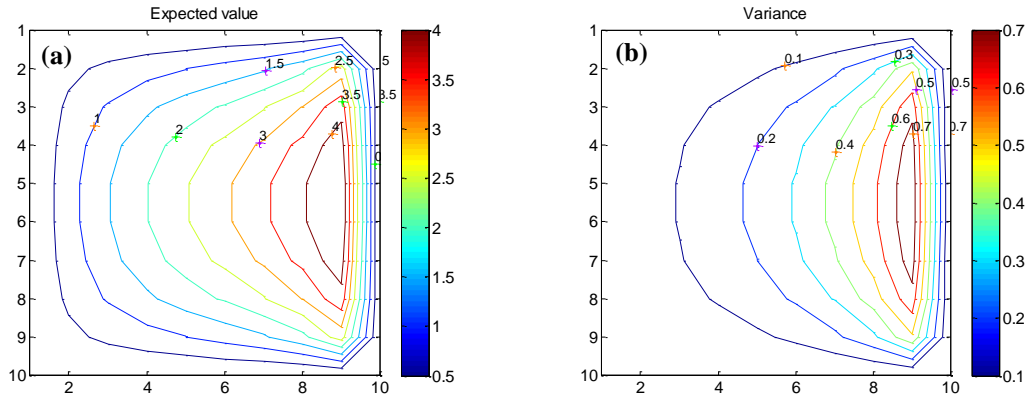


Figure 3.13 Mean and variance distribution over two-dimensional domain

(a) Expected value and (b) variance ($\bar{\kappa} = 20, \bar{T} = 5, Q = -100$)

Using the symmetry of the problem around a horizontal axis the analysis is conducted only for the top half of the square plate, as shown in Fig.3.14. In addition, as discussed in Section 3.5.1, the number and the placement of sensor have significant influence on both model calibration as well as fault detection. For simplicity, eight sensors shown in Fig.3.14 are used for model calibration. For simplicity, however, two different combinations of two sensor locations are investigated to study the influence of sensor placement on the observability of faults: (i) a sensor at the corner of the domain together with a sensor next to the boundary with changing condition (grid points 1 and 3 in Fig.3.14), and (ii) a sensor at the center of the domain together with a sensor next to the boundary with changing conditions (grid points 2 and 3 in Fig.3.14). The rationale is that the sensors positioned at the corner and center of the domain can serve to investigate the effect of the SNR, which is expected to be at extreme

values for these two locations. The sensor placed beside the stochastic boundary is used to obtain information about changes in this boundary.

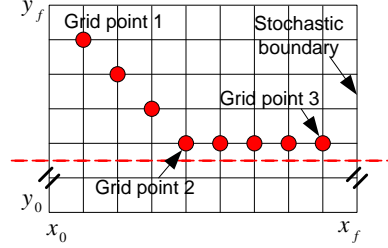


Figure 3.14 Sensor placements for Case II (one stochastic boundary)

Following modeling calibration procedures for two simultaneous faults as described in Section 3.4.2, eight sensors situated in Fig.3.14 and equal weights are used for optimization in Eq. 3.28, where P is 2 and N_{xy} is 8. Table 3.3 shows one group of model calibration results for the nine combinations of diffusivity and boundary values considered in the study, for which the variance of the measurement noise is 0.1. $\bar{\kappa}_{opt}$ and ζ_k are the optimum estimates for the stochastic diffusivity, \bar{g}_{opt} and ζ_g are the optimum values for the boundary condition as obtained from the solution of Eq. 3.28. It should be emphasized that eight sensors are utilized for model training to obtain better results of acceptable estimate rate A_{rate} as defined in Eq. 3.31 for both mean values in diffusivity and boundary condition. On the other hand, for simplicity, only two sensors at a time, i.e., either grid points 1 and 3 or grid points 2 and 3 as shown in Fig.3.14, are used for detecting the changes in diffusivity and boundary condition.

Using the calibrated results of diffusivity and boundary condition for all combinations of these two variables as summarized in Table 3.3, the gPC coefficients are calculated and the temperature distribution at a particular grid point (sensor) can be approximated. Using the gPC model, the JCR profiles are generated. For example, Fig.3.15 shows 99% confidence JCRs for the first sensor structure (grid points 1 and 3 depicted in Fig.3.14) for different combinations of diffusivity and boundary condition, respectively.

Table 3.3 Summary of model calibration results (noise variance $\sigma^2=0.1$)

Optimization results	Diffusivity		Boundary Condition	
	Class	$\bar{\kappa}_{opt}$	ζ_k	\bar{g}_{opt}
$\bar{\kappa} = 10, \bar{g} = 5$	10.0736	1.2553	4.9010	1.0348
$\bar{\kappa} = 10, \bar{g} = 10$	10.0737	0.8706	9.9010	1.0348
$\bar{\kappa} = 10, \bar{g} = 15$	10.3909	1.1910	15.0688	0.8455
$\bar{\kappa} = 20, \bar{g} = 5$	19.9999	1.6065	5.0131	0.9082
$\bar{\kappa} = 20, \bar{g} = 10$	20.1766	1.4162	9.9679	0.9002
$\bar{\kappa} = 20, \bar{g} = 15$	20.3064	1.1593	15.0603	0.9390
$\bar{\kappa} = 30, \bar{g} = 5$	30.0559	0.8277	4.8412	1.1195
$\bar{\kappa} = 30, \bar{g} = 10$	30.3387	0.9031	9.9736	0.9436
$\bar{\kappa} = 30, \bar{g} = 15$	30.4832	1.2802	15.0702	0.8430

As can be seen from Fig.3.15, there is no overlap between JCRs with a 99% confidence interval. Therefore, the gPC model based minimum distance approach, as defined in Eq. 3.29, can be used when the measurement is located between the calculated JCRs. By using a three-sigma confidence level, the solution domain of random variables (ξ) with a 99% confidence region is determined in Eq. 3.29.

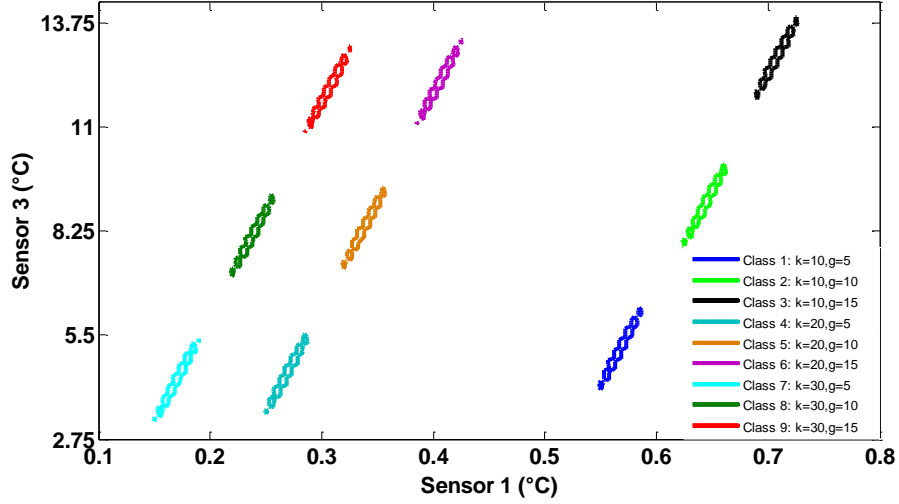


Figure 3.15 JCRs for two measurements at sensor 1 and 3 with a 99% confidence interval

The influence of the sensors' placement on the observability and distinguishability is studied for the two sensor structures under consideration. Table 3.4 shows the fault detection rate for $Q = -100$ by using either a JCR-profiles method or the gPC model based minimum distance method, where the measurement noise is assumed to be normally distributed with zero mean and variance of 0.1. There are 1000 pairs of testing samples (D_i in Eq.3.32) for each sensor placement strategy and for a specific combination of diffusivity and boundary condition thus resulting in a total of 9000 testing samples corresponding to nine different combinations of three means of diffusivity and three means of boundary condition values.

Table 3.4 Summary of results for fault detection rate for two simultaneous faults (noise variance $\sigma^2=0.1$)

Sensors	1&3		2&3	
	0	5	0	5
JCR profiles	0.5852	0.8148	0.8741	0.9630
Minimum distance	0.7481	0.9296	0.9741	0.9889

For both approaches, as seen in Table 3.4, the use of sensors 2 and 3 gives better results with respect to the fault detection rate (d_{rate}) due to higher SNR, as compared to the combination using sensor 1 and 3, which experiences larger changes on SNR. As done for Case Study I, replicates can mitigate the influence of SNR on the observability of faults as shown in Table 3.4 with 5 replicates. It is also found that the minimum distance

algorithm improves the ability to recognize and classify the faulty operations since many tested samples fall in between JCRs regions, as depicted in Fig.3.6 (b).

3.5.3 Computational Efficiency

The computational time of the proposed gPC methodology is compared with MC simulations. First, the expected value and variance for a particular grid point obtained with the gPC and MC methods are compared. The sensors are located along the diagonal direction from the top-left to the bottom-right corners, and the spatial discretization order N_{xy} is 15. Fig.3.16 shows the expectation and variance along the diagonal calculated by the gPC as well as MC with different number of samples. It is observed that there is no noticeable difference in the expectation between the gPC and MC. By contrast, the variances between the gPC and MC are slightly different when the number of samples used in MC is relatively small. The MC solution only approaches the gPC as the number of the samples increases, but at the cost of a significant increase in the computational time, which is further discussed later in this section.

To evaluate the efficiency of the gPC based algorithm for detecting faults as compared to MC, the detection rate, d_{rate} , is studied by MC. The mean and variances for the inputs using an MC based model for fault detection are calibrated by solving the optimization problem stated in Eq. 3.28 as done for the gPC based model. Once again, following symmetry considerations, only a quarter (top-left) of the solution domain, i.e., square plate, is studied. It is worth mentioning that the samples used to calculate the mean and variance approximated by MC in Eq. 3.28 are different in each optimization step, which results in a stochastic optimization problem. Thus, the genetic algorithm (GA) (Fouskakis & Draper, 2002) is employed to solve Eq. 3.28 when the MC model is used. This is fundamentally different from the solution of Eq. 3.28 when using gPC that is deterministic in terms of the expansions' coefficients.

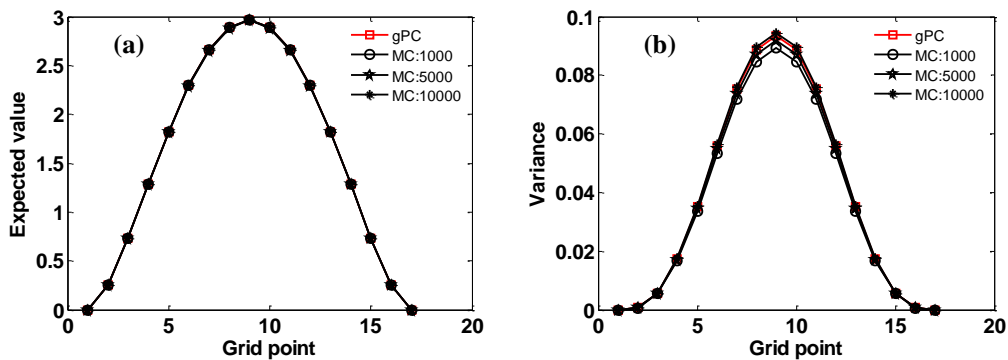


Figure 3.16 Comparisons of expected value (a) and variance (b) between gPC and MC

For consistency with respect to the gPC based approach, the same eight sensors along the diagonal as shown in Fig.3.8 are used for model calibration to compare the results between gPC and MC with a single fault in diffusivity. The same assumptions, for stochastic diffusivity, external heat, boundary condition and measurement

noise used in the gPC study are made for the MC study. Following the same criterion defined in Eq. 3.30, Fig.3.17 shows the estimation rate (A_{rate}) which is defined as in Eq. 3.31 and is equivalent to the ones obtained with gPC, and the computational time required for both gPC as well as MC to obtain these results. The computational times are shown in different scales in Fig.3.17 (b), due to the resulting orders of magnitudes differences between gPC and MC computation times.

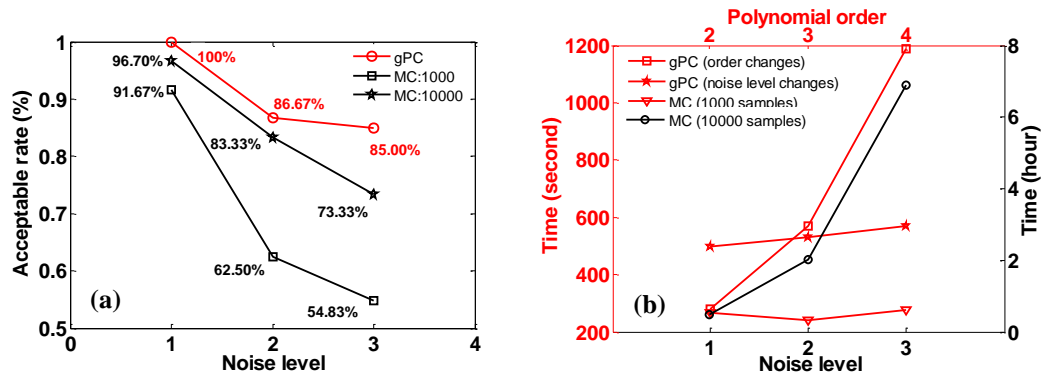


Figure 3.17 Comparison of model calibration results between gPC and MC (single fault)

For comparison purposes, the results of acceptable estimation rate A_{rate} by using gPC model is also given (red circled line in Fig.3.17 (a)), where sensor structure 6, as prescribed in Table 3.1, is used for consistency for both the gPC based method and the MC based method. Compared with MC, gPC gives better results with respect to the acceptable estimation rate A_{rate} . As seen from Fig.3.17 (a), A_{rate} is a function of noise level and it is highly dependent on the number of samples for the MC based method. For example, the acceptable estimation rate of the MC based method can be increased by ~18.5 percent point, from 54.83% to 73.33% for noise level 3, if 10,000 samples are used instead of 1,000 but at the cost of a significant increase in computation time, as can be observed in Fig.3.17 (b).

In general, the solution of the calibration step (Eq. 3.28) when using an MC model is highly sensitive to measurement noise. As shown in Fig.3.17 (b), it is observed that, to obtain an acceptable estimate rate level similar to gPC, 7 hours (black axes, black circled line) were needed for one single model calibration in MC, if the noise level is 3 as compared to the gPC based method that solves the problem in a matter of seconds. As compared with noise level 3, for example, only 1 hour was needed for the noise level 1 with the MC approach, which is still computationally demanding. By contrast, as shown in Fig.3.17 (b), the computational time for the gPC method is a function of the polynomial order, which is associated with the prior assumption of the probability distribution in random events. Even for a higher order polynomial, the computational load does not increase significantly, e.g., the average time for a fourth order polynomial is around 1200 seconds (red axes, red squared line), which is still significantly lower as compared with MC. All the methods are executed on a 2.66 GHz Intel^(R) Core Duo processor.

The MC approach is also used to study the fault detection efficiency and the results are compared with the gPC method at each grid point as depicted in Fig.3.8. Using MC, the PDF profiles are approximated for different mean values on diffusivity. For example, Fig.3.18 shows the PDF profile at grid point 8 as example, where 10,000 samples are used in MC. The horizontal axis in Fig.3.18 is the temperature, while the vertical axis is the normalized probability. Similar to the gPC approach, FDD tests are conducted for different measurement noise levels. As in the gPC study, each class is characterized by the expected value of diffusivity.

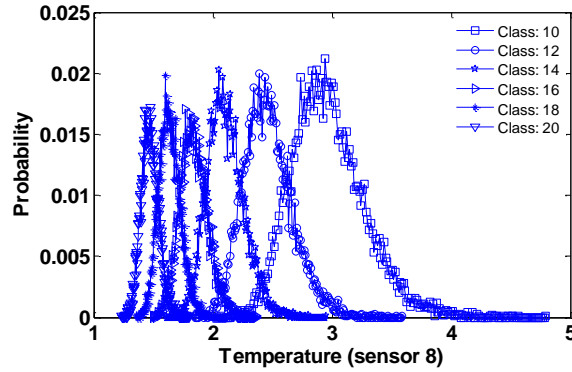


Figure 3.18 PDF profiles of six classes at grid 8 by MC ($Q = -100$, 10,000 samples)

Fig.3.19 shows the fault detection rate when the variance of the measurement noise is 0.1 and 0.2, and Q is -100. The same number of testing samples is used as for the gPC model based FDD. For comparison, the results obtained by gPC are also displayed. Compared with the gPC approach, the fault detection rate for these aforementioned eight sensors is approximately 2% lower for MC. Thus, a further slight increase in the number of samples might be needed for training the MC model, which will increase the computational effort. As shown, the observability of the fault is the best at grid point 8, thus confirming the result observed in the gPC study that the best sensor location is at the center of the domain. As in the gPC study, 10 replicates are used to mitigate the influence of lower SNR on the observability of faults. For the same noise level, the results corroborated that replicates can also improve the observability of faults with the MC modeling approach.

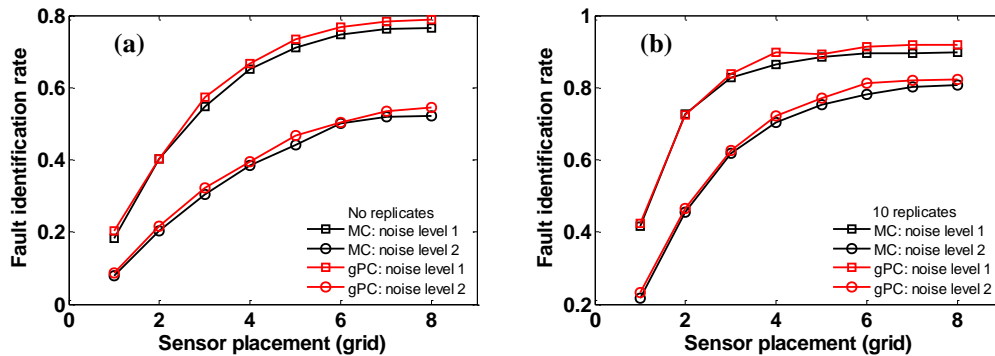


Figure 3.19 Comparison of result at each grid point between gPC and MC ($Q = -100$)

The gPC and the MC methodologies are further compared with respect to their abilities to detect a fault from measurements that lie in overlapping regions of the calculated PDFs with either one of the two methods. To test this point, the detection rate among different classes, i.e. different mean values of diffusivity, is categorized in terms of type I and type II, for both the gPC and the MC based approaches. A Type I error is the incorrect rejection of a true null hypothesis, while a type II error is the failure to reject a false null hypothesis (Montgomery & Runger, 1994). To quantify the distinguishability of adjacent classes with noisy measurement, the overlapping area as illustrated in Fig.3.20 for the training PDF profiles is approximated numerically by the following formulae:

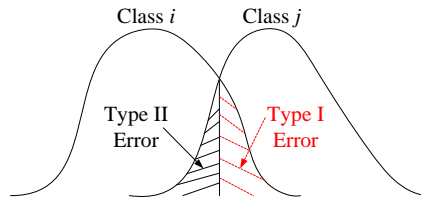


Figure 20 Type I and Type II error regions

$$E_I = n_{i,j}/N_s \quad (3.33)$$

$$E_{II} = n_{j,i}/N_s \quad (3.34)$$

where E_I is Type I error and E_{II} is Type II error, respectively. N_s is the total number of samples used to generate the PDFs for each class, $n_{i,j}$ is the number of samples in ‘Class i ’ that have been misclassified in ‘Class j ’, and $n_{j,i}$ is the number of samples in ‘Class j ’ that have been wrongly classified into ‘Class i ’. For a given measurement, Type I and II errors can provide the probability of misclassification.

Table 3.5 shows the type I and II errors of the gPC study confirms that the fault misclassification rate is higher for two groups that are adjacent to each other as compared to classes that are not adjacent to each other, since the Type I and Type II errors are relatively higher. For example, 21.86% of samples in *Class 20* may be misclassified as *Class 18*.

Table 3.5 Type I and Type II analysis for training set (gPC)

$\bar{\kappa}_{\text{simu}}$ (Class)	18		16		14		12		10	
	E_I	E_{II}	E_I	E_{II}	E_I	E_{II}	E_I	E_{II}	E_I	E_{II}
20	0.2186	0.1390	0.0415	0.0381	0.0151	0.0118	0.0037	0.0028	0.0041	4.00E-04
18	/	/	0.0015	0.1476	0.0050	0.0462	0.0020	0.0096	1.00E-04	0.0013
16	/	/	/	/	0.0026	0.2763	0.0040	0.0668	3.00E-04	0.0083
14	/	/	/	/	/	/	0.0090	0.1915	3.00E-04	0.0348
12	/	/	/	/	/	/	/	/	0.0070	0.3037

For comparison, Table 3.6 shows the results for the two errors with the MC model. As seen, most results exhibit relatively higher type I and type II errors as compared to gPC in Table 3.5. For instance, 40.94% of testing samples in *Class 20* may be identified as belonging to *Class 18*; and 27.75% of samples in *Class 18* may be misclassified into *Class 20*. Compared with the gPC, the Type I and Type II errors have almost doubled. An explanation for

the differences in types I and II errors between the gPC and MC model based detection approaches is the higher sensitivity to noise of the MC approach. A clear evidence for the higher sensitivity to noise is that the PDFs obtained by the MC approach are less smooth as shown in Fig.3.18, as compared to the PDF profiles calculated with the gPC expansions in Fig.3.9. Once again, a slight increase of the number of samples used for PDF profiles generation in MC may improve the accuracy but at the cost of a significant increase in computations as discussed in Fig.3.17 (b).

Table 3.6 Type I and Type II analysis for training set (MC)

\bar{K}_{simu} (Class)	18		16		14		12		10	
	E_I	E_{II}	E_I	E_{II}	E_I	E_{II}	E_I	E_{II}	E_I	E_{II}
20	0.4094	0.2775	0.0698	0.0369	0.0131	0.0021	5.00E-04	9.00E-04	0	0
18	/	/	0.1170	0.0782	0.0301	0.0058	0.0021	0.0015	2.00E-04	2.00E-04
16	/	/	/	/	0.2700	0.0416	0.0496	0.0127	0.0033	0.0011
14	/	/	/	/	/	/	0.4094	0.0859	0.1666	0.0058
12	/	/	/	/	/	/	/	/	0.2700	0.0511

To further verify the higher sensitivity to noise of the MC versus the gPC method, the effect of type I and type II errors on the detectability of faults is studied for a particular case where the temperature values used for fault detection correspond to measurements located in the overlapping regions of adjacent PDFs, since most misclassification happens near the class boundaries. It is found that most of the detection rates obtained by gPC are approximately 5% higher than for the MC based approach.

3.6 Conclusion

A new approach based on the gPC expansion of uncertainties quantification and propagation is proposed for the fault detection and diagnosis (FDD) problem. The efficiency has been demonstrated by a two-dimensional heat conduction problem, where the distributional uncertainties on diffusivity and boundary condition are considered. A key contribution is that the proposed methodologies are successful in detecting and diagnosing both individual as well as simultaneous occurrences of multiple stochastic faults. The proposed method was studied in terms of sensitivity to signal to noise ratio (SNR) and sensor location. The distinguishability of faults near the classes' boundaries is assessed from the Type I and Type II errors that quantify the overlap between two PDFs of observations under different stochastic fault modes. The key advantage of the proposed gPC approach is that the computational times are orders of magnitude shorter than for the MC simulations based approaches thus showing the potential of gPC for addressing problems of large dimensions.

Chapter 4

Fault Diagnosis for Nonlinear Dynamic Processes

(Adopted from Du et al., 2015, Industrial & Engineering Chemistry Research, in press)

4.1 Overview

This paper deals with detection and classification of intermittent stochastic faults by combining a generalized polynomial chaos (gPC) representation with either Maximum Likelihood or Bayesian estimators. The gPC is used to propagate stochastic changes in an input variable to measured quantities from which the fault is to be inferred. The fault detection and classification problem is formulated as an inverse problem of identifying the unknown input based on the Maximum Likelihood of fit between predicted and measured output variables, or on a Bayesian inference based estimator which recursively updates the gPC coefficients. Simulation studies compare the proposed methods with a Particle Filter (PF) to estimate the value of an unknown feed mass fraction of a chemical process. The proposed method is shown to be significantly more efficient in terms of computational effort and less sensitive to user defined tuning parameters than a PF.

Empirical fault detection methods have been often used for fault detection and classification. However such algorithms may be less accurate since they do not specifically address the stochasticity of the faults as discussed in the supplementary materials. To demonstrate this point, a comparison study has been conducted in Appendix A between the proposed gPC models based mechanistic algorithm and a Gaussian Process based statistical algorithm.

4.2 Introduction

An essential aspect of the economical and safe operation of chemical processes is rapid detection and removal of malfunctions or faults. A fault is defined as a deviation of one or more variables from an acceptable level (Isermann R. , 2006). If a fault is observable, the fault detection and diagnosis (FDD) system will provide symptomatic fingerprints, which in turn can be referred back to the fault diagnosis scheme to identify the root cause of the anomalous behaviour (Venkatasubramanian, Rengaswamy, Yin, & Kavuri, 2003). However, since FDD schemes are invariably based on models, a main restrictive factor of an efficient FDD system is the model uncertainties (Chiang, Russell, & Braatz, 2008). Such uncertainty may originate from either intrinsic time varying phenomena of model parameters or may result from inaccurate model calibration due to stochastic noisy data. The step of quantifying and propagating the uncertainties onto the variables used for detection is typically omitted in reported FDD studies, leading to a loss of information arising from these uncertainties.

In terms of applications of FDD algorithms, many industrial processes are intrinsically nonlinear systems (Gerlter, 1998) and they are operated at different operating conditions according to economic considerations (Haghani, Jeansch, & Ding, 2014). Due to nonlinearity, the performance of linear FDD algorithms often reported

in literature may be inaccurate especially when the process transits from one operating condition to another (Li & Yang, 2012). For example, FDD algorithms that are based on the steady state information will result in false alarms or missed detection of faults when performing detection with measurements collected during dynamic transients. Fault diagnosis (classification) that explicitly considers dynamic transients among different operating conditions in the presence of model uncertainties has not been substantially addressed. Classification of the cause of faults is equally critical to the detection of faults, since rapid classification will lead to a reduced economic loss. Additionally, most of the fault classification methods focus on classification using pattern recognition techniques, but provide little information about what is the probability that a fault has occurred in the presence of model uncertainties and noise.

Following the above, this paper presents two methods to improve fault detectability during transients in the presence of parametric uncertainties. Preliminary results of detecting and classifying stochastic faults with steady state measurements were outlined in the earlier work by the authors (Du, Duever, & Budman, 2015). A significant reduction in computational time was observed using the generalized polynomial chaos (gPC) expansion (Ghanem & Spanos, 1991; Xiu D. , 2010), as compared with Monte Carlo sampling based methods. As such our earlier work was not suitable to deal with dynamic fault scenarios. The current work expands upon our preliminary work by combining the gPC based model with either a Maximum Likelihood or a Bayesian based estimator to dynamically estimate the stochastic faults during transients. The proposed approaches are used to identify and classify the unmeasured stochastic intermittent faults for a nonlinear chemical plant. Specifically, the methods in this current work are developed to discriminate between specific fault classes and the normal process operation as well as between fault classes using dynamic transients.

The use of a gPC expansion to approximate the uncertainty of interest can reduce the computational complexity to a reasonable level. While the benefit of using the gPC models in parameter estimation problems has been reported (Chen-Charpentier & Stanescu, 2014; Madankan, Singla, Singh, & Scott, 2013; Pence, Fathy, & Stein, 2011; Emmanuel, Sandu, & Sandu, 2007): it has mainly been applied offline while this work proposes a gPC based estimators for real-time detection of intermittent faults. Unlike many traditional model-based methods such as the Kalman Filter (Daum, 2005), the proposed approach explicitly considers: (i) the nonlinear behaviour of the process, (ii) the stochastic nature of the parametric faults, and (iii) their effects on the measured quantities. A known alternative to solving this problem that specifically involved stochastic faults is to apply a particle filter (PF) (Arulampalam, Maskell, Gordon, & Clapp, 2002; Orchard & Vachtsevanos, 2009; Kadiramanathan, Li, Jaward, & Fabri, 2002): but it will be shown that the proposed gPC approach is significantly more efficient than PF in terms of computational time, thus making it more suitable for the real-time implementation in problems of large dimensions. Also, the proposed algorithm is shown to be less sensitive than PF to user selected tuning parameters. It is also shown that the proposed algorithm is suitable for selecting sensors to improve detection.

To summarize, the contributions of this chapter are: (i) The use, in the context of fault detection and diagnosis, of a gPC model based approach for uncertainty propagation and quantification applied directly to the first principles nonlinear model of a complex system; (ii) The use of maximum likelihood or Bayesian inference based estimators in combination with the gPC model for improved fault diagnosis; and (iii) Optimal selection of sensors used for fault detection based on sensitivity analysis of the gPC model.

This chapter is organized as follows. In Section 4.3, the theoretical background of gPC expansions is presented. The maximum likelihood and Bayesian inference based two-level fault detection algorithms, as well as a sensitivity analysis based approach for sensor selections are explained in Section 4.4. A nonlinear chemical plant with two continuously stirred tank reactors and a flash tank separator is introduced as a case study in Section 4.5. Analysis and discussion of the results are given in Section 4.6 followed by conclusions in Section 4.7.

4.3 Generalized Polynomial Chaos

A generalized polynomial chaos (gPC) expansion can be used to represent an arbitrary random variable of interest as a function of a polynomial series of another random variable of a given standard distribution (Ghanem & Spanos, 1991; Xiu D. , 2010). Let us assume a set of nonlinear ordinary differential equations (ODEs) describing the dynamic behaviour of a system:

$$\begin{aligned}\dot{\mathbf{x}} &= f(t, \mathbf{x}, \mathbf{u}; \mathbf{g}) \\ 0 \leq t \leq t_f, \mathbf{x}(0) &= \mathbf{x}_0\end{aligned}\quad (4.1)$$

where the vector $\mathbf{x} \in \mathbf{R}^n$ contains the system states (measured variables) with initial conditions $\mathbf{x}_0 \in \mathbf{R}^n$ over time domain $[0, t_f]$, and \mathbf{u} denotes the known inputs of the system. The vector $\mathbf{g} \in \mathbf{R}^{n_g}$ is the unknown stochastic time varying input. It will be assumed heretofore that the input vector \mathbf{g} is the stochastic fault/s that has to be detected by the FDD algorithm in this current work. The function f is assumed to be a fundamental model of the process that can be developed from first principles. To quantify the effect of stochastic inputs (faults) \mathbf{g} on the different measured variables \mathbf{x} , a gPC expansion can be employed. To that purpose each unknown input fault g_i ($i = 1, 2, \dots, n_g$) in \mathbf{g} is represented as a function of a set of random variables $\boldsymbol{\zeta} = \{\zeta_i\}$:

$$g_i = g_i(\zeta_i) \quad (4.2)$$

where ζ_i is the i^{th} random variable and the elements in the set ($\boldsymbol{\zeta} = \{\zeta_i\}$) are assumed to be independent and identically distributed. Using gPC expansions, the unknown stochastic faults (inputs) $\mathbf{g}(\boldsymbol{\zeta})$ and system states $\mathbf{x}(t, \boldsymbol{\zeta})$ are described in terms of orthogonal polynomial basis functions $\Phi_k(\boldsymbol{\zeta})$:

$$\mathbf{g}(\boldsymbol{\zeta}) = \sum_{k=0}^{\infty} \mathbf{g}_k \Phi_k(\boldsymbol{\zeta}) \quad (4.3)$$

$$\mathbf{x}(t, \boldsymbol{\zeta}) = \sum_{k=0}^{\infty} x_k(t) \Phi_k(\boldsymbol{\zeta}) \quad (4.4)$$

where \mathbf{x}_k and \mathbf{g}_k are the gPC coefficients of measured variables (states) and faults at each time instant t , $\Phi_k(\xi)$ are multi-dimensional orthogonal basis functions of ξ . It is assumed that the input (\mathbf{g}) can be measured or estimated and then the coefficients of the unknown input, \mathbf{g}_k , can be calculated such that Eq. 4.3 follows a statistical distribution determined from collected data. Then, after substituting the gPC representation of the input \mathbf{g}_k into the process model, it is possible to calculate gPC representations of the measured quantities (states) by applying a Galerkin projection procedure (Xiu D. , 2010). The latter permits to compute the expansion coefficients $\{x_k(t)\}$ by projecting Eq. 4.1 onto each one of the polynomial chaos basis functions $\{\Phi_k(\xi)\}$ as follows:

$$\langle \dot{\mathbf{x}}(t, \xi), \Phi_k(\xi) \rangle = \langle f(t, \mathbf{x}(t, \xi), \mathbf{u}(t), \mathbf{g}(\xi)), \Phi_k(\xi) \rangle \quad (4.5)$$

For practical application, Eqs. 4.3 and 4.4 are often truncated to a finite number of terms, i.e., P . Hence, the total number of terms in Eq. 4.5 is a function of an arbitrary order p in Eq. 4.3 that is deemed sufficient to represent a known priori distribution of \mathbf{g} for n_g different faults (inputs) in vector \mathbf{g} as follows:

$$P = ((n_g + p)! / (n_g! p!)) - 1 \quad (4.6)$$

From Eq. 4.6, the number of the gPC expansion terms for the measured variables in Eq. 4.4 increases as the polynomial order p in Eq. 4.3 and the number of unknown inputs n_g in Eq. 4.2 increase. The inner product in Eq. 4.5 between two vectors can be defined by:

$$\langle \psi(\xi), \psi'(\xi) \rangle = \int \psi(\xi) \psi'(\xi) W(\xi) d\xi \quad (4.7)$$

where the integration is conducted over the entire event domain generated by the random variables ξ , and $W(\xi)$ is a weighting normalizing function, which is chosen according to the polynomial basis function used to represent ξ so as the result of Eq. 4.7 is either one or zero. To enforce orthogonality of the basis functions, these have to be selected according to the choice of the statistical distribution of ξ . For example, Hermite polynomials are chosen as basis functions for normally distributed ξ and Laguerre polynomials can be used for Gamma distributed ξ (Xiu D. , 2010). Once the coefficients of the expansion in Eq. 4.4 are calculated, it is possible to compute statistical moments for the measured variables at any given time instant t as Eqs 4.8 and 4.9 as a function of the coefficients of the expansion \mathbf{x}_k in Eq. 4.4 as follows (Xiu D. , 2010):

$$E(\mathbf{x}(t)) = E \left[\sum_{i=0}^P x_i(t) \Phi_i \right] = \mathbf{x}_0(t) E[\Phi_0] + \sum_{i=1}^P E[\Phi_k] = \mathbf{x}_0(t) \quad (4.8)$$

$$\begin{aligned} Var(\mathbf{x}(t)) &= E \left[(\mathbf{x}(t) - E(\mathbf{x}(t)))^2 \right] = E \left[\left(\sum_{i=0}^P x_i(t) \Phi_i - \mathbf{x}_{(i=0)}(t) \right)^2 \right] \\ &= E \left[\left(\sum_{i=1}^P x_i(t) \Phi_i \right)^2 \right] = \sum_{i=1}^P x_i(t)^2 E(\Phi_i^2) \end{aligned} \quad (4.9)$$

Also, the probability density functions (PDFs) for the measured variables, $x(t)$, can be approximated by sampling the distribution of ζ and substituting the samples into Eq. 4.4. The availability of analytical formulae for calculating statistical moments as per Eq. 4.8 and 4.9 are the main rationale for using gPC, since it dramatically reduces the computational effort involved in repeated calculations of moments and the PDFs as required in this study. The fault detection procedure used in the current work consists of inferring the distribution of the stochastic parametric faults (inputs) g from measurements of the process measured variables x . Further details about this inference are given in Section 4.4.

4.4 Fault Detection and Diagnosis Methodology

4.4.1 Unknown Input Fault Detection and Classification Problem Formulation

The unknown input faults g to be considered in this work consist of stochastic perturbations around a specific set of mean values as described in Fig.4.1 (a), and given mathematically as follows:

$$g = \bar{g}_i + \Delta g_i \quad (i = 1, \dots, k) \quad (4.10)$$

where $\{\bar{g}_i\}$ are a set of constant mean values (operating modes), $\{\Delta g_i\}$ are stochastic variations around each mean value. The statistical distribution of Δg_i is assumed to be known a priori and time invariant. The changes in the mean values of \bar{g}_i follow a Multilevel Pseudo Random Signal (ML-PRS) (Ljung, 1999). The inputs described by Eq. 4.10 are typical in chemical processes that experience both changes in means of operating variables but also in additional continuous random perturbations in time. Then, the FDD problem is defined as detecting a change in the unknown input mean \bar{g}_i as well as diagnosing around which particular \bar{g}_i the system is being operated. Each particular mean \bar{g}_i will be referred heretofore as to an operating mode, and thus the goal in the current work is to classify the operating mode \bar{g}_i at any given time instant t .

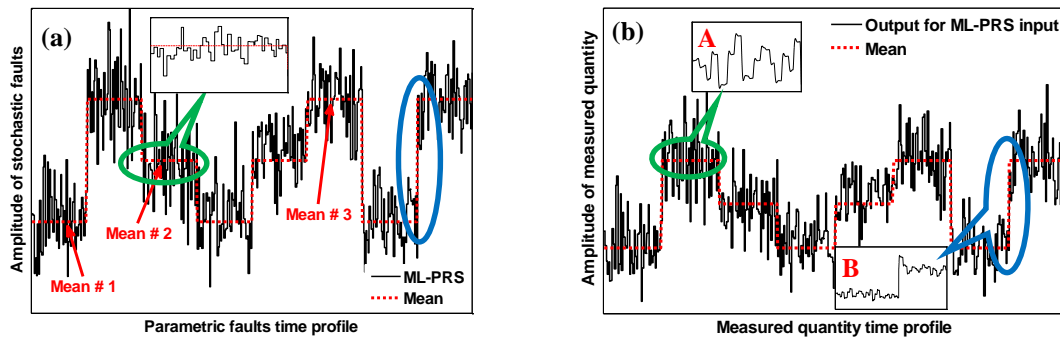


Figure 4.1 Fault profile representing an intermittent stochastic input fault and resulting measured variable

The fault detection and classification method in this work is formulated as a two-level procedure composed of a *Level-1 algorithm* and a *Level-2 algorithm*. This method is developed to discriminate between specific fault classes and the normal process operation as well as between fault classes.

Level-1 algorithm – For each mean value of \bar{g}_i given in eq 10, the corresponding PDF profile of the measured output variables (\mathbf{x}) can be calculated assuming that the mean value remains constant for a very long time, i.e., in the neighbourhood of a steady state. The PDF profiles of the measured variables are calibrated with simulated noisy measurements. The steady state at any given time can be initially inferred by testing the measured quantities with respect to the PDFs built around different steady states (operating modes). However, classification based on steady state information only, as done in this *Level-1 algorithm*, is not effective during transient changes among different steady states (multiple classes/operating modes). For that purpose the *Level-1 algorithm* is supplemented by the *Level-2 algorithm* explained next.

Level-2 algorithm – This algorithm is based on inferring the input fault \mathbf{g} from the application of a fitting criterion of the measured variables over a moving time window. To reduce computational effort, this step is only executed when large deviations from an input mean \bar{g}_i are detected with the *Level-1 algorithm* introduced above. In principle this second level algorithm (*Level-2 algorithm*) can be executed at each time interval but at the cost of increased computational time.

Two fitting criteria of the measured variables are proposed based on either a maximum likelihood function or a Bayesian inference estimator for detecting the average of the unknown input. The likelihood function is based on the error in mean and variance between a set of measurements and predictions calculated with a gPC model in Eq. 4.4. Similarly, a Bayesian inference based estimator is applied to dynamically infer the posterior gPC coefficients of the stochastic fault over a moving time window, which can be further used for fault diagnosis. Compared with the maximum likelihood based estimator, the objective is to recursively estimate the stochastic parametric faults during transients. Additional details on the *two-level algorithm* of fault detection are given below.

4.4.2 Level-1 Algorithm

For the purpose of calculating the PDF profiles, it is assumed that measurements of the certain variables (\mathbf{x}) around each mean value \bar{g}_i are available. It is also assumed in this step that the mean value of an input (fault) \bar{g}_i remains constant but its exact value is not known. The constancy of \bar{g}_i can be experimentally inferred from the constancy of measured and/or controlled variables through a steady state test (Seborg, Mellichamp, Edgar, & Doyle, 2010). In principle, in the absence of measurement noise and if the means and variances of the inputs (faults) \mathbf{g} would be known, the PDF profiles of the output variables (\mathbf{x}) that are measured and used for fault detection could be exactly calculated from a process model with the analytical expressions of a gPC as per the procedures shown in Section 4.3. Then, it could be possible to accurately infer the input from a measured output value by inverting the procedures outlined in Section 4.3.

However, in practice, due to noise and model error (e.g., gPC truncation error), the exact mean and variance of the input (fault) during steady state operation are not known and are unmeasured in FDD problems. Thus, the

PDF profiles of \mathbf{x} around each possible steady state (operating mode) have to be calibrated using actual process measurements. To this purpose, the mean and variance of the unknown input variable \mathbf{g} are calibrated from an optimization problem around each steady state (operating mode) shown in Fig.4.1 (a) as:

$$\min_{\lambda_{level-1}} J = \sum_{i=1}^n (\mathcal{G}_{1,i} - \nu_{1,i})^2 + \sum_{i=1}^n (\mathcal{G}_{2,i} - \nu_{2,i})^2 + \sum_{i=1}^n \sigma_{n,i}^2 \quad (4.11)$$

where $\mathcal{G}_{1,i}$ and $\mathcal{G}_{2,i}$ are the predicted mean and variance of a particular measured variable (\mathbf{x}) of the problem to be used for fault detection. These predicted means and variances are given explicitly by Eqs. 4.8 and 4.9 using the gPCs representations of \mathbf{x} given in Section 4.2, and are functions of the stochastic input as shown in Fig.4.1 (a). The terms $\nu_{1,i}$ and $\nu_{2,i}$ are the measured mean and variance of \mathbf{x} in Eq. 4.1. The last term $\sigma_{n,i}$ is utilized to represent the standard deviation of noise that is also expressed by a gPC expansion of the following form:

$$\sigma_n(t, \xi) = \sum_{k=1}^P \sigma_{n,k}(t) \Phi_k(\xi) \quad (4.12)$$

where $\sigma_{n,k}$ is the gPC coefficients of noise at time instant t , $\Phi_k(\xi)$ is the multi-dimensional polynomials in terms of ξ , and the variance of noise is assumed to be known a priori.

The decision variable $\lambda_{level-1}$ in Eq. 4.11 is a vector consisting of the mean and variance of the unknown fault (\mathbf{g}) and noise σ_n , and n is the number of the measured variables \mathbf{x} used to calibrate the gPC model. Due to noise and truncation error introduced by the gPC approximation, the mean and variance of the input variable (\mathbf{g}) defining $\lambda_{level-1}$ calculated from Eq. 4.11 deviate from the actual values entering the process. After obtaining $\lambda_{level-1}$, it is possible to calculate the actual gPC coefficients for the measured variables \mathbf{x} . Using these coefficients, the PDF profiles for \mathbf{x} 's around each constant mean value (operating mode) can be approximated by substituting samples (ξ) from a priori known distribution, e.g., normal or uniform, into the resulting gPC expansions given in Eq. 4.4. Following these substitutions the PDF profiles are calculated as a histogram composed of bins each corresponding to different ranges of values of \mathbf{x} (Du, Duever, & Budman, 2015).

Histograms are built for each of the mean values considered in Eq. 4.10. When the system is operated around a constant mean \bar{g}_i , the corresponding index i ($i = 1, \dots, k$) in Eq. 4.10 is detected from the PDF profiles for a given measurement as follows:

$$\text{Operating Mode}(\bar{g}_i) = \arg \max \{P_i\} \quad (4.13)$$

where i is the i^{th} operating mode as defined in Eq. 4.10. P_i means the probability that the process is operating about a particular mean \bar{g}_i for a given measurement. The solution of this problem is depicted in Fig.4.2 showing 3 PDF profiles that correspond to 3 different operating modes (input mean values). For example, three probabilities (red dots) can be found for a given measured variable shown in Fig.4.2, where the maximum probability can be used to indicate that the system is operating around the second mean value corresponding to 'Operating mode 2'. In practical problems, one of these operating modes can be defined as the normal operating

condition, while the rest can be defined as faulty operating conditions. The PDF profiles based *Level-1 algorithm* can discriminate between specific fault classes and the normal process operation as well as all other fault classes.

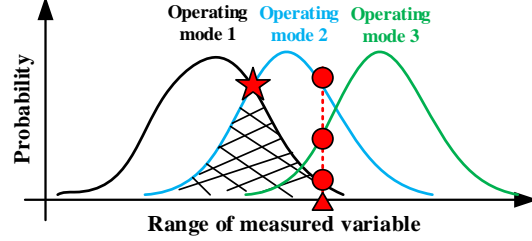


Figure 4.2 Visual interpretation of FDD with the level-1 algorithm

4.4.3 Level-2 Algorithm

The *Level-1 algorithm* presented in Section 4.4.2 assumes that the system is operated about a specific mean value \bar{g}_i in Eq. 4.10, but it does not take into account transient responses resulting from the step changes occurring among different \bar{g}_i as shown in Fig.4.1 (a). For instance, the step change indicated in the figure by a blue circle represents the dynamic change between *mean value 1* and *mean value 3*. Thus the *Level-1 algorithm* can only serve as a preliminary indicator that the system operates around a particular steady state but this diagnosis may be inaccurate during transients. This section explains the problem of estimating these step changes (faults) based on a fitting criterion between measured and predicted variables over a chosen moving time horizon. Two fitting criteria are tested: (i) a maximum likelihood and (ii) a Bayesian inference.

4.4.3.1 Classification of transient changes using a maximum likelihood criterion

The likelihood function between measured values and model predictions is maximized over a moving time window. Define $f_{x'}$ as the PDF of a measured output of interest, which can be estimated by a Gaussian kernel density function as (Wand & Jones, 1995):

$$f_{x'}(x', \pi(\xi)) = \frac{1}{n_i} \sum_{i=1}^{n_i} G_k [x' - \pi(t, x(\xi), u; g(\xi))] \quad (4.14)$$

where G_k denotes the Gaussian kernel, x' is the measured variable, n_i is the number of samples drawn from a priori known distribution of ξ . The π operator is the gPC model and can be obtained as explained in Section 2. Then, a likelihood function of output variables x over a moving time window of m measurements can be calculated as:

$$\ell(\pi, x') = \prod_{j=1}^m f_{x'}(x'_j; \pi) \quad (4.15)$$

Thus, for a moving time window of m measurements, an estimate of an average value of the fault (input) g can be obtained by maximizing the likelihood function as follows:

$$\max_{\lambda_{level-2}} \ell(\pi, x') = \prod_{j=1}^m f'_x(x'_j; \pi) \quad (4.16)$$

where the decision variable $\lambda_{level-2}$ in Eq. 4.16 is the average of the unmeasured time varying input \mathbf{g} and the corresponding confidence level over the moving time window of m measurements. It is worth noting that the same set of ξ used in Eq. 4.15 is also used to maximize Eq. 4.16. For classification purposes, the average value of the input \mathbf{g} resulting from Eq. 4.16 is compared to the mean values $\{\bar{g}_i\}$ to identify which operating mode is active. Although this maximum likelihood based *Level-2 algorithm* can be executed at each time interval, it is only used after a large deviation is detected by the *Level-1 algorithm* to reduced computational burden.

4.4.3.2 Level-2: Classification of transient changes using Bayesian inference

To diagnose the intermittent changes in the mean values of stochastic faults, a Bayesian inference based estimator is applied to infer the posterior gPC coefficients of the stochastic fault over a moving time window. The gPC coefficients can be then used for calculating the mean value and the variance. Compared with the maximum likelihood based estimator, the objective is to recursively estimate the stochastic parametric faults during transients.

The PDF of faults \mathbf{g} can be formulated using Bayesian inference with a set of measurements collected over a moving time window as follows:

$$p(\mathbf{g}|\mathbf{x}_k) = (p(\mathbf{g}|\mathbf{x}_{k-1})p(\mathbf{x}_k|\mathbf{g}))/p(\mathbf{x}_k) \quad (4.17)$$

where \mathbf{x}_k means the measurements collected over a moving time window up to time instant t_k . The prior PDF of \mathbf{g} at t_k , i.e., $p(\mathbf{g}|\mathbf{x}_{k-1})$, is based on all measurements available up to time interval t_{k-1} , $p(\mathbf{x}_k|\mathbf{g})$ is the likelihood that \mathbf{x}_k can be observed given \mathbf{g} at time t_k , and $p(\mathbf{g}|\mathbf{x}_k)$ denotes the posterior PDF (gPC coefficients) of \mathbf{g} given all measurements up to t_k . The marginal likelihood $p(\mathbf{x}_k)$ is the total probability of measurements at time instant t_k and can be estimated from (Emmanuel, Sandu, & Sandu, 2007):

$$p(\mathbf{x}_k) = \int p(\mathbf{g}|\mathbf{x}_{k-1})p(\mathbf{x}_k|\mathbf{g})d\mathbf{g} \quad (4.18)$$

Although the marginal likelihood can be evaluated by the integration in Eq. 4.18, in this work there is no need to calculate it since the probability of observing the measurements is assumed to be constant. To calculate the posterior PDF of \mathbf{x}_k , a differentiable scalar function defined as $\varphi(\mathbf{g}, \mathbf{x})$ is used that depends on the stochastic faults and measured variables. Multiplying both sides of Eq. 4.17 with $\varphi(\mathbf{g}, \mathbf{x})$ and integrating over \mathbf{g} yield:

$$\int p(\mathbf{g}|\mathbf{x}_k)\varphi(\mathbf{g}, \mathbf{x})d\mathbf{g} = \int \frac{p(\mathbf{g}|\mathbf{x}_{k-1})p(\mathbf{x}_k|\mathbf{g})\varphi(\mathbf{g}, \mathbf{x})}{p(\mathbf{x}_k)}d\mathbf{g} \quad (4.19)$$

It should be noted that both sides of Eq. 4.19 are only a function of \mathbf{x}_k , which can be represented as a function of gPC coefficients. To calculate the likelihood $p(\mathbf{x}_k|\mathbf{g})$, the Gaussian kernel function can be used for simplicity and the calculation is similar to Eq. 4.14.

The main challenge is to evaluate the integrals in Eq. 4.19 in a computationally efficient way. It should be noted that all components in Eq. 4.19, i.e., \mathbf{g} and \mathbf{x} , are functions of their gPC coefficients and a set of polynomial basis functions $\Phi(\xi)$. The prior PDF given by the gPC expansions of the faults and of the measured variables can be calculated offline and stored in memory. Using these gPC expansions, Eq. 4.19 can then be re-written as follows:

$$\int p(\mathbf{g}_{k,gpc} \Phi(\xi) | \mathbf{x}_{k,gpc} \Phi(\xi)) \varphi(\mathbf{g}_{k,gpc} \Phi(\xi), \mathbf{x}_{k,gpc} \Phi(\xi)) d\xi = G^* \int \frac{p(\mathbf{g}_{k-1,gpc} \Phi(\xi) | \mathbf{x}_{k-1,gpc} \Phi(\xi)) \varphi(\mathbf{g}_{k-1,gpc} \Phi(\xi), \mathbf{x}_{k-1,gpc} \Phi(\xi))}{D} d\xi \quad (4.20)$$

where $\mathbf{g}_{k,gpc}$ and $\mathbf{x}_{k,gpc}$ represent the posterior gPC coefficients at time instant t_k , $\mathbf{g}_{k-1,gpc}$ and $\mathbf{x}_{k-1,gpc}$ denote the gPC coefficients of \mathbf{g} and \mathbf{x} at time interval t_{k-1} , G is the prior density function calculated with a Gaussian kernel function, and D is a scaling denominator. The gPC coefficients $\mathbf{g}_{k,gpc}$ and $\mathbf{x}_{k,gpc}$ are continuously updated and optimized based on the available new measurements. For simplicity of the presentation, the left-hand side of Eq. 4.20 is defined as γ_1 and the right-hand side is defined as γ_2 . To update the gPC coefficients, the following likelihood function using a Gaussian kernel can be formulated:

$$\max_{\mathbf{g}_{k,gpc}, \mathbf{x}_{k,gpc}} J = \exp(-(\gamma_1 - \gamma_2)^2 / 2\sigma^2) \quad (4.21)$$

where $\mathbf{g}_{k,gpc}$ and $\mathbf{x}_{k,gpc}$ are the decision variables representing the posterior gPC coefficients at time instant t_k and σ is the estimate of the error. The likelihood is maximal when the error is minimal. A gradient descent algorithm is applied for finding a set of gPC coefficients that maximizes the likelihood function in Eq. 4.21.

For the detection of stochastic faults about different mean values (as seen in Fig.4.1), a set of symbolic gPC models are generated for each of these means $\{\bar{g}_i\}$. Then different sets of gPC coefficients can be obtained from Eq. 4.21 corresponding to each one of the mean values \bar{g}_i . Finally, the set of gPC coefficients that results in the maximum value of the likelihood in Eq. 4.21 is used to estimate the average value of faults (inputs) over the moving time window by using Eq. 4.8.

To obtain an explicit solution of the integral in Eq. 4.20, the scalar function is chosen as $\varphi(\mathbf{g}, \mathbf{x}) = \mathbf{g}^{s_1} \mathbf{x}^{s_2}$ which ensures matching of joint momenta for two sides of Eq. 4.20 up to order of $s_1 + s_2$ (sum of the orders of \mathbf{g} and \mathbf{x} in $\varphi(\mathbf{g}, \mathbf{x})$) (Madankan, Singla, Singh, & Scott, 2013). In this current work, s_1 and s_2 are selected as $s_1 = 1$ and $s_2 = 1$ for simplicity. For computational efficiency, a bi-level optimization, involving a *lower-level optimization* and an *upper-level optimization*, is developed for solving Eq. 4.21. Only the first coefficients of the gPC expansions, from which the mean of \mathbf{g} and \mathbf{x} can be calculated, are optimized and updated in the *lower-level optimization* while the remaining higher order gPC coefficients retain their prior value. The *upper-level optimization* is only launched when the decision variables in the *lower-level optimization* reached a constant value. Using the results obtained in the *lower-level optimization* and the same measurements collected over a moving time window, the higher order gPC coefficients can be updated when information about the higher moments is required. In principle,

all the gPC coefficients can be simultaneously updated and optimized but at the cost of a slightly increased computational time.

4.4.4 Summary of Level-1 and Level-2 Algorithm

The *two-level* fault detection and diagnosis (FDD) algorithm proceeds as per the following steps:

Step 1 - The PDF profiles of the measured variables x in Eq. 4.4 operating around each one of the mean values \bar{g}_i in Eq. 4.10 is approximated using the *Level-1 algorithm* in Section 4.4.1.

Step 2 - When a sample of measurements is available, the probabilities P_i in Eq. 4.13 are assessed ($i = 1, \dots, k$). The maximum probability can be used to infer a particular mean value \bar{g}_i (operating mode) as illustrated in Fig.4.2.

Step 3 - A potential change in the operating mode (mean value \bar{g}_i) is detected by the *Level-1 algorithm* when the probability of a given measurement switches across a limit between two adjacent PDF profiles, as depicted in Fig.4.2 (red star), corresponding to, $P_i = P_j$.

Step 4 - If a switch in operating mode has been detected in *Step 3*, the maximum likelihood or the Bayesian inference based fault estimation (*Level-2 algorithm*) in Section 4.4.3.1 or Section 4.4.3.2 are executed.

To evaluate the performance of the proposed algorithm, the Fault Detection Rate (*FDR*) is defined as:

$$FDR = n_d/n_{total} \quad (4.22)$$

where n_{total} is the total number of tested samples and n_d is the number of samples that have been correctly classified. The *FCR* is used to discriminate between specific fault classes and the normal process operation as well as between fault classes due to economic considerations.

Multivariate statistical techniques have been often used for fault detection and classification (Chiang, Russell, & Braatz, 2008; Raich & Cinar, 1996). However such algorithms may be less accurate since they do not specifically address the stochastic distribution of the faults as the proposed algorithm. To demonstrate this point a comparison has been conducted between the proposed algorithm and a Principal Component Analysis (PCA) based fault detection method. The results of this comparison are shown in the supplementary materials indicating that for a similar number of data points used for model training the gPC offers considerably better detection performance. The reasons are: i- the gPC method correctly accounts for the nonlinearity by explicitly using the first principle model and ii-the gPC method directly models the stochastic distribution of the fault whereas a larger amount of data will be needed by the PCA method to correctly describe the statistical faults' distributions.

4.4.5 Sensitivity Analysis based Sensor Selection

Appropriate selection of sensors (measured quantities) for enhanced fault detection is essential in the presence of uncertainty. Sensitivity analysis aims to quantify the effect of stochastic faults onto the variability of the

measured variables and provide reliable information about the stochastic faults. This section presents a sensitivity analysis algorithm based on differentiating the gPC model describing the measured variable in eq 4 with respect to the random variables ξ .

For that purpose, the partial derivatives of each of the measured quantities $\mathbf{x} = \{x_j\}$ ($j = 1, 2, \dots, n$) in eq 4 can be calculated with respect to the random variables $\xi = \{\xi_i\}$ ($i = 1, 2, \dots, n_g$). As each measured variable has different units and orders of magnitude, each variable is normalized by the first gPC coefficients, i.e., the mean values of the corresponding variables, and eq 4 can be accordingly re-written as follows:

$$\frac{x_j(t, \xi)}{x_{j,1}(t)} = \frac{x_{j,1}(t)}{x_{j,1}(t)} \Phi_0(\xi) + \frac{1}{x_{j,1}(t)} \sum_{k=1}^P x_{j,k}(t) \Phi_k(\xi) = y_j \quad (4.23)$$

where $x_{j,1}(t)$ denotes the first gPC coefficients of the j^{th} measured variable. The partial derivatives of each measured quantity can be calculated with respect to the n_g random variables, and the sensitivity matrix \mathbf{S} can be formulated as:

$$\mathbf{S} = \begin{pmatrix} \partial y_1 / \partial \xi_1 & \cdots & \partial y_1 / \partial \xi_{n_g} \\ \vdots & \ddots & \vdots \\ \partial y_n / \partial \xi_1 & \cdots & \partial y_n / \partial \xi_{n_g} \end{pmatrix} = \begin{pmatrix} s_{1,1} & \cdots & s_{1,n_g} \\ \vdots & \ddots & \vdots \\ s_{n,1} & \cdots & s_{n,n_g} \end{pmatrix} \quad (4.24)$$

where $s_{j,i}$ represents the sensitivity of the j^{th} measured variable to the i^{th} unknown fault. Although each element in \mathbf{S} can be also formulated over a time horizon, in this work for simplicity it is only evaluated around each of the mean values (operating modes).

4.5 Example: Reactor-Separator Process

Simulation studies of a nonlinear chemical process consisting of two reactors and a separator with recycle unit (Stewart, Venkat, Rawlings, Wright, & Pannocchia, 2010) are used to demonstrate the efficacy of the proposed two-level fault detection and classification algorithm. The proposed algorithm is also compared to a Particle Filter (PF), which is has been proposed as the optimal choice for stochastic faults in dynamic nonlinear systems. Fig.4.3 depicts a schematic of the system with three temperature control loops. A stream of reactant A is added to each reactor and converted to the product B by the first order reaction, C denotes the side-product of the process. The feed mass fraction of reactant A (x_{A0}) is assumed as the unknown (unmeasured) stochastic fault (\mathbf{g}) in this current work. x_{A0} is assumed to change as shown in Fig.4.1, i.e., normally distributed perturbations around three mean values (operating modes) as described in Eq. 4.10. The mathematical model of the process controlled with three *PI* controllers is described by the following set of equations:

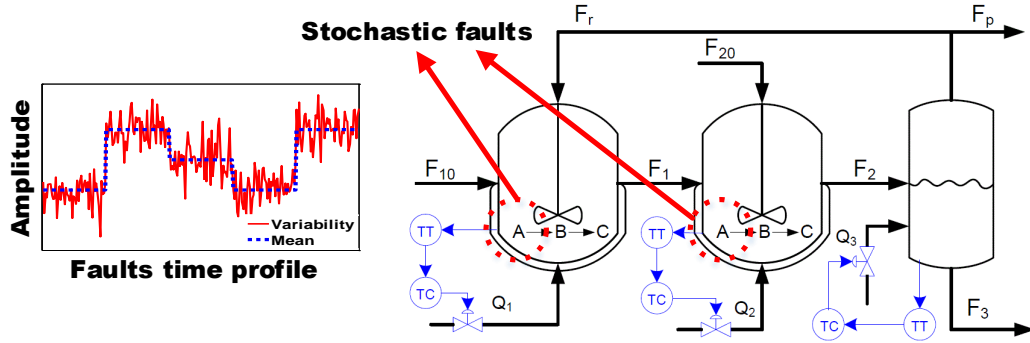


Figure 4.3 Two reactors in series with separator and recycle unit

$$\dot{H}_1 = (1/\rho A_1)(F_{f1} + F_R - F_1) \quad (4.25)$$

$$\dot{x}_{A1} = (1/\rho A_1 H_1)(F_{f1} x_{A0} + F_R x_{AR} - F_1 x_{A1}) - k_{A1} x_{A1} \quad (4.26)$$

$$\dot{x}_{B1} = (1/\rho A_1 H_1)(F_R x_{BR} - F_1 x_{B1}) + k_{A1} x_{A1} - k_{B1} x_{B1} \quad (4.27)$$

$$\dot{T}_1 = (1/\rho A_1 H_1)(F_{f1} T_0 + F_R T_R - F_1 T_1) - (1/C_p)(k_{A1} x_{A1} \Delta H_A + k_{B1} x_{B1} \Delta H_B) + (Q_1/\rho A_1 C_p H_1) \quad (4.28)$$

$$\dot{H}_2 = (1/\rho A_2)(F_{f2} + F_1 - F_2) \quad (4.29)$$

$$\dot{x}_{A2} = (1/\rho A_2 H_2)(F_{f2} x_{A0} + F_1 x_{A1} - F_2 x_{A2}) - k_{A2} x_{A2} \quad (4.30)$$

$$\dot{x}_{B2} = (1/\rho A_2 H_2)(F_1 x_{B1} - F_2 x_{B2}) + k_{A2} x_{A2} - k_{B2} x_{B2} \quad (4.31)$$

$$\dot{T}_2 = (1/\rho A_2 H_2)(F_{f2} T_0 + F_1 T_1 - F_2 T_2) - (1/C_p)(k_{A2} x_{A2} \Delta H_A + k_{B2} x_{B2} \Delta H_B) + (Q_2/\rho A_2 C_p H_2) \quad (4.32)$$

$$\dot{H}_3 = (1/\rho A_3)(F_2 - F_D - F_R - F_3) \quad (4.33)$$

$$\dot{x}_{A3} = (1/\rho A_3 H_3)(F_2 x_{A2} - (F_R + F_D) x_{AR} - F_3 x_{A3}) \quad (4.34)$$

$$\dot{x}_{B3} = (1/\rho A_3 H_3)(F_2 x_{B2} - (F_R + F_D) x_{BR} - F_3 x_{B3}) \quad (4.35)$$

$$\dot{T}_3 = (1/\rho A_3 H_3)(F_2 T_2 - (F_R + F_D) T_R - F_3 T_3) + (Q_3/\rho A_3 C_p H_3) \quad (4.36)$$

where the subscripts 'i' (i.e., 1, 2, 3) refers to the vessel, x_i is the mass fraction of A or B, T_i is temperature, H_i is the level, F_i is the flow rate and the reaction terms are:

$$F_i = k_{vi} H_i \quad (4.37)$$

$$k_{Ai} = k_A \exp(-E_A/RT_i) \quad (4.38)$$

$$k_{Bi} = k_B \exp(-E_B/RT_i) \quad (4.39)$$

The recycle flow and the weight percent factors satisfy:

$$F_D = 0.01 F_R \quad (4.40)$$

$$x_{AR} = \alpha_A x_{A3} / \bar{x}_3 \quad (4.41)$$

$$x_{BR} = \alpha_B x_{B3} / \bar{x}_3 \quad (4.42)$$

$$\bar{x}_3 = \alpha_A x_{A3} + \alpha_B x_{B3} + \alpha_C x_{C3} \quad (4.43)$$

$$x_{C3} = 1 - x_{A3} - x_{B3} \quad (4.44)$$

Each of the tanks in the process receives an external heat input that is determined by a *PI* controller:

$$Q_i(t) = Q_{(ss),i}(t) + K_{p,i}(T_{(set),i} - T_i(t)) + K_{p,i}/\tau_i \int_0^t (T_{(set),i} - T_i(t^*)) dt^* \quad (4.45)$$

The parameters used for the simulation are given in Table 4.1.

Table 4.1 Parameter declaration for the Reactor-Separator process

<i>Symbol</i>	<i>Value</i>	<i>Units</i>	<i>Symbol</i>	<i>Value</i>	<i>Units</i>	<i>Symbol</i>	<i>Value</i>	<i>Units</i>
F_{f1}	10	kg/s	k_{v1}	2.5	kg/m s	ρ	0.15	kg/m ³
F_{f2}	1	kg/s	k_{v2}	2.5	kg/m s	A_1	3	m ²
F_R	60	kg/s	k_{v3}	2.5	kg/m s	A_2	3	m ²
$T_{(set),1}$	315	K	k_A	0.02	1/s	A_3	1	m ²
$T_{(set),2}$	315	K	K_B	0.018	1/s	α_A	3.5	/
$T_{(set),3}$	400	K	E_A/R	-1000	K	α_B	1.1	/
T_0	310	K	E_B/R	-500	K	α_c	0.5	/
T_R	310	K	ΔH_A	-40	kJ/kg	$K_{p,i}$	0.25	/
C_p	2.5	kJ/kg K	ΔH_B	-50	kJ/kg	τ_i	0.0025	/

4.6 Results and Discussion

4.6.1 Model Formulation for the Reactor-Separator Process

The fault detection and classification (diagnosis) problem consists of diagnosing the mean value (operating mode) of the unknown feed mass fraction x_{A0} based on measurements such as $\{Q_i\}$. For simplicity, 3 mean values of the feed mass fraction (x_{A0}) are considered, i.e., 0.65, 0.75 and 0.85 ($k = 3$ in Eq 4.10). Thus, the objective in this work is to (i) rapidly identify the occurrence of any potential switches between different mean values (operating modes), and (ii) classify the operating mode that the process is being operated. Stochastic perturbations in x_{A0} occur around each of these mean values, and they follow a normal distribution with zero mean and a standard deviation of 0.1. Note that for general non-normal distribution, the Askey chaos polynomial basis rather than the original Hermite chaos polynomial basis can be used to improve the convergence rate for the model calibration. Since the solution of the gPC coefficients involved in the gPC expansions of each one of the states (\mathbf{x} in eq 1) as given in Section 4.3 requires the application of Galerkin projections, the employment of gPC is limited to monomial or polynomial terms. Hence, non-polynomial terms such as the reaction term (Arrhenius energy function) k_{Ai} , are approximated by a 2nd order Taylor expansion around each mean value on the input fault x_{A0} . Since the random variable ζ is normally distributed, the corresponding basis polynomial functions for gPC approximations are selected as Hermite as per the Askey scheme to maintain orthogonality (Xiu D. , 2010).

To test the accuracy of the Taylor approximation preliminary simulations are done with the gPC model resulting from this approximation and these are compared to Monte Carlo simulations of the full nonlinear model without approximations. Fig.4.4 shows the simulation results of the controlled variable T_1 in the first reactor, using the gPC method with a 2nd order Taylor expansion to approximate the reaction terms and the Monte Carlo

simulations with the nonlinear model described in Section 4.5, respectively. For the gPC method, the gPC coefficients of the measured quantities \mathbf{x} are calculated as outlined in Section 4.3. Then samples generated for the random variable ξ are substituted into these gPC expressions to predict the measured quantities and to estimate their upper (maximum) and lower (minimum) values at each time instant.

The MC simulations are conducted as follows: (i) A set of samples of the feed mass fraction x_{A0} following the same statistical properties as used for the gPC are generated; (ii) Each of these samples is substituted into the nonlinear model shown in Section 4.5; and (iii) The simulation results of the measured variables are stored for comparison. Several randomly chosen simulated trajectories with the MC simulations are shown in Fig. 4.4. The plot corroborates that the trajectories obtained with MC are bounded by the upper (*Maximum*) and lower (*Minimum*) bounds calculated with the gPC model. Thus, the gPC model with the Taylor approximation of the Arrhenius term provided correct bounds for the MC simulations.

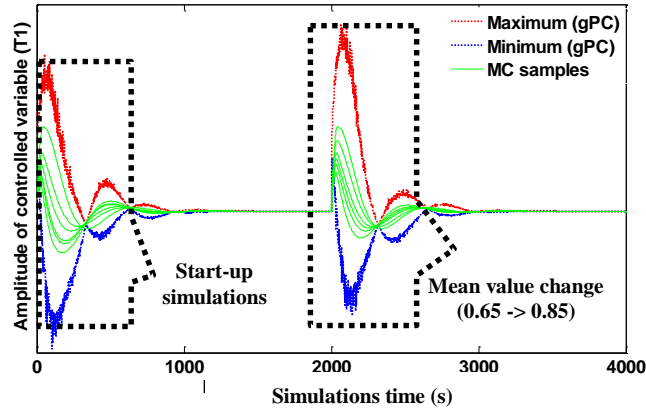


Figure 4.4 Comparisons of the gPC model and MC simulations using controlled variable T_1

4.6.2 Sensor Selection based on Sensitivity Analysis

A sensitivity analysis is conducted as described in Section 4.4.5 for the purpose of sensor selection. For each of the mean values of the feed mass fraction x_{A0} , the sensitivity matrix \mathbf{S} (Table 4.2 ~ Table 4.4) can be calculated for all the states defined by the mechanistic model in Section 4 with respect to the random variable $\xi = \{\zeta\}$. The dimension of the space of the random variables ξ is 1, since only one stochastic fault x_{A0} is considered in this current work.

As seen in Table 4.2 ~ Table 4.4, variations in the feed mass fraction contributes significantly to changes in the mass fractions of A and B in the reactors and separator. Despite its sensitivity however, they are not used for fault detection in the current work, since the objective is to detect faults using measurements that can be easily measured and concentrations are generally expensive to measure on-line. The sensitivity of temperatures $\{T_i\}$ to the variations in the feed are small as expected, since they are controlled variables. Instead, the manipulated

variables $\{Q_i\}$ are more sensitive to the random changes in the feed and consequently they are chosen for inferring the faults.

Table 4.2 Sensitivity analysis of reactor 1

Mean	Measured variables				
	H_1	x_{A1}	x_{B1}	T_1	Q_1
0.65	1.5e-64	0.1044	0.0157	1.6e-7	0.0177
0.75	1.8e-64	0.1044	0.0156	6.3e-7	0.0177
0.85	1.8e-64	0.0879	0.0032	8.9e-7	0.0165

Table 4.3 Sensitivity analysis of reactor 2

Mean	Measured variables				
	H_2	x_{A2}	x_{B2}	T_2	Q_2
0.65	2.5e-62	0.1050	0.0247	5.1e-7	0.0156
0.75	2.5e-62	0.1050	0.0246	1.4e-6	0.0157
0.85	2.5e-62	0.0886	0.0125	1.3e-6	0.0143

Table 4.4 Sensitivity analysis of separator

Mean	Measured variables				
	H_3	x_{A3}	x_{B3}	T_3	Q_3
0.65	2.5e-61	0.2150	0.1233	3.4e-7	0.0056
0.75	2.5e-61	0.2150	0.1232	3.4e-7	0.0011
0.85	2.5e-61	0.1930	0.1044	1.8e-7	0.0012

4.6.3 Level-1 Algorithm with PDF Profiles

In this case study, the model calibration of the PDF profiles of measured quantities for each operation around constant mean values of x_{A0} is studied. Following the above, 3 mean values of the feed mass fraction x_{A0} are studied, *i.e.*, 0.65, 0.75 and 0.85. The stochastic perturbations, added around these mean values (operating modes), are assumed to be normally distributed with zero mean and a standard deviation of 0.1. The step changes follow a ML-PRS signal (Ljung, 1999) as shown in Fig.4.5 (a) and (b), respectively.

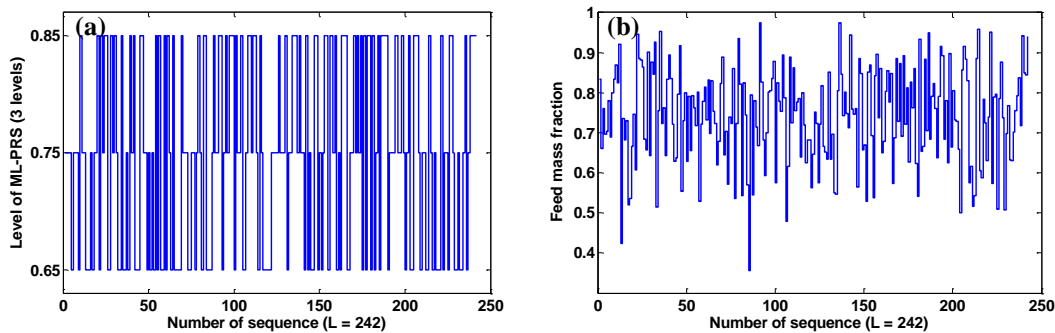


Figure 4.5 Multi-level pseudo random sequence

(a) Three-level-PRS and (b) application to the feed mass fraction superimposed with stochasticity

The number of step changes of the unknown input (x_{A0}) among the 3 selected mean values (operating modes) in the ML-PRS is 242 and the maximum number of measurements between two consecutive step changes in faults (inputs) is limited to 1000. The time interval between two measurements is set to 0.01s, which means that the simulation time between two consecutive step changes is limited to 10 seconds.

Table 4.5 shows the model calibration results with the *Level-1 algorithm* as described in Eq. 4.11 using the measurements of manipulated variables $\{Q_i\}$. To simulate actual data, Gaussian noise is added to the measurements of $\{Q_i\}$. Hermite polynomials are used and the highest order of polynomials used for the gPC models is 2 ($p = 2$ in Eq. 4.6).

Table 4.5 Model calibration result for the *level-1 algorithm*

x_{A0}	x'_{A0}	σ_{A0}	σ_n	time(s)
0.65	0.6370	0.0937	0.0188	992
0.75	0.7364	0.0979	0.0199	788
0.85	0.8319	0.0933	0.0201	871

In Table 4.5, the first column gives the mean values of x_{A0} used for simulations. x'_{A0} and σ_{A0} are the mean and standard deviation calculated from Eq. 4.11, σ_n is the standard deviation of measurement noise. As explained before, the mean and standard deviation of the faults (inputs) resulting from Eq. 4.11, i.e., x'_{A0} , σ_{A0} , are not identical to the actual simulated values used for calibration (x_{A0} , and 0.1), due to the measurement noise and the gPC series' truncation errors. For each operating mode (mean value), the last column shows the required computational time for the model calibration with Eq. 4.11. It should be noted that this calibration step can be performed off-line.

Once the gPC model is constructed, the gPC models and the PDF profiles of the measured variable (Q_i), estimated for each of the 3 mean values of the feed mass fraction (x'_{A0} in Table 4.5), can be obtained. Table 4.6 shows the gPC representations for the measured variables $\{Q_i\}$, where the statistical moment (mean values and standard deviation (*s.d.*)) are calculated with Eq. 4.8 and Eq. 4.9 as a function of the gPC coefficients in Eq. 4.4. Figure 4.6 shows the PDF profiles for the external heat Q_1 in the first reactor, in which the horizontal axis represents the range of Q_1 , and the vertical axis is the normalized probability.

Table 4.6 The gPC model representations for the *level-1 algorithm*

x_{A0}	Q_1		Q_2		Q_3	
	mean	s.d.	mean	s.d.	mean	s.d.
0.65	752.51	13.62	780.02	12.59	566.62	0.375
0.75	733.66	13.19	768.14	12.20	564.66	0.321
0.85	724.08	10.62	759.29	9.83	562.68	0.237

It should be emphasized that the *Level-1 algorithm* that is calibrated for steady states only, has been proposed in this work solely as a preliminary step to assess the possible occurrence of a step change in the fault variable

(feed composition). However, for the purpose of justifying the need for using the additional *Level-2 algorithm*, the efficiency of the *Level-1 algorithm* in detecting an operating mode is tested first.

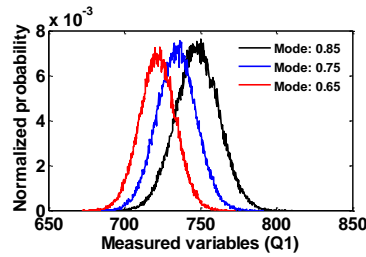


Figure 4.6 The PDF profiles of the measured variable (Q_1) at 3 operating modes

Two cases are considered for these simulations: (i) samples are collected during periods where the system is operating at steady state, and (ii) samples are collected immediately after the occurrence of a step change in the feed mass fraction.

Table 4.7 shows the result of Fault Detection Rate (*FDR*) with different noise levels using the PDF profiles each calculated based on the assumption of constant mean values of x'_{A0} . To comply with the assumption that the system is operated around a fixed mean value with perturbations, the detection efficiency is investigated using the measurements of $\{Q_i\}$ before a switch between means occurred (see inset Fig.4.1 (b)-A). The measurements denote that the system is operating at steady state with constant mean values.

Table 4.7 FDR with the *Level-1 algorithm* (steady state measurements)

x_{A0}	Noise level		
	1%	2%	3%
0.65	93%	90%	91%
0.75	92%	90%	87%
0.85	95%	93%	90%
<i>Average</i>	93%	91%	89%

Table 4.6 is based on 1000 test samples for each mean value on the feed mass fraction, and the averages of the fault detection rates decrease as expected when the noise level increases. It is worth noting that the model calibration as per the optimization in Eq. 4.11 would be time prohibitive if Monte Carlo (MC) simulations were to be used instead of a gPC approximation. For instance, the processor time required for one cost evaluation with MC (5000 samples) is ~ 15465 seconds. The search for the optimum in Eq. 4.11 for each mean value requires 40~60 iterations and takes approximately 171 ~ 257 hours on average. However, the proposed method takes ~ 15 minutes to calculate the optimum in Eq. 4.11 for all mean values, as can be seen in Table 4.5. Also, the use of 5000 samples for calibrating the PDF profiles of measured variables from MC simulations resulted in lower fault detection rates, as compared to the gPC method. Thus, a larger number of samples than 5000 would be required to obtain comparable fault detection rate as with the gPC approach, which would further increase the computational burden.

As mentioned above, the *Level-1 algorithm* is only suitable when the system is operating for long periods around a fixed mean value. Thus, it is expected to be less accurate during periods where changes between mean values occur. To demonstrate this point, the fault detection rate is studied with the *Level-1 algorithm* using measurements collected during the transition periods, i.e., immediately after the occurrence of a step change in the mean values of x_{A0} (see inset Fig.4.1 (b)-B).

A moving time window of 50 measurements of $\{Q_i\}$ is used for the tests and the fault detection rate is evaluated based on the average of the probabilities of these 50 measurements with respect to the PDF profiles generated in the *Level-1 algorithm*. For instance, each of these measurements inside the moving time window is separately referred to the PDF profiles of measured variables and the operating mode can be inferred from its maximum probability. The final fault detection result is based on the largest number of times a particular operating mode is detected within the moving time window. For each of the mean values of feed mass fraction x_{A0} , there are 1000 test samples in the ML-PRS and the fault detection rate between mean values on average is found to be as low as ~61%. This result justifies the necessity for the use of the *Level-2 algorithm* that does not assume operation around a steady state as in the *Level-1 algorithm*. In summary, the *Level-1 algorithm* is proposed only to evaluate the necessity for executing the *Level-2 algorithm* so as to avoid executing the *Level-2 algorithm* too frequently which would require excessive computational effort.

4.6.4 FDD with Level-2 Algorithm using Maximum Likelihood

The *Level-2 algorithm* is only executed after the *Level-1 algorithm* has indicated the occurrence of a change in the mean value of fault x_{A0} . Table 4.8 shows the detection rates obtained with the likelihood function based *Level-2 algorithm* for three case studies to evaluate the efficacy and computational time. In the first case study, a time moving window of 50 measurements of $\{Q_i\}$ ($m = 50$ in Eq. 4.16) is used to compare the results obtained by the *Level-1 algorithm* with the same time window. Only the average value of x_{A0} over the moving time windows is chosen as a decision variable for $\lambda_{level-2}$ in eq 4.16. For the other two case studies, 100 measurements of $\{Q_i\}$ are used ($m = 100$ in Eq. 4.16). The decision variable of the second case study in Eq. 4.16 is the average value of x_{A0} , while in the third case study both the average value and confidence interval of x_{A0} are optimized.

Table 4.8 FDR with the maximum likelihood based *Level-2 algorithm*

<i>Case studies</i>	<i>FDR</i>	<i>Time(s)</i>
1 ($m = 50$)	71%	225
2 ($m = 100$)	85%	498
3 ($m = 100$)	80%	1133

It can be seen that the *Level-2 algorithm* shows significantly better fault diagnosis performance, as compared to the *Level-1 algorithm* alone. For instance, the fault detection rate is ~71% for the first case study, which has been increased by ~10 percent point compared with the *Level-1 algorithm* (~61%) for the same number of measurements, thus confirming the necessity for the *Level-2 algorithm* to detect transitions among mean values

of x_{A0} . The third column shows the required computational time for each moving time window with different measurements. As seen, there is a trade-off between the classification accuracy and computational time.

4.6.5 FDD with Level-2 Algorithm using Bayesian Inference

Figure 4.7 shows a segment of the ML-PRS used for fault detection involving five consecutive step changes on the feed mass fraction x_{A0} , using the Bayesian inference based *Level-2 algorithm*. The *bi-level optimization* is conducted consecutively in the Bayesian inference based *Level-2 algorithm* to recursively and simultaneously estimate the mean and variance of a fault in x_{A0} . Figure 4.7 (a) shows five consecutive step changes of the stochastic fault around each of the three mean values.

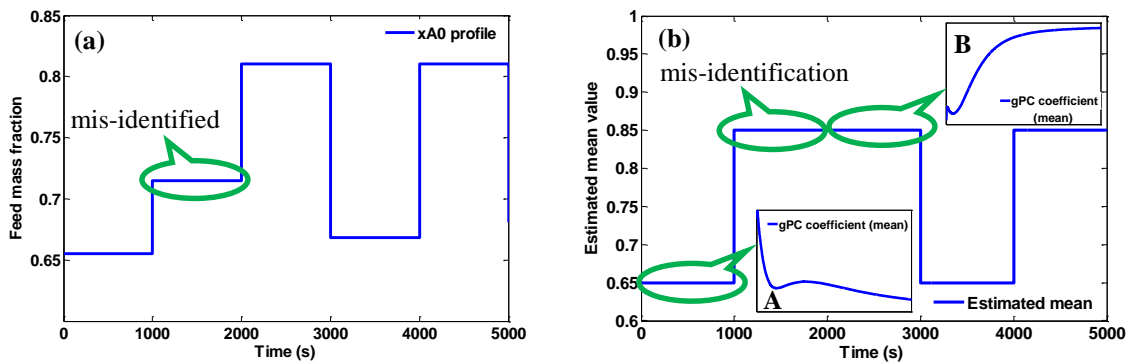


Figure 4.7 Illustration of Bayesian inference estimation based fault detection

Using measurements over a moving time window, the optimization of Eq. 4.21 is conducted for each of the gPC models generated with the three different mean values considered in this study for the feed mass fraction x_{A0} . The maximum value of Eq. 4.21, representing the likelihood of operating around a particular mean value, is calculated at each time interval and used for detecting the operating mode. Fig.4.7 (b) displays the estimated mean value of the fault using the first posterior moment (gPC coefficients) of feed mass fraction x_{A0} . For Fig.4.7 (b), the two insets, i.e., A and B, show the optimized posterior gPC coefficients (first posterior moment) of feed mass fraction at each time instant while optimizing Eq. 4.21 based on the measurements collected over a moving time window. The estimation of the higher order gPC coefficients can be updated using the first posterior moment and the same set of measurements collected over the moving time window.

As seen, the Bayesian inference based *Level-2 algorithm* can provide correct estimation and identification results. However, there is one misclassification of the mean value, which is highlighted in Fig.4.7 (b). The value of fault after the step change is ~ 0.72 in Fig.4.7 (a), but it has been misclassified as 0.85. To further investigate the fault detection performance using the Bayesian inference based *Level-2 algorithm*, Table 8 shows the fault detection rates and the corresponding computational time required for optimizing Eq. 4.21 at each time instant.

Table 4.9 FDR with the Bayesian inference based *Level-2 algorithm*

<i>Case studies</i>	<i>FDR</i>	<i>Time(s)</i>
1 ($m = 50$)	65.3%	~1
2 ($m = 100$)	70.2%	~1
4 ($m = 1000$)	81.4%	~3

As seen in Table 4.9, three cases are studied where the fault detection is based on the gPC coefficients estimated with the newest (last) measurement inside the moving time window. For the first two case studies, only the first posterior coefficient is chosen as decision variable and updated at each time instant. In the first case, the fault detection rate is evaluated with a moving time window of 50 measurements of $\{Q_i\}$, in order to compare the results with the *Level-1 algorithm* that used the same window length, while 100 measurements are used for the second case study. The decision variable of the third case study optimizes all the gPC coefficients at each time instant.

For testing the fault detection efficiency, 242 consecutive steps changes have been considered that follow a ML-PRS between the 3 mean values (operating modes) and the maximum number of measurements between two consecutive step changes in fault (input) is limited to 1000. As seen, an average of ~3 seconds is required at each time instant if all the posterior moments (gPC coefficients) are optimized simultaneously, which is slightly slower than the estimation if only the first gPC coefficient is updated. It can be also observed that the Bayesian inference *Level-2 algorithm* shows better fault detection rate, as compared to the *Level-1 algorithm* alone. For instance, the fault detection rate is ~65.3% for the first case study, which is an increase of ~5 percent point, as compared with the *Level-1 algorithm* for the same number of measurements. As compared with the maximum likelihood based *Level-2 algorithm*, the Bayesian inference based estimator performs much faster, but it results in a less accurate fault detection rate.

4.6.6 FDD Using Solely Level-2 Algorithm

In previous case studies, the *Level-2 algorithm* is triggered by the *Level-1 algorithm*, only if a potential change in the operating mode has been detected. In this section, the *Level-2 algorithm* is continuously applied by itself and the detection performance is compared to the approach that combines the *Level-1* and *Level-2 algorithms*.

The maximum likelihood based *Level-2 algorithm* is implemented for two consecutive step changes on the feed mass fraction x_{A0} as shown in Fig.4.8 (a), in which the maximum number of measurements between these step changes is limited to 1000. Fig.4.8 (b) and (d) show the simulated noise free external heat Q_I of the first reactor, corresponding to the ML-PRS in Fig.4.8 (a). To simulate actual data, Gaussian noise is added to the measurements of $\{Q_i\}$. Fig.4.8 (c) gives the point estimate with a moving time window of 100 measurements.

In Fig.4.8 (c), the first 10 point estimates are the results obtained using measurements of $\{Q_i\}$ before a step change, i.e., these measurements inside the moving time window represent a case where the system is operating at steady state with fixed mean values. The estimate of the fault (feed mass fraction x_{A0}) reaches ~0.62, which is

close to the simulated value ~ 0.64 as shown in Fig.4.8 (a). Using the maximum likelihood based *Level-2 algorithm*, the estimate of the fault starts to rise after a step change on the feed mass fraction that has been introduced at time 5000 seconds. The optimization of Eq. 4.16 is conducted at each of the time instants and the point estimate eventually reaches a plateau. The point estimate, however, is overestimated as seen in Figure 4.8 (c), stabilizing at ~ 0.99 . It should be remembered that when the *Level-2 algorithm* is applied on its own, the estimation of the feed mass fraction becomes an unsupervised learning problem. In such a case, the upper and lower bounds approximated with the gPC model as shown in Fig.4.4 itself cannot accurately predict the step change.

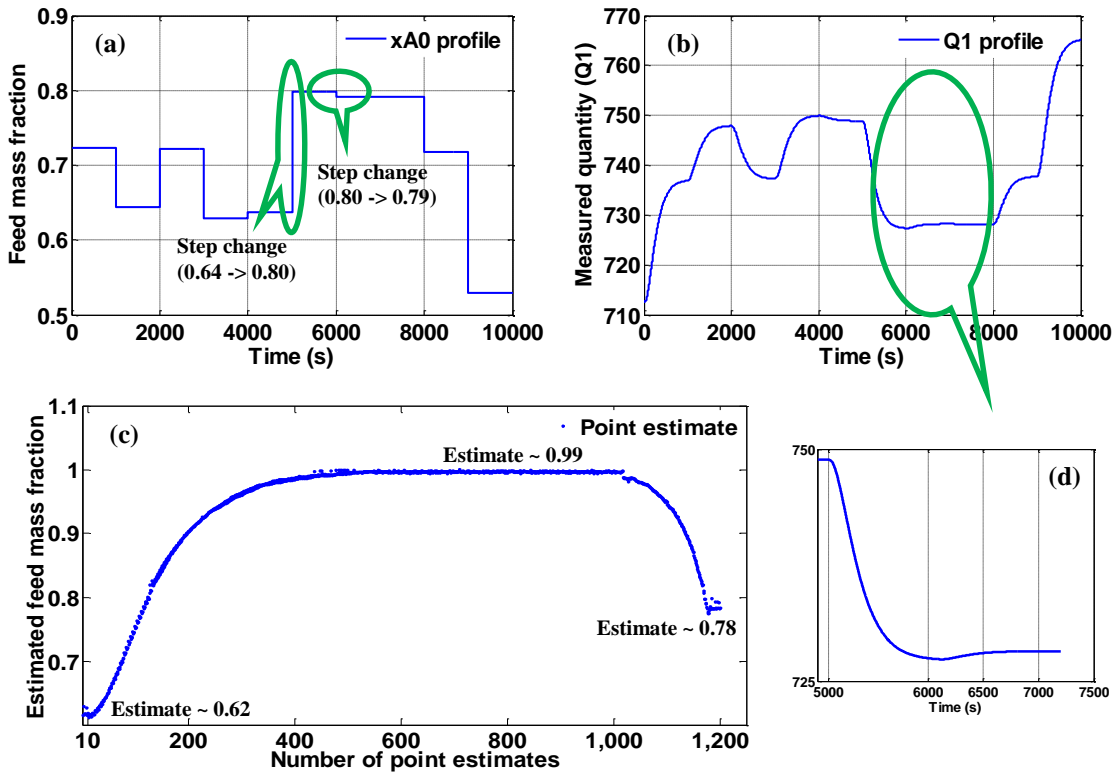


Figure 4.8 Illustration of Maximum likelihood based fault estimator

Additional studies are conducted to investigate the processor time with the *Level-2 algorithm* alone. For instance, ~ 400 point estimates are required for the step change indicated by a circle in Fig.4.8 (a), in which the feed mass fraction x_{A0} has been changed from ~ 0.64 to ~ 0.80 . By contrast, it takes ~ 200 point estimates to stabilize ~ 0.78 for the second step change, due to the relatively smaller change on the feed mass fraction. As discussed above, ~ 500 seconds are required for each of these point estimates as shown in Table 4.7. Thus, it is clear that the two-level fault detection algorithm improves the computational time, if the *Level-2 algorithm* is only executed once the *Level-1 algorithm* has detected a possible step change. An additional advantage of the two-level

algorithm is that the estimation problem is of supervised learning type thus identifying the accurate values of the step changes.

4.6.7 Application of the Two-level Algorithm to Detect An Incipient Fault

This case study illustrates the application of the proposed two-level fault classification methodology to a slow developing (incipient) fault (Isermann R. , 2005). As seen in Figure 4.9 (a), a ramp-like fault is simulated, which is characterized by a slowing increase on the feed mass fraction x_{A0} . Figure 4.9 (b) shows the simulated external heat Q_1 at each time interval during this transition region, which can be used for the detection and classification of faults. The dotted red line in Figure 4.9 (b) represents the simulated noise-free measurements, while the dots denotes the measurements collected at each time instants that are corrupted with 1% measurement noise. For simplicity, Figure 4.9 (c) shows two PDFs profiles of the measured external heat Q_1 operating around two mean values (operating modes), which are obtained using the *Level-1 algorithm*.

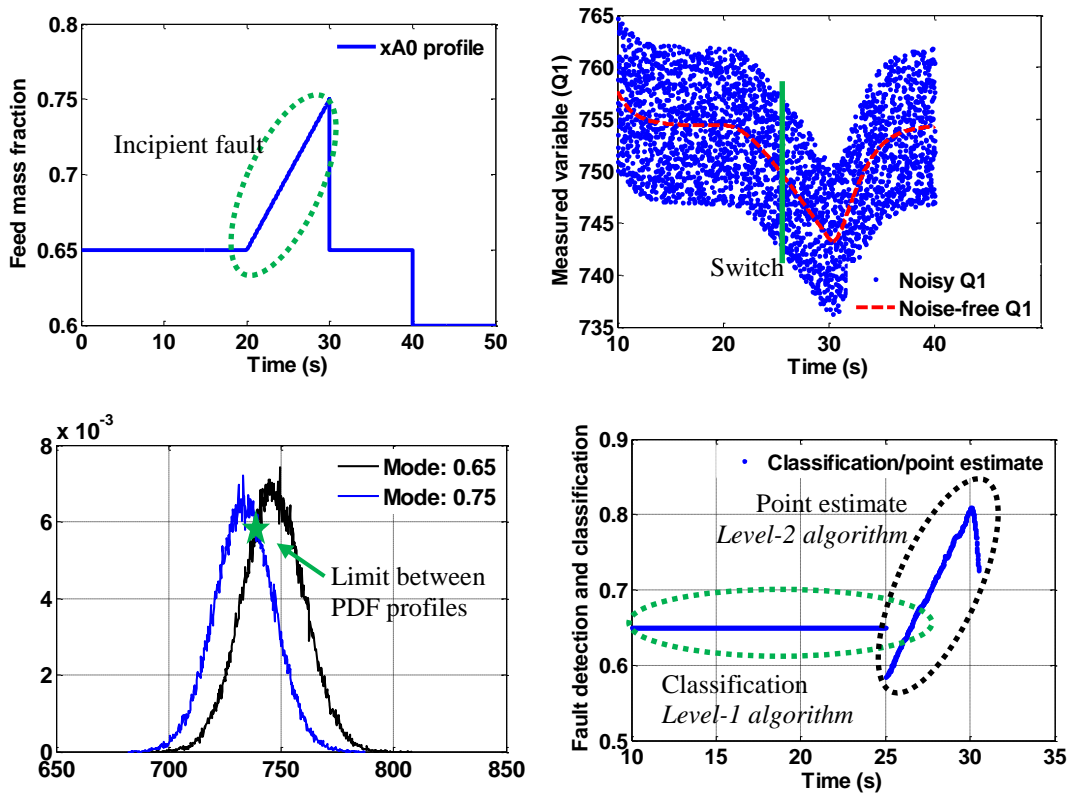


Figure 4.9 Application of the two-level algorithm to an incipient fault

To apply the *two-level algorithm*, each of the measurements collected during the transient is referred to the PDFs profile, and the maximum probability is used to infer a particular mean value that the system is being operated. When a switch in operating mode has been detected, i.e., the measurement value crosses the limit between two adjacent PDF profiles as shown by the star in Figure 4.9 (c) causing the *Level-2 algorithm* to be

launched. Figure 4.9 (d) shows the fault detection and classification results. Before the switch has been detected, the operating mode is determined by the *Level-1 algorithm*. When a switch was detected, the *Level-2 algorithm* is launched. The inference of operating mode can be classified using the estimation result based on a minimum distance criterion to the set of mean values that serve for calibration of the *Level-1 algorithm*. For this case study, a moving time window with 50 measurements is used for the Maximum Likelihood based *Level-2 algorithm*. As can be seen, the estimate of the fault starts to rise after the switch has been detected. It should be noted that when the *Level-2 algorithm* is applied alone, the estimation of the feed mass fraction becomes an unsupervised learning problem. Thus, there is a time delay associated with the correct classification of faults, which may be improved with an increase of measurements used within the moving time windows. An additional study is conducted where a step change occurs immediately after the incipient fault, for which the feed mass fraction has been changed back to 0.65. As seen, the *Level-2 algorithm* based estimation responds faster to this step change.

4.6.8 Comparison Studies to Particle Filter based Fault Detection

Finally, comparison studies are conducted between the proposed two-level algorithm and a particle filter (PF) (Arulampalam, Maskell, Gordon, & Clapp, 2002) for the transient measurements based fault detection. The PF based algorithm can be described as per the following steps. (1) Assume a set of prior particles at each time instant t_k , run each of these particles through the reactor-separator model up to time t_k and update the model to make a new set of transitioned particles. (2) Simulate and update the observations of measured quantities for each of new particles. (3) Calculate the weights for each particle, which are based on the likelihood function between the given observations for a particle and the measurements collected from the process. The Gaussian kernel function can be used to describe the probability distribution of errors. (4) Normalize these weights to formulate a probability distribution. (5) Generate new particles from the new distribution and simulate the observations using particles with larger weights on average. (6) Repeat Step 1~5 until the decision variable is stabilized. The noise for the PF algorithm is assumed to be equal to the one used for the gPC based *Level-2 algorithm*. All the methods are executed on a 2.66 GHz Intel^(R) Core Duo processor.

Fig.4.10 shows the dynamic value and posterior standard deviation (*s.d.*) calculated by the PF for one of the tested input values, i.e., $x_{A0} \approx 0.9130$, where three different initial states (conditions) and 100 particles are used. The legend in Fig.4.9 represents the three conditions used for the PF simulations. The PF is executed for a duration of $t = 50$ seconds to perform a fair comparison with the gPC model based algorithm that uses a time moving window of 50 measurements.

As seen in Fig.4.10, three sets of the initial state in PF provide very different posterior estimates on the average value of x_{A0} . For example, the PF method stabilizes at ~ 0.6358 with an initial state 0.65, which would indicate that the closest feed concentration, out of the 3 mean values considered in the case study, is 0.65 whereas the actual input value is 0.9130.

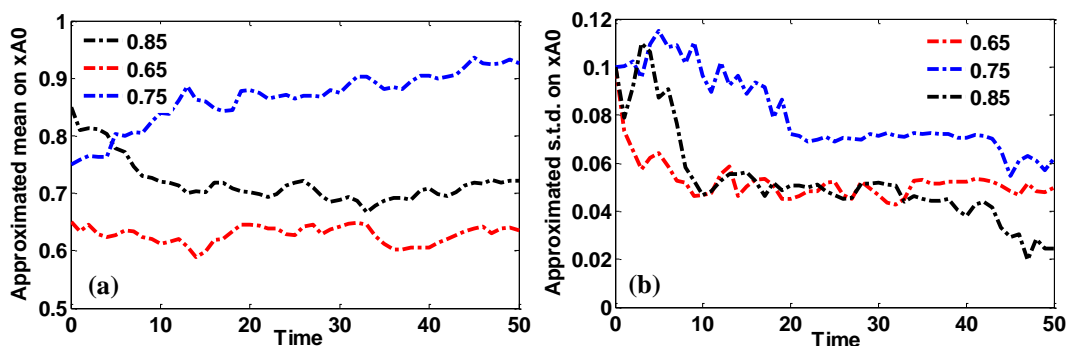


Figure 4.10 Posterior estimation of x_{A0} with Particle Filter (PF)

Additional studies are conducted to investigate the processor time with the PF. As done for the gPC method, 50 and 100 measurements are tested, respectively. The PF requires ~ 6800 seconds to run for 50 time intervals and ~ 13780 seconds for 100 time intervals. As shown in Table 4.7, the gPC based *Level-2 algorithm* requires significantly less computational effort, while it is not sensitive to the user choice of initial guesses as the PF. In order to reduce the sensitivity to initial guesses one can execute the PF algorithm for a larger set of initial guesses and then average the results. However based on the computation times discussed above, such approach will be prohibitive especially for real-time operation.

4.7 Conclusion

This chapter proposes a two-level fault detection and diagnosis approach for faults of a stochastic nature, by combining generalized Polynomial Chaos (gPC) theory with the maximum likelihood method and Bayesian inference, respectively. We propose an algorithm that comprises two levels: Level-1 based on steady state information and Level-2 based on dynamic information. The *Level-2 algorithm* is executed only when the *Level-1 algorithm* indicates the possibility of a step change. This significantly reduces the computational time as compared to using the *Level-2 algorithm* on its own. The proposed method is demonstrated using a simulation of a nonlinear multimode chemical plant with two continuously stirred tank reactors and a flash tank separator. The results show that the proposed methodology is computationally more efficient as compared to simulation based approaches such as Monte Carlo (MC) simulations and Particle filter (PF) and it is not sensitive to the user selected tuning parameters such as the PF.

4.8 Supplementary Material

The objective of this case study is to compare the gPC model based fault detection and classification method with the empirical model based methods for process monitoring. The principal component analysis (PCA) is used for comparison.

One of the most standard methods consists of constructing a single PCA model and defines regions in the lower dimensional space which classify whether a particular fault has occurred. Let us assume the matrix X is used to store measurements for all operating modes (mean values), and then the sample covariance matrix S can be calculated as:

$$S = \frac{1}{n-1} X^T X = VAV^T \quad (\text{S.1})$$

where the diagonal matrix A contains the nonnegative real eigenvalues of decreasing magnitude. The matrix A can be used to optimally capture the variations of the data in X , and the loading vectors P corresponding to the first a largest singular values can be then calculated.

Using the sample covariance matrix S and the loading vectors P , the maximum score discriminant ^[24] can be used to estimate the likelihood that an observation \mathbf{x} is the operating mode i , which can be calculated as:

$$f_i(\mathbf{x}) = \frac{1}{2}(\mathbf{x} - \bar{\mathbf{x}}_i)^T P(P^T S_i P)^{-1} P^T (\mathbf{x} - \bar{\mathbf{x}}_i) + \ln(p_i) - \frac{1}{2} \ln[\det(P^T S_i P)] \quad (\text{S.2})$$

$$\bar{\mathbf{x}}_i = \frac{1}{n_i} \sum_{\mathbf{x}_j \in \chi_i} \mathbf{x}_j \quad (\text{S.3})$$

where $\bar{\mathbf{x}}_i$ is the mean vector for operating mode i , n_i is the number of measurements in operating mode i , χ_i is the set of vectors \mathbf{x}_j which belong to the operating mode i , and S_i is the sample covariance matrix for operating mode i .

The score discriminant can also be used for multiple PCA models. Assuming the PCA models retain the important variations in discriminating between the faults (operating modes), and observations \mathbf{x} is classified as being in the operating mode i with the maximum score discriminant:

$$f_i(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T P_i \sum_{a,i}^{-2} P_i^T \mathbf{x} + \ln(p_i) - \frac{1}{2} \ln[\det(\sum_{a,i}^{-2})] \quad (\text{S.4})$$

where P_i is the loading matrix for the operating mode i , $\sum_{a,i}$ is the diagonal matrix for the operating mode i , and p_i is the overall likelihood of the operating mode i .

For comparison, the fault detection and classification algorithms defined in Eq S.3 and Eq S.4 are compared with the *Level-1 algorithm* when the system is operating at steady states. For the model calibration with eq S.3 and eq S.4, 100 measurements for each operating mode are used, while ~81 measurements for each operating mode are used for the gPC model calibration with Eq 4.11. The number of step changes of the unknown input (x_{A0}) among the 3 mean values in the *ML-PRS* is 300 for the model calibration with PCA. Thus a slightly larger number of measurements were selected for the calibration of the PCA algorithm as compared to our proposed gPC approach.

Three scenarios are considered: (i) measurements collected in the absence of measurement noise and variation on the feed mass fraction x_{A0} ; (ii) measurements collected with measurement noise but no stochastic variation on x_{A0} ; and (iii) both measurement noise and uncertainty on x_{A0} are considered. Table S.1 shows the result of Fault Classification Rate (*FCR*) for these three scenarios.

Table S.1 *FCR with PCA model (steady state measurements)*

x_{A0}	S.3			S.4		
	Case i	Case ii	Case iii	Case i	Case ii	Case iii
0.65	0.99	0.98	0.83	0.99	0.99	0.88
0.75	1	0.85	0.72	1	0.88	0.76
0.85	1	0.93	0.85	0.99	0.90	0.84
<i>Average</i>	0.997	0.92	0.80	0.993	0.923	0.827

In Table S.1, the variation on x_{A0} follows the same assumption as done for the gPC model and 1% measurement noise is used for simulations. To comply with the assumption that the system is operated around a fixed mean value with perturbations, the classification efficiency is investigated using the measured quantities before a switch between means occurred (see inset Figure 1 (b)-A). The measurements denote that the system is operating at steady state with constant mean values. It can be seen that the variation on x_{A0} and the measurement noise show strong influence on the classification of faults. As compared to the results in Table 7, the *FCR* is ~10 percent points lower than the gPC model based *Level-1 algorithm*. An explanation for the difference is that the principal component analysis (PCA) is a linear dimensionality reduction method. When the data components have nonlinear dependencies, PCA may require a larger dimensional representation than would be found by a nonlinear technique. Additionally, comparing Case-ii to Case-iii, the classification rate decreased by ~10 percent points, when the uncertainty on feed mass fraction x_{A0} is considered. One may argue that extra data is required for the model calibration with the PCA method to increase accuracy. The use of more training measurements may improve the classification rate but would increase the computational burden. The proposed gPC based method both addresses the nonlinearity by explicitly using a nonlinear model and necessitates less data, since it directly predict PDF profiles of the variables used for detection.

Chapter 5

Integration of Fault Diagnosis and Process Control

(Adopted from Du et al., 2015, *Journal of Process Control*, vol. 38, p. 42~53, 2016)

5.1 Overview

This chapter presents a novel methodology for simultaneous optimal tuning of a fault detection and diagnosis (FDD) algorithm and a feedback controller for a chemical plant in the presence of stochastic parametric faults. The key idea is to propagate the effect of time invariant stochastic uncertainties onto the measured variables by using a Generalized Polynomial Chaos (gPC) expansion and the nonlinear first principles model of the process. A bi-level optimization is proposed for achieving a trade-off between the fault detectability and the closed loop process variability. The goal of the *outer level optimization* is to seek a trade-off between the efficiency of detecting a fault and the closed loop performance, while the *inner level optimization* is designed to optimally calibrate the FDD algorithm. The proposed method is illustrated by a continuous stirred tank reactor (CSTR) system with a fault consisting of stochastic and intermittent variations in the inlet concentration. Beyond achieving improved trade-offs between fault detectability and control, it is shown that the computational cost of the gPC model based method is significantly lower than the Monte Carlo type sampling based approaches, thus demonstrating the potential of the gPC method for dealing with large problems and real-time applications.

5.2 Introduction

Equipment failures and abnormalities defined as faults are a major source of economic loss and safety hazards in many industries thus creating a need for fault detection and diagnosis algorithms. Most fault detection and diagnosis (FDD) systems are implemented at a supervisory hierarchical level above the control systems level and use measured variables that are also used for feedback control. While there is a large body of literature on FDD, the issue of integration between control and fault diagnosis algorithms has not been addressed as much in particular in the presence of stochastic faults.

A key challenge for integrating control and FDD is that they often have competing objectives. For instance, if the controlled variables are to be used for detection, better control means that the controlled variable deviates little from the set point, while FDD requires sufficiently large deviations for effective detection purposes (Davoodi, Golabi, Talebi, & Momeni, 2013; Meng & Yang, 2014). Similar trade-offs occur also when the manipulated variables are used since good detection generally translate into large control actions as shown in this work. Moreover, process disturbances, nonlinearity and model error make the integration of FDD with control a challenging task (Paulson, Raimondo, Findeisen, Braatz, & Streif, 2014). Several methods have been proposed for optimal simultaneous tuning of a FDD algorithm and a controller based on robust norms. To synthesize the controller and diagnosis algorithms, a four parameter controller setup as a generalization of the two degrees of

freedom controllers was proposed (Jacobson & Nett, 1991; Tyler & Morari, 1994). This method, however, did not explicitly address the cost of unobservable faults and their stochastic nature.

To improve the fault detectability in the presence of bounded uncertainties, set-based (separating inputs) FDD techniques have been used for active fault diagnosis (Scott J. , Findeisen, Braatz, & Raimondo, 2013; Raimondo, Marseglia, Braatz, & Scott, 2013). These methods inject auxiliary signals into the system to enhance the detectability of faults. Instead of introducing an auxiliary signal in the current study, the controller is synthesized together with the fault detection algorithm.

Following the above, the current work addresses the problem of optimal simultaneous tuning of a FDD algorithm and the controller's parameters in the presence of time varying stochastic intermittent parametric faults, where the FDD is based on a nonlinear first principle model. The proposed approach seeks a trade-off between the fault detectability and the closed loop performance. Since the stochastic parametric faults (inputs) are considered, it is necessary to quantify the effect of these inputs on both the variables used in feedback control and for fault detection. One option to do such propagation and quantification is by Monte Carlo (MC) type sampling based simulations, which are computationally demanding since they require a large number of simulations of the nonlinear process model to get accurate results. Computational efficiency is critical in the current problem, since the propagation of the stochastic faults on other variables of interest has to be performed repetitively within the optimization algorithms used to achieve a trade-off between detection and control. Uncertainty analysis and propagation using the generalized Polynomial Chaos (gPC) expansion has been studied by a number of authors in different areas, and has been reported to be more efficient as compared to MC simulations (Du, Duever, & Budman, 2015; Nagy & Braatz, 2007; Patz & Preusser, 2012). The advantage of gPC is that it can propagate a complex probability distribution into a variable of interest and explicitly calculate the statistics of the resulting outputs by analytical formulae (Ghanem & Spanos, 1991; Xiu D. , 2010).

The current work investigates the problem of optimal simultaneous tuning of a FDD algorithm and a controller in the presence of stochastic time varying disturbances by using the gPC expansions for stochastic parametric faults (inputs) and measured output variables. A bi-level optimization algorithm proposed in this work balances the fault detectability and the closed loop control performance. In both the works by Mesbah et al. (Mesbah, Streif, Rindeisen, & Braatz, 2014) and our previous work (Du, Budman, & Duever, 2014) presented at the same meeting, the PDF profiles generated with the gPC models were utilized to enhance the fault detectability by minimizing the overlap between the PDF profiles. Unlike the referenced work (Mesbah, Streif, Rindeisen, & Braatz, 2014), the previous study done by the authors (Du, Budman, & Duever, 2014) and the current work synthesize the fault detection algorithm together with the controller to seek an optimal trade-off between detection and control. Also, the current work differs from previous studies in the proposed fault detection algorithm that it is based on a maximum likelihood criterion to detect the fault using a gPC model. Preliminary results of seeking a trade-off between the fault detectability and the closed loop control performance were outlined in (Du, Budman,

& Duever, 2014). A significant reduction in computational effort was observed by using the gPC method, as compared with the MC sampling based approaches, which is further investigated in this work. Also, the earlier work by the authors (Du, Budman, & Duever, 2014) is extended by combining the gPC theory with the maximum likelihood based estimation to recursively estimate the stochastic parametric faults (inputs) during transients, while in our previous study only the steady state fault detection problem was considered. The application of the gPC model with maximum likelihood dynamically estimates the value of the stochastic fault over a time moving window. The estimation results can be used as a real-time process monitoring strategy for detection of stochastic faults in nonlinear systems. While previously reported parameter estimation approaches based on combinations of the gPC with Bayesian and maximum likelihood have been applied in an offline fashion, the current work proposes a gPC based methodology for online detection of faults.

To summarize, the novel contributions in this current work are: (i) The use, in the context of integration between fault diagnosis and control, of an intrusive gPC approach for uncertainty propagation and quantification by substituting the gPC directly into the first principles nonlinear model of the system; (ii) The use of the maximum likelihood based estimation in combination with the gPC model for fault detection; and (iii) The formulation of a bi-level optimization for achieving an optimal tradeoff between control and improved fault detection. The methodology is specifically targeted to: (i) Balance the control performance and the fault detectability, by synthesizing a FDD algorithm that is operated together with a feedback controller; and (ii) Diagnose the stochastic faults consisting of uncertainties around mean values that change intermittently, using measurements collected immediately after the occurrence of a step change on the mean values of the faults.

This paper is organized as follows. Section 5.3 presents the background and the principal methodologies used in this work. The optimization problems formulated for simultaneously tuning the FDD algorithm and the controller are given in Section 5.4. The presentation of the maximum likelihood based FDD algorithm is also presented in Section 5.4. An endothermic continuous stirred tank reactor (CSTR) is introduced as a case study in Section 5.5. Analysis and discussion of the results are presented in Section 5.6 followed by conclusions in Section 5.7.

5.3 Theoretical Background

The generalized polynomial chaos (gPC) expansion (Xiu D. , 2010) represents an arbitrary continuous random variable of interest as a polynomial series of another random variable with a given standard distribution. Assume a set of nonlinear ordinary differential equations (ODEs) describe the dynamic behaviour of a system:

$$\begin{aligned} \dot{x} &= f(t, x, u; g) \\ 0 \leq t \leq t_f, x(0) &= x_0 \end{aligned} \tag{5.1}$$

where the vector $x \in R^n$ contains the system states (measured variables) with initial conditions $x_0 \in R^n$ over time domain $[0, t_f]$, and u denotes the known inputs of the system. The vector $g \in R^{n_g}$ is the unknown stochastic time

varying input. Note that this work assumes that the input vector g contains the stochastic parametric faults of interest. The $\dot{\cdot}$ notation over x signifies the derivative with respect to time t . The function f is assumed to be the first principle model of the process. To quantify the effect of stochastic inputs (faults) g on the different measured variables, the gPC expansion can be employed. To that purpose each unknown input g_i ($i = 1, 2, \dots, n_g$) in g is represented as a function of a set of random variables $\zeta = \{\zeta_i\}$:

$$g_i = g_i(\zeta_i) \quad (5.2)$$

where ζ_i is the i^{th} random variable. The random variables ($\zeta = \{\zeta_i\}$) are assumed to be independent and identically distributed. Following the gPC expansion, the unknown stochastic faults (inputs) $g(\zeta)$ and system states $x(t, \zeta)$ are described in terms of orthogonal polynomial basis functions $\Phi_k(\zeta)$:

$$g(\zeta) = \sum_{k=0}^{\infty} g_k \Phi_k(\zeta) \quad (5.3)$$

$$x(t, \zeta) = \sum_{k=0}^{\infty} x_k(t) \Phi_k(\zeta) \quad (5.4)$$

where x_k and g_k are the gPC coefficients of measured variables (states) and faults at each time instant t , $\Phi_k(\zeta)$ are multi-dimensional orthogonal basis functions of ζ in the gPC theory. If the input (g) can be measured or estimated, the coefficients of the unknown input, g_k , can be calculated such that Eq. 5.3 follows an a priori measured statistical distribution. Then, the gPCs representing the measured quantities (states) resulting from this random input can be calculated using a model of the process combined with a Galerkin projection procedure (Xiu D. , 2010). By Galerkin projection it is possible to compute the expansion coefficients $\{x_k(t)\}$ by projecting Eq. 5.1 onto each one of the polynomial chaos basis functions $\{\Phi_k(\zeta)\}$ as described in Eq. 5.5:

$$\langle \dot{x}(t, \zeta), \Phi_k(\zeta) \rangle = \langle f(t, x(t, \zeta), u(t), g(\zeta)), \Phi_k(\zeta) \rangle \quad (5.5)$$

For practical application, Eqs. 5.3 and 5.4 are often truncated to a finite number of terms, i.e., P . Hence, the total number of terms in Eq. 5.5 is a function of an arbitrary order p in Eq. 5.3 that is necessary to represent an a priori known distribution of g and the number (n_g) of different faults (inputs) in vector g as follows:

$$P = ((n_g + p)! / (n_g! p!)) - 1 \quad (5.6)$$

From Eq. 5.6, the number of the gPC expansion terms for the measured variables in Eq. 5.4 increases as the polynomial order p in Eq. 5.3 and/or the number of unknown inputs n_g in Eq. 5.2 increase. The inner product in Eq. 5.5 between two vectors is defined by:

$$\langle \psi(\zeta), \psi'(\zeta) \rangle = \int \psi(\zeta) \psi'(\zeta) W(\zeta) d\zeta \quad (5.7)$$

where the integration is conducted over the entire event domain generated by the random variables ζ , and $W(\zeta)$ is the weighting function, which is the probability function of random variables and has to be chosen with respect to the polynomial basis function used to represent ζ so as the result of Eq. 5.7 is one or zero. To obtain

orthogonality the basis functions have to be selected according to the choice of the distribution of ζ . For example, Hermite polynomials are chosen as basis functions for normally distributed ζ and Laguerre polynomials are used for Gamma distributed ζ . Once the coefficients of the expansion in Eq. 5.4 are calculated, it is possible to compute statistical moments for the measured variables at any given time instant t as Eq. 5.8 and Eq. 5.9 as a function of the coefficients of the expansion x_k in Eq. 5.4 as follows:

$$E(x(t)) = E \left[\sum_{i=0}^P x_i(t) \Phi_i \right] = x_0(t) E[\Phi_0] + \sum_{i=1}^P E[\Phi_k] = x_0(t) \quad (5.8)$$

$$\begin{aligned} Var(x(t)) &= E \left[(x(t) - E(x(t)))^2 \right] = E \left[\left(\sum_{i=0}^P x_i(t) \Phi_i - x_{(i=0)}(t) \right)^2 \right] \\ &= E \left[\left(\sum_{i=1}^P x_i(t) \Phi_i \right)^2 \right] = \sum_{i=1}^P x_i(t)^2 E(\Phi_i^2) \end{aligned} \quad (5.9)$$

Also, the probability density functions (PDFs) for measured variables, $x(t)$, can be approximated by sampling from the distribution of ζ and substituting the samples into Eq. 5.4. The ability of analytical formulae for calculating statistical moments as per Eq. 5.8 and Eq. 5.9 and to rapidly calculate the PDF profiles of the measured variables are the main rationale for using the gPC, since it dramatically reduces the computational effort involved in repeated calculations of moments and the PDF profiles as required in this study.

The fault detection procedure used in the current work consists of the inverse of the procedures explained in this section, i.e., the distribution of the stochastic parametric faults (inputs) \mathbf{g} is to be inferred from measurements of the process measured variables \mathbf{x} . Further details about this inverse procedure are given below.

5.4 Optimal Tuning of Controller Supervised by a FDD Algorithm

Since the tuning of the controller affects both the fault detectability and measured variables, a simultaneous optimal tuning of a controller and a fault detection algorithm is formulated as a bi-level optimization problem. It comprises an *inner level optimization* where the fault detection algorithm is calibrated with simulated noisy data and an *outer level optimization* where optimal tuning parameters of the controller and/or the set-point of the controlled variable are optimized. The calibrated fault detection algorithm that involves the inverse of the procedure described in Section 5.3 is used to estimate the proportion of faults that will go undetected, i.e., the misdetection rate. Then, the estimated amount of undetected faults is given a cost that is a trade-off in the *outer level optimization* with the costs of control to lead to an optimal trade-off between control and detection. Details about the fault detection algorithm and the integration of this algorithm with the controller are given below.

5.4.1 Fault Detection and Diagnosis Algorithm

The faults considered in the current work consist of intermittent step changes in an input variable with superimposed stochastic noise (as seen in Fig.5.1 (a)). The step changes follow a Multi-Level Pseudo Random Sequence (ML-PRS). Fig.5.1 (a) shows a segment of the typical faults signal used, where the step changes occur among five different mean values on stochastic parametric faults. The unknown faults (inputs g in Eq. 5.1) are then described as follows:

$$g_i = \bar{g}_i + \Delta g_i \quad (i = 1, \dots, k) \quad (5.10)$$

where \bar{g}_i is a set of possible mean values (operating modes), Δg_i are stochastic variations around each mean value, k is the number of mean values in total. Each particular mean value will be referred heretofore as an operating mode. The fault detection and diagnosis (FDD) problem is then defined as detecting a change in the mean values of the stochastic parametric faults (inputs) from noisy process measurements such as manipulated or controlled variables (Fig.5.1 (b)). The insets in Fig.5.1 (a) and (b) respectively show additional details of inputs considered as faults and the corresponding measured variables to be used for inferring these faults. For instance, the inset in Fig.5.1 (a) represents a few step changes around a particular mean value of stochastic fault. The insets in Fig.5.1 (b) show the changes on the measured variable induced by the variations of faults (inputs).

In Section 5.3 above, we explained how to propagate stochastic inputs, such as the ones shown in Fig.5.1 (a), onto measured quantities (outputs) and how to calculate the probability density functions (PDFs) of these outputs by using gPC expansions. The main idea of the fault detection algorithm proposed in this study is to solve the inverse of the problem given in Section 5.3. In particular, the goal is to dynamically infer the mean and variance of the stochastic parametric faults (inputs) from the gPC models of the measured variables, i.e., states x in Eq. 5.1.

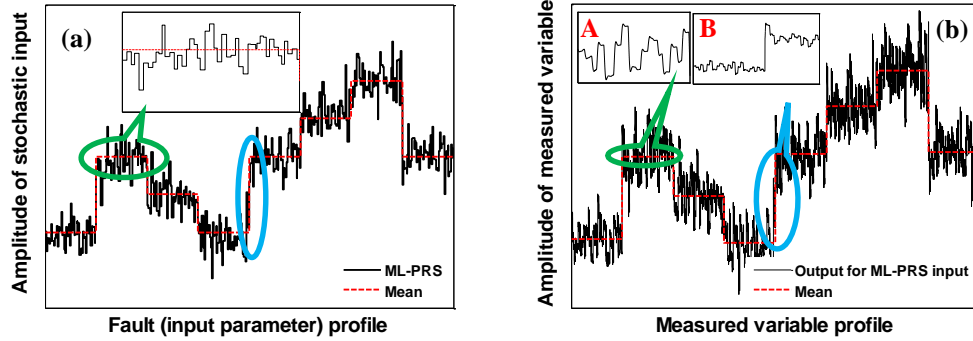


Figure 5.1 Fault profile representing an intermittent stochastic input fault and resulting measured variable

The method consists of four steps: (1) the stochastic parametric fault is propagated onto the measured variables thus providing the gPC expressions of the measured quantities (outputs) $x(t, \xi)$ around each possible mean value \bar{g}_i considered in the problem; (2) The PDF profiles of the outputs are calculated for each mean value \bar{g}_i by substituting samples from a priori known distribution of ξ into $x(t, \xi)$. The PDF profiles are used to quantify the

unobservable faults, which builds the connection between the *outer level optimization* and the *inner level optimization*; (3) The maximum likelihood based stochastic fault estimation combines the gPC model predictions with the measured data to maximize the likelihood with respect to the value of the random variable ξ ; and (4) The optimized values of ξ in Step (3) are substituted into a set of likelihood functions generated around each of the mean values \bar{g}_i , where the largest likelihood value is used to infer the corresponding operating mode (mean value of stochastic parametric faults) around which the system is operated. These 4 steps are further explained below.

Step 1:

The stochastic parametric faults (\mathbf{g} in Eq.5.10) are propagated with the gPC model into the measured variables (\mathbf{x}) to be used for fault detection following the method outlined in Section 2.

Step 2:

The PDF profiles of the measured quantities (variables) to be used for detection are calculated around each mean value of \bar{g}_i . The times for switching between different values of \bar{g}_i are simulated with a Multi-Level Pseudo Random Sequence (ML-PRS) in this step. The PDF profiles of the measured variables to be used for detection (\mathbf{x} in Eq. 5.1), can be calculated by the method described in Section 5.3. In practice, due to noise, model error (e.g., gPC truncation error in Eq. 5.3) and/or Eq. 5.4) and lack of exact knowledge about the stochastic faults, the PDF profiles of the measured variables have to be calibrated using actual process measurements. To this purpose, the mean and variance of the unknown stochastic parametric faults are calibrated for each of the mean values \bar{g}_i in Fig.5.1 (a) as follows:

$$\min_{\lambda_{inner}} J = \sum_{i=1}^n \omega_{1,i} (\vartheta_{1,i}(\lambda_{inner}) - v_{1,i})^2 + \sum_{i=1}^n \omega_{2,i} (\vartheta_{2,i}(\lambda_{inner}) - v_{2,i})^2 \quad (5.11)$$

where λ_{inner} is a decision variable vector consisting of the mean and variance of the stochastic unknown parametric faults (inputs). $\vartheta_{1,i}$ and $\vartheta_{2,i}$ are the mean and variance of the predictions of the particular variables (\mathbf{x}) to be used for fault detection and control. It should be noted that $\vartheta_{1,i}$ and $\vartheta_{2,i}$ are functions of the decision variables λ_{inner} , and they can be calculated numerically with Eq. 5.8 and Eq. 5.9 by substituting the gPC approximation into the first principle model and by conducting Galerkin projection as per the procedure illustrated in Section 5.3. The terms $v_{1,i}$ and $v_{2,i}$ are the mean value and variance of the sampled noisy measurements of \mathbf{x} , and n is the number of the manipulated and/or controlled variables used to calibrate the nonlinear first principles model. The weights, $\{\omega_{1,i}\}$ and $\{\omega_{2,i}\}$, determine the contribution of each term to the objective function Eq. 5.11. The measured values in Eq. 5.11 are collected for all step changes around each \bar{g}_i in the ML-PRS. Due to the presence of noise and truncation error introduced by the gPC approximation, the mean and variance of the input disturbance defining λ_{inner} and calculated from Eq. 5.11 will differ from the actual values entering the process. From the λ_{inner} resulting from Eq. 5.11, it is possible to calculate the calibrated gPC coefficients for the measured variables \mathbf{x} .

Step 3:

Using the calibrated gPC coefficients from *Step 2*, the PDF profiles of measured quantities x 's are estimated by substituting random samples (ξ) from the random events' space into the resulting gPC expansions Eq. 5.4. The PDF profiles for the measured variables are then approximated by using a binning algorithm where different ranges of probability values are assigned to each particular bin. Fig.5.2 shows a schematic of PDF profiles for one measured variable of interest, where each operating mode represents the mean value of the stochastic parameter faults \bar{g}_i .

Each of the PDF profiles calculated in *Step 2* and *Step 3* assumes that the fault mean value \bar{g}_i remains constant. If the system is operated around a constant mean \bar{g}_i , the corresponding operating mode i ($i = 1, \dots, k$) in Eq. 5.10 is detected from the PDF profiles for a given measurement as follows:

$$i = \arg \max\{P_i\} \quad (5.12)$$

where i is the operating mode as defined in Eq. 5.10, P_i represents the probability of being operated around a particular mean value \bar{g}_i for a given measurement at steady state. The steady state PDF profiles based fault detection was proposed in earlier work by the authors (Du, Budman, & Duever, 2014), while the process is operating around a particular mean value \bar{g}_i . As shown in Fig.2, for example, three probabilities (red dots) are found for a given measurement collected at steady state. The maximum probability is used to infer that the system is operating around the mean value corresponding to 'Mode 2'.

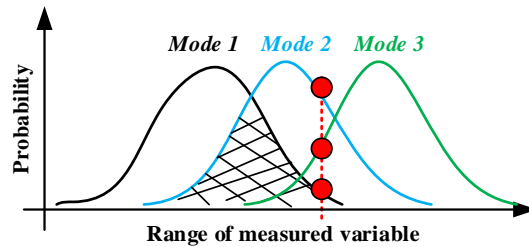


Figure 5.2 The PDF profiles of measured variables

Step 4:

The actual fault involves dynamic changes around the particular mean values as shown in Fig.5.1 (a). To diagnose changes in the mean values of stochastic faults following the step changes in the ML-PRS with transient measurements, i.e., measurements collected immediately after the occurrence of a step change on the mean value of faults, a maximum likelihood based estimator is applied to infer the dynamic value of the stochastic fault over a time moving window $[t_i, t_i]$. Compared with our earlier work, a moving time window of measurements is utilized rather than an individual measurement to add a filtering capability to the algorithm proposed in the current work. The objective is to recursively estimate the stochastic parametric faults (inputs) during transients.

$\ell_\tau(\xi, \mathbf{x}_{1:i})$ is defined as the likelihood function of the measured variables \mathbf{x} over time domain $[t_l, t_i]$, conditioned on the random variable ξ and the gPC coefficients of \mathbf{x} obtained in Section 2 and can be estimated by a suitable kernel density function at a particular time instant ($t_\tau, \tau = 1, \dots, i$) as:

$$\ell_\tau(\mathbf{x}|\pi(\xi), \mathbf{x}_\tau) \propto G_k[\mathbf{x}_\tau - \pi(t_\tau, \mathbf{x}(\xi), u; \mathbf{g}(\xi))] \quad (5.13)$$

where G_k denotes the kernel function, \mathbf{x}_τ are the measurements collected over the moving time window $[t_l, t_i]$, the π operator is defined as the gPC model conditioned on the gPC coefficients and ξ for a particular fault mean value \bar{g}_i as defined in Eq. 5.10. For the detection of stochastic fault changes, the likelihood of Eq. 5.13 is maximized at each time instant with respect to the random variable ξ . Then, a different value of the random variable ξ will be obtained corresponding to each gPC model generated around one of the mean values \bar{g}_i as follows:

$$\min_{\xi} J_{1:i} = \frac{1}{2} \sum_{\tau=1}^i (\mathbf{x}_\tau - \mathbf{x}_g(\xi))^T R_\tau (\mathbf{x}_\tau - \mathbf{x}_g(\xi)) \quad (5.14)$$

, where R_τ is the inverse covariance matrix used for kernel function, \mathbf{x}_g is calculated with the gPC model around a particular mean value of \bar{g}_i approximated in *Step 2*. For computational convenience, the gPC model of \mathbf{x} at each time instant over time domain $[t_l, t_i]$ is substituted into Eq. 5.14, and after some algebraic manipulations Eq. 5.14 can be rewritten for each mean value \bar{g}_i and at each time interval t_τ as:

$$\begin{aligned} \min_{\xi} J_{t_\tau=1:i} = & \frac{1}{2} \left(\sum_{k=1}^n \sum_{\tau=1}^i \mathbf{x}_\tau^T R_\tau \mathbf{x}_\tau - 2 \sum_{j=1}^P \sum_{\tau=1}^i \mathbf{x}_\tau^T R_\tau \mathbf{x}_{g,\tau} \Phi_j(\xi) + \sum_{j=1}^P \sum_{\tau=1}^i \mathbf{x}_{g,\tau} \Phi_j(\xi) R_\tau \mathbf{x}_{g,\tau} \Phi_j(\xi) + \right. \\ & \left. 2 \sum_{j=1}^P \sum_{k=1}^P \sum_{\tau=1}^i \mathbf{x}_{g,\tau} \Phi_j(\xi) R_\tau \mathbf{x}_{g,\tau} \Phi_k(\xi) \right) \end{aligned} \quad (5.15)$$

where $\mathbf{x}_{g,\tau}$ are the gPC coefficients at time instant t_τ , n is the number of the manipulated and/or controlled variables used in Eq. 5.15, P and $\Phi(\xi)$ are the truncation order and polynomial basis functions of the gPC model respectively. The gPC coefficients $\mathbf{x}_{g,\tau}$ are a priori calculated for each particular operating mode \bar{g}_i as explained in Section 5.3 and are independent of ξ . A gradient descent algorithm is used to solve Eq. 5.15. In practice, the term in Eq. 5.15 can be calculated offline to speed up calculations for each of the mean values \bar{g}_i (operating modes), since they are independent of ξ . The optimization of Eq. 5.15 is conducted for each of the gPC models generated with the set of mean values in *Step 2*. Thus, a set of ξ values can be obtained, each corresponding to a different mean value on stochastic parametric faults. Finally, the value of ξ that results in the smallest cost in Eq. 5.15 is substituted into the gPC approximation of the stochastic input in *Step 1* to estimate the average value of fault (input) over the moving time window.

5.4.2 Integration of Control and FDD Algorithm

An algorithm is proposed to simultaneously tune the fault detection algorithm presented above and a controller that involves measured variables to be used for fault detection. Since the tuning of the controller affects the detectability of the fault as well as the variability in the manipulated and controlled variables, the controller

parameters and/or the set-point at which the system should be operated can be optimized. A bi-level optimization problem, involving an *outer level optimization* where an optimal trade-off between control and detection is sought and an *inner level optimization* to optimally calibrate the fault detection algorithm with data (*Problem* Eq. 5.11)) is defined as follows:

$$\min_{\lambda_{outer}} J = \mu_1\gamma_1 + \mu_2\gamma_2 + \mu_3\gamma_3 + \mu_4\gamma_4 + \mu_5\gamma_5 \quad (5.16)$$

Subject to: Optimization problem in (5.11)

Stability constraints

where γ_1 is the cost of product quality related variables, γ_2 is the cost associated with variability in the controlled variables that are often associated with quality, γ_3 are the operating costs of the process, e.g., cost of utilities, γ_4 is the cost related to the variability in manipulated variables, i.e., deviation of control actions around nominal operating values, and γ_5 is the cost of unobservable faults which will be further discussed. The decision variables λ_{outer} are the tuning parameters of the controller or a combination of these tuning parameters and the set point, and the subscript *outer* indicates that Eq. 5.16 is the cost of the *upper level optimization* whereas the *lower level optimization* is given by Eq. 5.11. Stability constraints in Eq. 5.16 are imposed to ensure stability of the linearized model based on its eigenvalues. The weight coefficients, $\{\mu_i\}$, determine the contribution of each factor to the objective function Eq. 5.16. The effect of the choice of these weights is further discussed in the case study, since it is problem specific. The variabilities in objective function Eq. 5.16 account for the competing objectives between costs related to the tuning of controller and the cost incurred due to lack of detection of potential faults (γ_5).

A key simplifying assumption made in this work is that misidentification of faults occurs when the measurement values used for inferring the faults are located in the overlap regions of adjacent PDFs estimated as shown in Fig.5.2, since most misclassification will happen near the class boundaries. Accordingly the amount of fault misclassification is assumed to be correlated to the area of overlap between adjacent PDFs. For example, the shaded area in Fig.5.2 represents the overlap between *operating modes 1* and *2*. This assumption is directly justified by the fault detection algorithm presented in Eq. 5.15 where the objective is to calculate the maximum likelihood that a set of measurements within a moving time window correspond to operation around a particular mean \bar{g}_i and where this likelihood is assessed with respect to the PDF profiles generated around each mean. Thus, γ_5 in Eq. 5.16 is calculated by numerical integration of the total area of the overlaps between the PDF profiles shown in Fig.5.2. Note that the overlaps can also be estimated with Bayes error (Cha & Srihari, 2002) or Kullback-Leibler divergence (Press, Teukolsky, Vetterling, & Flannery, 2007), which quantifies the similarity of PDFs. The mean and variances of the product and control actions, as well as the estimate of the overlapping regions are all obtained from the PDF profiles calculated in *the inner level optimization*. This connects the *outer level optimization* to the *inner level optimization*, since all these values calculated from the PDF profiles participate in the cost of the *outer level optimization*. This is a two-way connection since the PDF profiles affect the cost of the

outer level optimization whereas the controller tuning parameters solved in the *outer level optimization* affect the PDF profiles.

The proposed bi-level optimization methodology and fault detection algorithm can be applied to other processes as per the following steps. (i) Assume stochastic parametric faults of interest for the process and formulate a gPC model from the first principles' model. (ii) Simulate the dynamic process with measurements noise, and identify the number of possible mean values for a ML-PRS (Multi-Level Pseudo Random Sequence) fault signal. (iii) Calibrate mean values and variances of the stochastic faults using the *inner level optimization* Eq. 5.11. (iv) Formulate stability constraints using the linearized closed loop gPC model to guarantee negative eigenvalues within the *outer level optimization* Eq. 5.16. (v) Calculate the unobservable fault from the numerical integration of the total area of the overlapping regions between the PDF profiles of the measured quantities. (vi) The trade-off between the control performance and the fault detectability is solved by the *outer level optimization* Eq. 5.16 with respect to the decision variable vector (controller parameters and/or set-point). (vii) Faults are detected by combining the gPC model with a maximum likelihood function using measurements obtained over a pre-specified moving time window.

5.5 Case Study

The fault detection algorithm proposed in Section 5.4.1 and the optimization problems in Section 5.4.2, defined by Eqs. 5.11, Eq. 5.15 and Eq. 5.16, respectively, are illustrated for a non-isothermal continuous stirred tank reactor (CSTR) system (Riggs, 1999). This process is considered sufficiently complicated to illustrate the methodology in terms of the presence of nonlinear behavior, uncertainty and disturbances (faults). Fig.5.3 depicts the CSTR with a concentration control loop as well as the variable (inlet concentration C_{A0}) for which faults are considered. The mathematical model of the process controlled with a *PI* controller is described by the following nonlinear model:

$$V_r \frac{dC_A}{dt} = \frac{F}{\rho} (C_{A0} - C_A) - V_r k_0 C_A e^{\frac{E}{RT}} \quad (5.17)$$

$$V_r \rho C_v \frac{dT}{dt} = F C_p (T_0 - T) - V_r \Delta H k_0 C_A e^{\frac{E}{RT}} + Q \quad (5.18)$$

$$\frac{dQ}{dt} = K_p \left(\frac{F}{\rho V_r} (C_{A0} - C_A) - k_0 C_A e^{\frac{E}{RT}} \right) - \frac{K_p}{\tau_i} (C_{A,set} - C_A) \quad (5.19)$$

where K_p and τ_i are the controller gain and integral time constant, respectively. The controller is used to control the outlet reactant concentration C_A by manipulating the external heat Q . To illustrate the proposed algorithm, faults are assumed to be intermittent step changes superimposed with variations in the inlet concentration C_{A0} of the type shown in Fig.5.1 (a). The parameter settings used for the CSTR simulation are given in Table 5.1.

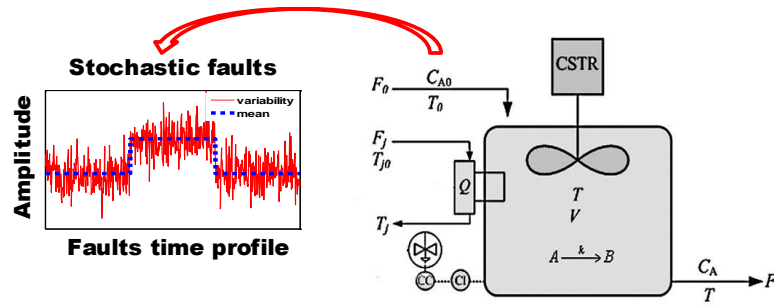


Figure 5.3 The CSTR with a concentration control loop and typical industrial stochastic faults

Table 5.1 Parameter declaration and setting used for CSTR

Symbol	Description	Symbol	Description
V_r	Reactor volume (100 L)	ΔH	Heat of reaction (160,000 cal/gmoles)
F	Mass feed rate (10 kg/s)	T_0	Feed temperature (400 K)
C_{A0}	Inlet concentration of component A (1.0 gmoles/L)	E/R	Normalized activation energy (2.0e5 K)
ρ	Density of the reactor feed and product (1.0 cal/g)	C_v	Assumed equal to C_p
C_p	Heat capacity, reactor feed and product (1.0 cal/g/K)	k_0	Rate constant (1.97e24 s ⁻¹)
Q_s	Initial steady state heat addition rate (7.0e5 cal/s)	T	Reactor temperature (350 K)

The objective is to solve the bi-level optimization Eq. 5.16 to seek the optimal tuning parameters for the *PI* controller and/or the optimal set-point for the reactant concentration C_A , while minimizing unobservable faults. The estimation of unobservable fault is subject to the optimal calibration of the fault detection algorithm in the *inner level optimization* Eq. 5.11. The goal of the fault detection algorithm defined by the *inner level optimization* is to identify changes in the mean value of inlet concentration C_{A0} based on the external heat measurements, while the *outer level optimization* in Eq. 5.16 is seeking to balance the competing objectives between the fault detectability and the closed loop performance. For this case study, the measured variable (x) used for the *outer level optimization* is the outlet concentration C_A (γ_1 in Eq. 5.16) and the external heat duty Q represents the operating cost (γ_3 in Eq. 5.16). Also, the cost related to the variability in product quality C_A (γ_2 in Eq. 5.16) and the cost related to the variability in the manipulated variable Q (γ_4 in Eq. 5.16) are approximated by the variance calculated using the gPC model with Eq. 5.9 in Section 5.3. The PDF profiles of the external heat Q is used in the current work to estimate the unobservable fault (γ_5) in Eq. 5.16, while the controlled variable (C_A) was not used for detection, since its variability around the set-point is generally small and thus it is less effective than the manipulated variable (external heat) for inferring stochastic concentration changes on C_{A0} . All the $\{\gamma_i\}$ values in the cost are calculated from the PDF profiles, thus connecting between the two levels of the bi-level optimization in Eq. 5.11 and Eq. 5.16, i.e., the *inner level optimization* and the *outer level optimization*. The weights $\{\mu_i\}$

determine the relative contribution of each cost to the total variability cost. Equal weights ($\mu_i = 1$) are assigned to guarantee each factor contributes evenly to the objective function Eq. 5.16. The effect of the choice of these weights is further discussed below in Section 5.6.7.

5.6 Results and Discussion

5.6.1 Formulation of gPC model

The application of Galerkin projection requires integrating the differential equations with respect to an appropriate selection of a polynomial for a particular random variable. Using the orthogonality property of the basis functions, for example, these integrations are possible for monomial or polynomial terms. However, the integration of non-monomial terms, such as the Arrhenius expression in Eq. 5.17, requires the use of an approximation. This problem is addressed in the current work by approximating the Arrhenius term with a 2nd order Taylor series expansion. The accuracy of the approximation of the Arrhenius term by the Taylor expansion is verified by comparing the gPC model predictions to the Monte Carlo (MC) simulations with the same operating conditions as listed in Table 5.1. Stochastic variations in the inlet concentration around a specific mean value and an additional inlet concentration perturbation at $t = 1000$ seconds are simulated.

For the gPC method, the coefficients of the gPCs describing measured quantities (outputs) are calculated. Then, samples from the random event ξ are substituted into these gPC expressions as outlined in Section 5.3 to approximate the range of the measurements for the measured quantities (x). Then, the maximum, minimum and mean of these measurements are obtainable at each time instant from the gPCs. For the MC simulations, (i) A set of samples of inlet concentration C_{A0} following the same statistical properties as used for the gPC are generated first; (ii) Each sample is substituted into the nonlinear CSTR model described as Eq. 5.17 ~ Eq. 5.19; and (iii) The resulting simulation of the measured variables are stored for comparison. The comparison results are depicted in Fig.5.4, where several randomly chosen simulated trajectories with the MC simulations are given. As expected, these trajectories are bounded by the upper (*Maximum*) and lower (*Minimum*) bounds calculated with the gPC approach. It is apparent from Fig.5.4 that the gPC model with the Taylor approximation of the Arrhenius term provided correct bounds for the MC simulations.

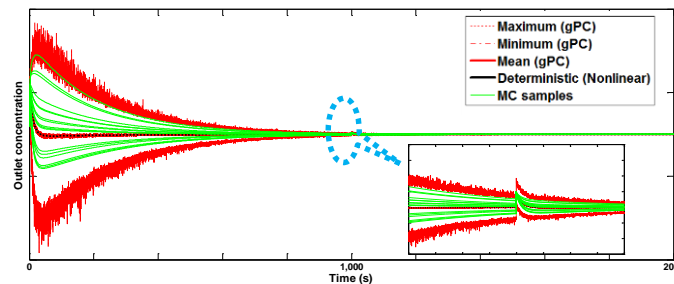


Figure 5.4 Simulation results of the gPC model, MC simulations and deterministic nonlinear model

5.6.2 Stability Constraints for the gPC model

The closed loop stability constraint in Eq. 5.16) is generated based on the linearization of the CSTR model of Eq. 5.17 ~ Eq. 5.19 with respect to all inputs and measured variables. The set point of reactant concentration $C_{A, set}$ and feed temperature T_0 are assumed to be deterministic and fixed, while the stochastic parametric fault C_{A0} and measured variables C_A , T and Q change with time. Following linearization and using deviation variables C'_{A0} , C'_A , T' and Q' , the following can be obtained:

$$\frac{dC'_A}{dt} = \frac{F}{\rho V_r} C'_{A0} - \left(\frac{F}{\rho V_r} + k_0 e^{\frac{E}{RT}} \right)_{ss} C'_A - \left(k_0 C_A \frac{E}{RT^2} e^{\frac{E}{RT}} \right)_{ss} T' \quad (5.20)$$

$$\frac{dT'}{dt} = - \left(\frac{\Delta H k_0}{\rho C_V} e^{\frac{E}{RT}} \right)_{ss} C'_A - \left(\frac{F C_p}{\rho V_r C_V} + \frac{\Delta H k_0}{\rho C_V} C_A \frac{E}{RT^2} e^{\frac{E}{RT}} \right)_{ss} T' + \left(\frac{1}{\rho V_r C_V} \right)_{ss} Q' \quad (5.21)$$

$$\frac{dQ'}{dt} = \left(\frac{K_p F}{\rho V_r} \right)_{ss} C'_{A0} - \left(\frac{K_p F}{\rho V_r} + k_0 K_p e^{\frac{E}{RT}} - \frac{K_p}{\tau_i} \right)_{ss} C'_A - \left(k_0 C_A \frac{E}{RT^2} e^{\frac{E}{RT}} \right)_{ss} T' \quad (5.22)$$

where the deviations on C'_{A0} in the linearized model are defined around each of the mean values \bar{g}_i in Eq. 5.10 using a gPC expression, according to the variability chosen for inlet concentration C_{A0} . After expanding all the variables of Eq. 5.20 ~ Eq. 5.22 with the gPC approximations and applying a Galerkin projection on both sides of Eq. 5.20 ~ Eq. 5.22, a coupled system of linearized ODEs is obtained in terms of the coefficients of the gPC expansions and the controller parameters. For stability, the eigenvalues of the linearized system of Eq. 5.20 ~ Eq. 5.22 are required to be negative (Seborg, Edgar, Mellichamp, & Doyle, 2011), which is the stability constraint of the *outer level optimization* in Eq. 5.16.

5.6.3 Calibration of the PDF Profiles for Inner Optimization

The PDF profiles of the measured variables (outputs) described in Section 5.4.2 are calibrated from a set of external heat measurements. Five mean values on inlet concentration are studied, i.e., 1.0, 1.25, 1.5, 1.75 and 2.0 (gmoles/L). The stochastic perturbations are added around these mean values which are assumed to be normally distributed with zero mean and a variance of 0.1 gmoles/L. The step changes follow a ML-PRS as shown in Fig.5.5 (a) and (b), respectively.

The ML-PRS used for calibration of the PDF profiles is defined by two parameters: L is the maximum number of step changes for all mean values (operating modes) and m is the maximum number of measurements between two consecutive step changes in the fault. To simulate actual data, Gaussian noise is added to the measurements. Table 5.2 shows the model calibration results for the *inner level optimization* Eq. 5.11 with a 1% measurement noise as an example, where Hermite polynomial is used and the highest order of the polynomial is 2 for different combinations of L and m .

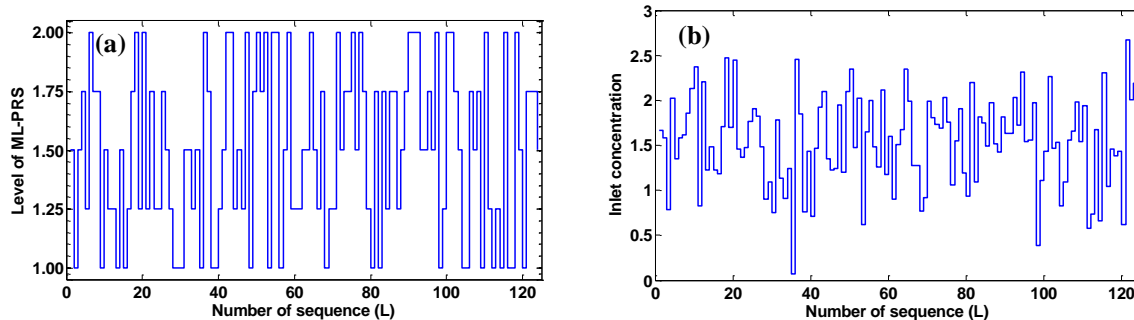


Figure 5.5 Multi-level pseudo random sequence

(a) Five-level PRS and (b) applied to the inlet concentration superimposed with stochasticity

Table 5.2 Comparison of the *inner level optimization* strategies (noise 1%)

$\bar{C}_{A0}^{\text{Simu}}$	ML-PRS							
	$L=124, m=2000$		$L=124, m=1000$		$L=124, m=500$		$L=624, m=500$	
	$\bar{C}_{A0}^{\text{opt}}$	ζ^{opt}	$\bar{C}_{A0}^{\text{opt}}$	ζ^{opt}	$\bar{C}_{A0}^{\text{opt}}$	ζ^{opt}	$\bar{C}_{A0}^{\text{opt}}$	ζ^{opt}
1.00	1.0534	0.1410	1.1064	0.1079	1.2713 ^F	0.1170	1.2647 ^F	0.1165
1.25	1.2757	0.1207	1.3079	0.1043	1.3720	0.1086	1.3739	0.1088
1.50	1.5039	0.1057	1.5030	0.1010	1.5111	0.1023	1.5170	0.1025
1.75	1.7357	0.0955	1.7211	0.0991	1.6867	0.0983	1.6867	0.0982
2.00	1.9695	0.0891	1.9391	0.0979	1.8667 ^F	0.0953	1.8668 ^F	0.0954

In Table 5.2, the first column is the actual mean value of inlet concentration C_{A0} used for simulations and the variance for each mean value is 0.1 gmoles/L. $\bar{C}_{A0}^{\text{opt}}$ and ζ^{opt} are the estimated mean value and variance with Eq. 5.11 for different values of L and m . As seen, the number of measurements in each sequence has a significant effect on the calibration results. For example, when using 500 measurements in each sequence some cases fail to converge to the correct mean values regardless of the values of L (denoted with superscript ‘F’ in Table 5.2). As shown in Table 5.2, the mean value and variance of the input resulting from Eq. 5.11 are not identical to the actual values used for simulation due to measurement noise and the gPC series truncation error.

Once the gPC models of the closed loop system around each mean value of fault are constructed, the corresponding PDF profiles for the external heat duty Q can be easily approximated. Fig.5.6 shows the PDF profiles obtained with 10,000 samples of the random variable (ζ), where the horizontal axis is the range of the external heat duty and the vertical axis represents the normalized probability. Each PDF profile corresponds to a different mean value and associated variance on the inlet concentration C_{A0} inferred from Eq. 5.11 with $L = 124$ and $m = 2000$ in Table 5.2. In Fig.5.6, “Modes” are referred to the mean values of inlet concentration C_{A0} . For instance, “Mode: 1.00” denotes that the expectation of the inlet concentration (C_{A0}) is 1.00 gmoles/L.

5.6.4 Case Study 1: Tuning of a Gain-Scheduled Controller

The cost described in the *outer level optimization* Eq. 5.16 is optimized with respect to the tuning parameters of a gain scheduled PI controller involving different tuning parameter values around the five different means

(operating modes) of inlet concentration. Table 5.3 shows the results of the optimum controller parameters, the cost (J^{opt}) of the objective function defined in the *outer level optimization* Eq. 5.16 and the normalized overlap (O^{opt}), for which the set point of the outlet concentration is fixed at 0.25 gmol/L.

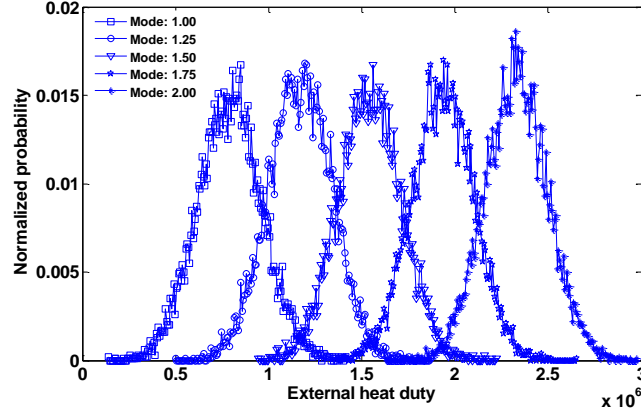


Figure 5.6 The PDF profiles of five operating modes on inlet concentration

Table 5.3 Summary of the results for the *outer level optimization* without tuning set point

\bar{C}_{A0}^{Simu}	K_p^{opt} [cal/s/gmol]	τ_i^{opt} [s]	J^{opt}	J^{ISE}	O^{opt}	O^{ISE}	Time (h)
1.00	81521.50	0.566	13.612	16.427	0.225	0.252	3.82
1.25	20175.84	0.082	12.233	14.943	0.217	0.272	3.44
1.50	19034.54	0.172	11.523	13.657	0.217	0.272	2.64
1.75	99773.78	0.340	12.136	14.904	0.234	0.264	3.97
2.00	14709.06	0.047	12.830	15.831	0.188	0.245	3.65

For comparison in Table 5.3, the value of the cost function is calculated for the case where the controller parameters are optimized by the minimization of the *integral squared error (ISE)*, i.e., $K_p = 75508$, $\tau_i = 0.505$. The corresponding cost and amount of overlap are referred to as J^{ISE} and O^{ISE} , respectively. The controllers designed based on the *ISE* criteria or the ones obtained from the *outer level optimization* in Eq. 5.16 is referred heretofore as the *ISE* and optimized controller (superscript *opt*), respectively.

As seen in Table 5.3, the largest improvements in the costs of J^{opt} versus J^{ISE} are observed around the largest and smallest mean values on inlet concentrations C_{A0} . An explanation is that the controller tuned according to the *outer level optimization* Eq. 5.16 seeks to minimize the overlaps of the PDF profiles corresponding to the different inlet concentrations C_{A0} shown in Fig.5.5. This is achieved at the cost of introducing larger variabilities in product quality and operating costs in objective function Eq. 5.16, since the controller attempts to shift the corresponding PDF profiles far apart from each other. The resulting overlap, as normalized by the total area of overlapping of the PDF profiles, is given in Table 5.3. It can be seen that the normalized total overlap is smaller with the tuning parameters optimized with the *outer level optimization*, as compared to the *ISE* controller. The decreased overlap

reflects the enhancement of the fault detectability, since the overlap is representative of the unobservable fault. The computational time required to solve the *outer level optimization* in Eq. 5.16 is given in the last column.

5.6.5 Case Study 2: Tuning of a Gain-Scheduled Controller and Set-point

A second study is investigated where the set-point of the outlet concentration C_A is chosen as an additional decision variable along with the tuning parameters of the gain-scheduled *PI* controller to minimize the cost function in the *outer level optimization* Eq. 5.16. Table 5.4 shows the optimum results of decision variables, the related cost and the normalized overlaps area between the PDF profiles of the external heat Q .

Table 5.4 Summary of the results for the outer level optimization with tuning set point

\bar{C}_{A0}^{Simu}	K_p^{opt} [cal/s/gmol]	τ_i^{opt} [s]	Set point	J^{opt}	O^{opt}	Time (h)
1.00	41375.12	1.304	0.498	8.978	0.111	3.21
1.25	83835.62	1.595	0.365	8.790	0.190	5.99
1.50	82061.19	1.697	0.348	8.501	0.181	11.01
1.75	69635.40	1.075	0.378	8.804	0.164	6.15
2.00	130554.09	1.242	0.362	9.912	0.157	5.21

It can be seen by comparing Table 5.3 and Table 5.4 that the cost of the objective function Eq. 5.16 is smaller when both the controller and set-point are simultaneously optimized. For example, the cost in Eq. 5.16 at an inlet concentration of 1.0 gmoles/L has been further decreased by ~34 percent point, from 13.612 to 8.978. Additionally, the overlap areas between the PDF profiles around each mean value of inlet concentration are smaller, as compared to the result where the set-point was not optimized. For instance, the overlap areas at an inlet concentration of 1.0 gmoles/L have been decreased by ~50 percent point, from 0.225 to 0.111. This confirms that the bi-level optimization can enhance the detectability of faults, since the overlap representing the unobservable fault has been decreased.

5.6.6 Case Study 3: Tuning of a Global Controller

In previous case studies, the controller parameters are tuned for each of the mean values (operating modes) of inlet concentration C_{A0} for the enhancement of fault detectability. In this study, one set of *PI* controller tuning parameters is optimized for all operating modes. The optimization results calculated from the *outer level optimization* Eq. 5.16 are $K_p = 9986.36$ and $\tau_i = 0.038$, respectively. For comparison, the cost defined in objective function Eq. 5.16 for three different tuning methods are calculated: (i) minimization of *ISE* ($J_{cost} = 69.894$); (ii) optimization of controller parameters using Eq. 5.16 with the gain-scheduled controllers ($J_{cost} = 12.467$) in case study 2, for which the controller parameters are switched among the operating modes; and (iii) optimization of Eq. 5.16 with a global controller ($J_{cost} = 33.387$) in this case study. As expected, the global controller improves the cost over the *ISE* based controller, but is inferior to the case that uses the gain-scheduled controllers optimized with Eq. 5.16. For the gain-scheduled controller, the cost is calculated as the average of J^{opt} given in Table 5.3.

5.6.7 Case Study 4: Investigation of Tuning Weights

The effect of the weights $\{\mu_i\}$ in the *outer level optimization* Eq. 5.16 is investigated to demonstrate a trade-off between different objectives. The weights determine the relative contribution of each factor to the total cost. The definition of $\{\mu_i\}$ is motivated by the proportionality constant in the Taguchi's quality loss function (Ross, 1988). These weights $\{\mu_i\}$ are problem specific. For instance, the operation cost of utilities (third term in the *outer level optimization* Eq. 5.16) is obtained from the expected cost of utilities per unit of variability for the utility source under consideration, e.g., manipulated external heat duty Q .

As each quality characteristic has different units and orders of magnitude, it is essential to normalize the quality cost. To assess the effect of the weights on the results, two sets of weights are investigated: (i) equal weights ($\mu_1 = \mu_2 = \mu_3 = \mu_4 = \mu_5 = 1$) and (ii) unequal weights ($\mu_1 = \mu_2 = \mu_3 = \mu_5 = 1, \mu_4 = 5$). In the first case, each factor gives an equal contribution to the total variability of the objective function Eq. 5.16. For the latter case, the variability of manipulated variable contributes more to the total cost, since the objective specifically is targeted to minimize the variability associated with the operating cost in this case study. Physically, the latter case is to avoid aggressive tuning of valves and minimize the cost of wear of the external heat actuator.

The cost described in the *outer level optimization* Eq. 5.16 is optimized with respect to one set of *PI* controller tuning parameters for all operating modes (five mean values on inlet concentration C_{A0}). For the case study with equal weights, the optimization results are given in Section 5.6.6. For the latter case with unequal weights, the optimization results obtained from (16) are $K_p = 998308.22$ and $\tau_i = 1.1640$, respectively. For comparison, Fig.5.7 shows several consecutive step changes on C_{A0} of the *ML-PRS* on the inlet concentration C_{A0} , and the corresponding simulation results of the controlled variable C_A and the manipulated variable Q , with equal weights and unequal weights.

As expected, the variability on the manipulated variable Q is smaller with unequal weights, as compared with the simulations with equal weights shown in Fig.5.7 (b). However, the variability on the controlled variable C_A has been increased, which is associated with the quality of product. These observations confirm that the weights are problem specific and can determine the relative contribution of each of the individual terms to the total cost function of Eq. 5.16. Additionally, the cost defined in the objective function Eq. 5.16 with unequal weights is calculated ($J_{\text{cost}} = 63.086$). Compared with equal weights ($J_{\text{cost}} = 33.387$), the cost has been increased by ~46 percent point. Finally, the normalized overlapping areas between the PDF profiles of the measured variable Q , representing the unobservable fault on C_{A0} , are 1.079 and 1.101 with equal weights and unequal weights, respectively. The overlap has not changed much for both case studies, since the weight penalizing the overlap is the same value ($\mu_5 = 1$).

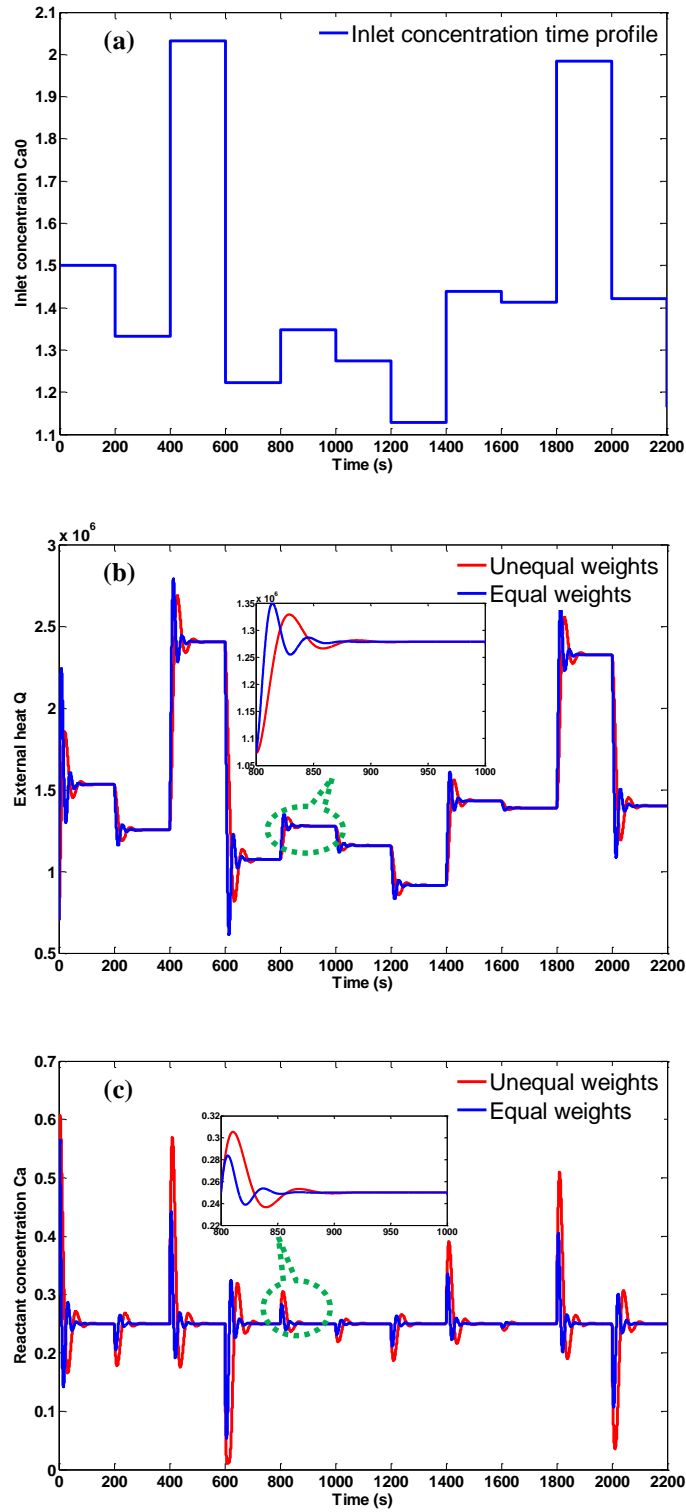


Figure 5.7 Illustration of the effect of weights on the control performance

5.6.8 Comparison of Fault Identification Results

The efficiency of the fault detection algorithm presented in Section 3.1 is investigated in order to verify that the reduced overlaps between the PDF profiles obtained in the optimal integration of control and detection Eq. 5.16 translate into better detection. The efficiency of the FDD algorithm is judged based on the fault detection rate (*FIR*) defined as follows:

$$d_{rate} = \frac{1}{n_m} \sum_{i=1}^{n_m} d_i/D_i \quad (5.23)$$

where n_m is the number of mean values (operating modes) on the inlet concentration C_{A0} , i is the i th operating mode, d_i is the number of samples that have been classified correctly and D_i is the total number of samples tested for in the i th mode.

Table 5.5 shows the *FIR* with the maximum likelihood based fault detection algorithm presented in Section 5.4. The measurements used for detection are collected during the transition period, i.e., immediately after the occurrence of a step change in ML-PRS, where the misdetection of a particular mean value is expected to be the highest since the system undergoes a transient response. To improve the *FIR* during transitions, pre-defined time moving window $[t_l, t_r]$ in Eq. 5.13 are used. Specifically, windows of 20, 50 and 100 measurements (n in Eq. 5.15) of the external heat Q are investigated, respectively. The *FIR* is compared for two cases: (i) a closed loop system that uses a controller tuned by the minimization of *ISE* (*ISE* system), i.e., without the *outer level optimization* Eq. 5.16, and (ii) a closed loop system that uses a controller based on the optimization of Eq. 5.16 (*optimized* system). An ML-PRS simulation is used to generate testing samples on inlet concentration C_{A0} , in which 100 consecutive step changes ($L = 100$), i.e., 100 testing samples are used for each of the measurement noise levels.

Table 5.5 Summary of the *FIR* using transient measurements

Case studies	Fault Identification Rate (<i>FIR</i>)			
	Optimized		ISE	
	Noise 1%	Noise 10%	Noise 1%	Noise 10%
$n = 20$	0.353	0.297	0.335	0.268
$n = 50$	0.629	0.601	0.619	0.591
$n = 100$	0.827	0.792	0.804	0.762

From Table 5.5, it can be seen that the *optimized* system shows better *FIR* performance as compared to the *ISE* based system. For example, for a time window of $n = 20$ the differences in *FIR* are of the order of 10%, thus confirming that the results obtained with optimized system based on the *outer level optimization* Eq. 5.16 translate into better detection. Additionally, as seen in Table 5.5, the *FIR* improves as the window is larger. For the *optimized* system, the *FIR* with 1% measurement noise has been increased by ~30%, from 35.3% to 62.9%, if 50 measurements of external heat Q are used instead of 20. It should be noticed that the differences in *FIR* obtained with the two closed loop systems considered in Table 5.5 are highly dependent on the weights used in the cost

function in Eq. 5.16, since these determine the overlap between the calculated PDF profiles. For example, if μ_5 in Eq. 5.16 is larger, the resulting overlap will be smaller resulting in larger differences in *FIR* values between the non-optimized (*ISE*) and the *optimized* systems but at the cost of higher values of the other terms in the cost of objective function Eq. 5.16, as discussed in Section 5.6.7.

Since a key performance indicator of a fault detection algorithm is the time required for detection elapsed after the occurrence of the fault, further comparison studies are conducted to investigate the required estimation (detection) time for the *ISE* and *optimized* systems. Fig.5.8 shows a segment of the *ML-PRS* used for fault detection involving five consecutive step changes on inlet concentration C_{A0} .

For the two systems, Fig.5.8 (a) shows the decision variable ξ at each time instant while optimizing Eq. 5.15 based on the measurements collected over a moving time window. The optimization of Eq. 5.15 is conducted for each of the gPC models generated with the five mean values on the inlet concentration C_{A0} . The minimum value of Eq. 5.15 can be calculated at each time interval and used for detecting the corresponding operating mode (mean). Also, the value of the decision variable ξ can be substituted into a particular gPC approximation of the stochastic input to provide a dynamic estimate of the inlet concentration C_{A0} at a particular time instant, i.e., Fig.5.8 (b).

Fig.5.8 (b) and (c) show the estimations of the inlet concentration and the corresponding normalized heat at each time interval, respectively. As can be seen, both systems, i.e. *ISE* system and optimal (optimized with Eq. 5.16), can provide correct estimation results. However, the optimal system can detect (estimate) the fault (inlet concentration) faster than the *ISE* system. The insets in Fig.5.8 (b) demonstrate the time difference for estimating the inlet concentration values between two consecutive step changes. For example, the system using the optimized controller is ~12 seconds faster than the *ISE* system for a step change from 1.2501gmoles/L to 1.2058 gmoles/L (inset A). For the step changes between operating modes (inset B), the optimal system performs better in terms of fault detection speed. As seen, the system optimized with Eq. 5.16 needs ~75 seconds to estimate the inlet concentration value and stabilizes at ~0.9903 gmoles/L, while ~100 seconds are required for the system that uses the *ISE* based controller. For the simulated *ML-PRS* on inlet concentration C_{A0} with 100 consecutive step changes ($L = 100$), the average of fault detection time with the *optimized* system is ~15 seconds faster than with the *ISE* based system. These observations above confirm that the results obtained with the *outer level optimization* Eq. 5.16 can translate into more robust and faster fault detection. It's worth mentioning that the differences in fault detection times may be very critical in some applications such as chemical reactors where small changes in inlet concentration may cause, if undetected, runaway conditions. Also, faster detection may be beneficial for the timely implementation of a gain-scheduled controller discussed above, where different sets of controller tuning parameters are used for different mean values of the inlet concentration.

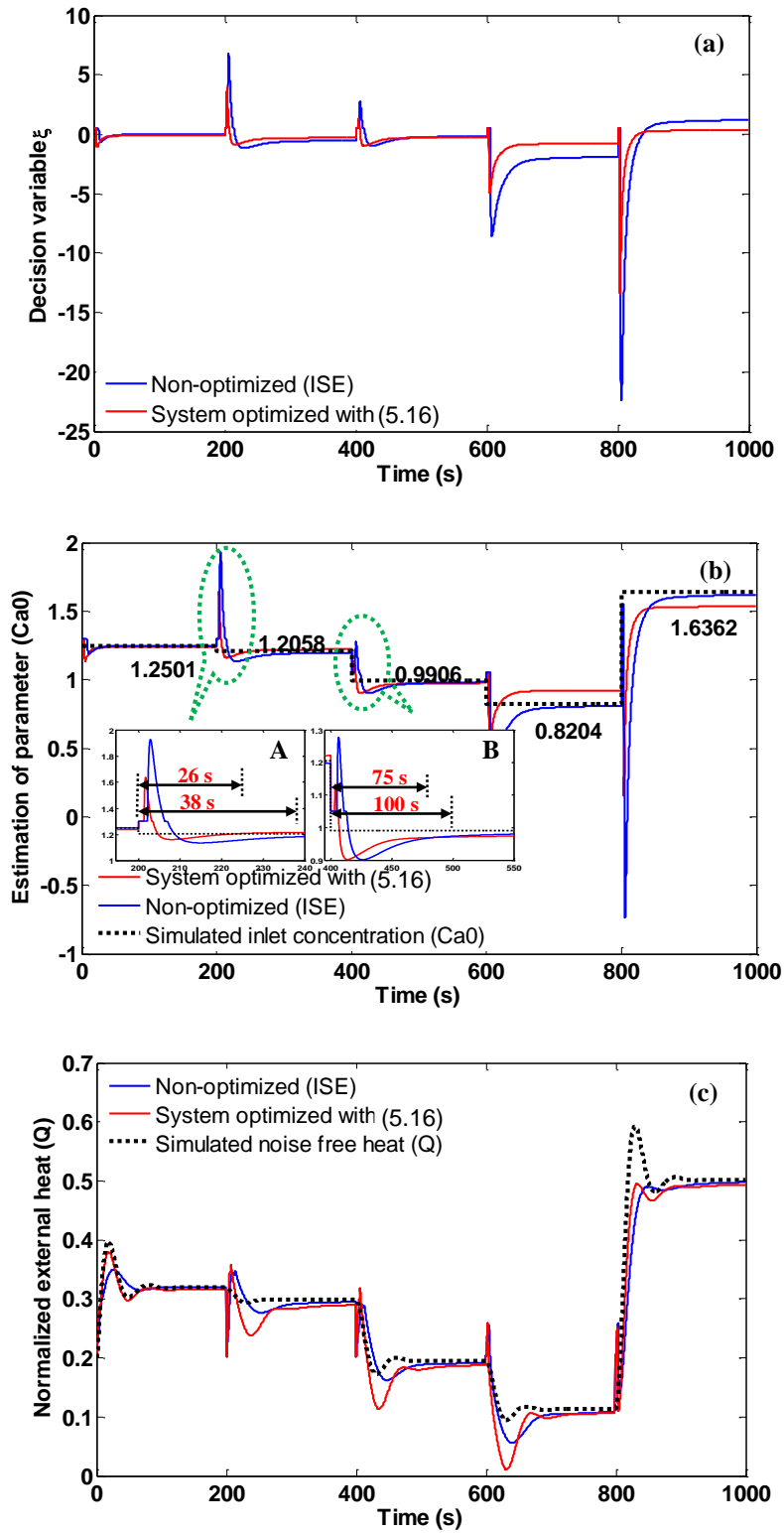


Figure 5.8 Illustration of maximum likelihood estimation based fault detection

5.6.9 Discussion of Computational Efficiency

The computational time of the proposed gPC based methodology is compared with the most widely used sampling methods. All the methods are executed on a 2.66 GHz Intel^(R) Core Duo processor. The processor time of the model calibration as per the *inner level optimization* in Eq. 5.11 is first evaluated by the Monte Carlo (MC) simulations. For instance, one set of evaluation takes ~ 3.89 hours for the MC based *ML-PRS* simulations, in which 10,000 samples ($L = 10000$) are used and the maximum number of measurements between two consecutive samples is chosen as 2000 (m). Additionally, 10,000 samples with MC simulations are found to be inaccurate as compared to the proposed gPC approach, thus an even larger number of samples are required to obtain better estimation of the mean and variance in Eq. 5.11, which would further increase the computational burden, since the search for the optimum in Eq. 5.11 for each mean value require many evaluations. If 60 iterations are required to optimize Eq. 5.11 for each mean value, for example, it takes approximately 233.4 hours on average. However, the proposed gPC method takes ~10 minutes to calculate the optimum in Eq. 5.11 for all mean values considered in the current work.

Additionally studies are conducted by using the Latin hypercube sampling method that can speed up the convergence of the MC by adding constraints on the samples (Dalbey, Patra, Pitman, Bursik, Sheridan, & F., 2008). Table 5.6 shows the optimization results for Eq. 5.11 with 50 sample points ($L = 50$) used to satisfy the Latin hypercube requirements. It should be noted that the samples used to calculate the mean and variance as per optimization in Eq. 5.11 are different for each search procedure, which results in a stochastic optimization problem. Thus, the genetic algorithm (Fouskakis & Draper, 2002) is used to solve Eq. 5.11, in which the maximum number of measurements between two consecutive samples is chosen as 2000 ($m = 2000$).

Table 5.6 Summary of inner level optimization with Latin hypercube sampling

\bar{C}_{A0}^{Simu}	\bar{C}_{A0}^{opt}	ζ_{opt}	Time (h)
1.00	1.1288 ^F	0.0143 ^F	77.67
1.25	1.2968	0.1333	77.12
1.50	1.5591	0.1125	72.07
1.75	1.8063	0.1420	76.49
2.00	2.0889	0.1207	77.99

Compared with the MC simulations method, the processor time with the Latin hypercube sampling for one mean value is ~76.26 hours on average, which is significantly lower than MC simulations. As compared with the gPC method, however, it is still very time demanding. Additionally, as seen in Table 5.6, 50 sample points in Latin hypercube sampling are still found to be less accurate. For example, there is an obvious discrepancy between the optimum and the values used for simulation, i.e., the mean and variance of inlet concentration used for simulation are 1.0 (gmoles/L) and 0.1, while the optimum obtained from Eq. 5.11 are 1.1288 and 0.0143, respectively. Increasing the number of samples used for model calibration may improve the efficacy, however, it still makes the evaluation of Eq. 5.11 time prohibitive, since this step has to be repeatedly optimized for solving

the simultaneous optimal tuning of controller and fault detection problem. This clearly illustrates that the use of gPC model is instrumental for solving the bi-level optimization problem in Eq. 5.16, since the problem has to be solved many times during the optimization search thus leading to a significant reduction in the computational time for the gPC based method as compared to the sampling based approaches.

5.7 Conclusions

In the present work, a methodology has been developed to simultaneously optimize closed loop performance and fault detection efficiency. The proposed approach is tested for an endothermic continuous stirred tank reactor (CSTR). The main novelty of the proposed approach is that it addresses the effect of stochastic inputs on the measured variables by using the gPC approximations and the first principle process model. The use of gPC is shown to be effective because the variabilities of the input and measured variables can be quickly calculated using analytical expressions. Since these variabilities have to be calculated repeatedly during the optimization search the dramatic reduction in computation time with gPC as compared to MC type sampling based methods makes this approach especially attractive for solving the simultaneous optimal tuning and fault detection problem for large systems. Also, a fault detection algorithm is formulated based on a maximal likelihood criteria and the gPC model. Since this algorithm is used in real time, the computational efficiency of gPC is instrumental for its implementation.

Chapter 6

Classification of Apoptotic and Normal Cells

(Adopted from Du et al., 2015, Microscopy and Microanalysis, under review)

6.1 Overview

Accurate automated quantitative analysis of living cells based on fluorescent microscopy images can be very useful for fast evaluation of experimental outcomes and cells culture protocols. In Chapter 3, first principles models described by Partial Differential Equations (PDEs) were used for fault detection. Image segmentation algorithms based on level set functions also involved the solution of PDEs of similar form to the ones solved in Chapter 3. Thus, in the current chapter, we capitalized on the methodology developed in Chapter 3 to introduce the effect of stochastic errors in edge detection problems by using gPC. The particular edge detection problem studied in this chapter is the fast differentiation of normal and apoptotic viable Chinese Hamster Ovary (CHO) cells. The stochastic image segmentation algorithm developed in this chapter can be described by a set of PDEs to dynamically evolve the boundary of cells. For the effective segmentation of cells' images, the stochastic segmentation algorithm is developed by combining a generalized polynomial chaos (gPC) expansion with a level set function based segmentation algorithm. This approach provides a probabilistic description of the segmented cellular regions along the boundary, from which it is possible to calculate the morphological changes related to apoptosis, i.e., the curvature and the length of cell's boundary. These features are then used as inputs to a support vector machine (SVM) classifier that is trained to distinguish between normal and apoptotic viable states of CHO cells' images. The use of morphological features obtained from the stochastic level set segmentation of cells' images in combination with the trained SVM classifier is shown to be more efficient in terms of differentiation accuracy as compared to the original deterministic level set method.

6.2 Introduction

Fluorescence microscopy is a well-developed tool for assessing characteristics of cells such as cells' number and physiological states (Waters, 2009). Studies of in-vitro cells behavior typically require analyzing thousands of cells' images for assessing the outcomes of experimental investigations. The interpretation and quantification of these data via manual analysis are either time consuming or prone to human errors. Also, the development of computationally efficient algorithms may facilitate future real-time implementation of these measurements.

Due to this growing demand for automated image processing tools, software packages such as CellProfiler (Carpenter, et al., 2006) and ImageJ (Schneider, Rasband, & Eliceiri, 2012) have been developed. These toolkits typically consist of segmentation algorithms that can be used to classify each pixel within an image as either cell or background. Based on the segmented regions, the characteristics of the cellular object such as size or shape can be inferred. Accurate segmentation of cells' images acquired using microscopy is generally challenging and

time consuming as reported in several studies on the subject (Veredas, Mesa, & Morente, 2010; Theriault, Walker, Wong, & Betke, 2012; Yin, Bise, Chen, & Kanade, 2010).

An image can be stored as a multi-dimensional matrix, in which elements are pixels' intensities (Chan & Shen, 2005). For most of the work on image processing and analysis, the intensities are assumed to be deterministic quantities, i.e., pixels have fixed gray or color values. However, such assumption ignores that in practice intensities are contaminated by noise and other uncertainties (Kybic, 2010). These may originate from either intrinsic measurement's limitations or as a result of inaccurate estimates of pixels' intensities. In addition, down-sampling of an image may be exploited, as done in the present work, to speed up the cell segmentation procedure but this operation will add uncertainty due to the lost information.

The main idea behind image segmentation is to detect the boundary of cells and separate the cells from the background. Any small measurement errors due to noise or uncertainty in the pixels' intensities may result in significant variations in the results of the segmentation procedures. To improve the robustness of image segmentation algorithms, uncertainty quantification has been proposed before for quantitative analysis of images (Chan & Shen, 2005). Accounting for uncertainties in images leads to the notion of stochastic images (Preusser, Scharr, Krajsek, & Kirby, 2008), where the pixels' intensities follow probability density distributions (PDFs) that describe the information about the measurements' uncertainties.

The speed and accuracy in propagating the effect of uncertainty onto the main morphological features to be extracted from an image is critical for automatic segmentation of stochastic images. The generalized polynomial chaos (gPC) expansion (Xiu D. , 2010) has been introduced into image processing and analysis to deal with random shapes or pixels' intensities in the presence of uncertainty (Stefanou, Nouy, & Clement, 2009). Following this idea, in the current work, the gPC theory is combined with the *active contours without edges* (Chan & Vese, 2001) method to differentiate cells according to their physiological status, i.e., normal versus apoptotic where the latter refers to cells undergoing apoptosis or programmed cell death. Mammalian cells are prone to apoptosis that can be characterized by significant levels of plasma membrane blebbing and nuclear condensation. Fig.6.1 shows a fluorescence microscopy image of Chinese Hamster Ovary (CHO) cells. As can be seen, apoptotic cells exhibit blebbing resulting in irregular shapes and blurry boundaries, as compared to the normal cells that exhibit smooth boundaries. Blebbing is due to swelling of the cell membrane following detachment of the cytoskeleton from the membrane when apoptosis occurs, which may introduce significant variations in the pixels numbers and their intensities along the boundary of cells.

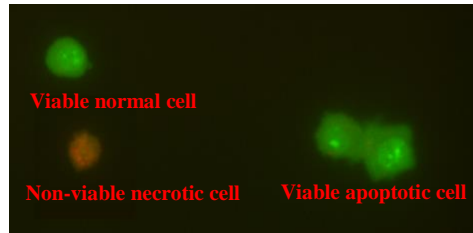


Figure 6.1 Fluorescent photomicrograph of CHO cells stained with AO and EB

In this work, the *active contours without edges* method (Chan & Vese, 2001) is modified to account for the uncertainty in a given measured image. The key idea is to represent the evolving contours of the cells with a probabilistic description. The gPC approximations of the level set function are directly calculated from the partial differential equation (PDE) used by the active contour method. The zero level (mean value) of the resulting stochastic level set function is interpreted as the mean value of the probabilistic active contour (boundary), while the curvature of fluorescence intensity at the boundary can be analytically derived with the gPC expansions. Then, the curvature together with the length (number of pixels) of the boundary are used to distinguish apoptotic cells from normal cells based on a support vector machine (SVM) classification algorithm (Burgers, 1998). The methodology in this current work is applied to fluorescence microscopy images of apoptotic and normal CHO cells grown in batch experiments, and the performance of the proposed algorithm is compared to the original *active contour without edge* (deterministic level set) method.

To summarize, the novel contributions in the current work are: (i) The development, in the context of image segmentation, of a gPC based approach for fast calculation of particular morphological features of CHO cells from fluorescence microscopy images; (ii) The formulation of an automated SVM differentiation algorithm based on two morphological features, i.e. curvature and length of cells' boundary, to distinguish apoptotic cells from normal cells.

This paper is organized as follows. Section 6.3 presents the background material and the main mathematical tools used in this work. The extraction of morphological features from the stochastic images and the classification algorithm based on these features are given in Section 6.4. Analysis and discussion of results are presented in Section 6.5 followed by conclusions in Section 6.6.

6.3 Segmentation of Stochastic Images

6.3.1 Fluorescence Imaging

Fluorescence microscopy has been used to differentiate and quantify apoptotic versus normal cells as well as determine the viability of cells (Mercille & Massie, 1994). This approach involves two types of fluorescent dyes, i.e., acridine orange (AO) and ethidium bromide (EB), which are mixed in a fixed ratio within the cell suspension and then they are analyzed by fluorescence microscopy. The AO can penetrate viable and nonviable cells and

make cells to appear green while the EB can only diffuse into nonviable cells and make them to appear orange or red (see Fig.6.1).

The objective of this work is to automatically differentiate apoptotic from normal viable cells while maintaining the processing time at a reasonable level. Since apoptotic cells exhibit irregular shapes and blurry boundaries (see Fig.6.1), a departure from smooth boundaries is a key morphological indicator for differentiating between normal and apoptotic cells. Also, due to the swelling of the boundary occurring with apoptosis longer boundaries are generally observed. Hence, this work builds on the hypothesis that the variability in curvature observed along the cells' boundary together the length of the boundary measured by the number of pixels on it can be utilized for doing this differentiation, i.e., lower variability in curvature and shorter lengths are associated to normal cells and higher variability in curvature and larger lengths are associated to apoptotic cells.

6.3.2 Active Contours without Edges

The key idea behind the active contour method is to progressively calculate a boundary for a given grayscale image U_0 , that separates objects from their background. Let us define the curve as C , subjected to the constraints of the image in an open bounded domain Ω of R^2 , which approximates the boundary φ , i.e., $C \approx \varphi$, $C \subset \Omega$, and $\varphi \subset \Omega$. To solve the curve C , the *active contours without edges* algorithm (Chan & Vese, 2001) seeks a best approximation of a segmentation C for a given image U_0 from the minimization of an energy function defined as follows:

$$\begin{aligned} \arg \min_{m_1, m_2, C} & \quad \mu_1 \cdot \text{Length}(C) + \mu_2 \cdot \text{Area}(\text{inside}(C)) \\ & + \lambda_1 \int_{\text{inside}(C)} |U_0(x, y) - m_1(C)|^2 dx dy \\ & + \lambda_2 \int_{\text{outside}(C)} |U_0(x, y) - m_2(C)|^2 dx dy \end{aligned} \quad (6.1)$$

where μ_1, μ_2, λ_1 and λ_2 are non-negative fixed parameters, m_1 and m_2 , depend on the evolving curve and are the mean values of U_0 inside C and outside C , respectively. The coordinates, defining the domain Ω , are determined by the x -axis and y -axis. The first term in Eq. 6.1 controls the regularity of C by penalizing the length. The second term penalizes the enclosed area to control the size of the segmented objects. The third and fourth terms penalize the discrepancy between pixels' intensities within and outside the active curve C in terms of their means.

The optimization problem in Eq. 6.1 can be formulated and solved by the level set method (Osher & Sethian, 1988), where the unknown curve C is replaced by the unknown level set function Z defined in the xy -plane. Instead of manipulating C , the minimization of Eq. 6.1 is represented as the evolution of a level set function Z . Then, the curve C corresponds to the geometric locus of the points with $Z(x, y) = 0$ in the xy -plane.

Provided that the level set function Z is smooth, the *active contours without edges* optimization (Chan & Vese, 2001) in Eq.6.1 can be equivalently rewritten in terms of the level set function Z as:

$$\begin{aligned}
\arg \min_{m_1, m_2, C} \quad & \mu_1 \cdot \int_{\Omega} \delta_{\varepsilon}(Z(x, y)) |\nabla Z(x, y)|^2 dx dy \\
& + \mu_2 \cdot \int_{\Omega} H_{\varepsilon}(Z(x, y)) dx dy \\
& + \lambda_1 \int_{\Omega} |U_0(x, y) - m_1(Z)|^2 H_{\varepsilon}(Z(x, y)) dx dy \\
& + \lambda_2 \int_{\Omega} |U_0(x, y) - m_2(Z)|^2 (1 - H_{\varepsilon}(Z(x, y))) dx dy
\end{aligned} \tag{6.2}$$

where H_{ε} represents the Heaviside function with respect to Z , and δ_{ε} is a regularized Dirac δ -function calculated by the derivative of the Heaviside function H_{ε} . The mean values m_1 and m_2 , depending on the evolution of the level set function, are calculated with the intensities of U_0 within and outside the unknown level set function Z as following:

$$\begin{cases} m_1(Z) = \text{average}(U_0) \text{ in } \{Z \geq 0\} \\ m_2(Z) = \text{average}(U_0) \text{ in } \{Z < 0\} \end{cases} \tag{6.3}$$

For simplicity, $m_1(Z)$ and $m_2(Z)$ will be used heretofore to represent these two values within and outside Z . For the purpose of the minimization with respect to Z , H_{ε} can be defined as:

$$H_{\varepsilon} = \frac{1}{2} \left(1 + \frac{2}{\pi} \arctan\left(\frac{Z(x, y)}{\varepsilon}\right) \right) \tag{6.4}$$

The minimization of Eq.6.2 can be solved by updating m_1 , m_2 and Z alternately. For a fixed Z value, the values of m_1 and m_2 are the region averages approximated by:

$$m_1 = \frac{\int_{\Omega} U_0(x, y) H_{\varepsilon}(Z(x, y)) dx dy}{\int_{\Omega} H_{\varepsilon}(Z(x, y)) dx dy} \tag{6.5}$$

$$m_2 = \frac{\int_{\Omega} U_0(x, y) (1 - H_{\varepsilon}(Z(x, y))) dx dy}{\int_{\Omega} (1 - H_{\varepsilon}(Z(x, y))) dx dy} \tag{6.6}$$

For fixed m_1 and m_2 values, a gradient descent equation can be formulated for the level set function Z with respect to an artificial time t as follows:

$$\frac{\partial Z}{\partial t} = \delta_{\varepsilon}(Z) \left[\mu_1 \cdot \text{div} \left(\frac{\nabla Z}{|\nabla Z|} \right) - \mu_2 \right. \tag{6.7}$$

$$\left. - \lambda_1 (U_0 - m_1(Z))^2 + \lambda_2 (U_0 - m_2(Z))^2 \right] + r(Z_t) \tag{6.8}$$

$$Z(0, x, y) = Z_0(x, y) \quad \text{in } \Omega \tag{6.8}$$

$$\frac{\delta_{\varepsilon}(Z)}{|\nabla Z|} \frac{\partial Z}{\partial \bar{e}} = 0 \quad \text{on } \partial \Omega \tag{6.9}$$

where \bar{e} is the outward normal to the boundary $\partial\Omega$. Then, the minimization of the discretized Eq.6.7 can be solved until convergence. A level regularization term r (Li, Kao, Gore, & Ding, 2008) is added to ensure the convergence of the level set function Z , which can be defined as:

$$r(Z_t) = \int_{\Omega} \frac{1}{2} (|\nabla Z_t(x, y)| - 1)^2 dx dy \quad (6.10)$$

where the subscript t means that the regularization term r is evaluated and updated with respect to the level set function Z at each time instant t .

6.3.3 Approximation of Intensity with gPC

In this work, the pixels' intensities are described as random variables to account for uncertainties in the cells' image. The generalized polynomial chaos (gPC) expansion (Xiu D. , 2010) is used to approximate the pixel' intensities U in a grayscale image U_0 with finite second order moment as follows:

$$U(\omega) = \sum_{i=0}^{\infty} u_i \Phi_i(\zeta(\omega)) \quad (6.11)$$

where U is the pixels' intensities of a given image, $\zeta = \{\zeta_1, \zeta_2, \dots\}$ is a set of independent, identically distributed random variables with known probability density functions (PDFs), which is defined by a random event ω . The $\Phi_i(\zeta)$ are multi-dimensional orthogonal basis functions of ζ , and u_i is the gPC coefficient multiplying each corresponding basis function. For practical implementation, Eq.6.11 is truncated to a finite number of terms such that:

$$U(\omega) \approx \sum_{i=0}^p u_i \Phi_i(\zeta(\omega)) \quad (6.12)$$

where p is the total number of terms used to represent a priori known distribution of pixels' intensities U .

For illustration, the segmentation result obtained with the *active contour without edges* (deterministic level set) method for an image containing three cells is schematically shown in Fig.6.2 (a); while Fig.6.2 (b) shows the pixels within a small inset in Fig.6.2 (a), displaying the cell boundary and the gPC approximations of pixels' intensities.

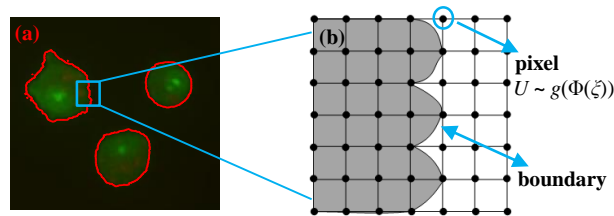


Figure 6.2 Visual interpretation of stochastic images

The explicit gPC representation of pixels' intensities U in Eq.6.12 is utilized to propagate the uncertainty onto the evolution of the level set function Z as follows. Assuming that the evolution of the level set function Z at each (artificial) time instant t is defined by Eq.6.7 as:

$$Z_t = f(U_0, t, Z_0) \quad (6.13)$$

where f represents the image segmentation operator U_0 denotes a grayscale image to be segmented, and Z_0 defines an initially assumed level set function for a given image U_0 .

To quantify the effect of stochastic changes in pixels' intensities on the evolution of the level set function Z , a gPC approximation of Z at each time instant t is employed and defined as follows:

$$Z_t(\omega) \approx \sum_{i=0}^P \zeta_{i,t} \Phi_i(\zeta(\omega)) \quad (6.14)$$

where $\zeta_{i,t}$ are the gPC coefficients used to describe the level set function at time t , P is the number of terms used to approximate the level set function Z and it is a function of a fixed polynomial degree p and a fixed number of random variables $\zeta = \{\zeta_1, \zeta_2, \dots, \zeta_n\}$. The total number of terms P in the gPC approximation in Eq. 6.14 is defined as:

$$P = ((p+n)!/(p!n!)) - 1 \quad (6.15)$$

As seen in Eq. 6.15, P increases significantly with respect to the number of random variables n in ζ and the required polynomial degrees p in Eq. 6.12.

6.3.4 Stochastic Level Set Function based Segmentation

Assuming the pixels' intensities are described by random variables as defined in Eq. 6.12, the stochastic level set function based segmentation can then be derived by replacing all quantities in Eq. 6.7 with their corresponding gPC expansions. Substituting the gPC expansions, i.e., Eqs. 6.12 and 6.14 into Eq. 6.13 results in the following expression:

$$Z_t(\omega) \approx f(U(\omega), t, Z_0(\omega)) \\ \sum_{i=0}^P \zeta_{i,t} \Phi_i(\zeta(\omega)) \approx f\left(\sum_{i=0}^P u_i \Phi_i(\zeta(\omega)), t, \sum_{i=0}^P \zeta_{i,0} \Phi_i(\zeta(\omega))\right) \quad (6.16)$$

where $\{\zeta_{i,0}\}$ are the gPC coefficients of the initially assumed level set function $Z_0 = \{\zeta_{i,0}\}$ ($t = 0$ and $i = 1, \dots, P$).

Using Galerkin projections, the gPC coefficients of level set function Z are calculated by projecting both sides of Eq. 6.16 onto each of the polynomial chaos basis function $\Phi_i(\zeta)$ as follows (Xiu D. , 2010):

$$\langle Z_t(\omega), \Phi_i(\zeta) \rangle = \langle f(U(\omega), t, Z_0(\omega)), \Phi_i(\zeta) \rangle \quad (6.17)$$

where ' $\langle \cdot, \cdot \rangle$ ' denotes the inner product between two vectors defined as below:

$$\langle \psi_1, \psi_2 \rangle = \int \psi_1 \psi_2 \varpi(\xi) d\xi \quad (6.18)$$

where the integration is conducted over the entire domain of the random variables ξ . The weighting function $\varpi(\xi)$ can be chosen with respect to the polynomial basis functions used to represent ξ so as the result of Eq. 6.18 is either 0 or 1. The type of polynomial is chosen for satisfying orthogonality according to the Wiener Askey scheme (Xiu D. , 2010). For example, Hermite polynomials are chosen as the optimal basis functions for normally distributed ξ and Laguerre polynomials are used for Gamma distributed ξ .

Once the gPC coefficients of the representation in Eq. 6.12 are available, it is possible to compute the gPC coefficients of the level set function Z at any given time t with Eq. 6.16 - Eq. 6.18. Then, the expectation, variance and other higher order statistical moments of the level set function Z can be calculated analytically. For example:

$$E(Z_t) = E\left[\sum_{i=0}^P \zeta_{i,t} \Phi_i\right] = \zeta_{0,t} E[\Phi_0] + \sum_{i=1}^P E[\Phi_i] = \zeta_{0,t} \quad (6.19)$$

$$\begin{aligned} \text{var}(Z_t) &= E\left[(\zeta_t - E(\zeta_t))^2\right] = E\left[\left(\sum_{i=0}^P \zeta_{i,t} \Phi_i - \zeta_{0,t}\right)^2\right] \\ &= E\left[\left(\sum_{i=1}^P \zeta_{i,t} \Phi_i\right)^2\right] = \sum_{i=1}^P (\zeta_{i,t})^2 E(\Phi_i)^2 \end{aligned} \quad (6.20)$$

The availability of analytical formulae that permits quick computations of the mean and variance of the level set function Z as described in Eq. 6.19 and Eq. 6.20 are the main rationale for using the gPC in this work, since the stochastic segmentation algorithm takes into account the variabilities of intensities while progressively updating the level set function Z . The first gPC coefficient $\zeta_{0,t}$ in Eq. 6.14, representing the mean value of the stochastic level set function Z , is used to segment the objects from the background, while the higher order gPC coefficients are used for estimating a probabilistic distribution of the curvature along the boundaries of cells.

From the substitution of Eq. 6.12 and Eq. 6.14 into Eq. 6.7, the stochastic level set segmentation model can be derived as follows:

$$\begin{aligned} \frac{\partial Z(\xi)}{\partial t} &= \delta_\varepsilon(Z(\xi)) \left[\mu_1 \cdot \text{div} \left(\frac{\nabla Z(\xi)}{|\nabla Z(\xi)|} \right) - \mu_2 \right. \\ &\quad \left. - \lambda_1 (U_0 - m_1(Z(\xi)))^2 + \lambda_2 (U_0 - m_2(Z(\xi)))^2 \right] \\ &\quad + r(Z(\xi)) \end{aligned} \quad (6.21)$$

where $Z(\xi)$ is the unknown stochastic level set function approximated with Eq. 6.14, $\delta_\varepsilon(Z(\xi))$ is the derivative of the stochastic Heaviside function $H_\varepsilon(Z(\xi))$ that is now defined as follows:

$$H_\varepsilon(Z_t(\xi)) = \frac{1}{2} \left(1 + \frac{2}{\pi} \arctan\left(\frac{Z_t(\xi)}{\varepsilon}\right) \right) \quad (6.22)$$

The mean values in Eq. 6.21, i.e., $m_1(Z(\xi))$ and $m_2(Z(\xi))$, can be calculated by averaging the pixels' intensities inside and outside the level set function $Z(\xi)$ as:

$$\begin{cases} m_1(Z(\xi)) = \text{average}(U(\xi)) \text{ in } \{\zeta_{0,t} \geq 0\} \\ m_2(Z(\xi)) = \text{average}(U(\xi)) \text{ in } \{\zeta_{0,t} < 0\} \end{cases} \quad (6.23)$$

where $U(\xi)$ is the gPC approximation of pixels intensities for a given image, $\zeta_{0,t}$ represent the mean value of the unknown stochastic level set function Z that is iteratively updated at each artificial time t .

The calculation of $Z(\xi)$ consists of two steps: (i) the calibration of mean and variance of pixels' intensities, and (ii) the evolution of $Z(\xi)$ with Eq. 6.16 - Eq. 6.18. In the calibration step, the mean and variance of the pixels' intensities have to be calibrated with part of the data, i.e., a subset of the available images. It is assumed for simplicity that the intensities of pixels inside the boundary (zero level set function $\zeta_{0,t}$) depend on one random variable and all the pixels' intensities outside the boundary depend on another random variable. Then, the mean values and variances of pixels' intensities in the segmented regions (within or outside the boundary) for a set of given stochastic images are calibrated from the following optimization problem:

$$\min_{\kappa} J = \sum_{i=1}^n (\mathcal{G}_{1,i} - \nu_{1,i})^2 + \sum_{i=1}^n (\mathcal{G}_{2,i} - \nu_{2,i})^2 \quad (6.24)$$

where the decision variables (κ) of this optimization consist of the means and variances of pixels' intensities inside (or outside) the boundary. For each iteration in Eq. 6.24, $\mathcal{G}_{1,i}$ and $\mathcal{G}_{2,i}$ are the mean and variance of pixels' intensities measured inside (or outside) the boundary corresponding to the zero values of the stochastic level set function $Z(\xi)$. The terms $\nu_{1,i}$ and $\nu_{2,i}$ are the measured mean and variance of the pixels' intensities inside (or outside) the boundary that is numerically calculated with the original deterministic active contour algorithm as defined in Eq. 6.7.

Following the calibration step, the gPC representations of the pixels intensities given by Eq. 6.12, where the mean and variance have been calibrated in the first step, are substituted into Eq. 6.21. Then, a Galerkin projection of each side of Eq. 6.21 onto each basis polynomial function $\{\Phi_i(\xi)\}$ can be conducted. The application of the Galerkin projection operation requires integrating Eq. 6.21 with respect to a set of appropriate selected polynomial basis functions. Using the orthogonality property of the basis functions $\Phi_i(\xi)$, these integrations are possible for monomial or polynomial terms. The integration of non-monomial terms, however, requires additional operations (Debusschere, Najm, Pebay, Knio, Ghanem, & Matre, 2004). For example, the term $\text{div}(\nabla Z(\xi)/|\nabla Z(\xi)|)$ in Eq. 6.21, representing the geometric attributes along the boundary, is approximated by forward, backward and central difference (Getreuer, 2012) in both x and y dimensions as below:

$$\begin{aligned}
\operatorname{div} \left(\frac{\nabla Z(\xi)}{|\nabla Z(\xi)|} \right) &= \partial_x \left(\frac{\partial_x Z(\xi)}{\sqrt{(\partial_x Z(\xi))^2 + (\partial_y Z(\xi))^2}} \right) \\
&\quad + \partial_y \left(\frac{\partial_y Z(\xi)}{\sqrt{(\partial_x Z(\xi))^2 + (\partial_y Z(\xi))^2}} \right) \\
&\approx \nabla_x^- \frac{\nabla_x^+ Z(\xi)}{\sqrt{(\nabla_x^+ Z(\xi))^2 + (\nabla_y^0 Z(\xi))^2 + \eta^2}} \\
&\quad + \nabla_y^- \frac{\nabla_y^+ Z(\xi)}{\sqrt{(\nabla_x^0 Z(\xi))^2 + (\nabla_y^+ Z(\xi))^2 + \eta^2}}
\end{aligned} \tag{6.25}$$

where ∇_x^+ denotes forward difference in the x direction, ∇_x^- is backward difference, and $\nabla_x^0 := (\nabla_x^+ + \nabla_x^-)/2$ means the central difference, and similarly in the y dimension. A small positive number η is introduced in the denominators in Eq. 6.25 to prevent division by zero.

Following the Galerkin projection operation and using the orthogonality properties, Eq. 6.21 is then transformed into a system of coupled deterministic equations as:

$$\begin{aligned}
\sum_{i=0}^P \frac{\partial \zeta_i(\xi)}{\partial t} \langle \Phi_i^2 \rangle &= \sum_{j=0}^P \sum_{i=0}^P \{ \delta_\varepsilon(\zeta_i(\xi)) [\mu_1 \cdot \operatorname{div} \left(\frac{\nabla \zeta_i(\xi)}{|\nabla \zeta_i(\xi)|} \right) - \mu_2 \\
&\quad - \lambda_1 (U_0 - m_1(Z(\xi)))^2 \\
&\quad + \lambda_2 (U_0 - m_2(Z(\xi)))^2] + r(\zeta_i(\xi)) \} \cdot e_{ij}
\end{aligned} \tag{6.26}$$

where $e_{ij} = \langle \Phi_i, \Phi_j \rangle$ and the results is either one or zero. The gPC coefficients of the stochastic level set function Z can be solved numerically from Eq. 6.26.

The steps of the proposed stochastic level set function based image segmentation algorithm, involving initialization, calibration and segmentation sections, are summarized in Fig.6.3.

Initialization

- (1) Initialize fixed parameters ($\mu_1, \mu_2, \lambda_1, \lambda_2, \gamma_1, \gamma_2$ and time-step Δt).
- (2) Choose polynomial basis function and the highest order of gPC approximation of random variables (pixel values).
- (3) Assign gPC coefficients to the initial level set function Z_0 .
- (4) Generate symbolic gPC model for the evolution of level set function Z with (26).
- (5) Determine the stop criterion (σ_{tol}) between $Z^{(n+1)}$ and $Z^{(n)}$, and n means the n^{th} iteration in the segmentation algorithm ($n = 0, 1, 2, \dots$).

Model calibration

- (i) Conduct segmentation for a given image with deterministic level set algorithm.
- (ii) Calculate the values of $v_{1,i}$ and $v_{2,i}$ in cost function (27) using the boundary obtained in (i).
- (iii) Assign initial values to the decision variable vector κ .
- (iv) Execute optimization (27).

Segmentation algorithm

- (a) Re-write each of the pixels inside (and/or outside) the initial level set function Z_0 with the gPC approximation.
- (b) Execute *for*-loop using the symbolic gPC model:

for $n = 0, 1, 2, \dots$

 Conduct the following procedures repeatedly:

 Compute means m_1, m_2, v_1 and v_2 using the gPC coefficients of Z_n .

 Update Z_{n+1} with one time-step with the stochastic level set cost function (26);

if $|Z_{n+1} - Z_n| < \sigma_{tol}$

 Terminate *for*-loop and display the final level set function, which is the first gPC coefficient representing the mean value of Z_n . The higher order gPC coefficients are utilized for classification algorithm as explained in next Section.

else

$n = n + 1$;

end

Figure 6.3 Stochastic segmentation algorithm

It should be noted that the stochastic evolution of Eq. 6.21 may be solved using a Monte Carlo (MC) approach based on sampling from statistical distributions representing the pixel intensities in Eq. 6.21. However, such calculations will be prohibitively long while computational speed is essential for the current application, since the goal is to analyze a large number of images automatically for throughput studies.

6.4 Automated Classification of Cells

6.4.1 Feature Extraction

The differentiation of cells into apoptotic or normal is based on two specific morphological features calculated from the images: (i) a measure of the length of cell's boundary and (ii) the variability in curvature along the boundary. For comparison purposes, these features are calculated with both the original deterministic level set algorithm and the stochastic level set method proposed in this work.

The length of boundary is given by the total number of pixels used to define the boundary. For the deterministic level set method, it is computed with the number of pixels corresponding to the zero level set function values. For

the stochastic level set method, the number of pixels is computed with the zero mean value of the stochastic level set function, i.e., $\zeta_0 \approx 0$ in Eq. 6.26.

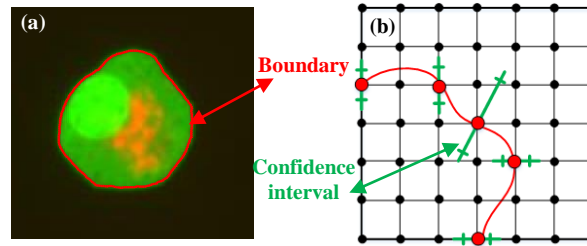


Figure 6.4 Sketch of the morphological feature along the boundary

The curvature is given by $div(\nabla Z(\xi)/|\nabla Z(\xi)|)$ which is one of the terms of the level set equation in Eq. 6.7 and Eq. 6.21. This quantity can be calculated both as a deterministic quantity and as a probabilistic quantity as given by its gPC representation. To assess the variance of the curvature with the deterministic level set, the difference between the largest value and the smallest value obtained along the boundary is calculated. When using the gPC representation of curvature, the largest curvature value was equal to the largest mean value plus one variance and the smallest curvature value was equal to the lowest mean value minus one variance. The variance can be computed with Eq. 6.20.

6.4.2 SVM based Differentiation

Using the values of contour length and variability in curvature, a support vector machine (SVM) (Burgers, 1998) classification model was developed to differentiate the physiological states of cells, i.e., distinguish apoptotic cells from normal cells. A set of training images were first collected. Following consensus decision-making ideas (Aurum & Wohlin, 2003), each image of cells is then characterized as either normal or apoptotic cells based on consensus among five different experimentalists in combination with measurements of percentage of apoptotic cells at the time the image was taken as determined by flow cytometry (Meshram, Naderi, McConkey, Budman, Scharer, & Ingalls, 2011). From the images used for training, the parameters of the SVM model were optimized with the Matlab[®] statistics and machine learning toolbox. Then, the trained SVM model was applied to new images that were not used for model training to classify new images according to their physiological state.

6.5 Results and Discussion

6.5.1 Model Calibration

The accuracy of the gPC representations used to solve the stochastic level set function based image segmentation depends on the polynomial basis function and the number of random variables. An appropriate basis function should be selected based on the statistical distribution of the uncertainty while ensuring orthogonality of the gPC expansions (Xiu D. , 2010).

Fig.6.5 shows the segmentation results with the deterministic level set algorithm and the probability density function (PDF) of intensities of pixels along the boundary. As seen in Fig.6.5 (a), the black line describes the boundary that segments the cell from the image background. The boundary corresponds to the points where $Z \approx 0$ in Eq. 6.7. The PDF profile for pixels' intensities defining this boundary can be obtained by using a binning algorithm, where each bin will include the number of pixels that exhibit intensities within a particular range of intensity values. Fig.6.5 (b) shows a PDF profile of pixels with 50 bins. It can be observed that the intensities of pixels follow approximately a normal distribution. Correspondingly, Hermite polynomials are chosen as basis function to capture the variability around the boundary since they are especially targeted to describe normal distributions (Xiu D. , 2010).

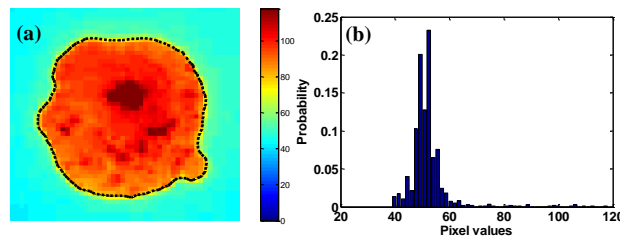


Figure 6.5 Segmentation results and PDF of pixel intensities defining boundary

6.5.2 Image Segmentation with One Random Variable

Fig.6.6 (a) shows an original fluorescence microscopy image with the corresponding pixels' dimensions indicated in the upper left corner. The segmentation result with 3 cells inside the grayscale image is schematically shown in Fig.6.6 (b). The green line denotes the results obtained with the stochastic level method, while the red line represents the boundaries computed with the deterministic method. Fig.6.6 (c) displays the pixels' intensities in a small window of the two-dimensional matrix that is used to store the image, and Fig.6.6 (d) shows the PDF profile of the intensities within the background based on the segmentation results.

As can be observed from Fig.6.6 (d), the intensities of pixels in the background vary very little and ~92.39% of them can be found in the first bin of the PDF profile. Therefore, for simplicity, it was assumed that the intensities in the background are constant. Accordingly only one random variable had to be used for describing the pixels' intensities within the boundary of the cells while the pixels' intensities in the background were represented by a constant value. Thus, the gPC representation in this case study is only applied to describe the intensities of pixels within cellular regions, while the intensities of pixels in the background are described by their original deterministic intensity values.

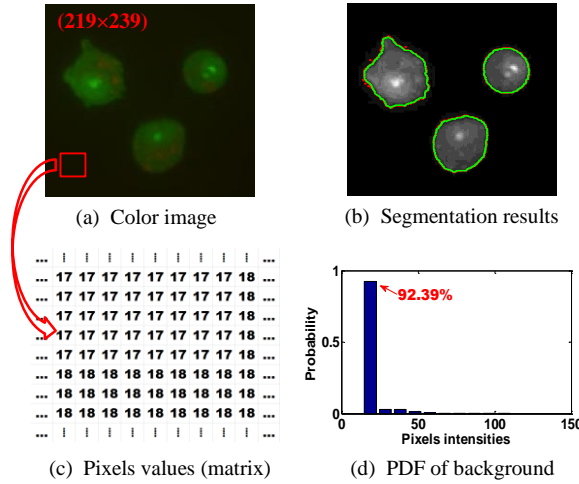


Figure 6.6 Visual illustration of pixels intensities in the background

As seen in Fig.6.6 (b), the results obtained with the stochastic and deterministic methods are in a good agreement. To compute the boundaries as shown in Fig.6.6 (b), the parameters for the stochastic method in this experiment are chosen as: $\mu_1 = \mu_2 = 1$, $\lambda_1 = \lambda_2 = 1$, and time-step Δt is 0.1.

To illustrate the efficiency of the proposed probabilistic segmentation approach, the stochastic level set function with one random variable is applied to a more complicated image. Fig.6.7 (a) displays an original image as obtained from the microscope with the corresponding pixels' dimensions. Fig.6.7 (b) shows the corresponding grayscale image and the initial level set function chosen to start the segmentation algorithms. For the stochastic level set method in this work, the initial level set function in Fig.6.7 (b) represents the mean values of the zero order gPC coefficients while the gPC coefficients of the higher order terms are set to zero. In Fig.6.7 (c), segmentation results obtained with 20 iterations are shown for both methods, and Fig.6.7 (d) shows the segmentation results with 150 iterations. The green line denotes the results obtained with the stochastic level function, while the red line is the results obtained with the deterministic level set method. It is clear that in terms of segmentation both the deterministic and stochastic methods provide similar accuracy. Further studies were conducted to investigate the computational time for both methods. For 150 iterations, the processor time for the deterministic level set method is ~ 158 s while ~ 220 s are required for the stochastic level set method with one random variable representing the variabilities of the intensities within the cells.

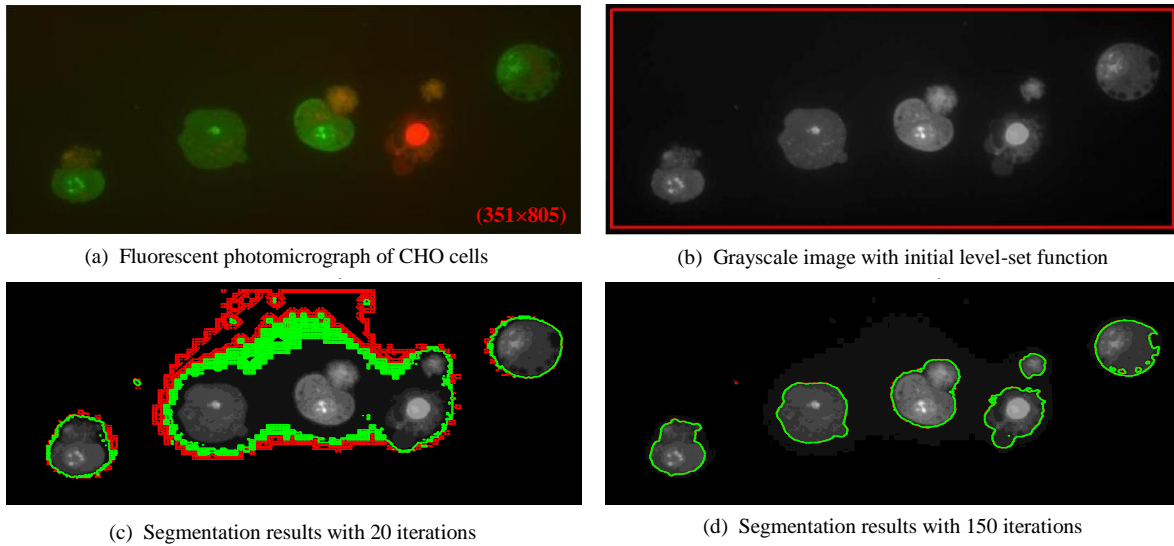


Figure 6.7 Segmentation results with deterministic and stochastic level set algorithms

6.5.3 Morphological Features

For the model training of the SVM differentiation model, the deterministic level set algorithm and the stochastic level set method are applied to 100 images of cells, in which 50 images are normal cells and 50 images are apoptotic cells. The pixels' dimensions of the original cells' images are $\sim 220 \times 220$. In view that down-sampling is an effective way to speed up computations the deterministic method and the proposed stochastic method are compared for different down-sampling frequencies.

In this case study, images of cells are down-sampled to pixels dimensions of 100×100 , 50×50 and 30×30 , respectively. Following the application of the segmentation algorithms, a feature vector is calculated for each image composed of the two morphological features mentioned above, i.e., the curvature along the boundary and the length of the boundary. Fig.8 shows the histogram of the curvature calculated with the stochastic level set method for a few images with the original pixels dimensions, and Tables 6.1 and 6.2 provides some of the feature vectors calculated with the deterministic and stochastic level set methods for the same down-sampling frequency.

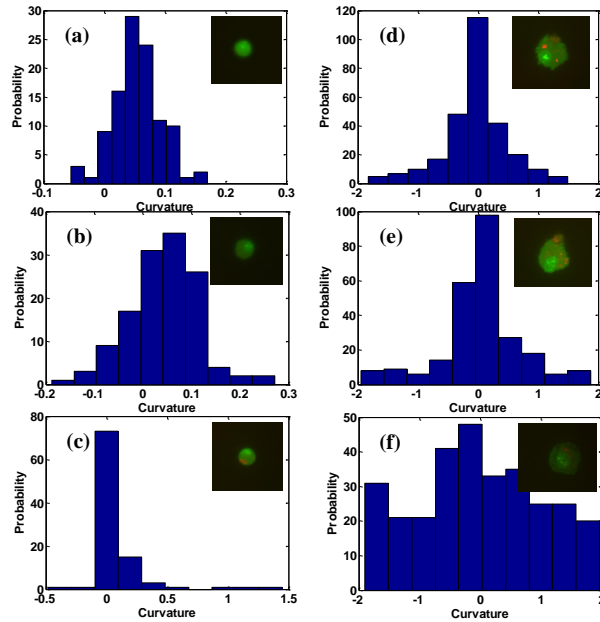


Figure 6.8 Histograms of curvature for apoptotic and normal cells

Table 6.1 Examples of feature vector (apoptosis)







Samples	Method	Length	Curvature
	Stochastic	373	3.9930
	Deterministic	403	3.3237
⋮	⋮	⋮	⋮
	Stochastic	299	3.8689
	Deterministic	323	3.7849
⋮	⋮	⋮	⋮
	Stochastic	405	4.3569
	Deterministic	437	4.3138

Table 6.2 Examples of feature vector (normal)

Samples	Method	Length	Curvature
	Stochastic	149	1.5208
	Deterministic	155	1.1930
⋮	⋮	⋮	⋮
	Stochastic	181	2.2618
	Deterministic	185	2.0592
⋮	⋮	⋮	⋮
	Stochastic	207	1.4592
	Deterministic	215	1.1546

Using cells' images with the original pixels' dimensions, the first column in Fig.6.8 shows the histogram of the curvature for three normal cells, while the second column displays the results for three apoptotic cells. As seen in Fig.6.8, the range of the curvature values for apoptotic cells is larger than for normal cells thus justifying the use of the maximal differences in curvature to differentiate cells.

6.5.4 Differentiation of Cells States

To compare the efficiency of the deterministic versus the stochastic segmentation methods for different down-sampling resolutions, a classification rate is defined as follow:

$$r_{rate} = d_i / D_T \quad (6.28)$$

where d_i means the number of testing images that have been correctly identified and D_T is the total number of samples used for experiments.

After model training is completed, 100 cells image samples that have not been used in the model training step are utilized to test the SVM classification model, where 50 images are of normal cells and 50 images are of apoptotic cells. The testing images with original pixels dimensions of $\sim 220 \times 220$ are down-sampled to a size of 100×100 , 50×50 and 30×30 , respectively.

For the purposes of illustration, Fig.6.9 shows the results of differentiation using 10 testing samples with the original pixels' dimensions. These testing samples are randomly chosen from the 100 cells images, 5 of them are normal cells and the others are apoptotic cells. In Fig.6.9, "N" denotes the normal cells and "A" means the apoptotic cells. For example, the testing sample in the red circle in Fig.6.9 (a) represents a misclassification, i.e., the normal cell has been misclassified as an apoptotic cell.

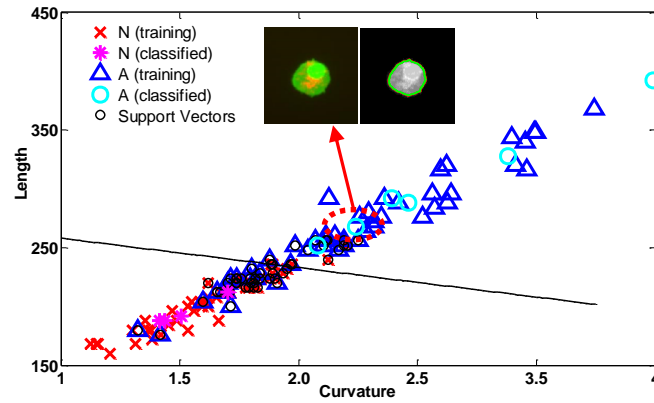


Figure 6.9 Visual illustration of normal and apoptotic testing cells

To compare the efficiency, the differentiation rates r_{rate} for four aforementioned different pixels dimensions are shown in Table 6.3 with the deterministic and the stochastic methods. The computational time and the differentiation rate are compared for different down-sampling resolutions and original sizes of images. The differentiation rate in Table 3 is calculated with respect to the total number of samples used for testing, i.e. 100.

From Table 6.3, it can be seen that the stochastic level set method shows better differentiation performance, as compared with the deterministic level set method. For example, as can be seen in the second row of results, when the images with the original sizes of $\sim 220 \times 220$ are down-sampled to pixels dimensions of 100×100 , 4 normal cells

and 9 apoptotic cells are misclassified with the stochastic method, which give a classification rate of ~87%. For the deterministic method, 15 testing samples (5 normal cells and 10 apoptotic cells) are misclassified that gives a differentiation rate of ~85%, which is ~2% lower than the stochastic method.

Table 6.3 Summary of differentiation rate

Method Size	Stochastic Level set		Deterministic Level set	
	r_{rate}	Time (s)	r_{rate}	Time (s)
~220*220	0.88	8.16	0.87	5.77
100*100	0.87	4.26	0.85	2.86
50*50	0.84	2.21	0.81	1.53
30*30	0.81	1.62	0.76	1.17

As expected, the differentiation rates decreased, when the testing images are further down-sampled to pixel dimensions of 30*30, due to the lost information. For example, as can be seen from the last row of results, the differentiation rate decreased by ~11%, from ~87% to ~76%, if the deterministic level set method is used to distinguish apoptotic versus normal cells where 9 normal cells and 15 apoptotic cells have been misclassified. On the other hand, the differentiation rate r_{rate} only decreased by ~7%, from ~88% to ~81%, when the stochastic level set method is utilized, for which 5 normal cells and 14 apoptotic cells are misidentified.

These observations confirm that the stochastic level set based algorithm provides a more robust differentiation of cells states. Studies are also conducted to investigate the computational time for each case study. When images of cells with original pixels dimensions of ~220*220 are down-sampled to the size of 30*30, it takes ~1.17s with the deterministic method for 50 iterations, while ~1.62s is required with the stochastic method for the same number of iterations.

6.6 Conclusion

In the present work, a methodology has been developed for throughput screening studies to distinguish apoptotic from normal viable Chinese Hamster Ovary (CHO) cells. The study addresses the identification of irregular boundaries as occurring for apoptotic cells by using a stochastic level set algorithm to calculate the evolution of a stochastic level set function. From the resulting gPC representations of the level set functions, it is possible to estimate the boundary of the cells, the length of the boundary and the variability in curvature along the boundary. Using the information, a support vector machine (SVM) classifier has been developed. The combination of the boundary length and curvature obtained from the stochastic level set segmentation of images and the SVM classifier is shown to be an efficient tool to classify cells into normal versus apoptotic. The proposed stochastic level set approach is shown to be more robust in terms of differentiation accuracy than the deterministic level set algorithm, when the images are down-sampled to reduce computation time.

Chapter 7

Concluding Remarks and Future Work

7.1 Overview

Models must account for uncertainty in order to provide precise results. Systematic and computationally efficient uncertainty analysis is the key to evaluate the performance of underlying engineering problems of interest. The statistical methods such as Monte Carlo simulations are one of the most popular approaches to solve problems with stochastic descriptions of uncertainty but they are computationally challenging (Spanos & Zeldin, 1998). Alternatively, stochastic spectral methods such as the generalized polynomial chaos (gPC) expansion have emerged as a promising computationally efficient technique that can be used for uncertainty quantification and propagation (Xiu D. , 2009). Using this technique, the probability distributions of the model outputs can be calculated analytically from the probability distributions associated with the input variables through mathematical models in combination with Galerkin projection operations.

The increasing size and complexity of modern process plants has made automation essential for their successful operations. Automated fault detection and diagnosis (FDD) is one of the key areas of chemical processes due to their potential for providing safer and more profitable operations (Isermann R. , 2006). However, disturbance such as parametric uncertainty can affect the performance of FDD algorithms (Chiang, Russell, & Braatz, 2008). Additionally, most of the available FDD tools are implemented at a supervisory hierarchical level above the control system and use measurements that are also used for feedback control. These two activities have competing objectives. For example, variables that are perfectly controlled in a closed loop control system do not exhibit large variability, which may results in lower detection rate of faults.

Fluorescence microscopy is a well-developed tool to study in vitro cells behavior. Accurate and automatic analysis of cells images such as Chinese Hamster Ovary (CHO) cells can be very useful. Mammalian cells are prone to apoptosis (programmed cell death), which is a key metabolic event that restricts the growth of cells and decrease the productivity in a bioreactor (Waters, 2009). However, apoptotic cells in images may exhibit highly variable values of the morphological features that characterize apoptosis due to the dynamic nature of this phenomenon (Taatjes, Sobel, & Budd, 2008). Thus, it is necessary to develop new image processing and quantitative analysis method that can automatically differentiate apoptotic from normal cells.

Motivated by the aforementioned discussion, the current work addresses these challenging problems through the following approaches:

- i- Formulated generalized polynomial chaos (gPC) method to propagate stochastic uncertainty from input faults and model parameters onto the outputs of the system. Then, faults are estimated by comparing the predicted and measured statistical properties of the outputs.

- ii- Formulated an optimization problem to find an optimal trade-off between a FDD algorithm and a feedback control, while taking into account model uncertainty and dynamic transients.
- iii- Developed of FDD algorithms based on the gPC approximate solution of nonlinear mathematical models with Maximum Likelihood and Bayesian Inference based estimators.
- iv- Developed an efficient algorithm to distinguish apoptotic versus normal cells using the identified morphological features of cells. The algorithm accounts for image uncertainty by propagating stochastic disturbances in the image through the level-set segmentation algorithm.

7.2 Concluding Remarks

One of the key limitations of model-based FDD algorithms is the presence of model uncertainty. The accuracy of fault detection can be affected by uncertainty in parameters of the model used for detection. Chapter 3 of this work addresses the topic by developing a FDD methodology for systems represented by first principles model where both parameters and faults are of a stochastic nature to account for uncertainty and for random disturbances. Using a generalized polynomial chaos (gPC) expansion, the proposed method allows for efficient quantification of stochastic changes and prompts propagation of these changes to the outputs that can be used for fault detection. To our knowledge, while the gPC has been applied before for modeling and control applications, it has not been used for FDD as in this current work. The key contribution of this work is that the proposed methods are successful in detecting and diagnosing both individual as well as simultaneous occurrences of multiple stochastic faults. Additionally, the detectability of fault/s near class boundaries is assessed with the Type I and Type II analysis. As compared with Monte Carlo simulations, the developed method is highly efficient in terms of computational time, thus showing the potential for addressing more complicated problems with large number of variables.

When a fault occurs, the objective is to detect and isolate it as promptly as possible. However, FDD methods that explicitly consider the dynamic transients in the presence of model uncertainty have not been addressed extensively in the literature. The detection of faults with steady state information based FDD algorithms may result in lower detection rate and higher mis-detection, when the measurements used for detection are collected during dynamic transients. Chapter 4 in this current work presented a two-level fault detection algorithm, i.e., (i) Level-1 algorithm based on steady state information and (ii) Level-2 algorithm based on dynamic transients. For the Level-2 algorithm, the gPC based solution is combined with either a Maximum Likelihood or a Bayesian Inference estimator to estimate the dynamic fault. The key feature of this method is the real-time detection of faults entering a system intermittently. The developed methods are demonstrated using a simulation of a nonlinear chemical plant with two continuously stirred tank reactors and a flash tank separator. As compared to the simulation based approaches such as MC simulations and Particle filter, the developed method is highly computationally efficient. This is a significant development since a main reason for the low acceptability of Particle Filter in industry is its high computational demand.

In Chapter 5, we address the problem of optimal simultaneous tuning of a FDD algorithm and the controller's parameters in the presence of time varying stochastic intermittent parametric faults. As compared with MC type sampling method such as Latin Hypercube sampling method, the main novelty of this chapter is that it addresses the effect of stochastic input on the measured variables explicitly by using the gPC approximations. This is shown to be very effective, since the variabilities of input and measured variables and the control laws have to be repeatedly computed within the optimization search. Thus, these computations will be prohibitive if they would be done with MC methods instead of gPC. In addition, the computational advantages open the possibility to perform this optimization online for re-configuring the controller according the occurring faults as in fault tolerant control (FTC) approaches. Additionally, for the purpose of fast fault detection, a set of dynamic gPC models can be generated based on a Maximum Likelihood criteria to recursively and dynamically diagnose the faults in a real-time fashion.

Chapter 6 investigates the classification problem of apoptotic versus normal cells. Since the image segmentation algorithms can be described by partial differential equations (PDEs), we realized that gPC solutions of the PDEs solving the segmentation problems can be used to address the effect of stochastic image noise on the segmentation results. The, particular morphological features identified from image segmentation are used to discern apoptotic cells from normal cells. Since the image segmentation can affect the accuracy of classification, a stochastic segmentation algorithm is developed by combining the gPC method with the active contours without edges method. Two specific morphological features, i.e., length of cell's boundary and curvature along cell's boundary, are computed and used as inputs to a support vector machine (SVM) classifier. As compared to the deterministic active contours without edges method, the developed method is shown to be more accurate in terms of differentiation accuracy between apoptotic and normal cells.

7.3 Future Work

7.3.1 Arbitrary Uncertainty Quantification and Propagation

The original homogeneous polynomial chaos expansion (Ghanem & Spanos, 1991) and the modified generalized polynomial chaos expansion (Xiu D. , 2010) can result in high computational efficiency and fast convergence. Both methods are based on an appropriate selection of orthogonal polynomials. For example, the gPC uses the Wiener-Askey polynomial chaos framework based on several orthogonal polynomials including the Hermite polynomial. For uncertain input distributions outside of the Wiener-Askey scheme, the Wiener-Askey polynomial chaos does converge, but the convergence rate might be slow for high dimensional complex system.

Thus, the appropriate selection of polynomial basis function for efficient quantification of uncertainty may be one of the possible direction to improve computational efficiency. For example, a multi-element generalized polynomial chaos has been recently developed to deal with stochastic input with arbitrary probability measures (Wan & Karniadakis, 2006). Based on a decomposition of the random space, a set of optimal orthogonal

polynomials using Stieltjes three-term recurrence procedure can be formulated. Another option is to combine Gram-Schmidt polynomial chaos with the polynomial chaos expansion (Witteveen & Bijl, 2006), in which the optimal set of orthogonal polynomials is computed for any type of input distribution. (The application of this method is illustrated as a case study in this work as can be seen in Appendix B). Moreover, uncertainty quantification of time varying uncertainty still poses a significant challenge, despite the success of the gPC methods (Gerritsma, Van der Steen, Vos, & Karniadakis, 2010). It is necessary to investigate the problem that the probability density function of uncertainty evolves as a function of time.

7.3.2 Integration of Plant Design, Control and Fault Diagnosis

The trade-off between control and fault detectability is investigated in this work to achieve a balance between these two activities, since they have competing objectives in particular in the presence of uncertainty. However, the objective is to seek the optimal controller's parameters to improve the detectability of intermittent faults. Further work should be conducted in the area of fault tolerant control, i.e., the optimal reconfiguration of control law in the event of faults to ensure the system to continue operating at a suboptimal levels, rather than breaking down completely.

Additionally, the consideration of the dynamic and control aspects during the early state of the plant design may lead to the improved controllability and operability. For example, appropriate design of chemical plants may reduce the effort in identifying and diagnosing the possible faults. Plant design criteria can be incorporated into the optimization to evaluate the effect of sensors' selection and distribution. This may maximize the information that can be ultimately used for the detection and control.

7.3.3 Image Segmentation and Classification

As an extension of active contour without edges method, the gPC expansion is combined with the level set functions to evolve the cells' boundaries. Such stochastic image segmentation can propagate the information about the gray value errors and uncertainty from the input image to the final segmentation results. With this tool, it is possible to provide information about the reliability and confidence intervals of the boundary. From the mathematical point of view, there are still areas for further improvement, since the gPC method tends to be computational demanding when the number of random variables increases. Another challenge is the visualization and segmentation of high dimensional stochastic color images. For example, the application of gPC in combination with image processing method in this work is a starting point and can be further improved.

In terms of feature extraction and selection, the identification of the most important feature is critical to minimize the classification error. For example, features can be selected based on mutual information criteria of maximize dependency and relevance (Peng, Long, & Ding, 2005).

Appendix A

Comparison of Stochastic Fault Diagnosis Algorithms

(Adopted from Du et al., 2015, Chemometrics and Intelligent Laboratory Systems, ready to submit)

A.1 Overview

This appendix presents a comparison study to identify and diagnose intermittent stochastic faults occurring in a dynamic multimode nonlinear process. The main objective is to develop efficient fault diagnosis algorithms in the presence of parametric uncertainty and to show the capabilities of each method. For the first principles' model based fault detection and diagnosis (FDD), a generalized polynomial chaos (gPC) expansion representing the stochastic input faults is employed to propagate the uncertainty onto the measured quantities. The resulting probability density functions (PDFs) of the measured variables can then be approximated and further used for fault diagnosis. For the statistical monitoring method, Gaussian process (GP) is used to map multivariate inputs into a univariate response, from which the fault can be inferred based on a minimum distance criterion. The performance of these methods is evaluated in terms of fault detection rate by applying them to a chemical plant of two continuously stirred tank reactors (CSTRs) and a flash tank separator. The proposed methods are successful in detecting and diagnosing intermittent faults in the presence of uncertainty.

A.2 Introduction

Early detection of abnormal events and malfunctions defined as faults is of great interest, since faults may affect the product quality and lead to economic losses (Gerlter, 1998). If a fault is detectable, the fault detection and diagnosis (FDD) system will provide symptomatic fingerprints, which in turn can be referred back to the FDD scheme to identify the root cause of the anomalous behaviour. Most of the available fault diagnosis algorithms can be broadly classified into three main classes (Isermann R. , 2005; Venkatasubramanian V. , Rengaswamy, Yin, & Kavuri, 2003): (i) Analytical methods that are solely based on first principles' models of process; (ii) Empirical models that use the historical process data; and (iii) Semi-empirical algorithms that combine these aforementioned two classes. Each of these methods has its own advantages and disadvantages depending on the specific problem (Isermann R. , 2006).

In terms of applications, many industrial processes are intrinsically nonlinear systems and they are operated at different operating conditions according to economic considerations (Haghani, Jeinsch, & Ding, 2014). Due to nonlinearity, the performance of linear FDD algorithms reported in literature (Li & Yang, 2012) may be inaccurate and lead to missed detection of faults, since the process model will change from one operating conditions to another. It is critical to develop new methodologies for the detection of faults in the context of nonlinear chemical processes with multiple operating conditions (Haghani, Jeinsch, & Ding, 2014).

Since most of the FDD schemes are invariably based on either first principle models or empirical models (Venkatasubramanian V. , Rengaswamy, Yin, & Kavuri, 2003), a main restrictive factor of an efficient FDD system is the model uncertainty. Such uncertainty may originate from either intrinsic time varying phenomena of model parameters or may result from inaccurate measurements due to noise. Models with large uncertainties make the detection and isolation of small faults very difficult. However, the step of quantifying and propagating the uncertainties onto the measured quantities that can be used for fault detection is typically omitted in reported FDD studies, leading to a loss of useful information arising from these uncertainties (Patton, Frank, & Clark, 2010). Moreover, the quantitative analysis of faults detectability in the presence of uncertainty provides more information to improve FDD algorithms. For example, engineering effort can be saved, if it is impossible to detect a fault due to uncertainties such as large measurement noise (Eriksson, Frisk, & Krysander, 2013).

To evaluate the effect of uncertainty on FDD, one possibility is to propagate stochastic variations with Monte Carlo (MC) simulations (Harrison, 2010), which involve drawing a large number of samples and running the models with each of these samples. However, approaches such as MC simulations are computationally prohibitive especially for complex processes as shown later in the manuscript. To improve the computational efficiency, this paper presents and compares two FDD algorithms in the presence of uncertainties. The uncertainty includes the parametric uncertainty of a process and measurement noise. In addition, the faults in this current work are stochastic perturbations superimposed on intermittent step changes in specific input variables for a nonlinear chemical plant. For the first FDD method, generalized polynomial chaos (gPC) (Ghanem & Spanos, 1991; Xiu D. , 2010) in combination with first principles' process models are used to quantify and propagate the uncertainty onto the measured quantities, which can be used for the detection of faults. For the second method, a surrogate metamodel is developed with Gaussian Process (GP) (Rasmussen & Williams, 2006), which is calibrated with a minimal model adjustment algorithm and can be used estimate the value of fault.

The objective in this work is to address the capabilities of these methods and propose a possible strategy to overcome their limitation by combing their outcomes. For this purpose, the performance of each method is evaluated in terms of fault detection rate in the context of stochastic parametric input faults. These faults occur intermittently with stochastic perturbations, i.e., the mean value of faults switch between the non-faulty and faulty operating conditions in a random fashion. For simplicity, the stochastic perturbations are assumed to be time-invariant uncertainties. Thus, the key is to identify and diagnose these step changes in the presence of the random perturbations in the parametric input faults, using available measurements corrupted with measurement noise.

To summarize, the contributions in this work are: (i) the use, in the context of fault detection and diagnosis, of a gPC model and a GP model for uncertainty propagation and quantification for a complex nonlinear system; (ii) the comparison of analytical and empirical methods for the detection of faults of a stochastic nature; and (iii) an ensemble of these methods to overcome limitations instead of the standalone application of each method.

This appendix is organized as follows. In Section A.3, the formulation of a fault detection problem is presented followed by the theoretical background of the gPC and GP theories. The fault detection and diagnosis (FDD) algorithms are explained in Section A.4. A nonlinear chemical plant with two continuously stirred tank reactors and a flash tank separator is introduced as a case study in Section A.5. Analysis and discussion of the results are given in Section A.6 followed by conclusions in Section A.7.

A.3 Problem Formulation and Theoretical Background

A.3.1 Formulation of Unknown Stochastic Faults

Assuming a system subject to stochastic parametric input faults can be described by a set of nonlinear ordinary differential equations (ODEs) as following:

$$\begin{aligned} \dot{\mathbf{x}} &= f(t, \mathbf{x}, \mathbf{u}; \mathbf{g}) \\ 0 \leq t \leq t_f, \mathbf{x}(0) &= \mathbf{x}_0 \end{aligned} \quad (\text{A.1})$$

where the vector $\mathbf{x} \in \mathbf{R}^n$ represents the system states (measured quantities) with initial conditions $\mathbf{x}_0 \in \mathbf{R}^n$ over time domain $[0, t_f]$, and \mathbf{u} denotes the known (measurable) inputs of the system. The vector $\mathbf{g} \in \mathbf{R}^{n_g}$ is the unknown (unmeasured) stochastic time varying input faults of interest, which has to be detected by a FDD algorithm. The function f is assumed to be a fundamental model of the process that can be developed from first principles. The input faults \mathbf{g} considered in this current work consist of stochastic perturbation around a specific set of mean values as described in Fig.A.1 (a).

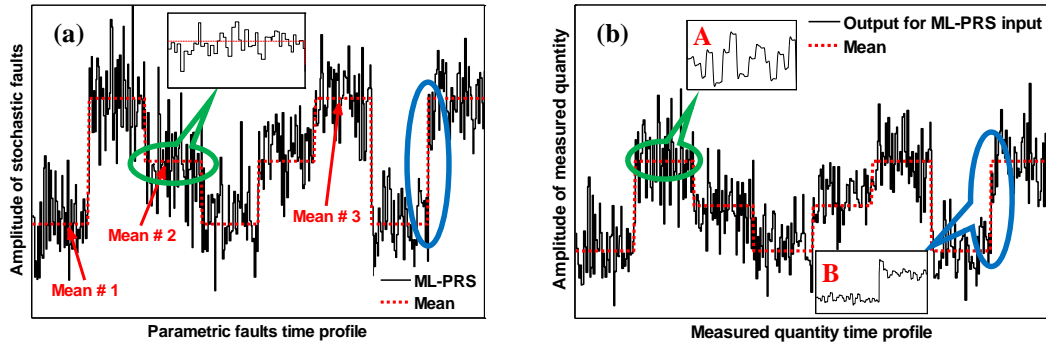


Figure A.1 Fault profile representing an intermittent stochastic input fault and resulting measured variable

It can be mathematically described as:

$$g_i = \bar{g}_i + \Delta g_i \quad (i = 1, \dots, n_g) \quad (\text{A.2})$$

where $\{\bar{g}_i\}$ are a set of constant mean values (operating modes), $\{\Delta g_i\}$ are stochastic variations around each mean value. The statistical distribution of Δg_i is assumed to be a priori and time invariant, which can be estimated from an offline model calibration algorithm. It is also assumed that the mean values $\{\bar{g}_i\}$ of faults remain constant. The

constancy of $\{\bar{g}_i\}$ can be experimentally inferred from the constancy of the measured quantities, such as the manipulated and/or controlled variables, through the steady state tests. As seen in Fig. A.1, the changes in the mean values of $\{\bar{g}_i\}$ follow a Multilevel Pseudo Random Signal (*ML-PRS*) (Ljung, 1999). The inputs described by Eq. A.2 are typical in chemical processes that experience both changes in means of operating variables but also in additional continuous random perturbations in time t . Then, the FDD problem can be defined as detecting a change in the unknown input mean \bar{g}_i and diagnosing around which particular \bar{g}_i the system is being operated. Each particular mean \bar{g}_i will be referred heretofore as to an operating mode, thus the goal in the current work is to diagnose the operating mode \bar{g}_i at a given time instant t .

A.3.2 Generalized Polynomial Chaos Expansion

The generalized polynomial chaos (gPC) expansion approximates a random variable as a polynomial series of another random variable following a standard distribution (Xiu D. , 2010). For the nonlinear chemical process defined by Eq. A.1, the gPC expansion can be used to quantify and propagate the effect of stochastic parametric inputs faults \mathbf{g} onto the measured quantities \mathbf{x} . The first step is to re-write each of the unknown input g_i ($i = 1, 2, \dots, n_g$) in \mathbf{g} as a function of a set of random variables $\xi = \{\xi_i\}$:

$$g_i = g_i(\xi_i) \quad (\text{A.3})$$

where ξ_i is the i^{th} random variable. The random variables ($\xi = \{\xi_i\}$) are further assumed to be independent and identically distributed for simplicity. Using the gPC expansion, the unknown stochastic faults $\mathbf{g}(\xi)$ and system states $\mathbf{x}(t, \xi)$ can be approximated in terms of orthogonal polynomial basis functions $\Phi_k(\xi)$:

$$\mathbf{g}(\xi) = \sum_{k=0}^{\infty} g_k \Phi_k(\xi) \quad (\text{A.4})$$

$$\mathbf{x}(t, \xi) = \sum_{k=0}^{\infty} x_k(t) \Phi_k(\xi) \quad (\text{A.5})$$

where x_k and g_k are the gPC coefficients of measured quantities and faults at each time instant t , $\Phi_k(\xi)$ are multi-dimensional orthogonal basis functions of ξ . If the faults (\mathbf{g}) can be measured or estimated, the coefficients, i.e., $\{g_k\}$ in Eq. A.4, can be computed such that Eq. A.3 follows a priori probability density function. Then, the gPCs coefficients, representing the responses of measured quantities (\mathbf{x}) resulting from the stochastic faults (\mathbf{g}), can be calculated using the first principle models of process in combination with a Galerkin projection (Ghanem & Spanos, 1991).

Using Galerkin projection, it is possible to calculate the gPC coefficients of the measured quantities $\{x_k(t)\}$ by substituting Eq. A.4 and Eq. A.5 into Eq. A.1, and then projecting Eq. A.1 onto each one of the polynomial chaos basis functions $\{\Phi_k(\xi)\}$ as defined in Eq. A.6:

$$\langle \dot{\mathbf{x}}(t, \xi), \Phi_k(\xi) \rangle = \langle f(t, \mathbf{x}(t, \xi), \mathbf{u}(t), \mathbf{g}(\xi)), \Phi_k(\xi) \rangle \quad (\text{A.6})$$

For practical implementation, Eq. A.4 is often truncated to a finite number of terms such as p , which is defined as the polynomial order. Hence, the total number of terms of measured quantities P in Eq. A.5 can be calculated as following:

$$P = ((n_g + p)! / (n_g! p!)) - 1 \quad (\text{A.7})$$

where p is the necessary terms used to approximate an a priori known distribution of \mathbf{g} , and n_g is the number of faults of interest defined in Eq. A.2. From Eq. A.7, the number of the gPC expansion terms for the measured variables in Eq. A.5 increases as the polynomial order p and/or the number of unknown inputs n_g increase. The inner product in Eq. A.6 between two vectors can be computed with:

$$\langle \psi(\xi), \psi'(\xi) \rangle = \int \psi(\xi) \psi'(\xi) W(\xi) d\xi \quad (\text{A.8})$$

where the integration is conducted over the entire event domain generated by the random variables ξ , and $W(\xi)$ is the weighting function, which is the probability function of random variables and has to be chosen with respect to the polynomial basis function used to represent ξ so as the result of Eq. A.8 is one or zero (Xiu D. , 2010). To obtain orthogonality the basis functions $\{\Phi_k(\xi)\}$ have to be selected according to the choice of the distribution of ξ . For example, Hermite polynomials are chosen as basis functions for normally distributed ξ . Once the gPC coefficients of the measured quantities \mathbf{x} in Eq. A.5 are available, it is possible to compute statistical moments for the measured variables at a given time instant t with Eq. A.9 and Eq. A.10 as following (Xiu D. , 2010):

$$E(\mathbf{x}(t)) = E \left[\sum_{i=0}^P x_i(t) \Phi_i \right] = \mathbf{x}_0(t) E[\Phi_0] + \sum_{i=1}^P E[\Phi_k] = \mathbf{x}_0(t) \quad (\text{A.9})$$

$$\begin{aligned} Var(\mathbf{x}(t)) &= E \left[(\mathbf{x}(t) - E(\mathbf{x}(t)))^2 \right] = E \left[\left(\sum_{i=0}^P x_i(t) \Phi_i - \mathbf{x}_{(i=0)}(t) \right)^2 \right] \\ &= E \left[\left(\sum_{i=1}^P x_i(t) \Phi_i \right)^2 \right] = \sum_{i=1}^P x_i(t)^2 E(\Phi_i^2) \end{aligned} \quad (\text{A.10})$$

In addition, the probability density functions (PDFs) for measured variables, $\mathbf{x}(t)$, can be approximated by sampling from the distribution of ξ and substituting the samples into Eq. A.5. The ability of analytical formulae for calculating statistical moments as per Eq. A.9 and Eq. A.10 and to rapidly calculate the PDF profiles of the measured variables are the main rationale for using the gPC. It can reduce the computational effort required to approximate the PDF profiles, which are further used for the detection of faults and for the evaluation of fault detectability.

The first principle models based fault detection procedure used in this current work consists of the inverse of the procedures explained in this section, i.e., the distribution of the stochastic parametric faults (inputs) \mathbf{g} is to be inferred from measurements of the process measured variables \mathbf{x} .

A.3.3 Gaussian Process

The Gaussian process (GP) extends multivariate Gaussian distributions to infinite dimensionality [12]. It can be used to generate a surrogate metamodel with the measurements \mathbf{x} in Eq. A.1 to provide a prediction of how the process is behaving without knowing the true generative system, i.e., the value of \mathbf{g} in this work. Assuming $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{g}_i)\} (i = 1, \dots, N)$ is N pairs of observations, then the GP regression model can be formulated as follows:

$$\mathbf{g}_i = \mathcal{G}(\mathbf{x}_i) + \varepsilon_i \quad (\text{A.11})$$

$$\varepsilon_i \sim N(0, \sigma_g^2) \quad (\text{A.12})$$

where \mathcal{G} denotes the GP metamodel and ε_i is a bias term. In other words, \mathbf{g}_i is related to \mathbf{x}_i nonlinearly through an unknown function \mathcal{G} that can be approximated with a GP. Moreover, each observation inside $\mathbf{X} = \{\mathbf{x}_i\}$ is related to another with the covariance function $k(\mathbf{x}_i, \mathbf{x}_j)$. A popular choice of the covariance function k is the squared exponential kernel function (Shi & Choi, 2011) that can be defined as:

$$k_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) = \sigma_G^2 \exp\left(-\frac{1}{2l^2}(\mathbf{x}_i - \mathbf{x}_j)^2\right) \quad (\text{A.13})$$

where (σ_G, l) are unknown parameters and heretofore referred as hyper-parameters. For the given observations, the covariance function k among all possible combinations of these N points can be computed with Eq. A.13. Let \mathbf{K} be the covariance matrix at all points of the N training observations, i.e., $\mathbf{K} = \{k_{ij}\}$ and $1 \leq i, j \leq N$. It can be proved that the marginal distribution of $\mathbf{G} = \{\mathbf{g}_i\}$ follows a multivariate normal distribution (Rasmussen & Williams, 2006; Shi & Choi, 2011):

$$\{\mathbf{g}_i\} \sim N(\mathbf{0}, \mathbf{K}_g) \text{ with } \mathbf{K}_g = \mathbf{K} + \sigma_g^2 \mathbf{I} \quad (\text{A.14})$$

where \mathbf{K}_g is the $N \times N$ covariance matrix and each element $(i, j)^{\text{th}}$ inside \mathbf{K}_g can be defined as:

$$\{\mathbf{K}_g\}_{ij} = \text{cov}(\mathbf{g}_i, \mathbf{g}_j) = k(\mathbf{x}_i, \mathbf{x}_j) + \sigma_g^2 \delta_{ij} \quad (\text{A.15})$$

where δ_{ij} is the Kronecker delta function. Training of the GP involves the determination of the values for the unknown parameters in Eq. A.13 and Eq. A.15, i.e., $\boldsymbol{\theta} = \{\sigma_G, l, \sigma_g\}$, based on the given observations \mathcal{D} . This can be solved with Empirical Bayes estimation algorithm by maximizing $\log p(\mathbf{g}|\mathbf{x}, \boldsymbol{\theta})$, which can be given as (Shi & Choi, 2011):

$$\arg \max \log p(\mathbf{g}|\mathbf{x}, \boldsymbol{\theta}) = -\frac{1}{2}N \log(2\pi) - \frac{1}{2} \log |\mathbf{K}_g| - \frac{1}{2} \mathbf{g}^T (\mathbf{K}_g)^{-1} \mathbf{g} \quad (\text{A.16})$$

Based on the training results, the GP model can estimate the prediction \mathbf{g}^* for a new set of observations \mathbf{x}^* , which has the mean and variance as below:

$$E(\mathbf{g}^*|\mathcal{D}, \boldsymbol{\theta}_{opt}) = \mathbf{k}^{*T} \mathbf{K}_g^{-1} \mathbf{G} \quad (\text{A.17})$$

$$\text{var}(\mathbf{g}^*|\mathcal{D}, \boldsymbol{\theta}_{opt}) = k(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{k}^{*T} \mathbf{K}_g^{-1} \mathbf{k}^* \quad (\text{A.18})$$

where $\mathbf{k}^* = (k(\mathbf{x}^*, \mathbf{x}_1), \dots, k(\mathbf{x}^*, \mathbf{x}_N))^T$ is the vector of covariance between the new measured quantities \mathbf{x}^* and the training observations $\mathbf{X} = \{\mathbf{x}_i\}$, $\boldsymbol{\theta}_{opt}$ is the model training results optimized with Eq. A.16.

A.4 Fault Detection and Diagnosis Algorithms

A.4.1 FDD using First Principle Models

A.4.1.1 FDD Algorithm

In Section A.3.2 above, we explained how to propagate stochastic parametric faults \mathbf{g} such as the ones shown in Fig.A.1 (a) onto measured quantities (\mathbf{x}), and how to calculate the probability density functions (PDFs) of these measured quantities by using gPC expansions. The main idea of the FDD algorithm with the first principle models is to solve the inverse problem given in Section A.3.2. Specifically, the goal is to infer the mean value (operating mode) of the faults from the gPC models of the measured variables, i.e., measured states \mathbf{x} in Eq.A.1.

For the purpose of calculating the PDF profiles of the measured quantities, it is assumed that measurements of the certain variables (\mathbf{x}) around each mean value \bar{g}_i are available. Note that in this step the mean value \bar{g}_i of a fault remains constant but its exact value is unknown. The constancy of $\{\bar{g}_i\}$ can be experimentally inferred from the constancy of the measured and/or controlled variables through a steady state test (Seborg, Mellichamp, Edgar, & Doyle, 2010). In principle, in the absence of measurement noise and if the means and variances of the faults \mathbf{g} would be known, the PDF profiles of the measured quantities (\mathbf{x}) that are measured and used for fault detection can be calculated with the gPC approximation as per the procedures shown in Section A.3.2. Then, it could be possible to accurately infer the mean value of fault from a measured output value by inverting the procedures outlined in Section A.3.2. In implementation, due to noise and model error (e.g., gPC truncation error), the exact mean and variance of the faults are unknown or unmeasurable. Thus, the PDF profiles of \mathbf{x} around each possible mean value (operating mode) have to be calibrated using actual process measurements. To this purpose, the mean and variance of the unknown faults \mathbf{g} can be calibrated from an optimization problem around each operating mode \bar{g}_i shown in Fig.A.1 (a) as:

$$\min_{\lambda_{gPC}} J = \sum_{i=1}^n (\vartheta_{1,i} - \nu_{1,i})^2 + \sum_{i=1}^n (\vartheta_{2,i} - \nu_{2,i})^2 + \sum_{i=1}^n \sigma_{n,i}^2 \quad (\text{A.19})$$

where $\vartheta_{1,i}$ and $\vartheta_{2,i}$ are the predicted mean and variance of a particular measured variable (\mathbf{x}) that is used for fault detection. These predicted means and variances are explicitly calculated with Eq. A.9 and Eq. A.10 using the gPCs coefficients of \mathbf{x} , which are functions of the stochastic faults as shown in Fig.1 (a). The terms $\nu_{1,i}$ and $\nu_{2,i}$ are the measured mean and variance of \mathbf{x} in Eq. A.1. The last term $\sigma_{n,i}$ is utilized to represent the standard deviation of noise that is also approximated by a gPC expansion of the following form:

$$\sigma_n(t, \xi) = \sum_{k=1}^P \sigma_{n,k}(t) \Phi_k(\xi) \quad (\text{A.20})$$

where $\sigma_{n,k}$ is the gPC coefficients of noise at time instant t , $\Phi_k(\xi)$ is the multi-dimensional polynomials in terms of ξ , and the variance of noise is assumed to be known a priori.

The decision variable λ_{gPC} in Eq. A.19 is a vector consisting of the mean and variance of the unknown fault (\mathbf{g}) and noise σ_n , and n is the number of the measured variables \mathbf{x} used to calibrate the gPC model. Due to noise and truncation error introduced by the gPC approximation, the mean and variance of the input variable (\mathbf{g}) defining λ_{gPC} calculated from Eq. A.19 deviate from the actual values entering the system. After obtaining λ_{gPC} , it is possible to calculate the gPC coefficients for the measured variables \mathbf{x} . Using these coefficients, the PDF profiles for \mathbf{x} 's around each mean value can be approximated by substituting samples (ξ) from an priori known distribution, e.g., normal or uniform, into the resulting gPC expansions given in Eq. A.5. Following these substitutions the PDF profiles of the measured quantities are calculated as a histogram composed of bins each corresponding to different ranges of values of \mathbf{x} .

Histograms are built for each of the mean values considered in Eq. A.2. When the system is operated around a constant mean \bar{g}_i , the corresponding index i ($i = 1, \dots, n_g$) in Eq. A.2 is detected from the PDF profiles for a given measurement as follows:

$$\text{Operating mode } (\bar{g}_i) = \arg \max \{P_i\} \quad (\text{A.21})$$

where i denotes the i^{th} operating mode as defined in Eq. A.2. P_i means the probability that the process is operating around a particular mean \bar{g}_i for a given measurement. The solution of this problem is depicted in Fig.A.2 showing 3 PDF profiles that correspond to 3 different operating modes (mean values of faults). For example, three probabilities (red dots) can be found for a given measured variable shown in Fig.A.2, where the maximum probability can be used to indicate that the system is operating around the second mean value corresponding to ‘‘Operating mode 2’’.

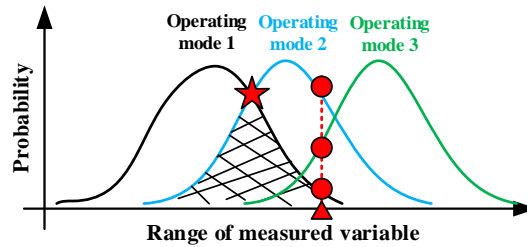


Figure A.2 Visual interpretation of FDD with the *level-1* algorithm

A.4.1.2 Sensitivity Analysis based Sensor Selection for FDD

Appropriate selection of sensors (measured quantities) for enhanced fault detection is essential in the presence of uncertainty. Sensitivity analysis in this work aims to propagate the effect of stochastic faults onto the variability of the measured variables and to maximize the obtainable information about faults. This section presents an

efficient sensitivity analysis algorithm based on differentiating the gPC approximation of measured variable in Eq.A.5 with respect to the random variables ξ .

For this purpose, the partial derivatives of the gPC models for the measured quantities $\mathbf{x} = \{x_j\}$ ($j = 1, 2, \dots, n$) in Eq.A.5 is calculated with respect to the random variables $\xi = \{\xi_i\}$ ($i = 1, 2, \dots, n_g$). Since each measured variable has different units and orders of magnitude, each of the measured quantities is normalized by its first gPC coefficient, i.e., the mean values of the gPC approximation. Thus, Eq.A.5 can be accordingly re-written as follows:

$$\frac{x_j(t, \xi)}{x_{j,1}(t)} = \frac{x_{j,1}(t)}{x_{j,1}(t)} \Phi_0(\xi) + \frac{1}{x_{j,1}(t)} \sum_{k=1}^P x_{j,k}(t) \Phi_k(\xi) = y_j \quad (\text{A.22})$$

where $x_{j,1}(t)$ denotes the first gPC coefficient of the j^{th} measured variable. The partial derivatives of each measured quantity can be calculated with respect to the n_g random variables, and the sensitivity matrix \mathbf{S} can be derived as:

$$\mathbf{S} = \begin{pmatrix} \partial y_1 / \partial \xi_1 & \cdots & \partial y_1 / \partial \xi_{n_g} \\ \vdots & \ddots & \vdots \\ \partial y_n / \partial \xi_1 & \cdots & \partial y_n / \partial \xi_{n_g} \end{pmatrix} = \begin{pmatrix} s_{1,1} & \cdots & s_{1,n_g} \\ \vdots & \ddots & \vdots \\ s_{n,1} & \cdots & s_{n,n_g} \end{pmatrix} \quad (\text{A.23})$$

where $s_{j,i}$ represents the sensitivity of the j^{th} measured variable to the i^{th} unknown fault. Although each element in \mathbf{S} can be also formulated over a time horizon, in this work for simplicity it is only evaluated around each of the mean values (operating modes). Based on \mathbf{S} , the measured variable with the maximum sensitivity value can be chosen to infer the faults, using the calibrated PDF profiles of measured quantities.

A.4.1.3 Quantitative Analysis of Fault Detectability

To quantify the detectability and isolability of faults, the Kullback-Leibler (K-L) divergence (Ullah, 1996) is used to measure the difference between two PDF profiles. The selection of the measured quantity used for fault detection and for K-L divergence analysis is based on the sensitivity analysis results as discussed in previous section. Assume two PDF profiles, i.e., p_{m1} and p_{m2} , describing the K-L divergence for two mean values (operating modes). The more the distribution of the two PDF profiles differs between two mean values, the easier it is to detect and isolate the operating mode. The K-L divergence D_{kl} of p_{m2} from p_{m1} can be defined as:

$$D_{kl}(p_{m1} || p_{m2}) = \sum_{i=1}^{n_{kl}} p_{m1}(i) \log \frac{p_{m1}(i)}{p_{m2}(i)} \quad (\text{A.24})$$

where n_{kl} denotes the number of bins used to approximate the PDF profiles. Generally, the right hand side of Eq.A.24 is the expected value of $\log(p_{m1}/p_{m2})$ given p_{m1} .

A.4.2 FDD using Empirical Models

A.4.2.1 FDD Algorithm

In Section A.3.3, we presented a brief outline of the GP regression model, which can be used to estimate the mean and variance of \mathbf{g} for a set of given observations of \mathbf{x} . However, the measured quantities \mathbf{x} of a process depend on the joint behaviour of a set of measured variables and the formulation of GP model with many variables may not be efficient. For this purpose, the principal component analysis (PCA) (Srinivasan & Qian, 2007) is first used to remove the inessential variables and the calculation of the GP model is then applied to the principal components obtained with the PCA model. For the purpose of model calibration in this method, it is assumed that a few measurements of faults are available. Note that this step can be replaced by a GP classification model (Rasmussen & Williams, 2006), in which the same assumption as done for the gPC model can be used, i.e., the mean values of the faults remain constant but their exact values are unknown.

The optimal selection of the principal components proceeds as per the following procedures: (i) The training set $\mathbf{X} = \{\mathbf{x}_i\}$ ($i = 1, \dots, N$) for all operating modes, consisting of N observations for n variables, are used to calculate the covariance matrix C of \mathbf{X} ; (ii) The eigenvalues and eigenvectors of the covariance matrix C are then calculated and sorted in a decreasing order; (iii) Compute the variances captured by each of the principal components and the corresponding eigenvectors matrix V ; (iv) Determine the number of principal components n_{PC} and build the transform matrix V_{PC} with respect to a predefined cumulative variance percent; (v) Based on the transform matrix V_{PC} , convert the training set \mathbf{X} onto the domain generated with the principal components and calculate the score matrix $\{X_{s,i}\}$ ($i = 1, \dots, N$) that can be used for the calibration of GP regression model.

Using the score matrix $\{X_{s,i}\}$, a transformed observation set $\mathcal{D} = \{(\mathbf{X}_{s,i}, \mathbf{g}_i)\}$ ($i = 1, \dots, N$) with N pairs of elements can be formulated, and then the GP regression model $\bar{\mathbf{G}}$ can be constructed following the procedures as described in Section A.3.3. The main idea of the FDD algorithm with the GP model is to estimate the values of parametric faults \mathbf{g}^* for a new set of observations \mathbf{x}^* . For this purpose, the detection of fault involves the following steps: (a) Determine the number of possible mean values $\{\bar{\mathbf{g}}_i\}$ by checking the constancy of the manipulated or controlled variable, and calculate the mean values $\{\bar{\mathbf{g}}_i\}$ using collected measurements in \mathcal{D} ; (b) The new observations \mathbf{x}^* are scaled and projected onto the principal components domain; (c) Estimate the mean value and the variance of \mathbf{g}^* given \mathbf{x}^* using Eq. A.17 and Eq. A.18; (d) The mean value obtained from step-c is compared based on a minimum distance criterion to the set of the mean values $\{\bar{\mathbf{g}}_i\}$ estimated in step-a and the corresponding operating mode (mean values) can be diagnosed.

A.4.2.2 Model Calibration through Minimal Adjustments

The GP regression model represents the relationship between the faults \mathbf{g} and the measured quantities \mathbf{x} based on the optimized hyper-parameters θ . For each given observation of \mathbf{x}^* , the GP model predicts the faults \mathbf{g}^* by calculating its mean value and the variance around the mean value with Eq. A.17 and Eq. A.18. However, the

model calibration and the prediction of faults are based on the initial training set $\mathbf{X} = \{\mathbf{x}_i\}$. The reliability of the training set \mathbf{X} might be insufficient due to the limited measurements. Moreover, more simulations results can be obtained than experimental measurements. Thus, a minimal model adjustment algorithm is developed in this work using the collected measurements and computer simulations. This algorithm combines a normal cumulative distribution function based probability improvement method (Jones, 2001) with an adaptive selection of new training observations.

For the purpose of calibrating the GP regression model, the model discrepancy between the GP prediction and the initial observations of $\{\mathbf{g}_i\}$ inside $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{g}_i)\}$ is defined as:

$$\varepsilon = \sum [\mathbf{g}_i - \mathcal{G}(\bar{\mathbf{g}}_i|\mathbf{x}_i, \boldsymbol{\theta})]^2 \text{ for } i = 1, \dots, N \quad (\text{A.25})$$

where \mathbf{g}_i and \mathbf{x}_i are the i^{th} set of observations, $\mathcal{G}(\bar{\mathbf{g}}_i|\mathbf{x}_i, \boldsymbol{\theta})$ denotes the estimated mean value of faults given \mathbf{x}_i and hyper-parameters $\boldsymbol{\theta}$. Based on the model discrepancy ε , the normal cumulative distribution function can identify new possible observations, with which the probability to minimize the model discrepancy beyond a target T can be maximized (Jones, 2001; Du, Yang, Ednie, & Bennett, 2015).

The optimal selection of a new observation set involve as per the following steps. (i) Calibrate the GP model to solve the hyper-parameters $\boldsymbol{\theta}_0$ with the initial observations set \mathcal{D}_0 , and determine the model discrepancy criterion ε^* . (ii) Generate a set of new observations \mathcal{D}_1 through simulations, consisting of N_1 observations for n variables. (iii) Calculate the model predictions for each observations \mathbf{x}_j in \mathcal{D}_1 ($j = 1, \dots, N_1$) with $\boldsymbol{\theta}_0$, i.e., the mean value and variance of \mathbf{g}_j . (iv) Compute the probability improvement P_j for each of the observations \mathbf{x}_j in \mathcal{D}_1 with Eq. A.26:

$$P_j = \psi[(T - E(\bar{\mathbf{g}}_j|\mathbf{x}_j)]/s(\bar{\mathbf{g}}_j|\mathbf{x}_j)] \quad (\text{A.26})$$

where ψ denotes the normal cumulative distribution function, T is a target value that can be used to adjust the model calibration. For any given observation \mathbf{x}_j in \mathcal{D}_1 , $E(\bar{\mathbf{g}}_j|\mathbf{x}_j)$ represents the mean value of faults and $s(\bar{\mathbf{g}}_j|\mathbf{x}_j)$ is the corresponding standard deviation. These two values are calculated with the hyper-parameters $\boldsymbol{\theta}_0$ calibrated in step-i. (v) The observation with the maximum probability improvement P_j can be added to the initial observations set \mathcal{D}_0 , and results in a new initial observations set. This new initial observations set now consists of $(N+1)$ observations for n variables. (vi) Calibrate the GP model with the new initial observations set consisting of $(N+1)$ observations to solve a new set of hyper-parameters $\boldsymbol{\theta}_0'$. (vii) Compute the model discrepancy ε with Eq. A.25 and $\boldsymbol{\theta}_0'$, repeat step-ii ~ step-vi and keep adding new observations into the initial observations until $\varepsilon < \varepsilon^*$.

The normal cumulative distribution function minimizes the model discrepancy by adding the best observations into the training set. However, it cannot provide information about the probability distribution of the observations. Note that for two observations \mathbf{x} and \mathbf{x}' , if \mathbf{x} is distant from \mathbf{x}' , the covariance function calculated from Eq. A.13 is tiny, e.g., $k(\mathbf{x}_i, \mathbf{x}_j) \approx 0$. These distant observations will have negligible effect on the interpolation of new observations. Thus, a second model adjustment criterion η is developed to check the value of covariance function

in Eq. A.13. Each of the possible new-to-be-added observations set is further evaluated based on the covariance function to guarantee that each value inside the covariance matrix is larger than η .

A.5 Case Study

Simulation studies of a nonlinear chemical process consisting of two reactors and a separator with recycle unit are used to illustrate the efficacy of the proposed two-level fault diagnosis algorithm (Stewart, Venkat, Rawlings, Wright, & Pannocchia, 2010). Fig.A.3 shows a schematic of the system with three temperature control loops. A stream of reactant A is added to each reactor and converted to the product B by the first order reaction, C denotes the side-product of the process. The feed mass fraction of reactant A (x_{A0}) is assumed as the unknown (unmeasured) stochastic fault (g) in this current work. x_{A0} is assumed to change as shown in Fig.A.1, i.e., normally distributed perturbations around three mean values (operating modes) as described in Eq. A.2. The first principles' model of the process controlled with three *PI* controllers is described by a set of ordinary differential equations (ODEs) as following:

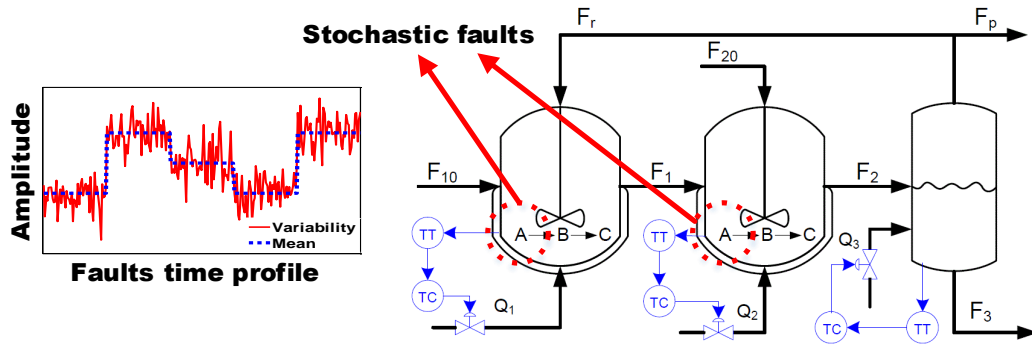


Figure A.3 Two reactors in series with a separator and a recycle unit

$$\dot{H}_1 = (1/\rho A_1)(F_{f1} + F_r - F_1) \quad (\text{A.27})$$

$$\dot{x}_{A1} = (1/\rho A_1 H_1)(F_{f1} x_{A0} + F_r x_{AR} - F_1 x_{A1}) - k_{A1} x_{A1} \quad (\text{A.28})$$

$$\dot{x}_{B1} = (1/\rho A_1 H_1)(F_r x_{BR} - F_1 x_{B1}) + k_{A1} x_{A1} - k_{B1} x_{B1} \quad (\text{A.29})$$

$$\dot{T}_1 = (1/\rho A_1 H_1)(F_{f1} T_0 + F_r T_R - F_1 T_1) - (1/C_p)(k_{A1} x_{A1} \Delta H_A + k_{B1} x_{B1} \Delta H_B) + (Q_1/\rho A_1 C_p H_1) \quad (\text{A.30})$$

$$\dot{H}_2 = (1/\rho A_2)(F_{f2} + F_1 - F_2) \quad (\text{A.31})$$

$$\dot{x}_{A2} = (1/\rho A_2 H_2)(F_{f2} x_{A0} + F_1 x_{A1} - F_2 x_{A2}) - k_{A2} x_{A2} \quad (\text{A.32})$$

$$\dot{x}_{B2} = (1/\rho A_2 H_2)(F_1 x_{B1} - F_2 x_{B2}) + k_{A2} x_{A2} - k_{B2} x_{B2} \quad (\text{A.33})$$

$$\dot{T}_2 = (1/\rho A_2 H_2)(F_{f2} T_0 + F_1 T_1 - F_2 T_2) - (1/C_p)(k_{A2} x_{A2} \Delta H_A + k_{B2} x_{B2} \Delta H_B) + (Q_2/\rho A_2 C_p H_2) \quad (\text{A.34})$$

$$\dot{H}_3 = (1/\rho A_3)(F_2 - F_D - F_r - F_3) \quad (\text{A.35})$$

$$\dot{x}_{A3} = (1/\rho A_3 H_3)(F_2 x_{A2} - (F_r + F_D) x_{AR} - F_3 x_{A3}) \quad (\text{A.36})$$

$$\dot{x}_{B3} = (1/\rho A_3 H_3)(F_2 x_{B2} - (F_R + F_D)x_{BR} - F_3 x_{B3}) \quad (\text{A.37})$$

$$\dot{T}_3 = (1/\rho A_3 H_3)(F_2 T_2 - (F_R + F_D)T_R - F_3 T_3) + (Q_3/\rho A_3 C_p H_3) \quad (\text{A.38})$$

where the subscripts 'i' (i.e., 1, 2, 3) refers to the vessel, x_i is the mass fraction of reactant A or product B, T_i denotes temperature in each tank, H_i is the level, F_i is the flow rate and the reaction terms are:

$$F_i = k_{vi} H_i \quad (\text{A.39})$$

$$k_{Ai} = k_A \exp(-E_A/RT_i) \quad (\text{A.40})$$

$$k_{Bi} = k_B \exp(-E_B/RT_i) \quad (\text{A.41})$$

The recycle flow and the weight percent factors satisfy:

$$F_D = 0.01 F_R \quad (\text{A.42})$$

$$x_{AR} = \alpha_A x_{A3} / \bar{x}_3 \quad (\text{A.43})$$

$$x_{BR} = \alpha_B x_{B3} / \bar{x}_3 \quad (\text{A.44})$$

$$\bar{x}_3 = \alpha_A x_{A3} + \alpha_B x_{B3} + \alpha_C x_{C3} \quad (\text{A.45})$$

$$x_{C3} = 1 - x_{A3} - x_{B3} \quad (\text{A.46})$$

Each of the vessels in the process receives an external heat input Q_i that is controlled by a *PI* controller:

$$Q_i(t) = Q_{(ss),i}(t) + K_{p,i}(T_{(set),i} - T_i(t)) + K_{p,i}/\tau_i \int_0^t (T_{(set),i} - T_i(t^*)) dt^* \quad (\text{A.47})$$

These parameters used for the simulation are given in Table 1.

Table A.1 Parameter declaration for the Reactor-Separator process

<i>Symbol</i>	<i>Value</i>	<i>Units</i>	<i>Symbol</i>	<i>Value</i>	<i>Units</i>	<i>Symbol</i>	<i>Value</i>	<i>Units</i>
F_{f1}	10	kg/s	k_{v1}	2.5	kg/m s	ρ	0.15	kg/m ³
F_{f2}	1	kg/s	k_{v2}	2.5	kg/m s	A_1	3	m ²
F_R	60	kg/s	k_{v3}	2.5	kg/m s	A_2	3	m ²
$T_{(set),1}$	315	K	k_A	0.02	1/s	A_3	1	m ²
$T_{(set),2}$	315	K	K_B	0.018	1/s	α_A	3.5	/
$T_{(set),3}$	400	K	E_A/R	-1000	K	α_B	1.1	/
T_0	310	K	E_B/R	-500	K	α_c	0.5	/
T_R	310	K	ΔH_A	-40	kJ/kg	$K_{p,i}$	0.25	/
C_p	2.5	kJ/kg K	ΔH_B	-50	kJ/kg	τ_i	0.0025	/

A.6 Results and Discussion

A.6.1 Model Formulation with gPC approximations

The detection of faults in this work is to diagnose the mean value (operating mode) of the unknown feed mass fraction x_{A0} based on measurements that can be easily measured such as Q_i . For simplicity, 3 mean values of the feed mass fraction (x_{A0}) are considered, i.e., 0.65, 0.75 and 0.85 ($n_g = 3$ in Eq. A.2). Stochastic perturbations in

x_{A0} occur around each of these mean values, and they follow a normal distribution with zero mean and a standard deviation of 0.1. Since the solution of the gPC coefficients involved in the gPC expansions of each one of the states (\mathbf{x} in Eq. A.1) as given in Section A.3 requires the application of Galerkin projection, the employment of gPC is limited to monomial or polynomial terms. Hence, non-polynomial terms such as the reaction term (Arrhenius energy function) k_{Ai} , are approximated by a 2nd order Taylor expansion around each mean value on input fault x_{A0} . Assuming that the random variable ζ is normally distributed, the corresponding basis polynomial functions for gPC approximations are chosen as Hermite as per the Askey scheme to maintain orthogonality (Xiu D. , 2010).

To test the accuracy of the Taylor approximation, simulations are conducted with the gPC model resulting from this approximation. These simulations are compared to Monte Carlo (MC) simulations (Harrison, 2010; Fouskakis & Draper, 2002) using the nonlinear model without the Taylor approximations. For comparison, Fig.A.4 shows the simulation results of the controlled variable T_1 in the first reactor, using the gPC method and the MC simulations, respectively. For the gPC method, a 2nd order Taylor expansion is used to approximate the reaction terms, while the nonlinear model described in Section A.5 is used for MC simulations. For the gPC method, the gPC coefficients of the measured quantities \mathbf{x} are calculated as outlined in Section A.3. Then samples generated for the random variable ζ are substituted into these gPC expressions to predict the distributions of measured quantities. Using these estimated distributions, it is possible to estimate their upper (maximum) and lower (minimum) values at each time instant t .

The MC simulations can be conducted as per the following steps: (i) A set of samples of the feed mass fraction x_{A0} following the same statistical distribution as used for the gPC are first generated; (ii) Each of these samples is then substituted into the nonlinear model as described in Section A.5; and (iii) The simulation results of the measured variables are stored for comparison. Several randomly chosen simulated trajectories with the MC simulations are shown in Fig.A.4. As seen, the trajectories obtained with MC simulations are bounded by the upper (*Maximum*) and lower (*Minimum*) bounds calculated with the gPC model. Thus, the gPC model with the Taylor approximation of the Arrhenius term can provide correct bounds for the MC simulations. To simplify the comparison of fault detection algorithm with different models, the diagnosis of fault in this work is performed to detect the mean value (operation mode) only when the system is operating for long periods around a fixed mean value. For example, the measurements can be collected at time instant t_s as shown in Fig.A.4.

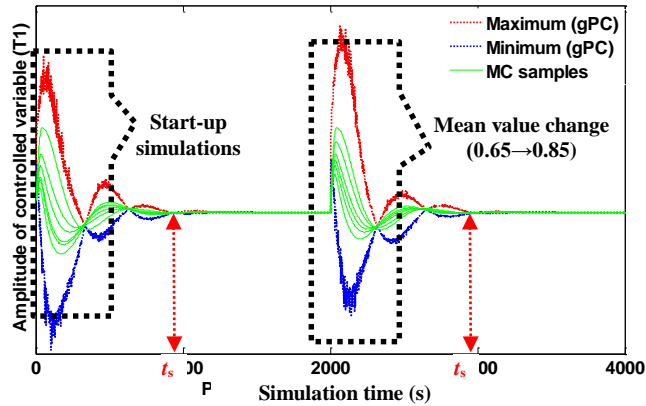


Figure A.4 Comparisons of the gPC model and MC simulations using controlled variable T_1

A.6.2 Sensitivity Analysis with gPC model

The sensitivity analysis described in Section A.4.1.2 is studied for the purpose of optimal selection of sensor. For each of the mean values on the feed mass fraction x_{A0} , the sensitivity matrix \mathbf{S} (Table A.2) can be computed for all the states defined by the first principles' model in Section 4 with respect to the random variable $\xi = \{\xi\}$. The dimension of the random variables ξ is 1 in this work, since only one stochastic fault x_{A0} is considered.

In principle, variations in the feed mass fraction x_{A0} can contribute significantly to changes in the mass fractions of A and B in the reactors and separators. Despite its sensitivity however, they are not used for the detection of faults in this work, since the objective is to detect faults using measurements that can be easily measured and concentrations are generally expensive to measure online. Thus, the sensitivity analysis is only investigated for the controlled and manipulated variables. As seen in Table A.2, the sensitivity of temperatures $\{T_i\}$ to the variations in the feed mass fraction x_{A0} is small as expected, since they are controlled variables. As compared with the temperatures $\{T_i\}$, the manipulated variables $\{Q_i\}$ are more sensitive to the random changes in the feed and consequently they can be chosen for inferring the faults. The measured variable Q_1 is used in this work because it has the largest sensitivity analysis value as seen in Table A.2.

Table A.2 Sensitivity analysis of Reactor-Separator process

Mean	Measured variables					
	T_1	Q_1	T_2	Q_2	T_3	Q_3
0.65	1.6e-7	0.0177	5.1e-7	0.0156	3.4e-7	0.0056
0.75	6.3e-7	0.0177	1.4e-6	0.0157	3.4e-7	0.0011
0.85	8.9e-7	0.0165	1.3e-6	0.0143	1.8e-7	0.0012

A.6.3 Model Calibration with the gPC model

Following the above, 3 mean values of the feed mass fraction x_{A0} are studied, i.e., 0.65, 0.75 and 0.85. The stochastic perturbations, added around each of these mean values (operating modes), are assumed to be normally

distributed with zero mean and a standard deviation of 0.1. The step changes of feed mass fraction x_{A0} follow a ML-PRS signal as shown in Fig.A.5 (a) and (b), respectively. The number of step changes of the unknown faults (x_{A0}) among the 3 selected mean values (operating modes) in the *ML-PRS* is 242 and the maximum number of measurements between two consecutive step changes in faults (inputs) is limited to 1000.

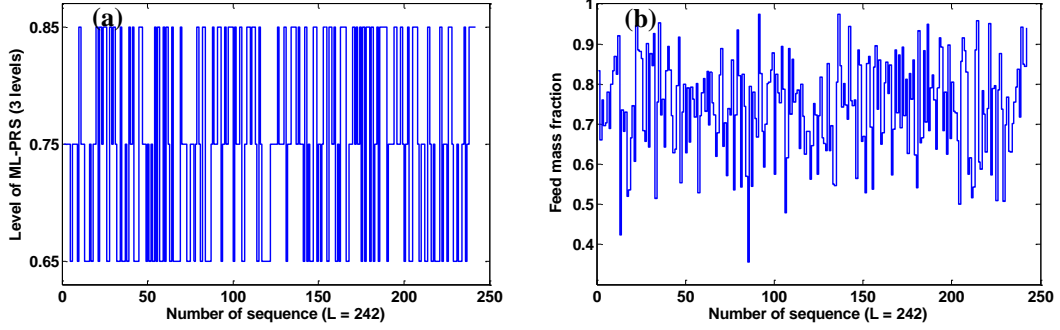


Figure A.5 (a) Three-level-PRS and (b) application to the feed mass fraction superimposed with stochasticity

Table A.3 shows the model calibration results calculated from Eq. A.19 using the measurements of manipulated variables $\{Q_i\}$, since they are more sensitive to the variation in the feed x_{A0} . To simulate actual data, Gaussian noise is added to the measurements of $\{Q_i\}$. Thus, Hermite polynomials are selected and the highest order of polynomials used for the gPC models is 2 ($p = 2$ in Eq. A.7), following the Askey scheme to maintain orthogonality (Xiu D. , 2010).

Table 3 Model calibration results for gPC model

x_{A0}	x'_{A0}	σ_{A0}	σ_n	time(s)
0.65	0.6370	0.0937	0.0188	992
0.75	0.7364	0.0979	0.0199	788
0.85	0.8319	0.0933	0.0201	871

As seen in Table A.3, the first column gives the mean values of feed x_{A0} used for simulations. The second and third columns (x'_{A0} and σ_{A0}) are the mean and standard deviation calibrated from Eq. A.19, σ_n is the standard deviation of measurement noise. As explained before, the mean and standard deviation of the faults (inputs) resulting from Eq. A.19, i.e., x'_{A0} , σ_{A0} , are not identical to the actual simulated values used for model calibration (x_{A0} , and 0.1), due to the measurement noise and the gPC series' truncation errors.

Based on the model calibration results, the PDF profiles of the measured variable $\{Q_i\}$, estimated for each of the 3 mean values of the feed mass fraction (x'_{A0} in Table 3), can be approximated. As an example, Fig.6 shows the PDF profiles for the external heat Q_j in the first reactor, in which the horizontal axis represents the range of Q_j , and the vertical axis is the normalized probability. Again the measurements of external heat Q_j is used for the detection of faults, since it is more sensitive to the random changes in the feed as seen in Table A.2 (the second column).

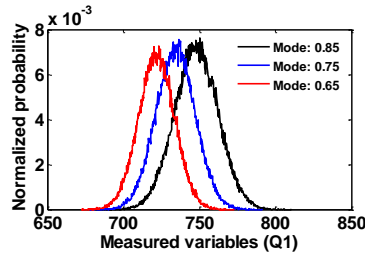


Figure A.6 The PDF profiles of the measured variable (Q_1) at 3 operating modes

A.6.4 Detectability Analysis with gPC model

The PDF profiles of the measured quantities can be further used to evaluate the fault detectability using the first principles' model of the system. For simplicity, the detectability of faults is computed with the PDF profiles of the external heat Q_1 of the first reactor and the results are summarized in Table A.4. The detectability is calculated with the K-L divergence D_{kl} as defined in Eq. A.24, which is based on the difference between two PDF profiles generated for different mean values (operating modes).

In Table A.4, the higher detectability of faults is related to the higher values of the K-L divergence D_{kl} . As can be seen, the fault detectability is lower for two operating modes that are adjacent to each other, as compared to operating modes that are not adjacent to each other. As observed from the second column for example, it is easier to detect the mean value changes occurring between 0.65 and 0.85 than those changes occurring between 0.65 and 0.75, since $0.03 > 0.01$. Additionally, when the system is operating around the second mean value 0.75, the detectability of faults is lower as compared to the case where the system is operating around the other two mean values. For example, the sum of the K-L divergence D_{kl} in the third column is 0.02, which is smaller than for the other columns, i.e., 0.04 and 0.08, respectively.

Table 4 Estimation of detectability with gPC model

D_{kl}	0.65	0.75	0.85
0.65	/	0.01	0.05
0.75	0.01	/	0.03
0.85	0.03	0.01	/

A.6.5 Minimum Adjustment of GP model

As discussed in Section A.4.2, the measured quantities of a process have a joint effect on the formulation of the GP model and the model calibration with many variables may not be efficient. For this purpose, the principal component analysis (PCA) (Srinivasan & Qian, 2007) is used to remove the inessential variables and the calculation of the GP model is applied to the principal components. For optimal selection of the principal component, a training set with 30 units of observations is used. For each observations' unit, measurements of the measured quantities $\{T_i\}$ and $\{Q_i\}$ are collected around each mean value (operating mode) of feed and each mean value has 10 samples. Following the procedure in Section A.4.2.1, it is found that one principal component can

capture ~80% of the cumulative variance. Thus, only the first principal component is used to build the transform matrix V_{PC} . The matrix V_{PC} transforms each observation unit onto the domain generated with the principal component and formulates a new score vector $\{X_{s,i}\}$, which can be further used for the training of the GP regression model.

Using the score vector $\{X_{s,i}\}$, a new initial observation set $\mathcal{D} = \{(X_{s,i}, \mathbf{g}_i)\}$ ($i = 1, \dots, 30$) with 30 pairs of observation units can be formulated, and then the GP regression model \mathcal{G} can be adjusted following the procedures described in Section A.4.3. As compared to the gPC model based method, in which it is assumed that the exact value of faults are unknown, the GP regression model needs a training set of the faults' values and the estimates of mean values of the feed. The estimation of mean values on the feed x_{A0} is used for the minimal model adjustment, since we typically have many simulation results than experimental results. These simulations results are used to determine a new set of observations that has to be added to the initial observation set. To simulate actual data, Gaussian noise is added to the measurements of $\{T_i\}$ and $\{Q_i\}$. The minimal model adjustment proceeds as per the steps described in Section A.4.2.2 and Table A.5 shows the model parameters.

As seen from Table 5, the model parameters computed with the minimal adjustment method are not identical to these without model adjustment. The efficiency of model adjustment will be further discussed in terms of fault detection rate as below. For comparison in this case study, the computational time required for the calibration of GP is also given. With the minimal model adjustment, 152 set of new observations are added to the initial training set and the computational time is significantly higher than the time needed without model adjustment, i.e., 3.61 hours verse 83.67 seconds. However, the minimal model adjustment can improve the performance of fault detection as shown below.

Table 5 Hyper-parameter of GP model

<i>GP model</i>	σ_G	l	σ_g	<i>Time</i>
<i>No adjustment</i>	0.9763	14.0979	0.1108	83.67 (s)
<i>Minimal adjustment</i>	1.1597	24.1909	0.0979	3.61 (h)

A.6.6 Summary of Fault Detection Rate

To evaluate the efficacy and compare the fault detection algorithm, a fault detection rate r_{rate} is defined as:

$$r_{\text{rate}} = d_i/D_T \quad (\text{A.48})$$

, where d_i denotes the number of testing samples that have been correctly identified and D_T is the total number of testing samples used in this case study.

For the gPC model, the PDF profiles of the measured quantity Q_1 are used. The testing samples of measurements are collected at time instant t_s as shown in Fig.A.4, where the system is operating around particular fixed mean values. For the detection of fault with the GP model, the detection rate is evaluated for two different

case studies: (i) model calibration without adjustment and (ii) model calibration in combination with the minimal adjustment algorithm. The detection rate r_{rate} is summarized in Table A.6.

Table 6 Summary of fault detection rate

<i>Method</i>	<i>Noise level</i>		
	<i>1%</i>	<i>2%</i>	<i>3%</i>
<i>gPC</i>	0.93	0.91	0.89
<i>GP</i>	0.80	0.79	0.76
<i>GP-adjustment</i>	0.88	0.86	0.85

In Table A.6, 1000 testing samples for each mean value on the feed mass fraction x_{A0} are used to evaluate the detection rate, and the fault detection rates decrease as expected when the noise level increases. As seen, the gPC model based fault detection method can provide the best performance in terms of fault detection. The fault detection rate for the GP model with minimal model adjustment is ~88% for the first level of noise, which is ~5% lower than the gPC model. The possible explanation is that the PCA pre-screening step is used to remove the inessential variables and one principal component accounting for ~80% of the variance is used for the GP model calibration. Additionally, it is evident that the minimal model adjustment can improve the accuracy of fault detection. This confirms that the statistical model based fault detection method is sensitive to the training set, i.e., the available measurements. Thus, we must combine the GP model with the minimal model adjustment algorithm, since the limited observations used for the model training cannot provide accurate identification of the faults in the presence of uncertainties.

A.6.7 Combination of the gPC model with GP model

In previous case studies, either the gPC model or the GP are used to detect the faults. In this section, one GP model is generated based on the sensitivity analysis results obtained with the gPC model. Note that the sensitivity analysis of the GP model could be also studied using MC simulations, but it may make the evaluation of Eq. A.23 computational prohibitive.

In this case study the external heat Q_1 is used for the calibration of GP model, since it is more sensitive to the variation in the feed x_{A0} . The model calibration proceeds as per steps followed in the minimal adjustment algorithm. The calculated hyper-parameters are: $\sigma_G = 1.1774$; $l = 194.7484$, $\sigma_g = 0.0987$, respectively. Using an initial set with 30 measurement units, 89 measurements of Q_1 are added to the initial training set. For comparison, the efficiency of this combination is evaluated in terms of fault detection rate. For 1000 testing samples, the fault detection rate is ~92% for the lowest noise level. As compared to the results obtained with the PCA pre-screening based GP model, this combination can improve the fault detection rate by ~4 percent points and provide similar results as obtained with the gPC model based method.

A.6.8 Discussion and Comparison

The comparison of the two fault detection algorithms has shown their different capabilities. The gPC model based method can provide the most accurate results in terms of fault detection, and can be further used to evaluate the detectability of faults in a computationally efficient way, but the formulation of the gPC model is more complex as compared to the GP model. For example, as discussed in Section A.6.1, the computation of the gPC coefficients requires the application of Galerkin projection and this operation is limited to monomial or polynomial terms. The GP model has shown to be more flexible in terms of model generation, and it can provide a closed form expression of the mean and variance of the measured quantities. However, many compromises are required. For example, it is difficult to express the input of a model in an explicit and efficient way. Also, the calibration of the model generally depends on the training set. To overcome these limitations and improve fault detection performance, a hybrid method can be developed by combining these two approaches. As discussed above, the GP model generated based on the sensitivity analysis results from the gPC model may bring benefits to more complicated chemical processes, while providing equivalent accuracy in fault detection.

Finally, studies are conducted to compare the proposed algorithms with Monte Carlo (MC) simulations for the gPC model based method. It is worth noting that the model calibration as per the optimization in Eq. A.19 would be time prohibitive if MC simulations were to be used instead of a gPC approximation. For instance, the processor time required for one cost evaluation with MC (5000 samples) is ~15465 seconds. The search for the optimum in Eq. A.19 for each mean value requires 40~60 iterations and takes approximately 171 ~ 257 hours on average. However, the proposed method takes ~15 minutes to calculate the optimum in Eq. A.19 for all mean values, as can be seen in Table A.3. Also, the use of 5000 samples for calibrating the PDF profiles of measured variables from MC simulations resulted in lower fault detection rates of ~83%, as compared to the gPC method. Thus, a larger number of samples than 5000 would be required to obtain comparable fault detection rate as with the gPC approach, which may further increase the computational burden.

The computational time is also evaluated for the GP model calibration. According to the sensitivity analysis results obtained from the gPC method, the measurements of Q_1 and Q_2 are used for model calibration, since they are more sensitive to the random changes on the feed (see Table 2). For the minimal model adjustment, it takes ~4.2 hours. In this step, 30 set of initial observation units are used and 188 set of additional units are added to the training set. Additional studies are conducted to compare the fault detection rate with the multivariate statistical analysis method in the presence of parametric model uncertainty. For example, the fault detection rate is ~78% by combining the PCA model with the score discriminant method (Chiang, Russell, & Braatz, 2008), for which six measured quantities $\{T_i\}$ and $\{Q_i\}$ are used to generate the PCA model. This shows the potential of the proposed GP model based method for dealing with large problems and complicated applications.

A.7 Conclusion

A comparison study of two stochastic fault detection and diagnosis (FDD) algorithms are presented in this work for a nonlinear chemical process. For the first FDD method, a generalized polynomial chaos (gPC) is used to quantify and propagate the uncertainty onto the measured quantities, which can be used for the detection of faults. For the second method, a surrogate (empirical) metamodel is developed with Gaussian Process (GP) to approximate the dynamic value of the fault and its confidence interval. These methods have been evaluated in terms of the resulting fault detection rates. The results show the different capabilities of each method. Specifically, the gPC model based method is the more accurate method, but it requires more efforts are required when the implicit intrusive method is used to approximate the gPC coefficients of measured quantities. On the other hand, the GP model is easier to implement, but it is less accurate and the model calibration step requires additional adjustment. To improve the overall efficiency and flexibility, the outcomes from both methods can be combined leading to equivalent results between the gPC approach and the hybrid approach.

Appendix B

Reactivity Ratio Estimation in Copolymerization

(Adopted from Du et al., 2015, Computers and Chemical Engineering, submitted)

B.1 Overview

In this appendix, a generalized polynomial chaos (gPC) based methodology is proposed for reactivity ratio estimation in copolymerization, where the reactivity ratio parameters are assumed to be stochastic unknowns and need to be determined by comparison model predictions with limited experimental data. The gPC representations of unknown parameters are propagated into the nonlinear copolymerization first principles' models, which is followed by a maximum likelihood based approach fitting of the predictions from the gPC model and the experimental data. Thus, the reactivity ratio estimation is formulated as a stochastic inverse problem of finding the distributional stochastic reactivity ratio parameters with a maximum likelihood function. To apply the method to arbitrary uncertainty distribution, the Gram-Schmidt orthogonalization is employed to compute the orthogonal polynomials, which is an extension of the gPC method. The results show that the gPC model based reactivity ratio estimation is efficient and powerful, since it simultaneously provides both true values as well as the best estimates of the confidence interval around these true values. Beyond achieving estimation results, it is shown that the computational cost of the gPC model based method is significantly lower than Markov Chain Monte Carlo (MCMC) simulations, thus demonstrating the potential of the gPC method for dealing with other large nonlinear problems and real-time applications.

B.2 Introduction

Most mechanistic models involve nonlinear elements that make the parameter estimation a very challenging problem, since the nonlinear components affect the way where the model is being interpreted with the available data. For example, polymerization models are complex and nonlinear with respect to the reactivity ratio parameters and the propagation of data to model parameters is difficult. The majority of parameter estimation in such system is based on linearization of model and the results are poor and biased sometimes. For instance, most of the Kalman filters based methods usually exert a restrictive assumption about the distribution of the parameters to obtain desirable estimates. In addition, the uncertainty (or model error) is often superimposed on the parameters and the stochasticity is not measureable in many situations. Thus, the nonlinear parameter estimation problems require propagating the uncertainty into the process and studying the effect of the relevant stochastic dynamic property. For such cases, estimates with basic nonlinear regression may yield imprecise and biased parameter estimations, which degrade the accuracy of computations as model parameters evolve.

To alleviate these facts, assimilating the available measurements to calibrate the nonlinear model and refine the model forecast in order to reduce the associated uncertainties is a logical improvement over purely model

based estimation. The nonlinear least square method has been proposed to accommodate negligible error in the independent variables by assuming the error is independent and identically distributed (Bates & Watts, 1988). This method, however, leads to erroneous results in certain problems if the error in the independent variable is not negligible. The error in variable model (EVM) is therefore proposed for cases where the dependent and independent variables do not need to be distinguished (Keeler & Reilly, 1991). It treats each measurement as an unknown true value plus an error term and can handle implicit models, for which the dependent variable cannot be separated to one side of the equations. The disadvantage of EVM is that it is only tractable if all the distributions describing variation in the measurement are normal distributed and the assumptions on the error variances are known. Another issue with these aforementioned algorithms is that they often converge to a local optimum rather than a global optimum. Markov Chain Monte Carlo (MCMC) provides an alternate approach for finding the parameter estimates and can overcome limitations of classical methods (Geyer, 1992). However, executing MCMC algorithms is computational demanding, especially when differential or implicit equations are involved, which is one of the possible reasons for the limited application of MCMC in chemical engineering problems.

A useful alternative is to employ the spectral representations with the generalized polynomial chaos (gPC) (Xiu, 2009) expansion to approximate the uncertain parameters for nonlinear stochastic processes. The gPC method is an extension of the polynomial chaos of Wiener-Askey family, which has been reported to be an efficient way to propagate and quantify uncertainty in the stochastic dynamic systems. For instance, the gPC is combined with maximum likelihood method to estimate parameters. Point estimates of the process parameters are developed by substituting the gPC expressions into a likelihood function to solve the resulting maximum likelihood problem, and the estimates of parameter are transformed into a best-fit problem of random variables (Pence, et al., 2011; Chen-Charpentier & Stanescu, 2014). However, the accuracy of these methods is highly related to the number of data points used in the likelihood function, which maximizes the likelihood by fitting the predictions obtained from the gPC model and the experimental data. The gPC based Bayesian approach is proposed to provide point estimates of parameters, in which the numerical technique is used for non-polynomial nonlinearity since difficulties may arise during the computation of gPC model (Madankan, et al., 2013).

However, these aforementioned methods only provide the point estimates of parameters rather than a complete description of the probability density function (PDF) for each parameter or joint confidence region (JCR) between point estimates. Most importantly, the parameter estimation for nonlinear system with limited measurements is lacking. Since the nonlinear uncertain propagation is difficult and the model parameters affect the way that the model is being interpreted by the data (Snieder, 1998), the available referenced approaches (Madankan, et al., 2013; Andrieu, et al., 2003) may fail to provide accurate estimates with limited data. Moreover, the gPC expansion builds the connection between the stochastic processes and the Wiener-Askey scheme by approximating uncertainties with other random variables following the standard distributions. In terms of application, however, uncertain parameters can have distribution outside the Wiener-Askey framework. For instance, the lognormal distribution is a standard uncertainty distribution outside the Wiener-Askey scheme. Therefore, taking the

probability distribution of uncertain parameters into account in principle can produce more reliable parameter estimates, but little work has been carried out to the knowledge of the authors.

The current work addresses the limitations outlined above by combing the maximum likelihood estimation and the gPC model, which has the potential to be applied as an online estimation approach with limited measurements. For the arbitrary uncertainty distribution, the gPC approximation of the parameter uncertainty is extended using the Gram-Schmidt polynomial chaos (Witteveen & Bijl, 2006). The main feature of the proposed method is that it can provide the parameter estimates (mean values) and its statistical confidence intervals (variances) associated with these estimates simultaneously using a few measurements. Additionally, the proposed methodology can quantify the stochasticity in the parameters and propagate the variability to the measured quantities in an explicit fashion. With the gPC expression of measured quantities, another possible application is to dynamically solve an inverse problem and recursively estimate state/parameter (Du, et al., 2015).

This appendix is organized as follows: the mathematical tools used in the current work are given in Section B.3, i.e., description of stochastic inverse problem, introduction of the generalized polynomial chaos (gPC), Gram-Schmidt orthogonalization and the maximum likelihood method, followed by the formulation of two proposed estimation algorithms. In Section B.4, the proposed methodologies are illustrated by estimating the reactivity ratio parameters in the copolymerization, followed by results and discussion in Section B.5 and conclusion in Section B.6, respectively.

B.3 Background and Methodology Formulation

In this section, some brief descriptions of the mathematical tools used in the current work are summarized, which is followed by the proposed nonlinear stochastic parameter estimation methodologies.

B.3.1 Stochastic Inverse Problem

Assume a stochastic mathematical model with uncertain parameters can be described as $Y = \Xi(X)$, where Ξ is a nonlinear algebraic function or derivative with respect to time, $Y = \{y_i | i = 1, \dots, m\}$ is the measured variables vector, and X denotes a set of parameters in the model. Suppose the parameters vector X can be divided into two subgroups such as $X = \{x_1, x_2\}$, in which x_1 signifies these known parameters and x_2 is the unknown ones will be estimated by an inverse algorithm.

In the framework of stochastic inverse problems, the known parameter means that the probability density functions (PDFs) of these parameters (x_1) are given, while the unknown parameter (x_2) means that the PDFs shall be estimated by using measurements and the nonlinear first principles' model Ξ . For simplicity, it is assumed that both subgroups are independent and identically distributed (*i.i.d.*), thus the PDFs of parameters X can be rewritten as:

$$f_X(X) = f_X(\mathbf{x}_1, \mathbf{x}_2) = f_{x_1}(\mathbf{x}_1)f_{x_2}(\mathbf{x}_2) \quad (\text{B.1})$$

where the sizes of \mathbf{x}_1 and \mathbf{x}_2 are defined as n_1 and n_2 respectively, and the number of parameters in total is $n = n_1 + n_2$. The unknown parameters \mathbf{x}_2 considered in current work affecting the system are further described as:

$$\mathbf{x}_2 = \{\alpha_1, \alpha_2, \dots, \alpha_{n_2}\} \quad (\text{B.2})$$

$$\alpha_i = \bar{\alpha}_i + \Delta\alpha_i \quad (\text{B.3})$$

$$\kappa = \{\{\bar{\alpha}_i\}, \{\Delta\alpha_i\}\} \quad (\text{B.4})$$

where $\bar{\alpha}_i$ is the mean values for a particular unknown parameter, $\Delta\alpha_i$ is the variation around the mean $\bar{\alpha}_i$, and i is the i^{th} unknowns in \mathbf{x}_2 . To estimate the unknown parameters, i.e., κ , a set of measurements ($\hat{Y} = \{\hat{y}_1, \dots, \hat{y}_m\}$) should be available. The m represents the number of measured variables and each of them has n_{obs} obtainable measurements. In reality, this means that n_{obs} different trials of the process have been observed, and these trials are all modeled by the same operator Ξ . The unknown parameters in subgroup \mathbf{x}_2 , however, are operating around specific mean values $\{\bar{\alpha}_i\}$ with variation $\{\Delta\alpha_i\}$. It is worth mentioning that the measurements \hat{Y} may not exactly match with the model predictions due to model uncertainties and measurements noise. Using Equations B.1, B.2 and B.3, the stochastic parameter estimation then has been transformed into a new estimation problem, in which the unknowns (κ) can be determined by finding a set of estimates (means and variances) that best fit the data \hat{Y} .

B.3.2 Generalized Polynomial Chaos Expansion

The generalized polynomial chaos (gPC) expansion (Xiu, 2009) is the essential methodology of this current work, which is built upon the polynomial chaos expansion (Ghanem & Spanos, 1991) and the conceptualization of homogenous chaos (Wiener, 1938). It represents an arbitrary random variable of interest as a function of another random variable ξ (i.e., basic variable) with a prior distribution, and that function can be represented as a polynomial expansion from the Wiener-Askey framework. When each of the components of \mathbf{x}_2 (unknown parameters $\{\alpha_i\}$) are independent, there is one to one correspondence between the unknown parameter (α_i) and random basic variable (ξ_i) and it is efficient to associate each basic variable ξ_i ($i = 1, \dots, n_2$) to each unknown parameter α_i . It can be interpreted as that the variation on each parameter α_i is introduced by the corresponding random basic variable ξ_i . Thus, each of the unknown parameters $\{\alpha_i\}$ in Equation (2) can be approximated as:

$$\alpha_i = \sum_{k=0}^p \alpha_{i,k} \Phi_k(\xi) \quad (\text{B.5})$$

where $i = 1, \dots, n_2$, and $\alpha_{i,k}$ are the gPC expansion coefficients of the i^{th} unknown parameter in subgroup \mathbf{x}_2 . The p is the number of terms in B.5 that is necessary to represent the prior known distribution of α_i . It should be noted that the gPC expansion in Equation B.5 just needs more approximation terms, if the distributions of unknown parameters \mathbf{x}_2 are unknown prior. The random basic variables, $\xi = \{\xi_i\} = (\xi_1, \dots, \xi_{n_2})$, defines a multi-dimensional random space. The Φ_k are appropriate polynomial basis function of the random basic variables ξ . It is worth

mentioning that the polynomial basis functions Φ_k and the random basic variables ξ are modeling choice. For instance, Hermite polynomials can be chosen for normally distributed random variables ξ , since they are orthogonal with respect to the normal distribution functions. By propagation the gPC expansions to the system, the measured variables ($Y = \{y_j | j = 1, \dots, m\}$) can also be approximated in terms of orthogonal polynomial basis functions $\{\Phi_k\}$ as:

$$Y_j = \sum_{k=0}^P Y_{j,k} \Phi_k(\xi) \quad (\text{B.6})$$

where j is the j^{th} measured variables in Y and $Y_{j,k}$ is the gPC coefficients of the measured variables. For practical application, Equations B.5 and B.6 are often truncated to a finite number of terms for computational efficiency. Assume the number of terms to approximate a prior distribution of Equation B.5 is p , the total number of terms P used to approximate the measured variables in Equation B.6 can be calculated as:

$$P = ((n_2 + p)! / (n_2! p!)) - 1 \quad (\text{B.7})$$

For all random variables such as x_2 and Y , the first coefficients in Equations B.5 and B.6 represent the mean value and the rest terms are used to approximate the variability around the mean value. For example, normally distributed variables can be properly approximated by using only the first two terms of the Hermite polynomial, considering the properties of the polynomial basis and the definition of random variables. Obviously, the expansion dimensions in Equations B.5 and B.6 increases for arbitrary random variables (Xiu, 2009).

To solve these coefficients, Galerkin projection can be employed to project the stochastic mathematical model $Y = \Xi(X)$ into each polynomial chaos basis function $\{\Phi_k(\xi)\}$ and conduct the inner product as:

$$\langle Y(\xi), \Phi_k(\xi) \rangle = \langle \Xi(x_1, x_2(\xi)), \Phi_k(\xi) \rangle \quad (\text{B.8})$$

The inner product in Equation B.8, for instance, inner product between two vectors $\varphi(\xi)$ and $\varphi'(\xi)$, is defined by:

$$\langle \varphi(\xi), \varphi'(\xi) \rangle = \int \varphi(\xi) \varphi'(\xi) w(\xi) d\xi \quad (\text{B.9})$$

where the integration is calculated over the entire domain expanded by the random variables ξ , and $w(\xi)$ is the weighting function chosen for normalization purposes with respect to the type of polynomial basis functions used in Equation B.5. For polynomial nonlinearity, the Galerkin projection as Equation B.8 transforms the original stochastic system into a nonlinear deterministic system described by a set of coupled equations, which can be solved by any numerical methods such as Runge-Kutta method. However, difficulties may arise during the calculation of Equation B.8 for non-polynomial nonlinearity. The polynomial chaos quadrature (PCQ) is employed to overcome this issue, which replaces the exact integration in Equation B.9 with respect to ξ by a numerical integration and yields:

$$\langle \varphi(\xi), \varphi'(\xi) \rangle = \int \varphi(\xi) \varphi'(\xi) w(\xi) d\xi \approx \sum_{i=1}^{n_Q} \varphi(\xi_i) \varphi'(\xi_i) w_i(\xi_i) \quad (\text{B.10})$$

where ξ_i is the quadrature points used for the approximation and n_Q is the number of quadrature points in total. The PCQ method can be treated as Monte Carlo based evaluation of the stochastic system with samples generated with quadrature rules.

With the aforementioned Galerkin projection or PCQ approach, the statistical moments of the measured variables Y and the unknown parameters \mathbf{x}_2 represented by the gPCs can be efficiently calculated. For instance, the mean value and the variance for the j^{th} measured variable Y_j can be calculated as:

$$E(Y_j) = E \left[\sum_{k=0}^P Y_{j,k} \Phi_k \right] = Y_{j,0} E[\Phi_0] + \sum_{k=1}^P E[\Phi_k] = Y_{j,0} \quad (\text{B.11})$$

$$\begin{aligned} \text{Var}(Y_j) &= E[(Y_j - E(Y_j))^2] = E \left[\left(\sum_{k=0}^P Y_{j,k} \Phi_k - Y_{j(k=0)} \right)^2 \right] \\ &= E \left[\left(\sum_{k=1}^P Y_{j,k} \Phi_k \right)^2 \right] = \sum_{k=1}^P Y_{j,k}^2 E(\Phi_k^2) \end{aligned} \quad (\text{B.12})$$

For stochastic inverse problems, most of the methods minimize the discrepancy (or error function) between the collected measurements and the model predictions with point estimates on parameters. The ability to calculate the mean and the variance is the main rationale of the gPC expansions in terms of stochastic parameter estimation, since these quantities have to be repeatedly estimated in an optimization algorithm. However, this method can be affected by the number of available measurements used to optimization. Generally, the error function in optimization will have multiple minima. For instance, it cannot find the global optimum sometimes if the available dataset is small. To overcome this issue, the concept of the gPC expansion is used on unknown input parameters \mathbf{x}_2 and the coefficients in Equation B.5 are optimized by a maximum likelihood optimization which is further explained below.

B.3.3 Gram-Schmidt Orthogonalization

The weighting function $w(\xi)$ in Equation B.9 has to be specified with respect to the uncertainty distribution of the uncertain parameters. In application, however, the distribution of stochastic parameters is unknown or outside the Wiener-Askey framework. The Gram-Schmidt orthogonalization is used to compute a set of orthogonal polynomials with respect to $w(\xi)$, according to the uncertainty distribution of stochastic parameters. Multidimensional orthogonal polynomials, i.e., $\{\Phi_k(\xi)\}$, can be constructed using the tensor product of the one-dimensional polynomials $\{\Phi_k(\xi)\}$ computed by the Gram-Schmidt algorithm (Witteveen & Bijl, 2006). A set of coupled one-dimensional monic orthogonal polynomials can be formulated as:

$$\Phi_k(\xi) = e_k(\xi) - \sum_{j=0}^{k-1} c_{kj} \Phi_j(\xi) \quad (\text{B.13})$$

where $\Phi_0 = 1$, $k = 1, 2, \dots, p$ and p is the number of terms used for approximation in Equation B.4. The definition of $c_{k,j}$ can be described as:

$$c_{k,j} = \frac{\langle e_k(\zeta) \Phi_j(\zeta) \rangle}{\langle \Phi_j(\zeta) \Phi_j(\zeta) \rangle} \quad (\text{B.14})$$

where the $\{e_k(\zeta)\}$ are polynomials of exact degree k . The inner product in Equation (14) is evaluated numerically in order to apply the gPC approximation to any arbitrary weighting functions. Thus, the gPC approximation has been generalized to any type of stochastic parametric uncertainty.

B.3.4 Maximum Likelihood Estimation with gPC model

As discussed in Section B.3.1, for the case that the joint PDF of \mathbf{x}_1 is $f_{x_1}(\mathbf{x}_1)$ and the joint PDF of \mathbf{x}_2 is $f_{x_2}(\mathbf{x}_2)$, let $f_Y(\hat{\mathbf{Y}}, \kappa)$ denote the joint PDF of the measurements. Once again, the κ is the counterpart of unknown parameters $f_{x_2}(\mathbf{x}_2)$, i.e., the means and variances of \mathbf{x}_2 , which can be calculated by Equations B.11) and B.12. Thus, the joint PDF of the measured variables $f_Y(\hat{\mathbf{Y}}, \kappa)$ is related to the unknown gPC coefficients of stochastic parameters, which is of course unknown analytically.

To evaluate how well the estimates (κ) fit the data and what sort of uncertainty is associated with the estimates, the error between the measurements and the model predictions is assumed to be normal distributed. Thus, a Gaussian kernel is employed and the kernel density approximation of the measured variable $f_Y(\hat{\mathbf{Y}}, \kappa)$ is defined as:

$$\hat{f}_Y(\hat{\mathbf{Y}}, \kappa) = \frac{1}{n} \sum_{k=1}^n K_h(\hat{\mathbf{Y}} - \Xi(\mathbf{x}_1, \kappa)) \quad (\text{B.15})$$

where n is the number of known samples of \mathbf{x}_1 used for approximation, K_h is a Gaussian kernel function that fits the normal distributed errors between measurements and model predictions, $\hat{\mathbf{Y}}$ is a set of measurements, and $\Xi(\mathbf{x}_1, \kappa)$ is the gPC approximation of measured variables conditioned on samples of \mathbf{x}_1 as well as a set of unknown parameters κ of \mathbf{x}_2 .

By using Equation B.15, the likelihood function of the measured variables can be estimated as:

$$\ell(\kappa; \hat{\mathbf{Y}}) = \prod_{j=1}^{n_{obs}} \hat{f}_Y(\hat{\mathbf{Y}}^m; \kappa) \quad (\text{B.16})$$

where j is the j^{th} set of measurements inside $\hat{\mathbf{Y}}$, the estimates of the unknown κ can be obtained by maximizing the likelihood estimator ℓ with respect to κ as:

$$J = \max_{\Omega_I} \prod_{j=1}^{n_{obs}} \hat{f}_Y(\hat{\mathbf{Y}}^m; \kappa) \quad (\text{B.17})$$

where Ω_I is decision variables and is a vector consisting of the gPC coefficients for the unknown parameters \mathbf{x}_2 . Once the optimization is done, the counterpart part κ in Equation B.5, i.e., estimation of unknown parameters and

their confidence intervals, can be calculated with Equations B.10 and B.11. Using the orthogonality property of the basis polynomial functions, the coefficients in Equations (5) and (6) are obtainable when dealing with polynomial terms by using Galerkin projection. However, the integration of non-polynomial terms is not straightforward. To manage the difficulties, two different gPC model based parameter estimation methodologies are proposed, and each algorithm is formulated with a series of procedures and further discussed as follows.

B.3.5 Parameter Estimation Algorithms

Two approaches are developed using the gPC methodology and the maximum likelihood function. The first method develops a gPC symbolic model to integrate multiple sources of known information and estimate the unknown parameters, i.e., means and variances of \mathbf{x}_2 in Equation B.1.

Algorithm 1 – For this method, the uncertainty quantification step on the measured variables with Galerkin project is skipped in the optimization problem of Equation B.17. Instead the samples associated with each basic variable ξ_i in Equation B.5 are directly used to perform Monte Carlo simulations, while maximizing the likelihood Equation B.17. The benefit is that the samples are randomly chosen from a prior standard distribution and tend to perform better in capturing the global structure. The *Algorithm 1* involves a series of steps as follows.

Inputs initialization:

- (1) Input the samples of known parameters \mathbf{x}_1 and the available data of the measured variables $\hat{\mathbf{Y}}$.
- (2) Choose the order of polynomials (p) used to approximate the unknown parameters \mathbf{x}_2 in gPC model, decide the polynomial basis function Φ_k , and then formulate the gPC symbolic approximations of \mathbf{x}_2 . Once again, the counterpart of unknown parameters κ in Equation B.4 can be calculated with the gPC coefficients by using Equations B.10) and B.11.
- (3) Substitute the gPC approximation of the unknown parameters \mathbf{x}_2 into the nonlinear first principles' model Ξ , and generate a new gPC symbolic model with respect to the unknown gPC coefficients.
- (4) Decide the number of samples (l) for each random variable ξ_i and generate samples from the standard basis distributions (ξ).
- (5) Initialize the initial guesses for $\{\alpha_{i,k}^{[0]}\}$ in Equation B.5, i.e., the gPC coefficients for each unknown stochastic parameter.

Optimization with Equation (B.17):

- i. Use each of the input samples of known parameters \mathbf{x}_1 and the initial values $\{\alpha_{i,k}^{[0]}\}$ to perform l Monte Carlo simulations with the nonlinear gPC symbolic model and the samples generated in (4), thus l model predictions are obtained for each input sample of \mathbf{x}_1 .
- ii. Calculate the mean value of the model predictions simulated in (i) for each of the input samples.

- iii. Calculate the Euclidean distance between the mean values in (ii) and the corresponding measurements of the measured variables \hat{Y} .
- iv. Solve the optimization problem B.17 to obtain the optimum of the unknown gPC coefficients $\{a_{i,k}^{[*]}\}$. Each optimization iteration entails the repeated evaluation of the gPC model and updates the prediction mean and the Euclidean distance in Steps (ii) and (iii).

The use of the gPC expansions and the samples of random basic variables ξ in the *Algorithm 1* significantly improve the efficiency while taking the probabilistic uncertainties into account, as compared with the standard Monte Carlo simulations. For nonlinear models, a major disadvantage of the Monte Carlo type sampling based method is the requirement for appropriate samples. To ensure samples prediction converges to the theoretical value, a large number of simulations are often required, which in turn may increase the computation burden, especially for high dimensional problems. In this method, however, the samples are generated from the random basic distribution of ξ , which can release the requirement on the number of samples and improve the computational efficiency.

The *Algorithm 1* cannot provide an explicit expression of the measured variables. To mathematically propagate and quantify the effect of parametric uncertainty onto the measured variables in a computational efficient fashion, the polynomial chaos quadrature (PCQ) is used in the current work. As discussed in Section B.3.2, all moments of random variables, i.e., x_2 and Y , are just functions of their gPC expansion coefficients. Hence, the optimization problem Equation B.14 can be reformulated with the statistical moments calculated from the measurement data and the gPC coefficients of the measured variables.

Algorithm 2 – For the purpose of estimating the unknown stochastic parameters and their confidence interval, as well as approximating the variation on measured variables introduced by unknown parameters and measurement noise, the PCQ is used to calculate the analytical gPC expression of the measured quantities. To this objective, the mean values and the variances of unknown parameters are obtained from optimizing a modified joint PDF $f_Y(\hat{Y}, \kappa)$ of measured variables as:

$$J = \max_{\Omega_2} \prod_{m=1}^{n_{obs}} \sum_{k=1}^n K_h((v_{1,k} - \gamma_{1,k})^2 + (v_{2,k} - \gamma_{2,k})^2) \quad (\text{B.18})$$

where n is the number of known samples of x_1 , K_h is a Gaussian kernel function, v_1 and v_2 are the predicted mean and variance of the measured variables that are calculated with the gPC models. Using Equations B.10 and B.11, these values can be explicitly computed. $\gamma_{1,k}$ and $\gamma_{2,k}$ are the mean and variance computed with the measurements, and Ω_2 is the decision variables vector consisting of the gPC coefficients for the unknown stochastic parameters x_2 . To solve the optimization as Equation B.18, the following procedures are preceded.

Inputs initialization:

- (1) Input the samples of known parameters \mathbf{x}_1 and the available data of the measured variables $\hat{\mathbf{Y}}$.
- (2) Choose the order of polynomials (p) used to approximate the unknown parameters \mathbf{x}_2 in Equation B.5, determine the polynomial basis function Φ_k , and then formulate the gPC approximations for both unknown parameters \mathbf{x}_2 and measured variables \mathbf{Y} .
- (3) Substitute the gPC approximations in Step (2) into the nonlinear first principles' model Ξ , and generate a new gPC symbolic model by using polynomial chaos quadrature (PCQ), which transform the original stochastic model into a set of coupled deterministic equations (gPC symbolic model).
- (4) Set initial values of $\{\alpha_{i,k}^{[0]}\}$ in Equation B.5, i.e., the gPC coefficients for each unknown parameter.

Optimization with Equation (B.15):

- i. Substitute each input sample of parameters \mathbf{x}_1 and the initial values $\{\alpha_{i,k}^{[0]}\}$ into the gPC symbolic model generated in Initialization Step (3).
- ii. Solve the gPC coefficients for the measured variables from the gPC symbolic model.
- iii. Using Equations B.10) and B.11, calculate the mean and variance of the measured variables with the gPC coefficients in Step (ii).
- iv. Calculate the Euclidean distance between the mean value in (iii) and the mean value computed from the collected measurements of the measured variables $\hat{\mathbf{Y}}$.
- v. Calculate the Euclidean distance between the variance in (iii) and the variance computed from the collected measurements of the measured variables $\hat{\mathbf{Y}}$.
- vi. Solve the optimization Equation B.18 to obtain the optimum of the unknown gPC coefficients $\{\alpha_{i,k}^{[*]}\}$. Each optimization iteration entails the repeated evaluation of the gPC expansion and the Euclidean distance as in Steps (iv) and (v).

As compared with the *Algorithm 1*, this method provides an explicit gPC expression of the measured variables, while estimating the unknown uncertain parameters. It can be further employed to evaluate how uncertainties of a dynamical system's parameters manifest the effect on the measured variables.

Gram-Schmidt orthogonalization – The Gram-Schmidt polynomial chaos can be applied to both approaches above, if the probability distribution of stochastic unknown parameters is outside of the Wiener-Askey scheme. A few more procedures can be performed to replace the Step (2) in the *Inputs initialization* for both algorithms, which involve as per the following steps. (i) Determine the weighting function $w(\xi)$ in Equation (9); (ii) Compute the polynomial basis function $\{\Phi_k(\xi)\}$ with respect to a pre-assigned weighting function $w(\xi)$ in (i), using the Gram-Schmidt algorithm; (iii) Choose the order of polynomials (p) used to approximate the unknown parameters \mathbf{x}_2 in Equation B.5, (iv) Formulate the gPC approximations for unknown parameters \mathbf{x}_2 in the *Algorithm 1* or formulate the gPC approximations for both unknown parameters \mathbf{x}_2 and measured variables \mathbf{Y} in the *Algorithm 2*.

The rest parts of algorithm follow the same procedures as described in the proposed methods. The employment of Gram-Schmidt orthogonalization algorithm thus extends our algorithms to estimate an unknown parametric input for any type of probability distribution.

B.4 Case Studies: Reactivity Ratio Estimations in Copolymerization

To demonstrate the proposed gPC model based parameter estimation methodologies, the reactivity ratio estimation in copolymerization is revisited as an example, which is a very active research topic and has been gained popularity in the literature throughout several decades. This process is considered sufficiently complicated to illustrate the methodology in terms of the presence of nonlinear behavior, uncertainty and disturbance (measurement noise). Moreover, the number of experiments is usually limited for the reactivity ratio estimation, which cannot provide a reliable source of information. In this work, different aspects for the reactivity ratios estimation in copolymerization system are discussed in terms of application.

To define reactivity ratio parameters, the reaction below is considered:



where M_j denotes the monomer j , $R_{n,i} \cdot$ represents a propagating copolymer radical with n monomer units and the last unit containing monomer i . For a copolymer system, i and j can take on values of 1 and 2. Using the four propagating rate constant, the reactivity ratio parameters can be defined as follows:

$$r_1 = \frac{k_{11}}{k_{12}} \quad (\text{B.20})$$

$$r_2 = \frac{k_{22}}{k_{21}} \quad (\text{B.21})$$

The most popular copolymerization model is Mayo-Lewis model, which describes the instantaneous copolymer composition. As shown in Equation B.22, the Mayo-Lewis model provides a relationship between the true values of the instantaneous copolymer composition (F_1), the unreacted monomer composition (f_1 and f_2) and the reactivity ratios (r_1 and r_2). However, the model assumes that the monomer composition does not change with conversion, which limits its validity to low conversion level, since compositional drift can occur as the polymerization reaction progresses.

$$F_1 = \frac{r_1(f_1)^2 + f_1 f_2}{r_1(f_1)^2 + 2f_1 f_2 + r_2(f_2)^2} \quad (\text{B.22})$$

In order to use a larger conversion range in the polymerization reaction, the Mayo-Lewis equation was integrated by Meyer and Lowry. The analytical integration of the equation assumes that the reactivity ratios are constant during the polymerization reaction. The solution is referred to as the Meyer-Lowry model and is given as:

$$X_n = 1 - \left(\frac{f_{10} - \bar{F}_1 X_n}{f_{10}(1 - X_n)} \right)^\alpha \left(\frac{1 - X_n - f_{10} - \bar{F}_1 X_n}{(1 - f_{10})(1 - X_n)} \right)^\beta \left(\frac{(1 - X_n)(\delta - f_{10})}{(\delta - \delta X_n + \bar{F}_1 X_n)} \right)^\gamma \quad (\text{B.23})$$

where the parameters α , β , γ , and δ satisfy the following equations:

$$\alpha = \frac{r_2}{(1 - r_2)} \quad (\text{B.24})$$

$$\beta = \frac{r_1}{(1 - r_1)} \quad (\text{B.25})$$

$$\gamma = \frac{1 - r_1 r_2}{(1 - r_1)(1 - r_2)} \quad (\text{B.26})$$

$$\delta = \frac{(1 - r_1)}{(2 - r_1 - r_2)} \quad (\text{B.27})$$

$$X_n = X_w \frac{M_{w1} f_{10} + M_{w2}(1 - f_{10})}{M_{w1} \bar{F}_1 + M_{w2}(1 - \bar{F}_1)} \quad (\text{B.28})$$

In Equation B.23, the variables \bar{F}_1 , X_n and f_{10} denote the cumulative copolymer composition, molar conversion and the initial monomer mole fraction respectively, the reactivity ratios are r_1 and r_2 . Also, X_n in Equation B.23 can be replaced by mass conversion with Equation B.28, where M_{w1} and M_{w2} are the molecular weights of monomer. The assumption of negligible compositional drift is no longer an issue, since the Meyer-Lowry model takes the conversion of the polymerization reaction into account. Thus, this model can be applied to measurement data obtained at low to moderate conversion levels.

To estimate the reactivity ratio with the Mayo-Lewis model and the Meyer-Lowry model, the data are taken from (Madruga & Fernandez-Garcia, 1994; Madruga & Fernandez-Garcia, 1995) for both low conversion level and high conversion level. These data are given in Table B.1 and Table B.2, respectively, where the free radical copolymerization was carried out in benzene at 50°C. For the low conversion level, 16 experimental trials in total were conducted at various monomers feed composition f_{10} , and the conversion levels X_w as well as the output copolymer compositions \bar{F}_1 were measured. For the high conversion levels, there are 33 experimental trials in Table B.2.

Table B.1 Experimental data for DBI/MMA copolymerization at low conversion level

No.	f_{10}	X_w (wt%)	\bar{F}_1	No.	f_{10}	X_w (wt%)	\bar{F}_1
1	0.035	6.46	0.025	9	0.491	3.86	0.411
2	0.064	6.04	0.043	10	0.547	4.18	0.459
3	0.065	8.05	0.056	11	0.599	3.17	0.512
4	0.199	5.34	0.165	12	0.698	4.47	0.623
5	0.301	5.09	0.246	13	0.798	4.64	0.732
6	0.301	5.97	0.259	14	0.301	4.82	0.272
7	0.499	3.61	0.377	15	0.492	4.05	0.416
8	0.453	8.77	0.382	16	0.700	4.75	0.638

Table B.2 Experimental data for DBI/MMA copolymerization at high conversion level

No.	f_{10}	X_w (wt%)	\bar{F}_1	No.	f_{10}	X_w (wt%)	\bar{F}_1	No.	f_{10}	X_w (wt%)	\bar{F}_1
1	0.3	22.2	0.237	12	0.5	12.0	0.401	23	0.7	26.7	0.634
2	0.3	43.8	0.273	13	0.5	37.0	0.439	24	0.7	31.0	0.631
3	0.3	48.8	0.268	14	0.5	42.2	0.448	25	0.7	34.1	0.632
4	0.3	47.4	0.272	15	0.5	47.3	0.448	26	0.7	40.9	0.661
5	0.3	54.8	0.265	16	0.5	54.5	0.454	27	0.7	42.5	0.684
6	0.3	58.9	0.270	17	0.5	56.8	0.466	28	0.7	44.0	0.645
7	0.3	55.4	0.269	18	0.5	58.7	0.464	29	0.7	49.1	0.649
8	0.3	65.3	0.286	19	0.5	65.8	0.487	30	0.7	52.7	0.675
9	0.3	70.6	0.269	20	0.5	79.6	0.427	31	0.7	54.5	0.666
10	0.3	86.6	0.324	21	0.7	16.3	0.645	32	0.7	64.3	0.675
11	0.5	8.1	0.405	22	0.7	23.1	0.634	33	0.7	71.8	0.690

B.5 Results and Discussion

The implementation of methodologies described in Section B.3.5 can now be applied to the Mayo-Lewis model and the Meyer-Lowry model, respectively. By using the monomer reactivity ratio example, our goal in the current work is to offer comparisons and indicate some of the limitations of existing approaches, while presenting new approaches for estimating the unknown parameters superimposed with uncertainties. The main objectives are specifically to: (i) provide the best possible estimates of the parameters as well as their confidence intervals, (ii) take into account all uncertain information, such as the model error and the measurement noise, which cannot be measured due to lack of exact knowledge, (iii) develop user friendly algorithms with computational efficiency, (iv) quantify and evaluate how the stochasticity on unknown parameters affects the measured variables, and (v) demonstrate how the proposed method can be applied to other nonlinear parameter estimation problems.

As a key point to achieve these objectives, the parameter estimation problem is first studied. It consists of estimating the mean values of the unknown reactivity ratio (r_1 and r_2) and their confidence interval based on the measurements. For simplicity, the mole fraction in the feed is assumed to be known parameter (x_1 in Equation B.1), and the ratio parameters r_1 and r_2 are assumed to be unknown uncertain parameters (x_2 in Equation B.1).

B.5.1 Reactivity Ratio Estimation

As discussed in Section B.3.2, the basic idea of the gPC is to approximate a random variable by another random event with the given prior information on probability density function (PDF). It is assumed that the uncertainty on ratio parameters (r_1 and r_2) is introduced by the normally distributed random events for simplicity. Thus, the Hermite polynomials are used. In addition, the dimension of random space is 2, since two random parameters are considered in this case study. Further assumption is made that the two random events are independently identically distributed (*i.i.d.*), i.e., $\xi = (\xi_1, \xi_2)$ and $f(\xi_1, \xi_2) = f(\xi_1) f(\xi_2)$, where f denotes the probability density function (PDF). Since the measurement noise in copolymerization is multiplicative, the natural logarithm is performed on both sides of Equations B.22 and B.23, to make a fair comparison with published results such as

previous studies by (Madruga & Fernandez-Garcia, 1994; Kazemi, et al., 2011). Compared with the referenced work (Kazemi, et al., 2011) where the error in variable model (EVM) was employed, the linearization of models is not a must in the current work. Also, the estimation results with proposed algorithms in this work are compared with the point estimates obtained by the Gauss-Newton method and the Markov Chain Monte Carlo (MCMC) simulations. The sampling procedure referred to as the Metropolis-Hastings (MH) algorithm is utilized for MCMC (Andrieu, et al., 2003). Using data in Table 1, Table B.3 shows the reactivity ratios estimation results for both low and high conversion levels.

Table B.3 Parameters estimations for reactivity ratios (r_1 and r_2)

	<i>Model</i>	<i>Conversion level</i>	<i>Method</i>	r_1	r_2
Madruga and Fernandez-Garcia	Mayo-Lewis	Low	-	0.7170	1.329
Kazemi, Duever and Penlidis	Mayo-Lewis	Low	EVM	0.7098	1.313
Kazemi, Duever and Penlidis	Meyer-Lowry	Low	EVM	0.7129	1.310
Kazemi, Duever and Penlidis	Meyer-Lowry	High	EVM	0.6794	1.229
Current work	Mayo-Lewis	Low	Gauss-Newton	0.7127	1.286
Current work	Meyer-Lowry	Low	Gauss-Newton	0.6509	1.243
Current work	Meyer-Lowry	High	Gauss-Newton	0.6278	1.234
Current work	Mayo-Lewis	Low	MCMC	0.6949	1.311
Current work	Meyer-Lowry	Low	MCMC	0.6817	1.319
Current work	Meyer-Lowry	High	MCMC	0.6649	1.248
Current work	Mayo-Lewis	Low	Algorithm 1	0.6929	1.311
Current work	Meyer-Lowry	Low	Algorithm 1	0.6738	1.293
Current work	Mayo-Lewis	Low	Algorithm 2	0.6917	1.294
Current work	Meyer-Lowry	Low	Algorithm 2	0.6778	1.289
Current work	Meyer-Lowry	High	Algorithm 1	0.6738	1.317
Current work	Meyer-Lowry	High	Algorithm 2	0.6700	1.288

For comparison, only the mean values of the unknown parameters are given in Table B.3, using the proposed methods. The mean values are the first terms in each of the gPC expansions of the stochastic unknown parameters. The other coefficients providing information about the variance will be further discussed. For the MCMC simulations, the convergence is diagnosed using the acceptance rates and the Markov Chain time series plots. Once the convergence is confirmed, the reactivity ratios estimates are calculated by averaging all the MCMC samples.

In Table B.3, the reactivity ratios of the low conversion range data are first estimated using the Mayo-Lewis model and the Meyer-Lowry model, respectively. According to the results, the estimates of the reactivity ratios obtained with different models are very similar and our results are in a good agreement with the referenced results. For instance, the point estimates reported with EVM method are 0.7098 and 1.313 with the Mayo-Lewis model at low conversion levels. Using the proposed *Algorithm 1* in the current work, the results are 0.6929 and 1.311 with the same model and experimental conditions.

Further verification is conducted by comparing the results in the current work with the Gauss-Newton algorithm and the MCMC simulations. According to this table, the values of the reactivity ratios obtained with

different methods are all in a good agreement, which confirms the efficiency of the proposed algorithms in terms of parameter estimations. Comparison of the gPC results with the Gauss-Newton as well as the MCMC shows that, the nonlinear regression method (Gauss-Newton) underestimates the unknown parameters, when the Meyer-Lowry is used. This observation demonstrates the importance of accurately estimating the parameters in nonlinear models, and provides motivation for apply more advanced techniques instead of approximation methods.

Compared with the other methods, the advantage of the gPC method is that it can provide the point estimates and the variance around these point estimates simultaneously. To verify this point, Table B.4 shows the results obtained from the optimization problems of Equations B.17 and B.18, respectively. In Table B.4, $\{r_{1i}\}$ and $\{r_{2i}\}$ represents the gPC coefficients of the unknown parameters calculated with the gPC models ($i = 0, 1$). The point estimates of parameters are r_{10} and r_{20} , while r_{11} and r_{21} represents the variations around these point estimates. Using Equation B.12, the last two columns provide the variabilities (variances) around each of the mean values of the reactivity ratio parameters.

Table B.4 Parameter estimation results (gPC coefficients)

Copolymerization model	Conversion level	Method	gPC coefficients				Variance	
			r_{10}	r_{11}	r_{20}	r_{21}	V_{r1}	V_{r2}
Mayo-Lewis	Low	Algorithm 1	0.6929	0.0580	1.311	0.0452	0.0034	0.0020
Meyer-Lowry	Low	Algorithm 1	0.6738	0.0325	1.293	0.0573	0.0011	0.0033
Mayo-Lewis	Low	Algorithm 2	0.6917	0.0618	1.294	0.0467	0.0038	0.0022
Meyer-Lowry	Low	Algorithm 2	0.6778	0.0401	1.289	0.0503	0.0016	0.0025
Meyer-Lowry	High	Algorithm 1	0.6738	0.0368	1.317	0.0356	0.0014	0.0013
Meyer-Lowry	High	Algorithm 2	0.6700	0.0392	1.288	0.0322	0.0015	0.0010

B.5.1 Parameter Estimation with Gram-Schmidt Polynomial Chaos

As discussed the gPC approximation employs the classical orthogonal polynomial basis function in the framework of Wiener-Askey, in which only a limited number of standard uncertainty distributions can be considered. In this section the Gram-Schmidt polynomial chaos is utilized to verify the efficiency by finding a set of polynomials basis functions orthogonal to a given weighting function. The uncertainty of parameters follows the same assumption as done in Section B.5.1. The weighting function used in the Gram-Schmidt orthogonalization is equal to the probability density function (PDF) that is used to transform the uncertain parameters to the standard normal distributed domain. For comparison, the Hermite and Legendre polynomial are used in the Wiener-Askey scheme respectively, since Hermite polynomials are the optimal polynomial for the normal distribution and Legendre is suitable for uniform distribution. Using the Mayo-Lewis model and the low conversion data in Table B.1, Table B.5 shows the point estimates of unknown reactivity ratio parameters and Fig.B.1 shows one set of the estimations of variances with the Gram-Schmidt polynomial chaos and Wiener-Askey polynomial chaos, using the *Algorithm 1* in the current work.

As seen in Table B.1, the point estimates of unknown parameters are in a good agreement with different polynomial basis functions, regardless of the total number of terms used in the gPC approximation. However,

the estimation of variances is different as shown in Fig.B.1. Compared with the ones calculated with Hermite and Gram-Schmidt polynomials, it is observed that there is noticeable difference using the Legendre polynomial, since it is the optimal choice for uniform distribution. By contrast, the difference between Hermite and Gram-Schmidt is relatively small, but the results are not identical. The possible explanation may be that samples of each random variables ξ_i generated from the basis distributions (ξ) are not the identical samples used in the *Algorithm 1*. Also there is model error which is induced by the truncated terms used in the gPC approximation, and lack of exact knowledge about the unknown parameters since a limited number of measurements is used. Furthermore, the Legendre polynomial approaches to the results obtained with Hermite and Gram-Schmidt as the number of polynomial order increases, but this may increase the computational time.

Table B.5 Point estimate results for normal distributed stochasticity

Polynomial chaos	p in Eq.B.5	Point estimates	
		r_1	r_2
Askey-Hermite	2	0.6929	1.311
Askey-Legendre	2	0.6978	1.284
Gram-Schmidt	2	0.6905	1.293
Askey-Hermite	3	0.6917	1.294
Askey-Legendre	3	0.6955	1.284
Gram-Schmidt	3	0.6877	1.335

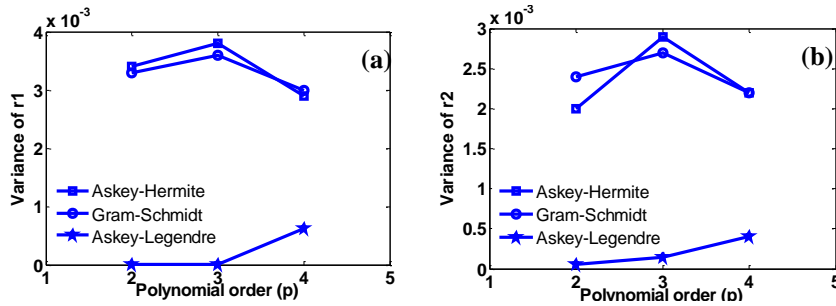


Figure B.1 Verification of Gram-Schmidt by comparison with Wiener-Askey framework

To evaluate the efficiency of the Gram-Schmidt polynomial chaos, a criterion is defined. The range of variability on unknown parameter is assigned with Hermite polynomial basis functions. If the estimates are outside the prescribed range, the corresponding estimations of variance are ruled out and defined as an inappropriate estimate. In addition, a percentage of the acceptable estimates defined as in Equation B.29 is utilized to evaluate the estimation results of variances on unknown stochastic parameters, where N_{trial} is the total number of trials and n_{trial} is the number of desirable estimates satisfying the criterion explained above.

$$A_{rate} = n_{trial}/N_{trial} \quad (B.29)$$

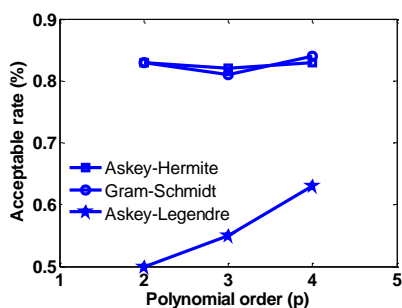


Figure B.2 Comparison of A_{rate} for three polynomials

In Fig.B.2, 100 trials are studied for each polynomial chaos order with three types of polynomial basis functions, using the *Algorithm 1* in the current work. As seen, the acceptable rate A_{rate} increases as more terms are used in the gPC approximation with the Legendre polynomial in the Wiener-Askey scheme. However, A_{rate} is ~20 percent point lower as compared with others, i.e., Hermite and Gram-Schmidt polynomials. As expected, the acceptable rate A_{rate} obtained with Hermite and Gram-Schmidt polynomials are similar. This verifies that the Gram-Schmidt is applicable to the other types of probability distribution. It is worth mentioning that a simplifying assumption in this work is that the uncertainty on unknown parameters is normal distributed, thus the Hermite polynomial is the optimal model choice. According the comparison above, the Gram-Schmidt can provide similar results in terms of the parameter estimations and the acceptable rate A_{rate} . Therefore, it renders the possibility that the methodology in this work can be extended to other parameter estimation problems, where the distribution of uncertain parameters is unknown.

B.5.3 Joint Confidence Regions for Parameter Estimation

The evaluation of parameter estimation results with confidence intervals allows us to determine whether the results are reliable. A joint confidence region (JCR) is usually utilized to visualize the measure of the uncertainty involved in the estimates, for cases where more than single parameter is considered. Further studies are conducted to investigate the effect of parameter estimates on the JCR. To efficiently choose the model while taking the levels of conversion into account, the performance of the proposed approaches is compared and verified with the MCMC simulations based on their calculated JCRs area. Fig.B.3 shows the random sample points and the estimated JCRs at lower conversion levels by using the proposed algorithm and the MCMC simulations with the Mayo-Lewis model, in which the point estimates are also provided. For brevity, only the results obtained with the *Algorithm 1* are utilized for illustration. The distribution of sample points and the estimated JCRs with the *Algorithm 2* have the similar shapes, since the mean values of unknown reactivity ratios do not change too much as seen in Table B.1.

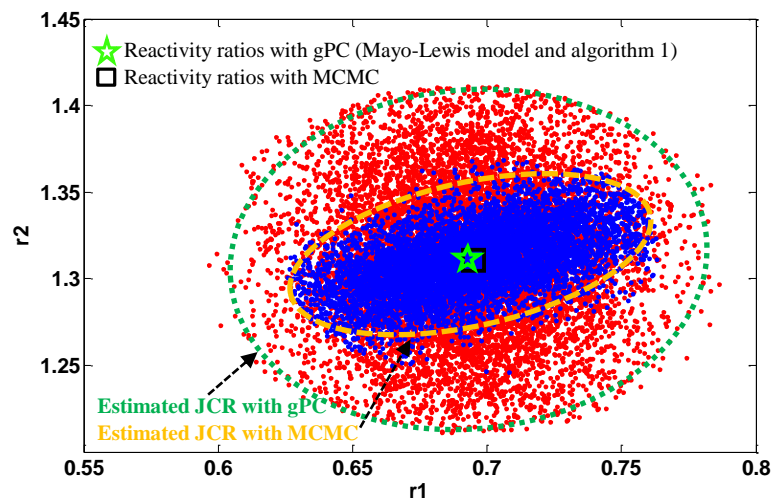


Figure B.3 Point estimates, random sample points and estimated Joint Confidence Regions (JCRs) for reactivity ratios using the Mayo-Lewis model in copolymerization of DBI/MMA at lower conversion levels

It can be seen in Fig.B.3, the point estimates obtained with the gPC method and the MCMC simulations are very similar. However, the area generated by the random sample points with the gPC model is larger than the MCMC method, which confirms that the gPC method can provide the upper and lower bounds of the point estimates. For parameter estimation, the smaller area of the sample points (JCR) is normally expected. However, it should be noted that the smaller JCR may not be able to provide enough information about the variation around the point estimates. For example, assumption is always made that the system is operated around a fixed parameter for most parameter estimation problems. However, the primary challenge is that all models are imperfect, either in their form, in the numerical values of parameters in equations or in the solution of these equations. The current study can provide both the magnitude of the variation in the parameter estimates and the impact of this variation on the estimation, while taking the worst case scenario simulations into account. Thus, the proposed approach of course introduces a relatively larger JCR, meanwhile provides desirable point estimates. Additionally, the current work can simply provide the probability at a particular estimation point.

Additional studies were conducted to compare the parameter estimation performance with different models using the low conversion data in Table B.1. Fig.B.4 shows the point estimates, random samples and the JCRs with the gPC method and the MCMC simulations for the Meyer-Lowry model.

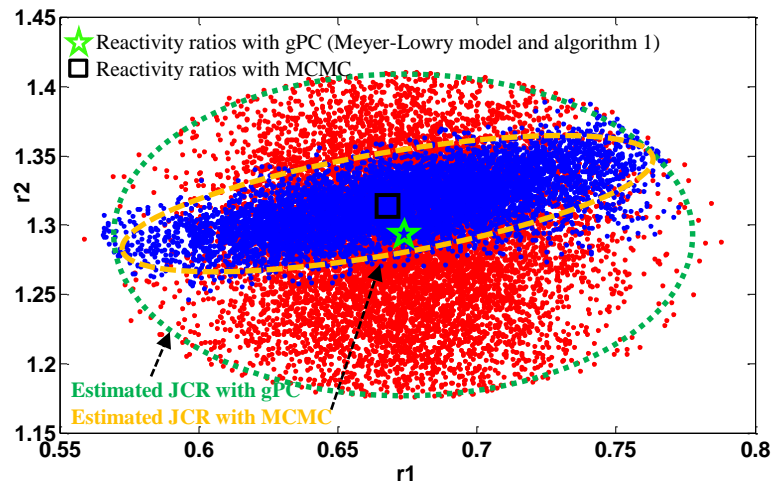


Figure B.4 Point estimates, random sample points and estimated Joint Confidence Regions (JCRs) for reactivity ratios using the Meyer-Lowry model in copolymerization of DBI/MMA at lower conversion levels

As seen in Figure B.4, the point estimates of the low conversion level with the Meyer-Lowry model are in a relatively good agreement between two aforementioned methods. The difference between the estimates is relatively bigger, as compared with the results in Fig.B.3, where the Mayo-Lewis model is used. This may arise from the fact that the nonlinearity of Equation B.13 is stronger than Equation B.12 and samples generated in MCMC simulations are less enough. Increasing the numbers of samples may improve the efficacy, however, it will make the evaluations of optimization problem more time consuming, which will be further discussed later. As compared with the referenced results (Kazemi, et al., 2011), the estimates obtained with the gPC and the MCMC in current work are slightly different. As seen in Table 1, the point estimates are $r_1 = 0.7129$ and $r_2 = 1.310$ with the error in variables model (EVM) method. The small difference arises from the fact that the gPC and the MCMC methods involve calculating the expected value while the EVM is based on finding the mode of a distribution of interest. Therefore if the posterior probability density function is not symmetric, then the two groups of method can produce different point estimates.

Using the proposed methods and the MCMC simulations, the 99% JCRs and the 95% JCRs at the low conversion level are generated and given in Fig.B.5 and Fig.B.6 respectively, where 100,000 samples are used. For the MCMC simulations, the convergence is first diagnosed using the acceptance rates and the Markov Chain time series plots. Once the convergence is confirmed, the MCMC samples can be used to generate the JCRs. For the gPC methods, samples from the random basic variables ξ are substituted into the gPC expansions of unknown parameters, thus generating corresponding parameters values (samples). The first step in generating the JCRs is to use the samples to create a three-dimensional histogram. This histogram with two parameters represents the actual posterior distribution function of the parameters. Therefore a contouring algorithm can be applied to the histogram to construct a particular confidence region.

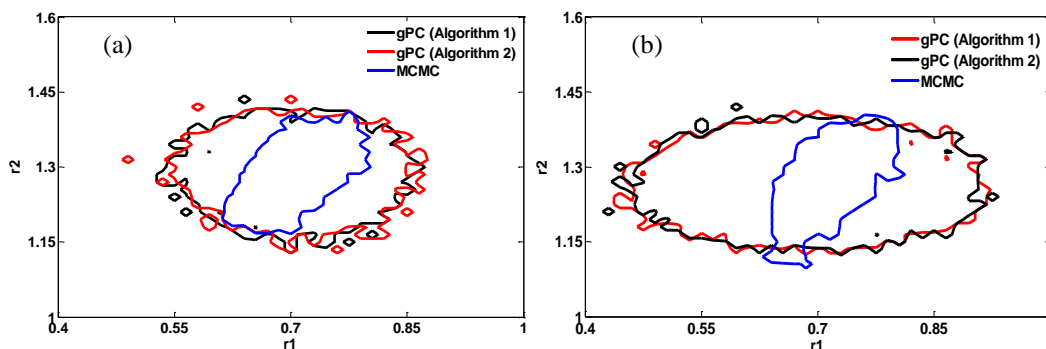


Figure B.5 99% Joint Confidence Regions (JCRs) for reactivity ratios using (a) the Mayo-Lewis model and (b) the Meyer-Lowry model in copolymerization of DBI/MMA at lower conversion levels

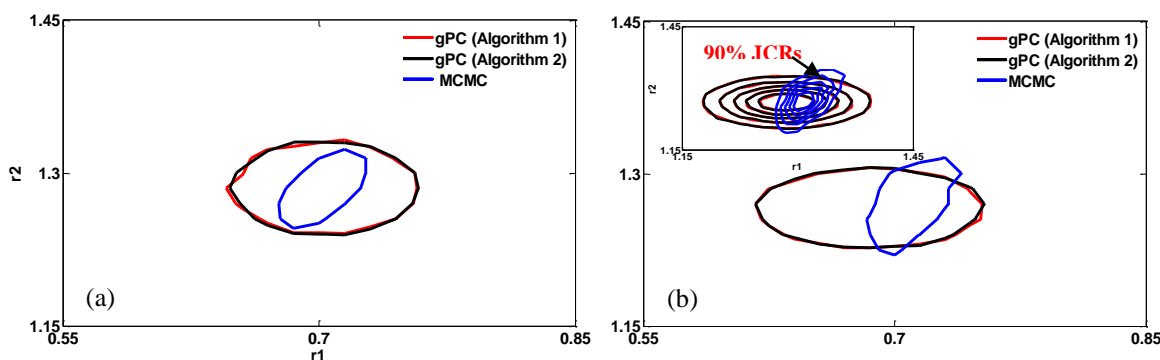


Figure B.6 95% Joint Confidence Regions (JCRs) for reactivity ratios using (a) the Mayo-Lewis model and (b) the Meyer-Lowry model in copolymerization of DBI/MMA at lower conversion levels

As can be seen in Fig.B.5 and Fig.B.6, the JCRs generated with the proposed methods have similar shape and can capture the uncertainty in the parameters. As expected, the area with the gPC models is bigger than the MCMC simulations. This is not surprising since, as discussed above, the gPC method takes into accounts most of the uncertainty information and can provide the bounds for parameter estimations. It should be noted that there are slightly difference between the JCRs generated with the gPC models and the MCMC techniques for the Meyer-Lowry model. This may arise from the fact that the highly nonlinearity of the Meyer-Lowry model and the number of experimental data is not enough, which cannot provide a completely reliable source of information needed for the variance estimation. Another possible explanation is that the normal distributed uncertainty on the stochastic parameters is assumed, thus the Hermite polynomial is used and only two terms are kept in the gPC approximation. Increasing the number of available measurements and approximation terms used in the gPC model may improve the performance. However, it is beyond the scope of the current work, since the objective is to introduce an alternative methodology for parameter estimation, especially for highly nonlinear problems with

limited measurements. Additionally, the small inset in Fig.B.6 (b) shows the JCRs at different confidence intervals. As seen, the gPC method basically covers the area generated by MCMC simulations with a 90% JCR.

B.5.4 Uncertainty Quantification of Measured Variables

Another advantage for the gPC based parameter estimation is that it can characterize the variability on the measured quantities, as introduced by the variation on the unknown parameters. For example, the mean and the variance of measured variables are the most common statistical properties. It is worth noting that these two moments of the measured variables can be exactly and simply calculated using the properties of the orthogonal polynomials, as shown in Equations B.10 and B.11.

In this case study the variability on one of the measured variable, i.e., the instantaneous copolymer composition \bar{F}_1 , is studied with the Meyer-Lowry model at high conversion levels. Different from the previous case studies, the data in Table B.2 are divided into three subgroups to further illustrate the efficiency of the proposed method with very limited measurements, since each experimental condition only 11 set of measurements are available. By using the *Algorithm 2*, three gPC models are generated with respect to three monomer mole fraction values f_{10} . Thus, three sets of parameters estimation results are obtained. Table 6 shows the gPC coefficients of the unknown reactivity ratio parameters for each of the monomer mole fraction values.

Table B.6 Optimization results from the gPC model with the high conversion data

f_{10}	gPC coefficients				Variance ($1e-5$)	
	r_{10}	r_{11}	r_{20}	r_{21}	V_{r1}	V_{r2}
0.3	0.6810	0.0019	1.2436	0.0025	0.361	0.625
0.5	0.6586	0.0022	1.2778	0.0034	0.484	1.156
0.7	0.7267	0.0016	1.2730	0.0019	0.256	0.361

For comparison, the variance defined in Equation B.11 is calculated for three sets of estimates. It can be seen that the values of the reactivity ratios obtained based on different gPC models at different conversion values are very similar. However, the variation of the parameter is relatively smaller than the results provided in Table B.4. For example, the magnitude of variance is not on the same order. The results show that at high conversion level, the choice of the modeling method has effect on the result of the point estimates. Another simplest use of the symbolic gPC model is to analytically approximate the probability density functions (PDFs) of the measured quantities. The gPC model can provide the mean and the variance of measured variables in an explicit closed form. Also, the range of the measured quantities is a function of uncertainty introduced by unknown parameters. Once the gPC model is constructed, the PDF profiles of measured quantities can be easily calculated, as shown in Fig.B.7 for three different monomer mole fraction values.

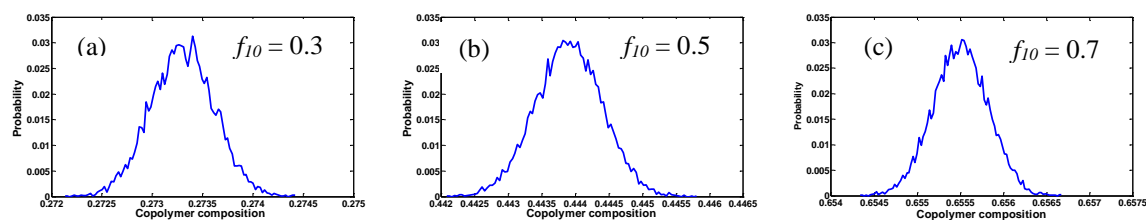


Figure B.7 Probability density functions (PDFs) of the high conversion levels with the Meyer-Lowry model

Each of the PDF profiles shown in Fig.B.7 assumes that the process is operating around one of the three monomer mole fraction values f_{10} . Besides estimating the reactivity ratio parameters, another possible application is to infer the mole fraction value using the measured copolymer composition. For a given measurement of copolymer composition, for instance, it can be compared with each of the PDF profiles in Fig.B.7. The probability of being operated around a particular monomer mole fraction value can be inferred by choosing the maximum probability. It's worth mentioning that the inverse inference provide opportunity for real-time monitoring, which may be very critical in other applications such as chemical reactors where small changes may cause, if undetected, runaway conditions.

B.5.5 Computational Efficiency

The computational time of the proposed gPC methodology is compared with the MCMC simulations, since the computational cost can be a limitation while applying techniques in the other chemical engineering problems. To better compare the time for parameter estimation, a summary of the simulation time for each model with the gPC method and the MCMC simulation is given in Table B.7. All the methods are executed on a 2.66 GHz Intel^(R) Core Duo processor.

Table B.7 The computational time required for the Mayo-Lewis model and the Meyer-Lowry Model

<i>Model</i>	<i>Method</i>	<i>Time (h)</i>
Mayo-Lewis	MCMC (10000 samples)	0.2250
Mayo-Lewis	MCMC (100000 samples)	1.2759
Mayo-Lewis	gPC Algorithm 1	0.0026
Mayo-Lewis	gPC Algorithm 2 (3 quadrature points)	0.0375
Mayo-Lewis	gPC Algorithm 2 (5 quadrature points)	0.0383
Mayo-Lewis	gPC Algorithm 2 (5 quadrature points, Gram-Schmidt)	0.0390
Meyer-Lowry	MCMC (10000 samples)	0.3072
Meyer-Lowry	MCMC (100000 samples)	9.7222
Meyer-Lowry	gPC Algorithm 1	0.0731
Meyer-Lowry	gPC Algorithm 2 (3 quadrature points)	0.3171
Meyer-Lowry	gPC Algorithm 2 (5 quadrature points)	0.3208
Meyer-Lowry	gPC Algorithm 2 (5 quadrature points, Gram-Schmidt)	0.3307

As can be seen in Table B.7, the computational time is not an issue when the MCMC simulations are applied to the weak nonlinear problems such as Mayo-Lewis model, with a careful selection of samples. However, there

is a significant increase on computational time, while the number of samples has been increased. For instance, ~ 9.7 h is required with 100,000 samples for the Meyer-Lowry model. Therefore, the MCMC technique is more sensitive to the model structure.

B.6 Conclusion

The parameter estimation methodologies in this work propose a reliable and user friendly method of estimating the reactivity ratios as well as their confidence intervals in copolymerization. The results show that the method can accurately capture the uncertainty in the Mayo-Lewis model. For the Meyer-Lowry model, it can produce desirable point estimates, although the correct shape of JCR is slightly different, as compared with the MCMC simulations. This may arise from the highly nonlinearity of Meyer-Lowry model or the number of samples used for the MCMC simulations. Most importantly, the measurement noise and uncertainty are assumed to be multiplicative in current work to make fair comparison with published work. The logarithm may reduce the nonlinearity in practice, for example, if the EVM is adopted. This kind of nonlinear transformation will affect the parameter estimation results, especially when the model uncertainty or measurement noise are additive. However, this is not an issue for the proposed method and the MCMC simulations, since the nonlinear components can be explicitly consider in the model without using any transformation. As compared with the MCMC simulations, the proposed methods perform better in terms of computation time. It may help in applying the techniques to a broader area, especially in cases where online parameter estimation is a must.

Appendix C

Quantitative Analysis of Normal and Apoptotic Cells

(Adopted from Du et al., 2016, IFAC Symposium on Dynamics and Control of Process System, including Biosystems)

C.1 Overview

Accurate and fast quantitative analysis of living cells from fluorescence microscopy images is useful for evaluations of experimental outcomes and cells culture protocols. An algorithm is developed in this work to automatically segment and discern apoptotic cells from normal cells. A coarse segmentation algorithm is proposed as a pre-filtering step that combines a range filter with a marching square method. This step provides approximate coordinates of cells' positions in a two-dimensional matrix used to store cells' image. With this information, the *active contours without edges* method is applied to identify cells' boundaries and subsequently it is possible to extract the mean value of intensity within the cellular regions, the variance of pixels' intensities in the vicinity of cells' boundaries and the lengths of the boundaries. These morphological features are then employed as inputs to a support vector machine (SVM) classifier that is trained to distinguish apoptotic from normal viable states of cells. The algorithm is shown to be efficient in terms of computational time, quantitative analysis and differentiation accuracy, as compared to the use of the active contours method without the proposed coarse segmentation step.

C.2 Introduction

Fluorescence microscopy is a well-developed tool to study in vitro cells' behaviour. However, microscopy experiments can generate a great amount of cells' images with varying image qualities (Waters, 2009). The manual quantification and analysis of these data is time consuming. Hence, accurate and automatic analysis of cells images such as Chinese Hamster Ovary (CHO) cells can be very useful.

Mammalian cells are prone to apoptosis (programmed cell death), which is a key metabolic event that restricts the growth of cells and decreases the productivity in a bioreactor (Rulter, et al., 2014). The accurate detection of apoptotic cells can help identifying the critical factors that trigger apoptosis. This knowledge may be used for delaying apoptosis and potentially increase the productivity (Taatjes, et al., 2008).

Morphological changes in cells are highly indicative of the occurrence of apoptosis (Henry, et al., 2013). For example, shrinkage and blebbing of the cytoplasmic membrane are found to be significant characteristics of apoptotic cells (see Fig.C.1), which cause cells to lose normal, smooth and circular shapes. Blebbing during apoptosis is generally associated to swell of the cell membrane into spherical bubbles. Hence, microscopic observation of morphological changes can be used to discern normal from apoptotic cells. However, cells may exhibit highly variable values of these morphological measures due to the dynamic nature of apoptosis.

This work presents a new image processing and quantitative analysis method that can automatically differentiate apoptotic from normal cells, while maintaining the computational time at a reasonable level. The proposed method involves three consecutive steps: (i) a *coarse segmentation* that can be used to identify the number of cells in a given image of cells; (ii) a *fine segmentation* step to detect the boundaries of cells and to identify particular morphological features related to these boundaries; and (iii) a support vector machine (SVM) based classification model that uses the morphological features identified in the *fine segmentation* step (step ii) to distinguish apoptotic cells from normal cells.

Our contributions in this current work are summarized as follows: (i) a computationally efficient *coarse segmentation* algorithm that combines a range filter and a marching square method to approximate cells' locations in an image; (ii) an automated differentiation algorithm to discern apoptotic from normal cells using three morphological features that can be extracted from the results of the *fine segmentation* algorithm. The method in this work can be easily extended to other studies for real-time monitoring of cells' cultures and for high throughput screening experiments upon appropriate tuning.

This appendix is organized as follows. Section C.3 reviews the background on fluorescence imaging and the challenges in analyzing cell morphology. The method developed is presented in Section C.4. Results and discussion are presented in Section C.5 followed by conclusions in Section C.6.

C.3 Fluorescence Imaging

Fluorescence microscopy has been used to differentiate and quantify apoptotic versus normal cells as well as to determine the viability of cells. This analysis involves two types of fluorescent dyes, i.e., acridine orange (AO) and ethidium bromide (EB), which are mixed in a fixed ratio within the cell suspension which is then analysed by fluorescence microscopy. The AO can penetrate viable and nonviable cells and make cells to appear green while the EB can only diffuse into nonviable cells and make them to appear orange (or red). Fig.C.1 shows a typical fluorescence photomicrograph of CHO cells stained with AO and EB. This image is stored as a multi-dimensional matrix, which elements are the intensities of pixels. The first task is to detect the edges of the cells based on pixels' intensities for a given image. Subsequently, it is desired to distinguish between apoptotic to normal cells.

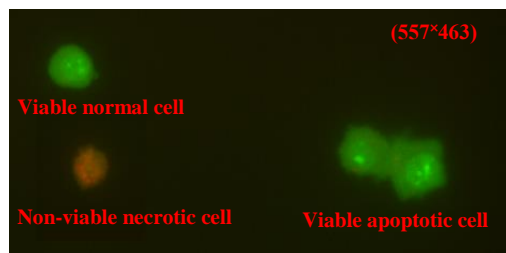


Figure C.1 Fluorescence photomicrograph of CHO cells

As observed in Fig.C.1, apoptotic cells usually exhibit irregular shapes and blurry boundaries, as compared to normal cells. Additionally, the appearance and size of cells belonging to the same class, i.e., normal versus apoptotic, can vary significantly (see Fig.C.2). This makes the automatic differentiation of cells status in an image very challenging.

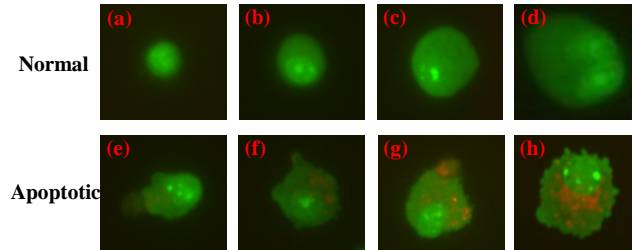


Figure C.2 Examples of cells in different states

In Fig.C.2, the cells have different shapes and boundaries. As seen in the first row of cells' images, normal cells can be characterized by rounded and smooth boundaries. However, the size of normal cells is very different from one another. Apoptotic cells, shown in the second row of images, have very irregular shapes and boundaries. Therefore, a departure from a smooth boundary (blebbing) is a key morphological indicator to discern apoptotic from normal cells. This work builds on the hypothesis that a combination of different indicators such as the average of pixels' intensities within the cell boundary, a measure of the variability of the pixels' intensities around the cells' boundary and the length of this boundary can be used for differentiating normal cells from apoptotic cells.

C.4 Image Processing Methodology

C.4.1 Image Segmentation

For images obtained with microscopy, the pixels' intensities within the cells' boundaries sometimes are very similar to the intensities measured within the background surrounding the cells. Thus, using strictly an intensity threshold to segment the cells from the background is not effective. Instead, edge-based methods such as the *active contour algorithm* ignore edges altogether and can handle segmentation more accurately. The central idea behind the active contour algorithms is to iteratively evolve a curve to segment objects from the background which upon convergence provides the boundary. A brief description is given for background.

Assuming a curve C , subjects to the constraints of a given grayscale image U_0 in an open bounded domain Ω of R^2 , which approximates the boundary φ , i.e., $C \approx \varphi$, $C \subset \Omega$, and $\varphi \subset \Omega$. To evolve C , the *active contours without edges* method (Chan & Vese, 2001) seeks a best approximation of C by minimizing an energy function defined as:

$$\begin{aligned}
\arg \min_{m_1, m_2, C} & \mu_1 \cdot \text{Length}(C) + \mu_2 \cdot \text{Area}(\text{inside}(C)) \\
& + \lambda_1 \int_{\text{inside}(C)} |U_0(x, y) - m_1(C)|^2 dx dy \\
& + \lambda_2 \int_{\text{outside}(C)} |U_0(x, y) - m_2(C)|^2 dx dy
\end{aligned} \tag{C.1}$$

where μ_1, μ_2, λ_1 and λ_2 are non-negative tuning parameters, m_1 and m_2 depend on the evolving curve C and are the mean values of intensities inside C and outside C , respectively. The coordinates, defining the domain Ω , are defined by the x -axis and y -axis. Intensities are available at each point in x and y coordinates. In Eq. C.1, the first component controls the regularity of C by penalizing its length. The second term penalizes the enclosed area to control the size of the cellular areas. The last two terms penalize the discrepancy between the active curve C and the given image.

The optimization problem Eq. C.1 can be formulated and solved by a level set method (Osher & Sethian, 1988), where the problem is rewritten in terms of an unknown level set function Z . Instead of manipulating C , the minimization of Eq. C.1 is formulated by an equation that progressively evolves the geometric locus of the zero value of the level set function Z . Assuming that the unknown level set function Z is smooth, the *active contours without edges* optimization Eq. C.1 is written in terms of the level set function Z as:

$$\begin{aligned}
\arg \min_{m_1, m_2, C} & \mu_1 \cdot \int_{\Omega} \delta_{\varepsilon}(Z(x, y)) |\nabla Z(x, y)|^2 dx dy \\
& + \mu_2 \cdot \int_{\Omega} H_{\varepsilon}(Z(x, y)) dx dy \\
& + \lambda_1 \int_{\Omega} |U_0(x, y) - m_1|^2 H_{\varepsilon}(Z(x, y)) dx dy \\
& + \lambda_2 \int_{\Omega} |U_0(x, y) - m_2|^2 (1 - H_{\varepsilon}(Z(x, y))) dx dy
\end{aligned} \tag{C.2}$$

where H_{ε} is the Heaviside function with respect to the level set function Z , and δ_{ε} denotes a regularized Dirac δ -function that for the purpose of minimization with respect to Z , H_{ε} is defined as:

$$H_{\varepsilon} = \frac{1}{2} \left(1 + \frac{2}{\pi} \arctan\left(\frac{Z(x, y)}{\varepsilon}\right) \right) \tag{C.3}$$

The minimization of (2) can be solved by updating m_1, m_2 and Z alternately as follows:

- i- For any fixed level set function Z , the values of m_1 and m_2 are the region averages approximated by:

$$m_1 = \frac{\int_{\Omega} U_0(x, y) H_{\varepsilon}(Z(x, y)) dx dy}{\int_{\Omega} H_{\varepsilon}(Z(x, y)) dx dy} \tag{C.4}$$

$$m_2 = \frac{\int_{\Omega} U_0(x, y) (1 - H_{\varepsilon}(Z(x, y))) dx dy}{\int_{\Omega} (1 - H_{\varepsilon}(Z(x, y))) dx dy} \tag{C.5}$$

- ii- And for fixed m_1 and m_2 values, a gradient descent equation is formulated for Z with respect to an (artificial iteration) time t as:

$$\frac{\partial Z}{\partial t} = \delta_\varepsilon(Z) [\mu_1 \cdot \text{div} \left(\frac{\nabla Z}{|\nabla Z|} \right) - \mu_2 - \lambda_1 (U_0 - m_1(Z))^2 + \lambda_2 (U_0 - m_2(Z))^2] \quad (\text{C.6})$$

$$Z(0, x, y) = Z_0(x, y) \quad \text{in } \Omega \quad (\text{C.7})$$

$$\frac{\delta_\varepsilon(Z)}{|\nabla Z|} \frac{\partial Z}{\partial \bar{e}} = 0 \quad \text{on } \partial\Omega = \varphi \quad (\text{C.8})$$

where \bar{e} is the outward normal to the boundary φ . The function r (Li, et al., 2008) is used to ensure the stability of the algorithm and is defined as follows:

$$r(Z_t) = \int_{\Omega} \frac{1}{2} (|\nabla Z_t(x, y)| - 1)^2 dx dy \quad (\text{C-9})$$

where the subscript t denotes that the regularization term r is evaluated and updated at each (artificial) time instant t .

C.4.2 Coarse Segmentation using a combination of a Range Filter and a Marching Square Algorithm

The level set method shown in Section C.4.1 is iterative in nature and therefore is generally very slow for the purpose of high throughput screening. Computations are especially slow for microscopy images, since the cells are usually sparsely distributed in fluorescence microscopy images. To tackle this problem efficiently, a *coarse segmentation* procedure is developed by combining a range filter with a marching square method, which provides an initial approximation of the coordinates of cells' positions in a xy -plane and the number of cells. This method will be referred heretofore as the *range filter with marching square (RFMS)* method.

The idea is to apply initially the *RFMS* algorithm to each image to find the approximate position of each cell within an image containing many cells before applying the level set method reviewed in the previous section. Using *RFMS*, it is possible to define sub-images around each identified cells and then apply the level set algorithm to each of these sub-images. This “*windowing*” process around each cell is intended to lead to overall reduction in processing time and increased accuracy, as compared to the application of the level set method to the original large images containing a large number of cells without the *RFMS* step.

Assume that a given image U_0 with pixels' dimensions of $p_1 \times p_2$ is scanned by a range filter with pixels' dimensions of $q_1 \times q_2$. This gives a new range map matrix U with the same pixels' dimension as U_0 , in which each pixel contains the range value of the $q_1 \times q_2$ neighbourhood around the corresponding pixel in U_0 . The range value for each pixel inside U is calculated as the difference between the maximum and minimum values within the $q_1 \times q_2$ neighbourhood of each pixel in the image. A schematic description of the range filtering operation is shown in Fig.C.3.

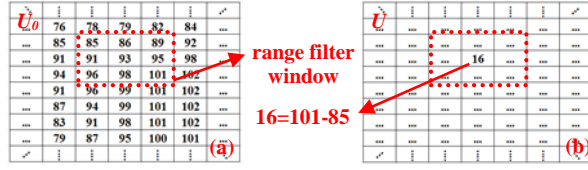


Figure C.3 Schematic description of the range filter operation for generating U from U_0

As seen in Fig.C.3 (a), U_0 is scanned by a range filter with pixels' dimensions of 3×3 . The difference of intensities between the maximum and minimum values captured by the range filter around a particular pixel is given in Fig.C.3 (b). Once the range filtered image U is calculated, a first approximation of the coordinates of cells in U_0 can be identified by comparing the intensities in the range filtered image U to a threshold value ζ .

Using this threshold a matrix U_B is generated which elements are binary intensities as follows:

$$U_B = \begin{cases} 1 & \text{if } U(i, j) \geq \zeta \\ 0 & \text{if } U(i, j) < \zeta \end{cases} \quad (\text{C.10})$$

where i and j represent the coordinates of pixels in the xy -plane, $1 \leq i \leq p_1$ and $1 \leq j \leq p_2$.

Subsequently a marching square algorithm is applied to U_B with two objectives: (i) approximate the bounds of cellular regions, (ii) count the number of cells. In principle the main focus of the paper is to distinguish apoptotic from normal cells but it will be shown in the Results' section that this algorithm is also useful for quickly count cells in the image.

The steps used to approximate contours with the marching square algorithm are schematically shown in Fig.C.4. Each 2×2 block of pixels in U_B can be used to construct a contouring grid. The dash line in Fig.C.4 (a) denotes one contouring grid element generated with the first 2×2 block of pixels. Fig.C.4 (b) shows a contouring grid in U_B (dash lines) made of individual contouring grid pieces, which can be used to find a line that all its points have the same intensity value. In our case, since the application of the marching square method follows the thresholding step in (10), this value is 1 (stars in Fig.C.4 (b)). Since each contouring grid element has 4 corners, there are exactly 2^4 possible patterns describing portions of the cell contour crossing within each element as shown in Fig.C.4 (c). By finding a match between the observed lines within each grid element with one of the possible patterns in Fig.C.4 (c), the contour (dot line) in Fig.C.4 (b) can be formed.

Once the contours corresponding to each cell in an image are obtained, the marching square method can provide an approximate location of the cell and the number of cells in a given image can be found by checking the number of contours. Note that the marching square method only provides a *coarse approximation* of cells' boundaries, since the contour lines are assumed to be straight between edges of a grid element as seen in Fig.C.4 (c). It will be shown in the Results' section that the contours resulting from this method are inaccurate for differentiating normal from apoptotic cells. Thus, the *RFMS* algorithm is only used as a *pre-filtering* tool to locate the approximate cells' positions but it must be complemented by the *fine segmentation* method shown in Section

C.4.1 to do the final differentiation. It will also be shown in the Results' section that the *RFMS* algorithm is very effective and more accurate for the counting of cells, as compared to the level set method applied alone without the *RFMS* pre-filtering step.

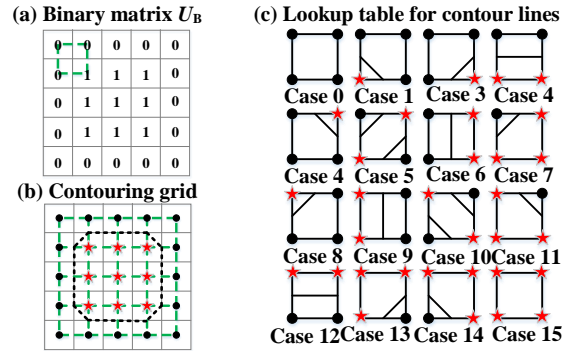


Figure C.4 Visual interpretation for generating contours

Based on the coordinates of contours (the pixels' location in xy -axis) identified by the *RFMS* method, the image is divided into sub-images each containing one cell. Then, the segmentation method explained in Section C.4.1 is applied to each of the sub-images to perform a finer identification of cells' boundaries.

C.4.3 Feature Extraction

The differentiation of cells into apoptotic or normal is based on a set of morphological features calculated from the images. Three features are used: (i) the mean value of pixels' intensities within the cellular regions; (ii) the variance of pixels' intensities in the vicinity of the boundary and (iii) a measure of the size of the boundary. The choice of these features is justified by a priori knowledge of the phenomena. Apoptotic cells exhibit blebbing due to swell of the cell membrane. This swelling process results in variable fluorescence intensities in the neighbourhood of the cell contour and generally longer contours of apoptotic cells as compared to normal ones thus justifying the properties used here for differentiating cells.

The mean value of pixels intensities of cellular regions is calculated with the level set function as defined in Eq. C.4, based on the segmentation results obtained in Section C.4.1. The calculation of the variance proceeds as per the following steps. (i) Perform the level set algorithm to segment cells from the background. The boundary shown as a solid line in Fig.C.5 (a) is obtained by connecting the points with a level set function value of $Z \approx 0$ in Eq. C.10. (ii) Set a value N_e that is the number of pixels in the immediate neighbourhood of each point on the boundary to be used for the calculation of the variance (see Fig.C.5 (b)). (iii) Connect the points neighbouring the boundary defined by N_e to build a fuzzy region around the boundary given by the dash lines in Fig.C.5 (b). (iv) Calculate the variance of all pixels intensities inside this fuzzy region. (v) Calculate the length of the boundary as the total number of pixels defining the boundary corresponding to pixels with level set function values of zero as calculated by the algorithm in Section C.4.1.

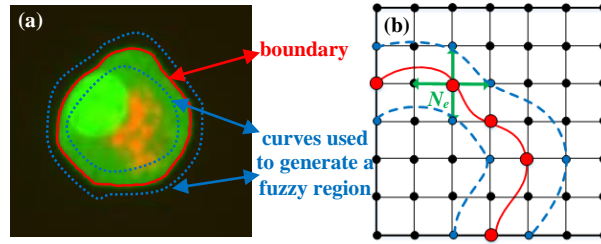


Figure C.5 Sketch of the calculation of the second feature

C.4.4 SVM based Classification

Based on the three features proposed above, a support vector machine (SVM) (Burgers, 1998) classification model is developed to distinguish apoptotic cells from normal cells. SVM was arbitrarily chosen as one possible regression technique among many possible ones such as PLS, etc. A set of training images are selected, and each of the images used for model training is first processed with the *RFMS* method to approximate the number of cells and determine the sub-images, each containing a cell. Each observed cell is then characterized as either normal or apoptotic cells based on consensus among five different experimentalists and based on the percentage of apoptotic cells at the time the image was taken as determined by available independent flow cytometry data (Meshram, et al., 2011). The morphological features of cells are computed using the method described in Section C.4.3.

From the images used for model training, the parameters of the SVM model can be optimized with the Matlab[®] statistics and machine learning toolbox. The trained SVM model is applied to new images that were not used for model training to classify new images into apoptotic or normal. These testing images are also pre-processed with the *RFMS* method.

The methodology in this work can be summarized as follows: (i) Calibrate the *RFMS* based *coarse segmentation* to estimate optimal parameters, i.e., pixels' dimensions ($q_1 \times q_2$) of the range filter and an intensity threshold ζ in Eq. C.10. (ii) Generate contouring grids for each training image with marching square method. (iii) Estimate the number of cells and the coordinates of cells. (iv) Construct sub-images with the information obtained in step iii. (v) Characterize manually each of the sub-images as either normal or apoptotic, using experimentalists' consensus and cytometry. (vi) Perform image segmentation using the level set method and calculate the three morphological features for each image classified in step v. (vii) Use the features obtained in step vi as inputs to an SVM classifier that is trained to discern apoptotic from normal cells.

C.5 Results and Discussion

C.5.1 Coarse Segmentation Results

One way to assess the advantages of the *RFMS* algorithm as a pre-filtering tool is by testing its ability to detect the number of cells within an image. To that purpose, the *RFMS* is applied to a dataset containing 187 cells in 46 images to study the accuracy of the algorithm in terms of cell counts. The range filter with pixels' dimensions of 3×3 is used to calculate the range map matrix U . The intensity threshold ζ , used to generate the binary matrix U_B , is chosen as 5. Based on U_B , the marching square method is utilized to generate contours, which provide the coordinates and estimated boundaries of cells for a given image U_0 . Fig.C.6 shows the boundaries for two images with different sizes, shapes and physiological states of cells.

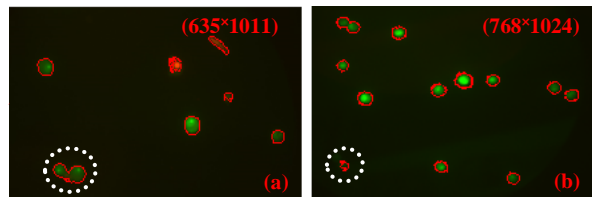


Figure C.6 Summary of quantitative analysis

For comparison purposes, the quantitative analysis of cells is also conducted with the level set method described in Section C.4.1 without applying the *RFMS* step. Following the finding in previous studies (Chan & Vese, 2001; Getreuer, 2012), the parameters for the level set algorithm are chosen as: $\mu_1 = \mu_2 = \lambda_1 = \lambda_2 = 1$, and the time-step Δt is 0.1. Then, the accuracy in cell counting by the *RFMS* with the level set method without the *RFMS* step is compared.

When the *RFMS* is used, 181 cells are counted correctly and 6 cells are missed. For example, the cell in the circle in Fig.C.6 (a) was counted as one cell instead of two due to overlapping between two boundaries of neighbouring cells. By contrast, when the level set method without *RFMS* is applied to the images only 175 cells are counted correctly and 12 cells are undetected. For example, the region of cells in the circle in Fig.C.6 (b) was detected as one cell using the *RFMS* algorithm. However, it is misclassified as background with the level set method (see Fig.C.7 (a)). The explanation is that for some cells the importance of the first two terms in Eq. C.6 which penalize the enclosed area of cells and the regularity of boundaries may be compromised in the level set method as compared to the other two terms in the cost Eq. C.6.

Further studies are conducted to investigate the cell counting accuracy when the *RFMS* is combined with the level set method. For the region in the circle in Fig.C.6 (b), the level set method is applied to the sub-image generated with the *RFMS* method. The sub-image is created by expanding it from both sides by 20 pixels on both x and y directions, based on the coordinates obtained with *RFMS*. The result is shown in Fig.7. As seen in Fig.C.7 (a), the cells in the circle cannot be detected, if the level set method is applied without the *RFMS* to process the

original image. However, when the level set method is applied to the sub-image generated with the *RFMS* (see Fig.C.7 (b)), the level set method can successfully segment cells from the background. This confirms that the combination of the *RFMS* and level set method is very advantageous, as compared to the level set method applied on its own without the *RFMS* pre-filtering step.

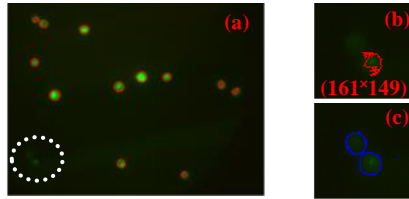


Figure C.7 Comparison of segmentation methods

C.5.2 Comparison of Image Segmentation Results

Studies are conducted to investigate the accuracy in terms of segmentation. Fig.C.8 shows the segmentation results for a few images with the level set method and the *RFMS* method. The analysed cells in Fig.C.8 have different sizes, shapes and blurry boundaries. The blue lines represent the boundary that is calculated with the level set function, while the red lines are the results approximated with the *RFMS* method in this work. As shown in the figure, both methods can successfully segment cells from the background. However, the boundary generated with the *RFMS* is less smooth, as compared to the level set method without the *RFMS*. For example, the first two images of cells (Fig.C.8 (a) and (b)) have regular shapes and smooth boundaries, but the boundaries calculated with the *RFMS* are fuzzy. The differentiation of cells in this work is built upon the hypothesis that low variability is associated with normal cells while higher variability is indicative of apoptotic cells. Thus, the *RFMS* must be combined with the level set method since the *RFMS* alone generally results in very fuzzy boundaries and provides inaccurate differentiation as shown in the Results' section.

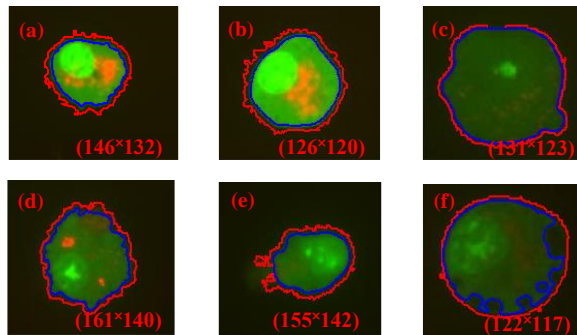


Figure C.8 Summary of segmentation results

C.5.3 Comparison of Computational Time Required for Segmentation

The computational time is studied for cells' images that have different sizes with respect to pixels dimensions. Three cases are investigated: $\sim 800 \times 1100$, $\sim 550 \times 450$ and $\sim 150 \times 150$ pixels of cells images. Using the dataset containing a total of 187 cells in 46 images, the average computational times required to only count cells within the images by the level set on its own or the *RFMS* on its own are shown in Table C.1.

Table C.1 Summary of computational time

Method	Time (s)		
	800×1100	550×450	150×150
Level set	601.7	20.87	2.80
<i>RFMS</i>	1.18	0.87	0.31

As seen in Table C.1, if the sole objective is to count cells, the computational time of the *RFMS* algorithm is significantly lower than the level set based segmentation. Also, as shown in Table C.1 the computational times are highly related to the pixels' dimensions of cells' images.

The computational time is further investigated by combining the *RFMS* with the level set method. Note that the solution of the level set algorithm requires initial conditions (in Eq. C.6). We found that an additional benefit of the *RFMS* pre-filtering algorithm is that it can provide a good initial guess for the solution of the level set algorithm.

We compared the computational cost of using an initial guess from the *RFMS* algorithm versus using a random initial guess for an image containing 3 cells. The computational time is evaluated by the number of iterations that are required to progressively evolve the boundary of cells. It was found that approximately ~ 5 iterations are needed for the level set method to converge to the boundary when the results obtained with *RFMS* are chosen as the initial values whereas approximately ~ 10 iterations are required for the randomly chosen initial values. This observation confirms that the ability of the *RFMS* to provide an initial guess for the level set algorithm is an additional benefit of using the *RFMS* as a pre-filtering step before applying the level set method.

C.5.4 Feature Extraction

The proposed method combines the *coarse segmentation* step achieved with the *RFMS* with the *fine segmentation* achieved with the level set method for the images from dataset. For the training of SVM model, the level set based segmentation is applied to a training set with 100 samples of cells obtained with the *RFMS* based *coarse segmentation*. In this training set, 50 images are normal cells and 50 images are apoptotic cells. A feature vector is calculated for each of the training images composed of the 3 proposed morphological features, i.e., the mean value of pixels intensities of cells, the variance of pixels' intensities in the vicinity of the boundary and the complexity of the boundary. A few training feature vectors are shown in Table C.2, where the variance is normalized with respect to the mean value of the cellular regions.

Table C.2. Examples of feature vectors

<i>States</i>	<i>Variance</i>	<i>Complexity</i>	<i>Mean</i>
<i>Apoptotic</i>	9.84	87	74.87
<i>Apoptotic</i>	9.60	91	81.45
<i>Apoptotic</i>	6.71	115	64.09
\vdots	\vdots	\vdots	\vdots
<i>Normal</i>	5.94	79	74.23
<i>Normal</i>	8.77	77	84.46
<i>Normal</i>	3.61	103	91.24

C.5.5 Differentiation Results using a SVM Classifier

Using the trained SVM classifier model, 60 images of cells that were not used for the model training are used to test the classification rate, which contain 30 samples of normal cells and 30 samples of apoptotic cells. To evaluate the efficacy of the classification between normal cells and apoptotic cells, a *differentiation rate* is defined as:

$$r_{rate} = d_i / D_T \quad (C.11)$$

where d_i denotes the number of testing images that have been correctly identified and D_T is the total number of images used for the experiments. To test whether it is necessary to complement the *RFMS* method with the level set algorithm, two scenarios are studied to discern apoptotic from normal cells, i.e., the combination of the *RFMS* with level set and the *RFMS* without the level set method. Table C.3 shows the results of *differentiation rate* r_{rate} .

Table C.3. Differentiation rates r_{rate}

<i>Methods</i>	<i>Normal</i>	<i>Apoptotic</i>	<i>Average</i>
<i>Combination</i>	0.96	0.93	0.945
<i>RFMS</i>	0.93	0.83	0.880

From Table C.3, it can be seen that the combination of the *coarse segmentation* achieved with the *RFMS* with the *fine segmentation* provides high accuracy. The average of r_{rate} is $\sim 94.5\%$, and 1 normal cell's image is misclassified and 2 testing images of apoptotic cells are misidentified. However, 7 of the testing samples of cells (2 normal and 5 apoptotic cells' images) are misclassified with the *RFMS* alone that provides a differentiation rate of $\sim 88\%$.

Hence, it is evident that we must combine the *RFMS* with the level set algorithm, since the standalone application of the level set method without the *RFMS* cannot provide accurate counting of cells and is time demanding while the use of *RFMS* without complementation with the level set method results in a differentiation rate that is $\sim 6\%$ lower than the algorithm combining the *RFMS* and level set methods.

C.6 Conclusion

In this work, a methodology has been developed for high throughput screening studies to distinguish apoptotic from normal CHO cells. A simple *coarse segmentation* algorithm, which combines a range filter and a marching square method (*RFMS*), is used as a *pre-filtering* step to provide the approximate positions of cells within each image. Using the information obtained from the *RFMS*, the level set method is used to achieve the *finer segmentation* of cells from the background. Based on these segmentation results, three morphological features are computed and used as inputs to train a support vector machine (SVM) classifier, which can accurately classify cells into normal versus apoptotic. The developed algorithm that combines the *RFMS* with the level set method is shown to be more accurate and significantly faster than the standalone application of the level set method in terms of cell counting or the standalone use of the *RFMS* in terms of differentiation of apoptotic and normal cells.

Bibliography

- Acton, S., Yang, C., Hossack, J., & Wamhoff, B. (2009). Poisson inverse gradient approach to vascular myocyte detection and segmentation. *Proceedings of the IEEE International Symposium on Biomedical Imaging*, (pp. 1298-1301). Boston, MA.
- Adams, R., & Bischof, L. (1994). Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 641-647.
- Andrieu, C., Freitas, N., Doucet, A., & Jordan, M. (2003). An introduction to MCMC for machine learning. *Machine learning*, 50, 5-43.
- Arulampalam, M. S., Maskell, S., Gordon, N., & Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2), 174-188.
- Aurum, A., & Wohlin, C. (2003). The fundamental nature of requirements engineering activities as a decision making process. *Information and Software Technology*, 45(14), 945-954.
- Bakshi, B. R. (1998). Multiscale PCA with application to multivariate statistical process monitoring. *AIChE Journal*, 44, 1596-1610.
- Bates, D. M., & Watts, D. M. (1988). *Nonlinear regression: iterative estimation and linear approximations, in nonlinear regression analysis and its application*. Hoboken, NJ, USA: John Wiley & Sons.
- Blanke, M., Kinnaert, M., & Lunze, J. S. (2006). *Diagnosis and fault tolerant control* (2nd ed.). Berlin, Germany: Springer.
- Burgers, C. J. (1998). A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2, 121-167.
- Carpenter, A. E., Jones, T. R., Lamprecht, M. R., Clarke, C., Kang, I. H., Friman, O., et al. (2006). CellProfiler: image analysis software for identifying and quantifying cell phenotypes. *Genome Biology*, 7(10), R100.
- Cha, S.-H., & Srihari, S. N. (2002). On measuring the distance between histograms. *Pattern Recognition*, 35, 1355-1370.
- Chan, T. F., & Shen, J. (2005). *Image processing and analysis: variational, PDE, wavelet, and stochastic methods*. Philadelphia, PA: The Society for Industrial and Applied Mathematics.
- Chan, T. F., & Vese, L. A. (2001). Active contours without edges. *IEEE Transactions on Image Processing*, 10(2), 266-277.

- Chen, J., Bandoni, J., & Romagnoli, J. A. (1996). Robust PCA and normal region in multivariate statistical process monitoring. *AIChE Journal*, 42, 3563-3566.
- Chen, M., Chen, S., & Guan, Q. (2009). Hybrid contour model for segmentation of cell nucleolus and membranes. *International Conference on Biomedical Engineering and Informatics*, (pp. 1-5). Tianjin.
- Chen-Charpentier, B., & Stanescu, D. (2014). Parameter estimation using polynomial chaos and maximum likelihood. *International Journal of Computer Mathematics*, 91(2), 336-346.
- Chetouania, Y., Mouhaba, N., Cosmaoa, J. M., & Estela, L. (2002). Application of extended kalman filtering to chemical reactor fault detection. *Chemical Engineering Communications*, 189(9), 1222-1241.
- Chiang, L., Russell, E. L., & Braatz, R. D. (2008). *Fault Detection and Diagnosis in Industrial Systems* (1st ed.). London: Springer-Verlag.
- Chilin, D., Liu, J., Pena, D. M., Christofides, P. D., & DavisJames, F. (2010). Detection, isolation and handling of actuator faults in distributed model predictive control systems. *Journal of Process Control*, 20, 1056-1075.
- Dalbey, K., Patra, A. K., Pitman, E. B., Bursik, M. I., Sheridan, M., & F. (2008). Input uncertainty propagation methods and hazard mapping of geophysical mass flows. *Journal of Geophysical Research*, 113(B5), 1-16.
- Daum, F. (2005). Nonlinear filters: Beyond the Kalman filter. *IEEE A&E System Magazine*, 20(8), 57-69.
- Davoodi, M., Golabi, A., Talebi, H. A., & Momeni, H. (2013). Simultaneous fault detection and control design for switched linear systems based on dynamic observer. *Optimal Control Applications and Methods*, 34(1), 35-52.
- Debeir, O., Van Ham, P., Kiss, R., & Decaestecker, C. (2005). Tracking of migrating cells under phase contrast video microscopy with combined mean shift processes. *IEEE Transactions on Medical Imaging*, 24(6), 697-711.
- Debuschere, B. J., Najm, H. N., Pebay, P. P., Knio, O. M., Ghanem, R. G., & Matre, O. P. (2004). Numerical challenges in the use of polynomial chaos representations for stochastic processes. *SIAM Journal on Scientific Computing*, 26(2), 698-719.
- Du, D., Yang, H., Ednie, A. R., & Bennett, E. S. (2015). Statistical metamodeling and sequential design of computer experiments to model Glyco-altered gating of sodium channels in cardiac myocytes. *IEEE Journal of Biomedical and Health Informatics*, in press.

- Du, Y., Budman, H., & Duever, T. (2014). Integration of fault diagnosis and control by finding a trade-off between the observability of stochastic fault and economics. *The 19th World Congress of the International Federation of Automatic Control*. Cape Town, South Africa: The 19th World Congress of the IFAC.
- Du, Y., Duever, T. A., & Budman, H. (2015). Fault detection and diagnosis with parametric uncertainty using generalized polynomial chaos. *Computers and Chemical Engineering*, 76(8), 63-75.
- Emmanuel, B., Sandu, A., & Sandu, C. (2007). *A polynomial chaos based Bayesian approach for estimating uncertain parameters of mechanical systems – part I: theoretical approach*. Blacksburg, VA: Technical Report, Computer Science, Virginia Tech.
- Eriksson, D., Frisk, E., & Krysander, M. (2013). A method for quantitative fault diagnosability analysis of stochastic linear descriptor models. *Automatica*, 49, 1591-1600.
- Eterno, J., Weiss, J., Looze, D. P., & Willsky, A. S. (1985). Design issues for fault tolerant restructurable aircraft control. In proceedings of the 24th IEEE conference on decision and control.
- Foo, J., Yosibash, Z., & Karniadakis, G. E. (2007). Stochastic simulation of riser-sections with uncertain measured pressure loads and/or uncertain material properties. *Computer Methods in Applied Mechanics and Engineering*, 196, 4250-4271.
- Fouskakis, D., & Draper, D. (2002). Stochastic Optimization: a Review. *International Statistical Review*, 70(3), 315-349.
- Frank, P. M. (1990). Fault diagnosis in dynamic systems using analytical and knowledge based redundancy. *Automatica*, 26, 459-474.
- Gerlter, J. (1998). *Fault detection and diagnosis in engineering systems*. NJ, USA: Taylor & Francis.
- Gerritsma, M., Van der Steen, J.-B., Vos, P., & Karniadakis, G. (2010). Time dependent generalized polynomial chaos. *Journal of Computational Physics*, 8333-8363.
- Gerstner, T., & Griebel, M. (1998). Numerical integration with sparse grids. *Numerical Algorithms*, 18, 209-232.
- Gertler, J., & Cao, J. (2004). PCA-based fault diagnosis in the presence of control of control and dynamics. *AIChE Journal*, 50(2), 1-15.
- Getreuer, P. (2012). Chan-Vese segmentation. *Image processing online*, 2, 214-224.
- Geyer, C. J. (1992). Practical markov chain monte carlo. *Statistical science*, 7(4), 473-511.

- Ghanem, R., & Spanos, P. (1991). *Stochastic finite elements: A spectral approach* (2nd ed.). NY: Springer-Verlag, Berlin.
- Ghantasala, S., & El-Farra, N. H. (2009). Robust actuator fault isolation and management in constrained uncertain parabolic PDE systems. *Automatica*, 45, 2368-2373.
- Gonzalez, R., & Woods, R. (2009). *Digital image processing using Matlab*. Upper Saddle River, New Jersey: Prentice Hall.
- Haghani, A., Jeansch, T., & Ding, S. X. (2014). Quality related fault detection in industrial multimode dynamic processes. *IEEE Transactions on Industrial Electronics*, 61(11), 6446 - 6453.
- Hansen, L. k., & Salamon, P. (1997). Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 993-1001.
- Harrison, R. L. (2010). Introduction to Monte Carlo simulation. *AIP conference proceedings*. Bratos;ava.
- Henry, C. M., Hollville, E., & Martin, S. J. (2013). Measuring apoptosis by microscopy and flow cytometry. *Methods*, 61(2), 90-97.
- Huerta, S., Goulet, E. J., Huerta-Yepey, S., & Livingston, E. H. (2007). Screening and detection of apoptosis. *Journal of Surgical Research*, 143-156.
- Isermann, R. (2005). Model based fault detection and diagnosis - status and applications. *Annual reviews in control*, 29, 71-85.
- Isermann, R. (2006). *Fault diagnosis systems: An introduction from fault detection to fault tolerance*. Berlin, Germany: Springer.
- Jacobson, C. A., & Nett, C. (1991). An integrated approach to controls and diagnosis using the four parameter controller. *IEEE control systems*, 11(6), 22-29.
- Jones, D. R. (2001). A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21, 345-383.
- Kass, M., Witkin, A., & Terzopoulos, D. (1998). Snakes: active contour models. *International Journal of Computer Vision*, 321-331.
- Kazemi, N., Duever, T. A., & Penlidis, A. (2011). Reactivity ratio estimation from cumulative copolymer composition data. *Macromolecular reaction engineering*, 5, 385-403.
- Keeler, S. E., & Reilly, P. M. (1991). The error-in-variable model applied to parameter estimation when the error covariance matrix is unknown. *The canadian journal of chemical engineering*, 69, 27-34.

- Ko, B. C., Seo, M. S., & Nam, J. Y. (2007). Microscopic image segmentation for the clinical support system. *Computer Graphics, Imaging and Visualisation*, 489-494.
- Kramer, M. A. (1991). Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, 1991, 333-343.
- Kybic, J. (2010). Bootstrap resampling for image registration uncertainty estimation without ground truth. *IEEE Transactions on Image Processing*, 19(1), 64-73.
- Lagaros, N. D., & Papadopoulos, V. (2006). Optimum design of shell structures with random geometric material and thickness imperfections. *International Journal of Solids and Structures*, 43(22-23), 6948–6964.
- Lee, J. M., Yoo, C., Choi, S. W., Vanrolleghem, P. A., & Lee, I. B. (2004). Nonlinear process monitoring using kernel principal component analysis. *Chemical Engineering Science*, 59(1), 223-234.
- Li, C., Kao, C.-Y., Gore, J. C., & Ding, Z. (2008). Minimization of region-scalable fitting energy for image segmentation. *IEEE Transactions on Image Processing*, 17(10), 1940-1949.
- Li, W., Yue, H. H., Valle-Cervantes, S., & Qin, S. J. (2000). Recursive PCA for adaptive process monitoring. *Journal of Process Control*, 10(5), 471-486.
- Li, X., & Yang, G. (2012). Fault detection for linear stochastic systems with sensor stuck faults. *Optimal control applications and methods*, 33(1), 61-80.
- Liang, J., Chaudhuri, S., & Shinozuka, M. (2007). Simulation of non-stationary stochastic processes by spectral representation. *Journal of Engineering Mechanics*, 133(6), 616-627.
- Ljung, L. (1999). *System Identification - Theory for the user* (2nd ed.). N.J.: Prentice-Hall.
- MacGregor, J. F., & Kourti, T. (1995). Statistical process control of multivariate processes. *Control engineering practice*, 3, 403-414.
- Madankan, R., Singla, P., Singh, T., & Scott, P. D. (2013). Polynomial chaos based Bayesian approach for state and parameter estimation. *Journal of Guidance, Control and Dynamics*, 36(4), 1058-1074.
- Madruga, E. L., & Fernandez-Garcia, M. (1995). High conversion copolymerization of DI-n-butyl itaconate with methyl methacrylate in benzene solution. *European Polymer Journal*, 11(31), 1103-1107.
- Madruga, E., & Fernandez-Garcia, M. (1994). Free-radical homopolymerization and copolymerization of di-n-butyl itaconate. *Polymer*, 35(20), 4437-4442.
- Mandur, J., & Budman, H. (2014). Robust optimization of chemical process using Bayesian description of parametric uncertainty. *Journal of Process Control*, 24(2), 422-430.

- Maulud, A., Wang, D., & Romagnoli, J. (2006). A multi-scale orthogonal nonlinear strategy for multivariate statistical process monitoring. *Journal of process control*, 16, 671-683.
- Meng, X., & Yang, G. (2014). Simultaneous fault detection and control for stochastic time-delay systems. *International Journal of Systems Science*, 45(5), 1058-1069.
- Mercille, S., & Massie, B. (1994). Induction of apoptosis in nutrient deprived cultures of hybridoma and myeloma cells. *Biotechnology and Bioengineering*, 44(9), 1140-1154.
- Mesbah, A., Streif, S., Rindeisen, R., & Braatz, R. D. (2014). Active fault diagnosis for nonlinear systems with probabilistic uncertainties. *the 19th World Congress, The international Federation of Automatic Control*. Cape Town, South Africa.
- Meshram, M., Naderi, S., McConkey, B., Budman, H., Scharer, J., & Ingalls, B. (2011). Population based modeling of the progression of apoptosis in mammalian cell culture. *Biotechnology and Bioengineering*, 109(5), 1193-1204.
- Montgomery, D. C., & Runger, G. C. (1994). *Applied Statistics and Probability for Engineers*. New York: John Wiley & Sons.
- Mulvey, C., Curtis, A., Singh, S., & Bigio, I. (2007). Elastic scattering spectroscopy as a diagnostic tool for apoptosis in cell cultures. *IEEE Journal of Selected Topics in Quantum Electronics*, 1663-1670.
- Murat, L. C. (2012). *Two-dimensional Change Detection Methods: Remote Sensing Application*. Springer.
- Mylaraswamy, D., & Venkatasubramanian, V. (1997). A hybrid framework for large scale process fault diagnosis. *Computers & Chemical Engineering*, 21, 935-940.
- Nagy, N., & Braatz, R. D. (2007). Distributional uncertainty analysis using power series and polynomial chaos expansion. *Journal of Process Control*, 17, 229-240.
- Negiz, A., & Cinar, A. (1997). Statistical monitoring of multivariable dynamic processes with state space models. *AIChE journal*, 48(3), 2002-2020.
- Osher, S., & Sethian, J. A. (1988). Fronts propagating with curvature dependent speed: algorithms based on Hamilton Jacobi formulation. *Journal of Computational Physics*, 79, 12-49.
- Patton, R. J. (1997). Fault tolerant control: the 1997 situation. In proceedings of the 3rd IFAC symposium on fault detection, supervision and safety for technical processes.
- Patton, R. J., Frank, P. M., & Clark, R. N. (2010). *Issues of fault diagnosis for dynamic systems*. Springer.

- Patz, T., & Preusser, T. (2012). Segmentation of stochastic Images with a stochastic random walker method. *IEEE Transactions on Image Processing*, 21(5), 2424-2433.
- Paulson, J. A., Raimondo, D. M., Findeisen, R., Braatz, R. D., & Streif, S. (2014). Guaranteed Active Fault Diagnosis for Uncertain Nonlinear Systems. *European Control Conference (ECC)*. Strasbourg, France: European Control Conference.
- Pence, B. L., Fathy, H. K., & Stein, J. L. (2011). Recursive maximum likelihood parameter estimation for state space systems using polynomial chaos theory. *Automatica*, 47, 2420-2424.
- Peng, H., Long, F., & Ding, C. (2005). Feature selection based on mutual information: criteria of max-dependency, max-relevance and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1226-1238.
- Phoon, K. K., Huang, S. P., & Quek, S. T. (2002). Implementation of Karhunen-Loeve expansion for simulation using a wavelet-Galerkin scheme. *Probabilistic Engineering Mechanics*, 17(3), 293-303.
- Prashant, M., Charles, M., Adiwinata, G., Panagiotis, D. C., & James, F. D. (2008). Isolation and handling of actuator faults in nonlinear systems. *Automatica*, 144(1), 53-62.
- Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (2007). *Numerical recipes: the art of scientific computing*. Cambridge, England: Cambridge University Press.
- Preusser, T., Scharr, H., Krajsek, K., & Kirby, R. M. (2008). Building blocks for computer vision with stochastic partial differential equations. *International Journal of Computer Vision*, 80(3), 375-405.
- Raimondo, D. M., Marseglia, G. R., Braatz, R. D., & Scott, J. K. (2013). Fault tolerant model predictive control with active fault isolation. *Conference on Control and Fault-Tolerant Systems (Sys Tol)*. Nice, France: Conference on Control and Fault-Tolerant System.
- Rasmussen, C. E., & Williams, C. K. (2006). *Gaussian processes for machine learning*. Cambridge, MA: The MIT Press.
- Rawlings, J., & Bakshi, B. (2006). Particle filtering and moving horizon estimation. *Computers and chemical engineering*, 30, 1529-1541.
- Riggs, J. (1999). *Chemical process control*. Texas: Ferret Publishing.
- Rodenacker, K., & Bengtsson, E. (2003). A feature set for cytometry on digitized microscopic images. *Analytical Cellular Pathology*, 1-36.
- Ross, J. P. (1988). *Taguchi techniques for quality engineering*. New York: McGraw-Hill.

- Rulter, M., Spearman, M., & Braasch, K. (2014). Monitoring cell growth, viability and apoptosis. In R. Portner (Ed.), *Animal cell biotechnology, method and protocols* (pp. 169-192). Hamburg, Germany: Springer.
- Said, A., Karam, L., Berens, M., Lacroix, Z., & Renaut, R. (2007). Migration and proliferation analysis for bladder cancer cells. *IEEE International Symposium on Biomedical Imaging*, (pp. 320-323). Boston.
- Schneider, C. A., Rasband, W. S., & Eliceiri, K. W. (2012). NIH image to ImageJ: 25 years of image analysis. *Nature Methods*, 9, 671-675.
- Schwab, C., & Todor, R. A. (2006). Karhunen-loeve approximation of random field by generalized fast multipole methods. *Journal of Computational Physics*, 27(1), 100–122.
- Scott, J. K., Findeisen, R., Braatz, R. D., & Raimondo, D. M. (2013). Design of Active Inputs for Set-Based Fault Diagnosis. *the American Control Conference*. Washington, DC, USA: 2013 American Control Conference.
- Seborg, D. E., Mellichamp, D. A., Edgar, T. F., & Doyle, F. J. (2010). *Process dynamics and control*. Hoboken, NJ: John Wiley & Sons, Inc.
- Sethian, J. A. (2002). *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision and materials science*. Cambridge, UK: Cambridge University Press.
- Shams, M. B., Budman, H., & Duever, T. (2011). Finding a trade-off between observability and economics in the fault detection of chemical processes. *Computers and chemical engineering*, 35, 319-328.
- Shi, J. Q., & Choi, T. (2011). *Gaussian process regression analysis for functional data*. London: Chapman & Hall CRC.
- Shinozuka, M., & Deodatis, G. (1996). Simulation of multi-dimensional Gaussian stochastic field by spectral representation. *Applied Mechanics Reviews*, 49(1), 29-53.
- Snieder, R. (1998). The role of nonlinearity in inverse problems. *Inverse Problems*, 14, 387-404.
- Spanos, P. D., & Zeldin, B. (1998). Monte Carlo treatment of random fields: A broad presepctive. *Applied Mechanics Reviews*, 51(3), 219-237.
- Spanosa, P. D., Tezcanb, J., & Tratskasc, P. (2005). Stochastic processes evolutionary spectrum estimation via harmonic wavelets. *Computer Methods in Applied Mechanics and Engineering*, 194(12-16), 1367–1383.

- Srinivasan, R., & Qian, M. (2007). State specific key variables for monitoring multi-state processes. *Chemical Engineering Research and Design*, 85(12), 1630-1644.
- Stefanou, G. (2009). The stochastic finite element method: Past, present and future. *Computer methods in applied mechanics and engineering*, 198, 1031-1051.
- Stefanou, G., Nouy, A., & Clement, A. (2009). Identification of random shapes from images through polynomial chaos expansion of random level-set functions. *International Journal of Numerical Methods in Engineering*, 79(2), 127-155.
- Stewart, B. T., Venkat, A. N., Rawlings, J. B., Wright, S. J., & Pannocchia, G. (2010). Cooperative distributed model predictive control. *Systems and Control Letters*, 59, 460-469.
- Taatjes, D. J., Sobel, B. E., & Budd, R. C. (2008). Morphological and cytochemical determination of cell death by apoptosis. *Histochemistry and Cell Biology*, 129(1), 33-43.
- Theriault, D. H., Walker, M. L., Wong, J. Y., & Betke, M. (2012). Cell morphology classification and clutter mitigation in phase contrast microscopy images using machine learning. *Machine Vision and Applications*, 23(4), 659-673.
- Tyler, M. L., & Morari, M. (1994). Optimal and robust design of integrated control and diagnosis modules. *American Control Conference (ACC)*. Baltimore, Maryland, USA: Proceedings of the American Control Conference.
- Ullah, A. (1996). Entropy, divergence, and distance measures with econometric applications. *Journal of Statistical Planning and Inference*, 49, 137-162.
- Venkatasubramanian, R., & Kavuri, S. N. (2003). A review of process fault detection and diagnosis, part III: process history based methods. *Computers and chemical engineering*, 27, 327-346.
- Venkatasubramanian, V., Rengaswamy, R., Yin, K., & Kavuri, S. (2003). A review of process fault detection and diagnosis Part I: Quantitative model-based methods. *Computers and Chemical Engineering*, 27, 293-311.
- Veredas, F., Mesa, H., & Morente, L. (2010). Binary tissue classification on wound images with neural networks and Bayesian classifiers. *IEEE Transactions on Medical Imaging*, 29(2), 410-427.
- Wan, X., & Karniadakis, G. E. (2006). Beyond Wiener-Askey expansions: handling arbitrary PDFs. *Journal of Scientific Computing*, 455-464.
- Wand, M. P., & Jones, M. C. (1995). *Kernel smoothing* (First Edition ed.). New York: Chapman and Hall.

- Wang, D., & Romagnoli, J. A. (2005). Robust multi-scale principal components analysis with applications to process monitoring. *Journal of process control*, 15(8), 869-882.
- Wang, X., Kruger, U., & Lennox, B. (2003). Recursive partial least squares algorithms for monitoring complex industrial processes. *Control Engineering Practice*, 11, 613-632.
- Waters, J. C. (2009). Accuracy and precision in quantitative fluorescence microscopy. *The Journal of Cell Biology*, 185(7), 1135-1148.
- Wiener, N. (1938). The homogenous chaos. *American Journal of Mathematics*, 60, 897-936.
- Witteveen, J., & Bijl, H. (2006). Modeling arbitrary uncertainties using Gram-Schmidt polynomial chaos. Reno, Nevada: the 44th AIAA aerospace sciences meeting and exhibit.
- Witteveen, J., & Bijl, H. (2006). Modeling arbitrary uncertainties using Gram-Schmidt polynomial chaos. *44th AIAA Aerospace Sciences Meeting and Exhibit* (pp. 1-17). Reno, Nevada: American Institute of Aeronautics and Astronautics.
- Xiu, D. (2009). Fast numerical methods for stochastic computations: a review. *Communications in computational physics*, 5(2-4), 242-272.
- Xiu, D. (2010). *Numerical methods for stochastic computations: a spectral method approach*. Princeton, New Jersey: Princeton University Press.
- Xiu, D., & Karniadakis, G. (2002). The Wiener-Askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing*, 24(2), 619-644.
- Xiu, D., & Karniadakis, G. E. (2003). Modeling uncertainty in flow simulations via generalized polynomial chaos. *Journal of computational physics*, 187, 137-167.
- Yin, Z., Bise, R., Chen, M., & Kanade, T. (2010). Cell segmentation in microscopy imagery using a bag of local Bayesian classifiers. *IEEE International Symposium on Biomedical Imaging*. Rotterdam.
- Zhang, Y., & Jiang, J. (2008). Bibliographical review on reconfigurable fault tolerant control systems. *Annual reviews in control*, 32, 229-252.