Robust Algorithms for Optimization of Chemical Processes in the Presence of Model-Plant Mismatch

by

Jasdeep Singh Mandur

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, 2014

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.

Jasdeep Singh Mandur

Abstract

Process models are always associated with uncertainty, due to either inaccurate model structure or inaccurate identification. If left unaccounted for, these uncertainties can significantly affect the model-based decision-making. This thesis addresses the problem of model-based optimization in the presence of uncertainties, especially due to model structure error. The optimal solution from standard optimization techniques is often associated with a certain degree of uncertainty and if the model-plant mismatch is very significant, this solution may have a significant bias with respect to the actual process optimum. Accordingly, in this thesis, we developed new strategies to reduce (1) the variability in the optimal solution and (2) the bias between the predicted and the true process optima.

Robust optimization is a well-established methodology where the variability in optimization objective is considered explicitly in the cost function, leading to a solution that is robust to model uncertainties. However, the reported robust formulations have few limitations especially in the context of nonlinear models. The standard technique to quantify the effect of model uncertainties is based on the linearization of underlying model that may not be valid if the noise in measurements is quite high. To address this limitation, uncertainty descriptions based on the Bayes' Theorem are implemented in this work. Since for nonlinear models the resulting Bayesian uncertainty may have a non-standard form with no analytical solution, the propagation of this uncertainty onto the optimum may become computationally challenging using conventional Monte Carlo techniques. To this end, an approach based on Polynomial Chaos expansions is developed. It is shown in a simulated case study that this approach resulted in drastic reductions in the computational time when compared to a standard Monte Carlo sampling technique. The key advantage of PC expansions is that they provide analytical expressions for statistical moments even if the uncertainty in variables is non-standard. These expansions were also used to

speed up the calculation of likelihood function within the Bayesian framework. Here, a methodology based on Multi-Resolution analysis is proposed to formulate the PC based approximated model with higher accuracy over the parameter space that is most likely based on the given measurements.

For the second objective, i.e. reducing the bias between the predicted and true process optima, an iterative optimization algorithm is developed which progressively corrects the model for structural error as the algorithm proceeds towards the true process optimum. The standard technique is to calibrate the model at some initial operating conditions and, then, use this model to search for an optimal solution. Since the identification and optimization objectives are solved independently, when there is a mismatch between the process and the model, the parameter estimates cannot satisfy these two objectives simultaneously. To this end, in the proposed methodology, corrections are added to the model in such a way that the updated parameter estimates reduce the conflict between the identification and optimization objectives. Unlike the standard estimation technique that minimizes only the prediction error at a given set of operating conditions, the proposed algorithm also includes the differences between the predicted and measured gradients of the optimization objective and/or constraints in the estimation. In the initial version of the algorithm, the proposed correction is based on the linearization of model outputs. Then, in the second part, the correction is extended by using a quadratic approximation of the model, which, for the given case study, resulted in much faster convergence as compared to the earlier version.

Finally, the methodologies mentioned above were combined to formulate a robust iterative optimization strategy that converges to the true process optimum with minimum variability in the search path. One of the major findings of this thesis is that the robust optimal solutions based on the Bayesian parametric uncertainty are much less conservative than their counterparts based on normally distributed parameters.

Acknowledgements

First, I would like to express my sincere gratitude to Prof. Hector Budman for all his support and guidance throughout this research. Hector, one of your qualities that really touched my heart is your patience. I will never forget the times when I used to enter a deadlock and the way you handled those situations is truly inspiring. Thank you and really, it was a pleasure working with you!

Next, I would like to thank my PhD committee members: Prof. Thomas Duever, Prof. Kumaraswamy Ponnambalam, Prof. Ali Elkamel and Prof. Fraser Forbes for their valuable comments and suggestions in the exam.

I am also very thankful to the administrative staff especially, Liz Bevan, Judy Caron, Rose Guderian, Ingrid Sherrer and Lorna Kelly for their support in handling all the paperwork over these years. Here, I don't want to miss the technical staff; Dennis Herman and Ravindra for their help in keeping my computers up and running.

For the financial support, I am very grateful to the Natural Science and Engineering Research Council (NSERC) for awarding me the Postgraduate Scholarship and also the University of Waterloo for the President's Graduate Scholarship. In addition, I also thank the chemical engineering department for the teaching assistantships.

Before I close, I would like to thank my friends and colleagues specially Mukesh Meshram, Prashant Mutyala, Walid Al-Gherwi, Rosendo Diaz, Kaveh Ohadi and Ali Nikdel for being there and making these years more enjoyable. Thank you guys and I hope we'll all stay in touch.

In the end, this day would not have been possible without the support and encouragement from my family. I am really blessed to have them as a part of my life.

Dedication

To my parents, my brother and my wife

Table of Contents

AUTHOR'S DECLARATION **ii**

Abstract **iii**

Acknowledgements **v**

Dedication **vi**

Table of Contents **vii**

List of Figures **x**

List of Tables **xii**

Chapter 1 Introduction **1**

1.1 Research Objectives 4

1.2 Organization of thesis.....4

Chapter 2 Literature Review **7**

2.1 Parameter Estimation 7

2.2 Uncertainty Propagation.....14

2.2.1 Taylor Series Expansion.....14

2.2.2 Monte Carlo based Uncertainty Analysis.....16

2.2.3 Polynomial Chaos Expansions 17

2.3 Optimization under Uncertainty.....23

Chapter 3 Robust Optimization of Chemical Processes using Bayesian description of parametric uncertainty **29**

3.1 Introduction 29
3.2 Background and methodology 33
3.2.1 Polynomial Chaos Expansions 33
3.2.2 Bayesian framework 34
3.2.3 Uncertainty Propagation using PC expansions 38
3.3 Case Study: Fed-batch bioprocess 44
3.3.1 Problem formulation 44
3.3.2 Results and discussion 47
3.3.3 Computational Efficiency 54
3.4 Conclusions 59

Chapter 4 Simultaneous model identification and optimization in presence of model-plant mismatch 60

4.1 Introduction 60
4.2 Preliminaries 64
4.3 Proposed Methodology 69
4.3.1 Conditions for Convergence 73
4.3.2 Termination Criteria 74
4.3.3 Summary of algorithm 76
4.4 Case Study 78
4.4.1 Problem formulation 78
4.4.2 Results and discussion 81
4.5 Conclusions 90

Chapter 5 On improving the convergence of simultaneous model identification and optimization algorithm **91**

5.1 Introduction 91

5.2 Model Correction 95

5.2.1 Preliminaries.....95

5.2.2 Quadratic Model Correction.....97

5.3 Robust Optimization 100

5.3.1 Parametric Uncertainty.....100

5.3.2 Polynomial Chaos Expansions 102

5.4 Proposed optimization methodology.....105

5.5 Case Study.....107

5.5.1 Results and discussion.....110

5.6 Conclusion.....123

Chapter 6 Conclusions and Future Work **124**

6.1 Robust Optimization based on Bayesian parametric uncertainty.....124

6.2 Model correction for optimization 126

6.3 Robust run-to-run optimization.....127

6.4 Future Work 128

Copyright Permissions **131**

Bibliography **132**

Appendix A Matlab Codes **139**

List of Figures

Figure 3-1: (a) Comparison of Bayesian and Normal uncertainty for K_I; (b) 2-D representation of joint Bayesian uncertainty for K_X and K_I (c) 2-D representation of joint normal uncertainty for K_X and K_I 49

Figure 3-2: Distribution of the penicillin at the end of batch, predicted at the optimum, based on the (a) Bayesian uncertainty and (b) Normal uncertainty 52

Figure 3-3: (a) Penicillin, at end of the batch, as a function of K_X and K_I for (a) 0.2 threshold and (b) 0.5 threshold 57

Figure 3-4: 2-D representation of the joint probability distribution of K_X and K_I obtained by (a) PC approximation for 0.2 threshold, (b) Multi-resolution approximation for 0.2 threshold (c) PC approximation for 0.5 threshold and (d) Multi-resolution approximation for 0.5 threshold 58

Figure 4-1: Illustration of lack of fit 70

Figure 4-2: Proposed Algorithm with linear model corrections 77

Figure 4-3: Comparison of the effect of ϵ_{max}^{T} on convergence of optimalS_o 82

Figure 4-4: Predictions of corrected model at converged optimal solution ($-*-$ Predictions; $-\mathrm{o}-$ Noise-free measurements) 83

Figure 4-5: Convergence of optimal S_o 84

Figure 4-6: Comparison of proposed and modifier adaption algorithms on convergence of optimal S_o 87

Figure 4-7: Average convergence of optimal S_o for (a) Modifier Adaptation algorithm and (b) Proposed algorithm 88

Figure 5-1: Proposed Algorithm with quadratic correction and robust formulation 106

Figure 5-2: Comparing the effect of linear vs. quadratic model correction on the convergence of optimal S_o 111

Figure 5-3: Convergence of optimal S_o based on (a) linear model correction and $\epsilon_{max}^{T} = 1\%$, (b) quadratic model correction and $\epsilon_{max}^{T} = 1\%$, (c) linear model correction and $\epsilon_{max}^{T} = 5\%$ and (d) quadratic model correction and $\epsilon_{max}^{T} = 5\%$ 113

Figure 5-4: Robust convergence of optimal S_o for linear model correction and (a) $\epsilon_{max}^{T} = 1\%$ and (b)$\epsilon_{\max}^{\mathrm{T}} = 5\%$..... 118

Figure 5-5: Comparing the effect of Bayesian vs normal parametric uncertainty on the robust convergence of optimal S_o for (a) $\epsilon_{max}^{T} = 1\%$ and (b) $\epsilon_{max}^{T} = 5\%$; ($-\mathrm{o}-$ Normal with $w = 2\%$; $-\diamond-$ Bayesian with $w = 2\%$; $-*-$ Normal with $w = 1\%$; $-\mathrm{x}-$ Bayesian with $w = 1\%$)..... 121

Figure 5-6: Comparing normal (dotted) and Bayesian (solid) descriptions of parametric uncertainty in K_X and K_I..... 122

List of Tables

Table 3-1: Initial concentrations and input feed profile for the simulated data 45

Table 3-2: Parameters' values for process simulator (Equations (3.29)-(3.32)) 46

Table 3-3: List of means and variances, comparing Bayesian and Normal distributions for each example 48

Table 3-4: Comparison of optimal solution based on Bayesian and Normal description of uncertainty 51

Table 4-1: Parameters' values for process simulator (Equations 4.22-4.25)..... 79

Table 4-2: Initial set of input conditions used to estimate the parameters in uncertain model..... 80

Table 4-3: Comparison of convergence properties for the proposed algorithm vs modifier adaptation 89

Table 5-1: Set of initial input conditions 109

Table 5-2: Comparison of model correction based on linear vs quadratic approximation 112

Table 5-3: Comparison of nominal vs. robust iterative optimization (Linear model correction) 117

Chapter 1
Introduction

In the face of growing competition, limited resources and strict environmental regulations, optimization is an important tool for process industries to maximize their production and profits while keeping the use of available resources to the minimum. Since the exact relation between performance and the input conditions of a process may not be known a priori, the optimization techniques often rely on some mathematical representation of the process, also referred to as "process models". However, the successful implementation of model-based optimal solution depends on how accurately the model can predict the process behavior over the entire space of input conditions. Any inaccuracy in model predictions may result in non-optimal operating policies, leading to a significant loss in performance or even the violation of constraints. The model development is usually an iterative procedure where as a first step the proposed model is calibrated using the measurements at some initial set of operating conditions. Then, in the second step, also known as a validation step, the prediction accuracy of this model is evaluated at operating conditions different from the ones used in the first step. This iterative procedure is repeated until a model with required prediction accuracy is obtained. However, in most practical situations, it is difficult to measure all the states and the ones that can be measured are often associated with significant level of noise. The problem becomes even more critical if the cost and duration of the experiments are very high. In this case, the number of experiments that can be performed in a given timeframe is limited and may not provide sufficient excitation to estimate all the model parameters accurately. As a result, one has to compromise for a model that will be accurate only in a limited space of input conditions. Since it is practically impossible to come up with an accurate model for a nonlinear process, it is very important to investigate the effect of model uncertainties on the optimal solutions. If the resulting optimal solution is highly

inaccurate and uncertain, the strategies must be implemented to either reduce the model uncertainties that affect the optimal solution or search for an optimal solution that is robust to these uncertainties.

In the search for robust optimal solutions, the effect of model uncertainties is included explicitly in the optimization objective function. At a given set of operating conditions, the model uncertainties are generally quantified in terms of uncertainty in the parameter estimates. The standard technique is based on the linearization of model outputs around the parameter estimates, which for normally distributed measurement errors results in normal distribution for the parameters. Unlike the nominal optimization where the objective function is evaluated once for the nominal parameter estimates, the robust approach involves the calculation of objective function for different realizations of parameter values within the uncertainty region. Then, the goal is to minimize a weighted sum of expectation and variance of the objective function.

For highly nonlinear problems, the normal description for parametric uncertainty is valid only if the degree of measurement noise is such that the assumption of model linearity holds within the uncertain parameter region. If this assumption does not hold, the normal distribution may result in conservative optimal solutions. To alleviate this limitation, in this thesis it is proposed to implement a more accurate description based on the Bayes' Theorem. However, one of the major challenges in its implementation is the computational time involved in propagating non-normal uncertainty descriptions. Since, for nonlinear models, these descriptions do not have analytical expressions, the conventional approach is to use Monte Carlo sampling techniques where the parameter values are selected randomly from their distribution and then the corresponding values for the desired output variable are computed. Since this procedure requires repetitive simulations of full nonlinear model and considering the fact that this propagation step has to be performed a number of times during the optimization, this approach may become computationally prohibitive. Although the

computational time can be reduced by using some approximation of the model, e.g. first-order or second-order Taylor expansion, the question arises whether these approximations are valid within the uncertain parameter region. As an alternative, an approach based on the Polynomial Chaos (PC) expansions is proposed in this thesis. In the PC framework, any random variable with a finite variance can be expanded in terms of a set of independent random variables with standard distributions. The first term in the expansion is a constant, representing the mean value of the random variable whereas the remaining terms capture the random behavior. The most attractive property of these expansions is that the basis functions are orthogonal to each other and thus any statistical measure such as mean and/or variance can be calculated analytically.

In another class of optimization algorithms, where the aim is to reduce the model uncertainties, the standard procedure is to update the model iteratively around new optimal operating conditions until a convergence is achieved. This procedure, often referred to as "two-step approach", involves two sequential steps; first, the model is updated at previously calculated optimal solution and, in the subsequent step, the updated model is re-optimized for the next iteration. In the presence of model mismatch, since the model identification and optimization steps are independent of each other, it cannot be guaranteed that the above iterative approach will converge to a true process optimum. The reason is that when the inaccurate model is calibrated over different operating conditions, the parameter estimates have to compensate for the unmodelled dynamics. As a result, a unique set of parameter estimates that can predict the process behavior accurately over the entire space of operating conditions does not exist. It is also possible that, in an attempt to satisfy the identification objectives, the change in parameter estimates between two operating conditions can be of an extent that the model based optimization no longer proceeds in the direction of true process optimum. To address this problem, a class of algorithms has been proposed in the past where the optimization problem is corrected by using the measured gradients for the optimization objective and/or the constraints as a

feedback. However, in these algorithms, these corrections are external to the model, i.e. the model is not modified specifically for these corrections. Instead, the model parameters are updated using the standard estimation problem and, then, the differences between the predicted and measured gradients of the optimization objective and/or constraints are added to the respective quantities. In this thesis, an alternative approach is presented where the model structure is corrected such that, with the updated parameter estimates, the model simultaneously satisfies the identification and optimization objectives. In other words, the objective becomes identifying the model structure that is accurate around the process optimum.

1.1 Research Objectives

Based on the discussion in this chapter, the following three objectives were identified:

1. To develop a computationally efficient algorithm based on Polynomial Chaos expansions to solve a Robust Optimization problem when the parametric uncertainty is given by the Bayes' Theorem

2. To develop an iterative optimization algorithm to correct the model progressively for structural uncertainty with a guaranteed convergence to a true process optimum

3. To investigate and reduce the effect of model uncertainties on convergence of the algorithm developed in Step 2, using the tools from Step 1.

1.2 Organization of thesis

Overall, this thesis is organized in six chapters as follows:

After a brief introduction and outlining the research objectives in Chapter 1, Chapter 2 reviews the theory and existing literature on different topics relevant to this

research. The topics include parameter estimation, uncertainty quantification, propagation of uncertainty and optimization in the presence of model uncertainties.

In Chapter 3, a novel approach based on Polynomial Chaos (PC) expansions is developed (1) to quantify the parametric uncertainty using the Bayes' Theorem and (2) to propagate this uncertainty onto the optimization objective. The computational efficiency of the overall robust optimization approach is compared with the standard Monte Carlo sampling technique. The optimization results are also compared with those obtained by normal description of uncertainty.

In Chapter 4, an iterative optimization framework is developed where the model is corrected iteratively for model-plant mismatch as the algorithm progresses towards the true process optimum. Upon convergence, the updated model simultaneously satisfies the identification and optimization objectives. The corrections to the model are based on linearization of model outputs. The conditions for guaranteed convergence have also been formalized. A comparative study is presented where the algorithm is compared with the standard "two step" approach and a "modifier adaptation" algorithm proposed in the literature.

Chapter 5 presents two modifications to the iterative algorithm, developed in Chapter 4, that improve its rate of convergence and robustness towards the modeling errors and the measurement noise. In the first modification, a new correction term based on quadratic approximation of the model outputs is proposed. In the second modification, the measure for robustness is added explicitly to the optimization objective and the resulting robust optimization problem is solved as per the procedure proposed in chapter 3. The results are compared for both Bayesian and normal descriptions of parametric uncertainty.

Finally, Chapter 6 reviews the key contributions of this thesis and provides the directions for future research in this subject.

Chapters 3-5 are presented in a manuscript format. The contents of Chapter 3 have already been published in the ADCHEM 2012 Special Issue of Journal of Process Control (Mandur et al., 2013a). The manuscript in Chapter 4 has been submitted to Chemical Engineering Science and is currently under review and Chapter 5 is to be submitted in the near future. The parts of these chapters have also been published in two refereed conference proceedings and have been presented in oral presentations in national/international conferences as follows:

Refereed Conference Proceedings:

- Mandur, J. & Budman, H. (2013b). A Robust algorithm for Run-to-run Optimization of Batch Processes. In 10th IFAC International Symposium on Dynamics and Control of Process Systems, Vol. 10, No. 1, pp. 541-546

- Mandur, J. & Budman, H. (2012). A Polynomial-Chaos Based Algorithm for Robust Optimization in the Presence of Bayesian Uncertainty. In 8th IFAC Symposium on Advanced Control of Chemical Processes, Vol. 8, No. 1, pp. 549-554

Non-referred Conference Presentations:

- Mandur, J. & Budman, H. (2013). A Robust algorithm for Run-to-run Optimization of Batch Processes. 63rd Canadian Chemical Engineering Conference, Fredericton, NB, October 20-23

- Mandur, J. (2012). A Polynomial-Chaos Based Algorithm for Robust Optimization in the Presence of Bayesian Uncertainty. Statistics and Control Meeting 2012, McMaster University, May 23-24

- Mandur, J. & Budman, H. (2012). A Polynomial-Chaos Based Algorithm for Robust Optimization in the Presence of Bayesian Uncertainty. 62nd Canadian Chemical Engineering Conference, Vancouver, October 14-17

Chapter 2
Literature Review

This chapter reviews the existing methodologies in the following research areas: (1) estimation of parameters and associated parametric uncertainty; (2) propagation of uncertainty and (3) optimization in the presence of model uncertainties. Since an extensive research has been carried out in all these areas, the work that is specifically relevant to the research objectives is reviewed here.

2.1 Parameter Estimation

A model of a nonlinear process can be described by a set of ordinary differential equations as follows:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \boldsymbol{\theta}, \mathbf{u}, \mathbf{t}) + \boldsymbol{\upsilon}$$

$$\mathbf{y} = \boldsymbol{h}(\mathbf{x}) \qquad (2.1)$$

Where, $\mathbf{x} \in \mathbb{R}^{n_x}$ is a vector of model states, $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}$ is a vector of model parameters, $\mathbf{u} \in \mathbb{R}^{n_u}$ is a vector of process inputs or operating conditions, $\mathbf{y} \in \mathbb{R}^{n_y}$ is a vector of predicted output variables, $f \in \mathbb{R}^{n_x}$ is a set of differential equations based on the mass and energy balances, $h \in \mathbb{R}^{n_y}$is a mapping between the model states and the predicted outputs and $\boldsymbol{\upsilon}$ is a vector of uncertainties representing the model-plant mismatch.

After formulating a model structure, the next task is to estimate the unknown model parameters using the measurements from the process. Here, the goal is to search for the parameter estimates such that, at any given set of operating conditions, the model can predict the measured output variables with a minimum possible error. However, it is often the case that only a subset of parameters can be estimated from a given set of measurements. The reason is that, most of the time, not all the states can be measured

and the ones that are measured are often associated with a significant level of noise. As a result, it is very important to identify this subset of parameters before implementing any estimation strategy (Vajda et al., 1989). In general, the parameters are considered estimable if the derivatives of the output variables with respect to these parameters are linearly independent (Beck et al., 1977). The simplest method to test for estimability is to plot the evolution of the derivatives as a function of time. If the sum of derivatives corresponding to the effect of any two parameters on the same output variable is zero for the entire period, the two parameters are considered to be perfectly correlated and cannot be estimated simultaneously. The graphical approach is suitable only for simpler models with very few parameters. A more generic approach is based on an orthogonalization procedure proposed by Yao et al. (2003). This method uses the parametric sensitivity matrix, defined as follows:

$$Z = \begin{bmatrix} \left.\frac{\partial y_1}{\partial \theta_1}\right|_{t=t_1} & \cdots & \left.\frac{\partial y_1}{\partial \theta_{\mathrm{n}_\theta}}\right|_{t=t_1} \\ \vdots & \ddots & \vdots \\ \left.\frac{\partial y_{\mathrm{n}_y}}{\partial \theta_1}\right|_{t=t_1} & \cdots & \left.\frac{\partial y_{\mathrm{n}_y}}{\partial \theta_{\mathrm{n}_\theta}}\right|_{t=t_1} \\ \left.\frac{\partial y_1}{\partial \theta_1}\right|_{t=t_N} & \cdots & \left.\frac{\partial y_1}{\partial \theta_{\mathrm{n}_\theta}}\right|_{t=t_N} \\ \vdots & \ddots & \vdots \\ \left.\frac{\partial y_{\mathrm{n}_y}}{\partial \theta_1}\right|_{t=t_N} & \cdots & \left.\frac{\partial y_{\mathrm{n}_y}}{\partial \theta_{\mathrm{n}_\theta}}\right|_{t=t_N} \end{bmatrix} \tag{2.2}$$

Where $\left.\frac{\partial y_i}{\partial \theta_j}\right|_t$ represents the derivative of the i^{th} output variable with respect to j^{th} parameter at t^{th} time point.

If there is no correlation between the parameters, it is straightforward to identify a set of estimable parameters by simply rearranging the sum of squares in each column of Z in a descending order. Then, the parameter corresponding to a column with the highest sum of squares is the most identifiable as its overall effect on the outputs is

the highest. However, when the parameters have a correlated effect on the outputs, this effect has to be adjusted before their ranking. Let X_L be the column of Z corresponding to the most estimable parameter. Then, the orthogonalization procedure is as follows (Yao et al., 2003):

1. Assuming a linear relation between Z and X_L, solve a linear regression problem where the solution is given by a matrix Z_L as follows:

$$Z_L = X_L(X_L^T X_L)^{-1} X_L^T Z \tag{2.3}$$

2. Calculate a residual matrix R_L, given by $Z - Z_L$. Then, the next most estimable parameter corresponds to a column of this residual matrix with the highest sum of squares.

3. Augment X_L with the column of Z corresponding to the most estimable parameter, from Step 2 and let this matrix be X_{L+1}.

4. Resolve a linear regression problem assuming Z as a linear function of X_{L+1} as follows:

$$Z_L = X_{L+1}(X_{L+1}^T X_{L+1})^{-1} X_{L+1}^T Z \tag{2.4}$$

5. Recalculate the residual matrix R_L and identify the next most estimable parameter as outlined in the Step 2.

The procedure is repeated until the sum of squares corresponding to the next estimable parameter in R_L is below a predefined threshold.

Once a subset of estimable parameters is obtained, the next step is to formulate an objective to obtain maximum agreement between the predicted and measured outputs. One of the most commonly used formulations is based on sum of squared errors, also referred to as least squares estimation (LSE).

If $\mathbf{y_m} \in \mathbb{R}^{n_y}$ is a vector of measured output variables, then the parameter estimates by LSE are given as:

$$\widetilde{\boldsymbol{\theta}} = arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} \|\mathbf{y_m}(\mathbf{u}) - \mathbf{y}(\mathbf{u}, \boldsymbol{\theta})\|^2 \tag{2.5}$$

Here, the underlying assumption is that the residuals, given by $\mathbf{y_m}(\mathbf{u}) - \mathbf{y}(\mathbf{u}, \widetilde{\boldsymbol{\theta}})$, are normally distributed and are uncorrelated with a constant variance. For a correlated and non-constant variance, the objective function has to be modified as follows:

$$\widetilde{\boldsymbol{\theta}} = arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} \big(\mathbf{y_m}(\mathbf{u}) - \mathbf{y}(\mathbf{u}, \boldsymbol{\theta})\big)^T W^{-1} \big(\mathbf{y_m}(\mathbf{u}) - \mathbf{y}(\mathbf{u}, \boldsymbol{\theta})\big) \tag{2.6}$$

Where, W represents the variance-covariance matrix for the residuals

In this thesis the residuals are assumed to be uncorrelated but have unequal variances and in this case, Equation (2.6) reduces to a weighted least squares estimation problem.

Since the measurements are always corrupted with a noise, it is very important to quantify the confidence in the above estimates. A standard approach is based on the linearization of model outputs around these estimates and then, for normally distributed residuals, the covariance matrix for the estimates can be approximated as follows (Bard et al., 1974):

$$V_{\boldsymbol{\theta}}^{-1} = \sum_{i=1}^{N} \mathbf{B_i^T} W^{-1} \mathbf{B_i} \tag{2.7}$$

Further, the $(1-\alpha)$ joint confidence region can be approximated as:

$$\big(\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}}\big)^{\boldsymbol{T}} V_{\boldsymbol{\theta}}^{-1} \big(\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}}\big) \leq \chi^2_{n_\theta}(\alpha) \tag{2.8}$$

Here, $\mathbf{B}$ represents a Jacobian matrix of measurable states with respect to the uncertain parameters and $\chi^2_{n_\theta}$ represents the chi-square distribution with n_θ degrees of freedom.

However, these approximations (Equations (2.7)-(2.8)) are only accurate if the underlying linear assumption is valid over the uncertain parameter space. For nonlinear models, these approximations may result in inaccurate and misleading analysis (Watts, 1994). This is also true in the context of model-based optimization where the robust optimal solution based on the above parametric uncertainty may be inaccurate and/or too conservative.

To this end, the Bayesian framework provides an accurate and more realistic representation of uncertainty as no linear approximation is required. In fact, the Bayesian approach provides an integrated framework to estimate both parameters as well as their associated uncertainty. Unlike other estimation techniques, it considers the model parameters to be random and as a result, the model predictions are also random. Then, the aim is to learn the distribution of parameters in terms of their ability to predict the given measurements within some predefined error structure. The measure of this ability is expressed by a likelihood function that, basically, is a conditional probability density function.

According to the Bayes' theorem, the distribution of parameters is given by:

$$P(\boldsymbol{\theta}|\mathbf{D}) \propto L(\boldsymbol{\theta}|\mathbf{D})P(\boldsymbol{\theta}) \tag{2.9}$$

Where, $P(\boldsymbol{\theta}|\mathbf{D})$ is the posterior distribution conditional on the given measurements $\mathbf{D}$, $L(\boldsymbol{\theta}|\mathbf{D})$ is the likelihood function and $P(\boldsymbol{\theta})$ is the prior distribution, representing any information available prior to collecting the measurements.

The proportionality constant in the above Equation can be eliminated by standardizing the expression as follows:

$$P(\boldsymbol{\theta}|\mathbf{D}) = \frac{L(\boldsymbol{\theta}|\mathbf{D})P(\boldsymbol{\theta})}{\int L(\boldsymbol{\theta}|\mathbf{D})P(\boldsymbol{\theta})\mathbf{d}\boldsymbol{\theta}} \tag{2.10}$$

For normally distributed errors or residuals, the likelihood function is given by a multivariate normal distribution as follows:

$$L(\boldsymbol{\theta}|\mathbf{D}) = \frac{1}{(2\pi)^{k/2}|W|^{1/2}} \exp\left(-\frac{1}{2}\left(\mathbf{y_m} - \mathbf{y}(\boldsymbol{\theta})\right)^T \mathrm{W}^{-1}\left(\mathbf{y_m} - \mathbf{y}(\boldsymbol{\theta})\right)\right) \tag{2.11}$$

Since Equation (2.11) requires only the simulated values of the model outputs, it is clear that the linearization of the model is no longer required to calculate the distribution in parameters. Furthermore, no prior estimate is required as for the case where linear approximations are used (Equations (2.7)-(2.8)). With respect to the prior distribution $P(\boldsymbol{\theta})$, unlike the conventional estimation where the only information that can be provided is in the form of lower and upper bounds, the Bayesian framework allows to incorporate more structured information in the form of a distribution. This is a key feature that makes the Bayesian approach suitable for sequential learning where the posterior distribution based on one set of measurements can be used as the prior for the next set. Therefore, as more experiments are performed, the posterior distribution can be improved iteratively.

Although this approach is ideal for nonlinear problems, the main challenge in its application is the computational time. When the model is nonlinear in parameters, the likelihood function and the corresponding posterior distribution do not have an analytical solution. In order to sample from such distribution, one has to rely on numerical approximations. The methods based on the Markov Chain Monte Carlo (MCMC) are the most promising in sampling from a complex posterior distribution. A Markov chain is a sequence of random numbers where the probability of a current sample depends only on the last sample (Robert et al., 2004). Then, the idea behind MCMC is to construct a Markov chain such that it draws more samples from the target distribution as the chain grows. Since, in the beginning, the chain may not be

sampling from the desired distribution, the first few samples are usually discarded and are referred to as a *"burn-in"* sequence. Typically, this number is 1%-2% of the total number of samples provided enough samples have been generated to obtain an accurate estimate of some statistical measure, e.g. expectation (Robert et al., 2004). The Metropolis-Hastings (M-H) is the most widely used MCMC sampler where the candidate $\boldsymbol{\theta}_{i+1}$ is, first, drawn from some proposal distribution with probability $q(\boldsymbol{\theta}_{i+1}|\boldsymbol{\theta}_i)$ and, then, is accepted with a probability α, defined as follows (Metropolis et al., 1953, Hastings et al., 1970, Chib et al., 1995):

$$\alpha = \min\left\{1, \frac{\pi(\boldsymbol{\theta}_{i+1})q(\boldsymbol{\theta}_i|\boldsymbol{\theta}_{i+1})}{\pi(\boldsymbol{\theta}_i)q(\boldsymbol{\theta}_{i+1}|\boldsymbol{\theta}_i)}\right\} \tag{2.12}$$

Where, π represents the target distribution. In the Bayesian approach, π is given by the posterior distribution $P(\boldsymbol{\theta}|\mathbf{D})$ (Equation 2.10) and as a result, the acceptance probability can be expressed as follows:

$$\alpha = \min\left\{1, \frac{\frac{L(\boldsymbol{\theta}_{i+1}|\mathbf{D})P(\boldsymbol{\theta}_{i+1})}{\int L(\boldsymbol{\theta}|\mathbf{D})P(\boldsymbol{\theta})\mathbf{d}\boldsymbol{\theta}}\ q(\boldsymbol{\theta}_i|\boldsymbol{\theta}_{i+1})}{\frac{L(\boldsymbol{\theta}_i|\mathbf{D})P(\boldsymbol{\theta}_i)}{\int L(\boldsymbol{\theta}|\mathbf{D})P(\boldsymbol{\theta})\mathbf{d}\boldsymbol{\theta}}\ q(\boldsymbol{\theta}_{i+1}|\boldsymbol{\theta}_i)}\right\} \tag{2.13}$$

If the candidate $\boldsymbol{\theta}_{i+1}$ is rejected, the chain will remain at its last value $\boldsymbol{\theta}_i$. Clearly, if $\boldsymbol{\theta}_{i+1}$ has a higher probability in the target distribution it will always be selected. There are several other sampling strategies to formulate the Markov Chain, e.g. Importance Sampler (Tierney et al., 1994), Gibbs Sampler (Gilks et al., 1996) and Random-Walk Metropolis (Gustafson et al., 1998). However, comparing these techniques is beyond the scope of this discussion. Instead, the key point is that for nonlinear models, the Bayesian based distributions are more accurate and in the absence of an analytical expression for the posterior, the sampling strategies have to be used where the M-H sampler is one of the methods.

2.2 Uncertainty Propagation

Once the description of parametric uncertainty is obtained, the next goal is to propagate its effect into the desired output that, in this thesis, is the optimization objective function. This effect is usually measured in terms of some statistical measure such as a mean and/or variance of the optimization objective. If ϕ is a desired objective with probability p(ϕ), the mean E and variance V are given by;

$$E(\phi) = \int \phi p(\phi)\, d\phi \tag{2.14}$$

$$V(\phi) = \int \left(\phi - E(\phi)\right)^2 p(\phi)\, d\phi \tag{2.15}$$

2.2.1 Taylor Series Expansion

The simplest and the most straightforward approach to calculate the above mean and variance is based on model linearization where the desired output is expanded using a first-order Taylor Series expansion around the parameter estimates as follows:

$$\phi = L\boldsymbol{\theta} \tag{2.16}$$

Where, L is the vector of derivatives, defined as: $L_i = \left.\frac{\partial \phi}{\partial \theta_i}\right|_{\tilde{\boldsymbol{\theta}}}$

When there is an analytical expression for parametric uncertainty, as is the case in standard linearization approach (Equation (2.7)-(2.8)), the above linear approximation (Equation 2.16) also results in an analytical expression for p(ϕ) as follows:

$$p(\phi) = \frac{1}{V_\phi^{1/2}\sqrt{2\pi}} \exp\left(-\left(\phi - \tilde{\phi}\right)^2 / 2V_\phi\right) \tag{2.17}$$

Where, $\tilde{\phi}$ is the mean value corresponding to the nominal parameter estimates $\widetilde{\boldsymbol{\theta}}$ and V_{ϕ} is the variance given by:

$$V_{\phi} = LV_{\boldsymbol{\theta}}L^{T} \tag{2.18}$$

Since the mean and variance have analytical expressions, this approach could be a reasonable choice for the robust optimization problem. However, as mentioned in the previous sub-section, the above parametric uncertainty is only valid if the level of measurement noise and disturbances is such that the underlying linear assumption is valid over the uncertain parameter space. This problem could be addressed by using an uncertainty description based on the Bayesian approach (Equation 2.10). However, since for nonlinear models the posterior distribution does not have an analytical form, it is no longer possible to obtain analytical expressions for any statistical measure of the output (e.g. Equations (2.14)-(2.15)).

Another major limitation is related to the linearization of desired output (Equation 2.16). If the optimization objective is nonlinear in parameters, this approach will provide inaccurate estimates for statistical measures irrespective of the description for parametric uncertainty. Accordingly, the robust optimal solution will also be inaccurate.

To summarize, if the assumption of linearity is valid in both steps, i.e. (1) the estimation of uncertainty using Equations (2.7)-(2.8) and (2) the approximation of optimization objective using Equation (2.16), the first-order Taylor series approach will provide a good approximation with the least computational efforts.

2.2.2 Monte Carlo based Uncertainty Analysis

For nonlinear models, a more generic approach is to use Monte Carlo simulations. In the standard MC approach, a large number of parameters are randomly sampled from their joint distribution and then, for each one of these sampled parameter values, the desired output is calculated by simulating the full nonlinear model. The estimates for the mean and variance in Equations (2.14)-(2.15) are approximated using Monte Carlo integration as follows:

$$E(\phi) = \frac{1}{N}\sum_{i=1}^{N} \phi_i \tag{2.19}$$

$$V(\phi) = \frac{1}{N}\sum_{i=1}^{N} \left(\phi_i - E(\phi)\right)^2 \tag{2.20}$$

According to the law of large numbers, the accuracy of these estimates depends on the number of samples, N. This is one of the major bottlenecks in the implementation of this approach within a robust optimization framework. Since the optimization search requires the calculation of the above metrics in each function evaluation, the overall calculations can become computationally prohibitive. Moreover, when the distribution has long tails, sampling from low probability regions becomes even more challenging and if the parameter values within these tails have significant effect on the output variable, ignoring these regions may result in inaccurate solutions.

Although there are sampling techniques that are more efficient and can improve the above estimates, the number of samples is still very high thus limiting their use in the optimization. Perhaps a very high computational demand is one of the reasons that, despite its many advantages, the Bayesian distributions have not received significant attention in the area of optimization. Here, one possibility would be to use higher order approximations of the model, e.g. second-order Taylor series expansions, but the question remains whether these approximations are valid over the uncertain

parameter space. Another alternative is to use Polynomial Chaos (or PC) expansions and is reviewed next.

2.2.3 Polynomial Chaos Expansions

In a PC expansion, the underlying idea is to project a given random variable on a space of independent random variables, defined by one of the standard distributions.

Let us define $\{\xi_i(\omega)\}_{i=1}^{\infty}$ as a set of independent random variables with some probability *P*. Then, if *X* is a random variable on the same probability space and has a finite variance, it can be expanded in terms of $\{\xi_i(\omega)\}_{i=1}^{\infty}$ as follows:

$$X(\omega) = a_0\Gamma_o + \sum_{i_1=1}^{\infty} a_{i_1}\Gamma_1(\xi_{i_1}) + \sum_{i_1=1}^{\infty}\sum_{i_2=1}^{i_1} a_{i_1 i_2}\Gamma_2(\xi_{i_1},\xi_{i_2}) + \sum_{i_1=1}^{\infty}\sum_{i_2=1}^{i_1}\sum_{i_3=1}^{i_2} a_{i_1 i_2 i_3}\Gamma_3(\xi_{i_1},\xi_{i_2},\xi_{i_3}) + \cdots \tag{2.21}$$

Where, Γ_p is a Polynomial Chaos of order *p*, ω is a random event and $a_{(.)}$ is a deterministic constant. A more compact representation is as follows:

$$X(\omega) = \sum_{k=0}^{\infty} \hat{a}_k \Psi_k(\xi_1, \xi_2, \dots) \tag{2.22}$$

Where, there is a one-to-one correspondence between the functionals and the coefficients of (2.21) and (2.22).

The key property of these expansions is that all basis functions $\{\Psi_k\}_{k=0}^{\infty}$ are orthogonal to each other or in other words;

$$\langle \Psi_i \Psi_j \rangle = \int \Psi_i(\boldsymbol{\xi})\Psi_j(\boldsymbol{\xi})P(\boldsymbol{\xi})\mathrm{d}\boldsymbol{\xi} = \delta_{ij}\langle \Psi_i^2 \rangle \tag{2.23}$$

Based on this property of basis functions, the coefficients $\hat{a}_k$ can be calculated using Galerkin projection as follows:

$$\hat{a}_k = \frac{\langle \mathrm{X}\, \Psi_k \rangle}{\langle \Psi_k^2 \rangle} = \frac{\int X\, \Psi_k\, P(\boldsymbol{\xi})\, \mathrm{d}\boldsymbol{\xi}}{\int \Psi_k^2\, P(\boldsymbol{\xi})\, \mathrm{d}\boldsymbol{\xi}} \tag{2.24}$$

In the pioneering work of Ghanem et al. (1991), the independent random variables were considered to be normally distributed and, therefore, the corresponding basis functions were Hermite Polynomials. When *X* is a normally distributed random variable, the expansion requires only the first two terms where the coefficient of the first term will represent the mean of *X* and the second coefficient will represent the variance. On the other hand, if the *X* has a different distribution, the expansion will need several higher orders terms depending on the random behavior. In general, these expansions, also termed as Hermite-Chaos, can be used to represent both the Gaussian and the non-Gaussian random variables. In a subsequent study by Xiu et al. (2002), the authors illustrated that a PC expansion will have an optimal rate of convergence if the independent random variables also belong to the same family of distributions as the random variable *X*. Accordingly, the authors generalized these expansions by proposing a broader class of orthogonal polynomials from the Askey Scheme. If the *X* belongs to any standard random variable in this scheme, the corresponding orthogonal polynomials can be used to formulate the expansion with minimum number of terms. In case there is no exact match in the scheme, the polynomials from the closest family can be used.

Over the past two decades, the PC expansions have been applied extensively in the area of computational fluid dynamics (Najm, 2009; Knio et al. 2006), structural mechanics (Hosder et al., 2006; Ghanem et al., 1991; Ghanem, 1998a, 1998b; Ghanem et al., 1998) and applied mathematics (Xiu et al., 2002b; Xiu et al., 2005; Xiu, 2007). Recently, these expansions have also been introduced in the area of optimal control (Nagy et al., 2007).

For PC-based uncertainty propagation, the idea is to expand both the uncertain parameters and the resulting uncertain output in terms of the same set of independent random variables. The first step is to expand the parameters. If the parameters have standard distribution, e.g. normal as obtained by the linearization approach (Equations (2.7)-(2.8)), they can be expanded in terms of normally distributed independent variables without any further computation as the first coefficient will be the mean and the second coefficient will be the variance of parameters. However, if the parameters conform to some non-standard distribution, a mapping between the parameters and the independent random variables is required, which can be obtained by transforming both variables to a uniformly distributed probability space (Xiu et al., 2002b) as follows;

$$u = \int_0^{\xi} p(\xi_i) d\xi = \int_0^{\theta} p(\theta_i) d\theta \tag{2.25}$$

Once this map is available, the coefficients of the expansion can be calculated using Equation (2.24).

The next step is to expand the desired output with respect to the independent random variables used in the previous step. In literature, there are two methodologies for this task referred to as intrusive and non-intrusive. In the intrusive approach (Xiu et al., 2002a, 2003; Najm et al., 2009), the governing equations are modified by replacing the parameters and the desired output by their PC expansions. Then, using Galerkin projections, these equations are converted into a set of equations for each coefficient of PC expansion of the output. Let us consider the following example (Xiu et al., 2003):

$$\frac{dx}{dt} = -\theta x \tag{2.26}$$

Where, θ is uncertain with some distribution.

Then, the task is to propagate the uncertainty in θ onto the x. Since both θ and x are random variables, they can be expanded as follows:

$$\theta = \sum_{k=0}^{\mathrm{R}} \hat{\theta}_k \Psi_k(\xi_1) \tag{2.27}$$

$$x = \sum_{l=0}^{\mathrm{R}} \hat{x}_l \Psi_l(\xi_1) \tag{2.28}$$

Then, the first step in the intrusive approach is to substitute these expansions in the governing equation (Equation (2.26)) as follows:

$$\sum_{l=0}^{\mathrm{R}} \frac{d\hat{x}_l}{dt} \Psi_l(\xi_1) = -\sum_{k=0}^{R} \sum_{l=0}^{\mathrm{R}} \hat{\theta}_k \hat{x}_l \Psi_k(\xi_1)\, \Psi_l(\xi_1) \tag{2.29}$$

The next step is to perform Galerkin projections onto each basis function:

$$\begin{aligned} &\left\langle \left(\sum_{l=0}^{\mathrm{R}} \frac{d\hat{x}_l}{dt} \Psi_l(\xi_1) \right), \Psi_m(\xi_1) \right\rangle \\ &= \left\langle -\left(\sum_{k=0}^{R} \sum_{l=0}^{\mathrm{R}} \hat{\theta}_k \hat{x}_l \Psi_k(\xi_1)\, \Psi_l(\xi_1) \right), \Psi_m(\xi_1) \right\rangle \end{aligned} \tag{2.30}$$

Since $\langle \Psi_l(\xi_1), \Psi_m(\xi_1) \rangle = 0$ for $m \neq l$, the LHS in the above expression reduces to a single term corresponding to the basis function $\Psi_m(\xi_1)$ and by repeating this operation for $m = 0,1, \dots, R$, the above expression is converted into a set of differential equations for coefficients $\hat{x}_m$ as follows:

$$\dot{x}_m = -\frac{1}{\langle \Psi_m(\xi_1)^2 \rangle} \sum_{k=0}^{\mathrm{R}} \sum_{l=0}^{\mathrm{R}} \hat{\theta}_k \hat{x}_l \langle \Psi_k(\xi_1) \Psi_l(\xi_1) \Psi_m(\xi_1) \rangle \tag{2.31}$$

The major challenge in the above approach is whether the governing equations can be modified for applying the Galerkin projections, which may not be possible for complex nonlinear models. As an alternative, in the non-intrusive approach, the model is used as a black box to generate a mapping between the desired output and the set of independent random variables and, then, the coefficients of the expansion are calculated using Equation (2.24) (Hosder et al., 2006; Xiu, 2007; Xiu et al., 2005).

To summarize the procedure:

1. Obtain the values of independent random variables corresponding to a quadrature rule to be used for calculating the integrals in Equation (2.24).

2. Calculate the corresponding values of the parameter from its PC expansion (Equation (2.27)).

3. Solve the model equation (Equation 2.26) to calculate the output for each realization of parameter values.

4. Use the set of independent random variables from Step 1 and the outputs from Step 3 and calculate the coefficients of PC expansion for the output using Equation (2.24).

Since there is no restriction on the type of governing equations, this approach can be applied to a wide range of applications. However, it is quite clear that the accuracy of the coefficients depends on the quadrature rule used in the numerical integrations (Equation (2.24)) and this has been one of the key areas of research in this class of algorithms. In one of the earlier methods (Tatang et al., 1997 and Pan et al., 1997), the selection of collocations points was based on the roots of orthogonal polynomials with a degree higher than the corresponding basis functions. This approach was later refined by Isukapalli et al. (1999) where the authors improved the selection of collocation points, especially in the regions where the independent random variables have higher probability of occurrence. When it comes to multidimensional integrals,

the standard approach of taking a full tensor product of one-dimensional quadrature rule may become computational prohibitive. The reason is that the number of collocation points in the tensor rule grows exponentially with the dimensions. To this end, an alternative approach, often implemented in non-intrusive literature, is to use sparse grids as originally proposed by Smolyak in 1963. It has been shown that the collocation points based on the sparse grid quadrature rules have weak dependency on the number of dimensions, thus making them suitable for large multidimensional integrals.

Besides the selection of collocation points, the nonlinearity of the integrand is also a major factor. The standard quadrature rules assume the integrand to be a smooth function. However, when there is a bifurcation or discontinuity in this function, these methods fail to give accurate estimates for the coefficients. This problem can be addressed by formulating multi-element PC representations (Le Maitre et al., 2004a, 2004b, 2007; Wan et al., 2005) where the space of random variables is divided into sub-regions in which the integrand is relatively smoother and these regions are then approximated by individual PC expansions. The procedure is repeated until no further divisions are required.

The most important property that makes the PC expansions especially suitable for repetitive calculations is that, irrespective of uncertainty descriptions, the mean and variance can be calculated analytically as follows:

$$E(X) = \hat{a}_0 \tag{2.32}$$

$$V(X) = \sum_{k=1}^{\mathrm{P}} \hat{a}_k \langle \Psi_k^2 \rangle \tag{2.33}$$

This is clearly a significant advantage as compared to the both Taylor expansions and Monte Carlo methods, discussed in the previous sub-sections. The first-order Taylor series expansion can give the analytical expressions only if model is linear

whereas for Monte Carlo methods, there are no analytical expressions. It is worth noticing that the PC approximations are accurate over the uncertain space and therefore, they can be applied to any nonlinear problem efficiently.

Another area where the PC expansions have seen a major application is Bayesian Inference (Marzouk et al., 2007; Ma et al., 2009 & Balakrishnan et al., 2003). Here, the basic idea is to propagate the prior uncertainty in parameters onto the model outputs. Then, by using the PC expansions instead of full model, the computation of the likelihood function becomes much faster, speeding up the overall sampling significantly. In the studies reported so far, the expansions are formulated over the entire parameter space defined by the prior uncertainty. When the model is highly nonlinear, this requires a large number of collocation points to estimate the coefficients accurately. It is quite possible that, in the posterior distribution, only a part of this parameter space has higher probability and therefore, any accuracy in the approximation outside this partial region is of lesser importance. In fact, if the information about the posterior distribution can be used in formulating the approximation, the collocation points can be relocated to the parameter regions of higher posterior probability. To this end, a novel adaptive approach is presented in this thesis and will be discussed in Chapter 3.

2.3 Optimization under Uncertainty

When a process is optimized with an uncertain model, it is quite possible that the predicted optimal solution may have a significant variability and, in a worst-case scenario, there could be a significant bias between the predicted and the actual process optimum. In such scenario, one must implement appropriate strategies to improve the accuracy of the model-based optimal solution.

One of the strategies is robust optimization where the goal is to search for optimal solutions where the effect of model uncertainties is the minimum (Beyer et al., 2007;

Samsatli et al., 1998; Diwekar et al, 1996; Srinivasan et al., 2003; Nagy et al., 2004). In a standard optimization, also referred to as "nominal optimization", the objective function is calculated using the nominal parameter values that are already uncertain. Instead, in the robust optimization, the objective function is calculated over the entire uncertain parameter space and then, some combination of statistical measures is minimized. The most common cost function is a weighted sum of expectation and variance, to provide a trade-off between the maximum performance and the robustness. If there are any constraints to be considered, they can be formulated either as the worst-case or in a probabilistic sense where one can define a probability for a constraint violation. Either way, the feasibility can be guaranteed at-least within the measured uncertainty. The major limitations in this class of algorithms are related to the uncertainty quantification and propagation steps, as reviewed in the previous sub-sections. In this thesis, these limitations are addressed by using a PC based methodology to propagate the Bayesian description of parametric uncertainty.

Although the robust approach can result in an optimal solution with a minimum variability, this solution could be very conservative if the model uncertainties are quite significant. It is worth mentioning here that the robust approach does not attempt to reduce the bias between the predicted and actual process optimum, for which case an iterative approach has to be adopted. However, a robust approach can be combined with iterative algorithms for minimizing the bias as well as increasing the robustness of the solution (Zhang et al., 2002).

When new measurements can be obtained, they can be used to formulate an iterative procedure where it is expected that, by continuously updating and re-optimizing the model, it might be possible to eliminate the bias between the predicted and the actual process optimum. When the model is updated as soon as the measurements are collected, the approach falls within the category of real-time optimization (RTO) and it is widely implemented in the industry as a supervisory control layer. In a batch process, the update step can be carried out at the end of

batch, in which case the approach is often referred to as "run-to-run" optimization. However, in both applications, the standard approach is the same whereby in the first step, the model parameters are estimated using new measurements at previously calculated optimal solution and then, the updated model is optimized for the next iteration (Ruppen et al., 1998; Chen et al., 1987; Marlin et al., 1997; Astrom, 1970). This two-step procedure is repeated until the algorithm converges. Mathematically, these steps can be expressed as follows:

$$\boldsymbol{\theta}_k = \arg\min_{\boldsymbol{\theta}} \varphi\,(\mathbf{u}_k, \boldsymbol{\theta})$$

$$\mathbf{u}_{k+1} = \arg\min_{\mathbf{u}} \phi\,(\mathbf{u}, \boldsymbol{\theta}_k) \qquad \text{(P.2.1)}$$

Where,φ and ϕ are the objective functions corresponding to the parameter estimation and optimization problems respectively and k is the iteration number.

In this approach, the convergence to a process optimum can be guaranteed if; (1) the optimal solutions during the iterative search provide sufficient excitation and (2) there is no model-plant mismatch. The first requirement for the convergence can be addressed by incorporating an optimal design of experiments' criterion within the optimization objective. In this way, it is possible to obtain a trade-off between the optimal conditions and the operating conditions that provide sufficient excitation for the next identification. However, this topic is beyond the scope of this thesis and will not be discussed further.

Regarding the second requirement for the convergence, model-plant mismatch is inevitable in almost all practical situations and addressing this mismatch is a major objective in this thesis. When the model has inaccurate structure, there is a lack of synergy between the identification and optimization objectives. The reason is that the parameters that minimize the prediction error at a given operating conditions may not minimize the error in neighboring conditions. As a result, the model may predict the gradients of the optimization objective inaccurately.

In model identification, a given model is said to be adequate if it is possible to estimate the model parameters that can predict the available measurements with a minimum possible error. Similarly, for a combined identification and optimization problem, the model is considered adequate if there is a set of parameter estimates that can satisfy both identification and optimization objectives at the process optimum. In the context of real-time optimization (RTO), Forbes et al., (1994, 1996) proposed a following test for Model Adequacy:

If $\mathbf{u}^*$ is the unique process optimum and there exists at least one set of parameter values $\widehat{\boldsymbol{\theta}}$ for which;

$$\nabla_{\mathbf{u}}\phi(\mathbf{u},\widehat{\boldsymbol{\theta}})\big|_{\mathbf{u}^*} = 0 \tag{2.34a}$$

$$\nabla_{\boldsymbol{\theta}}\varphi(\mathbf{u},\widehat{\boldsymbol{\theta}})\big|_{\mathbf{u}^*} = 0 \tag{2.34b}$$

$$\nabla^2_{\mathbf{u}}\phi(\mathbf{u},\widehat{\boldsymbol{\theta}})\big|_{\mathbf{u}^*} \text{ is positive definite} \tag{2.34c}$$

$$\nabla^2_{\boldsymbol{\theta}}\varphi(\mathbf{u},\widehat{\boldsymbol{\theta}})\big|_{\mathbf{u}^*} \text{ is positive definite} \tag{2.34d}$$

Then, the model is adequate.

Now, if the given model is inadequate, the optimization strategies have to be modified such that the model-based optimization search will be forced to proceed towards the process optimum. For convergence to a process optimum, it is necessary that the predicted optimality conditions, also known as Karush-Kuhn-Tucker or KKT conditions, should match the ones measured from the process (Beigler et al., 1985). In one of their earlier studies, Robert et al. (1979) introduced an integrated approach where the optimization objective was augmented with an additional term that cancels out the differences between the predicted and measured gradients of the optimization objectives for the current iteration. The augmented objective function in this approach is as follows:

$$\mathbf{u_{k+1}} = \min_{\mathbf{u}} \phi(\mathbf{u}, \mathbf{y}, \boldsymbol{\theta}_k) - \boldsymbol{\lambda}_k^{\mathrm{T}}\mathbf{u}$$

With,

$$\boldsymbol{\lambda}_k = [\nabla_{\mathbf{u}}\mathbf{y_m} - \nabla_{\mathbf{u}}\mathbf{y}]\nabla_{\mathbf{y}}\phi(\boldsymbol{\theta}_k, \mathbf{u_k}) \quad \text{(P.2.2)}$$

Where, $\nabla_{\mathbf{u}}\mathbf{y_m}$ and $\nabla_{\mathbf{u}}\mathbf{y_m}$ are the gradients of measured and predicted outputs respectively. In this approach, the correction was introduced only in the objective function. Since then, there have been several extensions to this idea, involving different ways in which modifications can be implemented, to address a wide range of problems (Tatjewski et al., 2002; Gao et al., 2005; Chachuat et al., 2009; Marchetti et al., 2009 and Costello et al., 2011). These extensions also differ in a way the model update step is implemented. For example, while Robert et al. (1979) always updated the model parameters to satisfy the identification objectives, Tatjewski et al. (2002) and Gao et al., (2005) replaced the parameter estimation step by a linear correction to the outputs that accounts for the prediction error. Later on, in the studies by Chachuat et al., 2009; Marchetti et al., 2009 and Costello et al., 2011, the authors eliminated the model update step altogether. These latter studies argued that, since the convergence to a process optimum depends only on the ability of a model to predict the gradients of the optimization problem accurately, eliminating the model update step has no effect on the convergence. However, following the elimination of this step, the model can no longer be used for predictions around the optimum and this may have a significant importance in many practical applications. It should also be noted, here, that the corrections in the optimization problem have to be filtered in order to ensure the convergence to a unique solution. Without enough filtering, the corrections are often too aggressive and may result in oscillatory convergence.

Another key point in these algorithms is that the corrections are external to the model. In other words, the model parameters are never updated with respect to the measured gradients of the optimization problem. Even when the parameters are updated in some algorithms, the update is with respect to the prediction error. In this

thesis, a novel approach is developed where the goal is to correct the model for its structural uncertainty in a way that, upon convergence, it satisfies both identification and optimization objectives. In other words, the final corrected model should satisfy the adequacy conditions in Equations ((2.34a)-(2.34d)) It is also shown that the proposed approach eliminates the need for an external filter as the corrected model itself provides for much improved filtering properties.

Chapter 3
Robust Optimization of Chemical Processes using Bayesian description of parametric uncertainty[1]

(Published in Journal of Process Control)

3.1 Introduction

Model-based optimization methodologies rely primarily on the accuracy of the model used to predict outputs over the entire space of operating conditions. Any uncertainty in the model, if not accounted for, may result in non-optimal operating policies which may lead to significant loss in the economic objectives or even violations of environmental and safety constraints. Due to either noise in measurements or model structure error, the model parameters are always uncertain. In that case, it becomes very important to quantify the effect of the associated parametric uncertainty on the optimization objectives and if this effect is significant, it is necessary to either reduce the uncertainty in parameters by changing the model structure or obtaining additional data, or to search for an optimum that is robust to these uncertainties.

The current study considers the problem of finding an optimal solution that is robust to parametric uncertainties assuming that additional data, beyond a limited initially available set, are either very difficult to obtain or not available to further reduce the uncertainty. Although the optimization problem considered in this study is off-line, the proposed methodology can also be applied to online optimization problems where, as new measurements become available, the re-estimation of uncertainty description and/or re-estimation of the robust optimal solution is required

[1] Adapted from Mandur et al., 2013a (License Number: 3370961239953)

such as Robust Real-time Optimization and Robust Nonlinear Model Predictive Control.

In contrast to nominal optimization, where the objective function calculated at the nominal values of parameters is minimized, in robust optimization some statistical metrics of the objective function calculated over the uncertain parameter space is minimized instead (Diwekar et al., 1996; Samsatli et al., 1998; Nagy et al., 2004; Beyer et al., 2007). A typical robust optimization formulation involves the minimization of a weighted sum of both the expected value of the cost and its variance due to uncertainty, thus providing a trade-off between maximum performance and robustness as follows:

$$\min_{\mathrm{u}} \quad W_1 * E\big(f(\mathbf{x},\mathbf{u},\boldsymbol{\theta})\big) + W_2 * Var\big(f(\mathbf{x},\mathbf{u},\boldsymbol{\theta})\big)$$

$$s.t. \quad \mathbf{g}(\mathbf{x},\mathbf{u},\boldsymbol{\theta}) \leq \mathbf{0} \tag{3.1}$$

Where, $\boldsymbol{\theta}$ is a vector of uncertain model parameters, u is a vector of decision variables, x is a vector of states, f is an objective function, g is a vector of equalities or inequalities which includes the model equations and additional process limits, E is the expected value of the objective function f, Var is the variability in the objective function f and W_1 and W_2 are weights that are problem specific.

The calculation of any statistical metrics to be performed in the robust optimization framework, involves integrals related to the calculation of the functions E and Var in Equation (3.1) which generally do not have analytical solutions. The most common approach is to approximate the objective function f in Equation (3.1) by either a first order or second order Taylor Series Expansion around the nominal parameter values (Nagy et al., 2004; Darlington et al., 1999). Although these approximations work well when the uncertainty in the parameters is not too large and the objective function is nearly linear or quadratic, for most nonlinear processes, these assumptions are not valid. For general nonlinear cases, an alternative is to use a Monte Carlo approach

where the parameter values are randomly selected from the joint probability distribution and then the corresponding objective function is calculated (Diwekar et al, 1996; Beyer et al., 2007). However, in this approach, a large number of samples are required for obtaining an accurate estimate of the above integrals. Thus, when using Monte Carlo techniques, the need for extensive sampling combined with the fact that the cost function in Equation (3.1) has to be computed at each function evaluation during an iterative search for an optimum, results in a large computational burden.

In recent years, uncertainty propagation using Polynomial Chaos (PC) expansions has been studied by several authors from different areas (Najm, 2009; Knio et al., 2006; Nagy et al., 2007; Xiu et al., 2002, 2003; Ghanem et al., 1991) and has been shown to be much more efficient and accurate when compared to Monte Carlo sampling approaches. The two major advantages of the PC expansions are that they can be used to propagate any complex probability distribution into the desired output and that the mean, variance and any other higher order moments can be calculated analytically. Although few studies have also implemented the PC expansions within the robust optimization framework (Molina-Cristobal et al., 2006; Xiong et al., 2011), in these studies, very simplistic descriptions of uncertainty, e.g. normal or exponential distributions, were used. However, for nonlinear processes, such descriptions often fail to provide accurate results. Instead, in this work, we considered more accurate description of the uncertainty by using the Bayes' Theorem which gives a probability distribution instead of a point estimate. A preliminary description of the proposed approach was recently presented by the authors in Mandur et al. (2012). In the Bayesian approach, the posterior distribution of the parameters is proportional to a likelihood which is a function of the errors between the measurements and the model predictions and a prior probability which represents any information about the parameters available prior to collecting the data. The calculation of the likelihood function is generally a major time consuming step because it requires multiple simulations of the nonlinear dynamic model. Thus, the use of some reduced form or

an approximation of the process models can result in significant reduction of the computational time (Balakrishnan et al., 2003; Marzouk et al., 2007). In the previous studies, the approximations based on PC expansions were proposed that describe the model over the entire uncertain parameter space. The number of polynomials in these expansions depends on the degree of non-linearity of the model with respect to parameters. However, in the parameter space, there could be some regions where the degree of nonlinearity is relatively less and as a result, lower order polynomials can be used in these regions. In addition, there can also be some regions where the posterior probability of the parameters is low and in such regions, there is no need to achieve higher degree of accuracy in the approximation. Motivated by these two scenarios, in this work an adaptive approach based on multi-resolution analysis (MRA) is proposed which progressively decomposes the parameter space into sub-regions, where the change in posterior probability is above a pre-specified threshold value. The proposed approach results in more model runs in the parameter region with relatively higher posterior probability. After obtaining the posterior probability of the parameters, an approach based on PC expansions is used to propagate this uncertainty into the objective function of the optimization problem (Mandur et al., 2012). The results are compared with those obtained when a normal representation of uncertainty is used.

The rest of the chapter is organized as follows. Section 3.2 presents the background and the overall methodology. Section 3.3 illustrates the proposed methodology for a penicillin manufacturing process followed by Section 3.4 with summary and conclusions.

3.2 Background and methodology

In this section, we will begin with a brief description of the mathematical tools used in the methodology.

3.2.1 Polynomial Chaos Expansions

Consider a probability space defined by $(\Omega, \mathcal{F}, P)$, where Ω is the sample space, $\mathcal{F}$ is the σ-algebra over Ω and P is a probability measure on $\mathcal{F}$. If $\{\xi_i(\omega)\}_{i=1}^{\infty}$ is a set of independent random variables with probability measure P, then any random variable X with a finite variance can be expanded as follows:

$$X(\omega) = a_0\Gamma_o + \sum_{i_1=1}^{\infty} a_{i_1}\Gamma_1(\xi_{i_1}) + \sum_{i_1=1}^{\infty}\sum_{i_2=1}^{i_1} a_{i_1 i_2}\Gamma_2(\xi_{i_1}, \xi_{i_2}) + \sum_{i_1=1}^{\infty}\sum_{i_2=1}^{i_1}\sum_{i_3=1}^{i_2} a_{i_1 i_2 i_3}\Gamma_3(\xi_{i_1}, \xi_{i_2}, \xi_{i_3}) + \cdots \tag{3.2}$$

Where, Γ_r is the PC of order r (Ghanem et al., 1991; Xiu et al., 2002), ω is the random event and $a_{(.)}$ is the deterministic coefficient. The above expansion can further be re-written in a simpler form (Ghanem et al., 1991) as:

$$X(\omega) = \sum_{k=0}^{\infty} \hat{a}_k \Psi_k(\xi_1, \xi_2, \dots) \tag{3.3}$$

Where, there is a one-to-one correspondence between the functions and the coefficients of the above two representations (Equation (3.2)-(3.3)). For computational feasibility, the expansion is considered in a truncated form as follows:

$$X(\omega) = \sum_{k=0}^{\mathrm{N}} \hat{a}_k \Psi_k(\xi_1, \xi_2, \dots, \xi_{\mathrm{n}}) \tag{3.4}$$

Where, n is the total number of independent random variables and N is the total number of terms in the expansion and is given by:

$$N + 1 = \frac{(n + r)!}{n!\, r!} \tag{3.5}$$

An underlying property of a PC expansion is that all basis functions are orthogonal to each other with respect to the probability distribution of independent random variables, $\boldsymbol{\xi}$ and accordingly the following applies:

$$\langle \Psi_i \Psi_j \rangle = \int \Psi_i(\boldsymbol{\xi}) \Psi_j(\boldsymbol{\xi}) p(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} = \delta_{ij} \langle \Psi_i^2 \rangle \tag{3.6}$$

Where, $\langle \cdot \rangle$ is the inner product operator. To satisfy Equation (3.6), different distributions of independent random variables require different orthogonal basis functions. For example, Hermite polynomials are the basis functions for normal random variables, Legendre polynomials for uniform random variables, Laguerre polynomials for gamma random variables and so on (Xiu et al., 2002).

Orthogonality of the basis functions can be used to calculate the k^{th} coefficient by projecting the expansion onto the corresponding basis function as follows:

$$\hat{a}_k = \frac{\langle \mathrm{X}\, \Psi_k \rangle}{\langle \Psi_k^2 \rangle} = \frac{\int X\, \Psi_k\, p(\boldsymbol{\xi})\, \mathrm{d}\boldsymbol{\xi}}{\int \Psi_k^2\, p(\boldsymbol{\xi})\, \mathrm{d}\boldsymbol{\xi}} \tag{3.7}$$

The integrals in Equation (3.7) can then be calculated using quadrature rules.

3.2.2 Bayesian framework

In the Bayesian framework, the model parameters and hence the model outputs are both considered as random variables. Given a set of measurements, the Bayes' Theorem defines the posterior probability of the parameters as follows:

$$p(\boldsymbol{\theta}|D) = \frac{L(\boldsymbol{\theta}|D)\, p(\boldsymbol{\theta})}{\int L(\boldsymbol{\theta}|D)\, p(\boldsymbol{\theta}) d\boldsymbol{\theta}} \tag{3.8}$$

Where, D represents the set of measurements, $L(.)$ is the likelihood of parameters given the measurements and $p(\boldsymbol{\theta})$ is the prior probability of parameters, representing any information about the parameters available *a priori*. In this work, the errors between predictions and measurements are assumed to be independent and normally distributed with zero mean and finite variance. As a result, the likelihood function is given by a *k*-dimensional multivariate normal distribution as follows:

$$L(\boldsymbol{\theta}|D) = \frac{1}{(2\pi)^{k/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}\big(\mathbf{y} - \mathbf{x}(\boldsymbol{\theta})\big)^T \Sigma^{-1}\big(\mathbf{y} - (\boldsymbol{\theta})\big)\right) \tag{3.9}$$

Where, $\mathbf{x}$ and $\mathbf{y}$ are the vectors of model predictions and corresponding measurements respectively and Σ is the covariance matrix.

In calculating the posterior distribution or any integral over it, the major computational time is spent in the calculation of likelihood function which requires repeated simulations of the nonlinear dynamic model for different values of the uncertain parameters. Accordingly, this calculation can be improved significantly if an approximation of the nonlinear model is available which can provide a quick calculation of outputs' predictions as a function of parameters' values. In this work, an adaptive approach based on multi-resolution analysis (MRA) is proposed that progressively refines the approximation of the model in high probability regions of the parameter space. The level of successive refinements depends on the nonlinearity of the model with respect to parameters.

3.2.2.1 Multi-resolution analysis

Let $V_o^{N_o}$ be the subspace of polynomials $\phi^m(x) \in L^2(\mathbf{R}),\ m = 0,1,\dots,N_o - 1$. Then, a subspace $V_k^{N_o}$, where the set of polynomials is obtained by translation and dilation of the polynomials in $V_o^{N_o}$ can be defined as follows:

$$V_k^{N_o} = \{\phi_{k,l}^m(x) :\ \phi_{k,l}^m(x) = \phi^m(2^k x - l), \qquad m = 0,1,\dots,N_o - 1, \quad l = 0,1,\dots,2^k - 1\} \tag{3.10}$$

Based on the above definitions, a multi-resolution approximation for any function $f \in L^2(\mathbf{R})$ is given by the sequence of subspaces $V_k^{N_o}, k = 1,2,3,\dots$ satisfying the following properties:

$$V_0^{N_o} \subset V_2^{N_o} \subset \cdots \subset V_k^{N_o} \subset \cdots$$

$$\bigcup_{k\in\mathbf{Z}} V_k^{N_o} \text{ is dense in } L^2(\mathbf{R}) \tag{3.11}$$

$$\bigcap_{k\in\mathbf{Z}} V_k^{N_o} = \{0\}$$

Accordingly, at a particular resolution level k, any function can be approximated as follows:

$$f_k(x) = \sum_{l=0}^{2^k-1} \sum_{m=0}^{N_o} a_{m,l}\, \phi_{k,l}^m(x) \tag{3.12}$$

We consider orthonormal Legendre Polynomials, rescaled over $[0,1]$, as a set of basis functions for $V_o^{N_o}$ with the restriction that the polynomials vanish outside $[0,1]$. Therefore, the polynomials in successive subspaces $\phi_{k,l}^m(x)$ are defined only in the interval $[2^{-k}l, 2^{-k}(l+1)]$. Since due to orthonormality of $\phi^m(x)$;

$$\langle \phi_{k,l}^{m}, \phi_{k',l'}^{m'} \rangle = \delta_{kk'}\delta_{ll'}\delta_{mm'} \tag{3.13}$$

The coefficients of the expansion (Equation (3.12)) can then be calculated by projection as follows:

$$a_{m,l} = \langle f_k, \phi_{k,l}^{m} \rangle \tag{3.14}$$

It should also be noted that a particular polynomial $\phi_{k,l}^{m}(x)$ approximate the function only in $[2^{-k}l, 2^{-k}(l+1)]$ and at the subsequent higher resolution $k+1$, this region decomposes into two halves where each half can be then approximated by appropriate polynomials from $V_{k+1}^{N_o}$. Thus, in different regions of the parameter space, a function can be approximated by polynomials from different subspaces depending on the accuracy required in that region. Based on the above, an iterative procedure is proposed to refine the approximation only in the parameter regions where the change in posterior probability between the successive approximations is above some pre-specified threshold.

The comparison of the probability distribution from two successive approximations is based on the Kullback-Leibler (K-L) divergence (Cover et al., 1991), given by:

$$D(P_{k+1}\,||P_k) = \int P_{k+1}(\boldsymbol{\theta}|D)\log\left(\frac{P_{k+1}(\boldsymbol{\theta}|D)}{P_k(\boldsymbol{\theta}|D)}\right)d\boldsymbol{\theta} \tag{3.15}$$

$$= \int \frac{L_{k+1}(\boldsymbol{\theta}|D)\,p(\boldsymbol{\theta})}{\int L_{k+1}(\boldsymbol{\theta}|D)\,p(\boldsymbol{\theta})d\boldsymbol{\theta}}\log\left(\frac{\frac{L_{k+1}(\boldsymbol{\theta}|D)}{\int L_{k+1}(\boldsymbol{\theta}|D)\,p(\boldsymbol{\theta})d\boldsymbol{\theta}}}{\frac{L_k(\boldsymbol{\theta}|D)}{\int L_k(\boldsymbol{\theta}|D)\,p(\boldsymbol{\theta})d\boldsymbol{\theta}}}\right)d\boldsymbol{\theta} \tag{3.16}$$

$$\approx \sum_{i=n}^{N}\frac{L_{k+1}(\boldsymbol{\theta}|D)}{\sum_{\mathrm{i=n}}^{\mathrm{N}} L_{k+1}(\boldsymbol{\theta}|D)}\log\left(\frac{\frac{L_{k+1}(\boldsymbol{\theta}|D)}{\sum_{\mathrm{i=n}}^{\mathrm{N}} L_{k+1}(\boldsymbol{\theta}|D)}}{\frac{L_k(\boldsymbol{\theta}|D)}{\sum_{\mathrm{i=n}}^{\mathrm{N}} L_k(\boldsymbol{\theta}|D)}}\right) \tag{3.17}$$

The iterative procedure to approximate the nonlinear dynamic model is as follows:

1. Transform the uncertain model parameters, based on prior probability distribution, into uniformly distributed random variables defined over the interval [0,1].

2. Obtain two separate approximations of the model outputs as a function of transformed random variables first using the basis functions from subspace $V_o^{N_o}$ and then from subspace $V_1^{N_o}$. The first approximation is basically a polynomial chaos representation, where the approximation is over the entire parameter space. The second approximation is more refined as it approximates the sub-regions of parameter space separately using locally supported basis functions.

3. Calculate the posterior distribution using both the approximations and compare them using K-L divergence in sub-regions defined by $V_1^{N_o}$ subspace. Since the polynomial approximations are used to calculate the likelihood, the required expression in Equation (3.17) is calculated very quickly.

4. Approximate the sub-regions where the difference is above a specified threshold using the basis functions from subspace $V_2^{N_o}$.

5. Repeat step 3 and 4 until the difference between the posterior distributions from last two successive approximations is less than the pre-specified threshold.

3.2.3 Uncertainty Propagation using PC expansions

Once the posterior distribution of the parameters $\boldsymbol{\theta}$ is obtained, the next step is to propagate this uncertainty into the objective function f. In the PC framework, the

simplest approach would be to formulate a PC expansion directly relating the objective function to the parameters. Here, the first step would be to obtain a map between the parameters and the corresponding objective function by solving the model for different realizations of parameters. Then using this map and Equation (3.7), the coefficients for different basis functions, to be used in the expansion, could be calculated. However, this approach would require the parameters to be independent and to have a probability distribution corresponding to the orthogonal basis functions being used. However the parameters may not meet these conditions, as in this work. In that case, appropriate transformations must be carried out and the PC expansion for the objective function is then formulated in terms of transformed random variables as follows.

For the one parameter case, the transformation is as follows:

$$\xi = F^{-1}\left(\int_0^{\theta} p(\theta|D)d\theta\right) \tag{3.18}$$

Where, F^{-1} is the inverse cumulative density function for the independent random variable ξ and $p(\theta|D)$ is the posterior probability of the model parameter θ calculated by Equation (3.8).

For more than one parameter, if the parameters are uncorrelated, the transformations are straightforward where each model parameter can be transformed into a separate independent random variable according to Equation (3.18). However, for the case of correlated parameters which is generally the case for model parameters within a system of coupled differential equations, the transformation based on marginal and conditional probabilities can be used. For example, for two correlated parameters, the transformations are as follows:

$$\xi_1 = {F_1}^{-1}\left(\int_0^{\theta_1} p(\theta_1|D)d\theta_1\right) \tag{3.19}$$

$$\xi_2 = {F_2}^{-1}\left(\int_0^{\theta_2} p(\theta_2|\theta_1, D)\, d\theta_2\right) \tag{3.20}$$

Where, $p(\theta_1|D)$ is the marginal posterior probability of parameter θ_1 and $p(\theta_2|\theta_1, \mathrm{D})$ is the posterior probability of parameter θ_2 conditional on parameter θ_1.

Once the transformation of parameters **θ** into a set of independent random variables **ξ** is obtained, the next step is to obtain a map between the objective function f in Equation (3.1) and the set of independent random variables **ξ**. A straightforward method is to select values for parameters **θ** and then calculate for them, the corresponding objective function f by solving the model equations and the values for a set of independent random variables **ξ** using the transformations in Equations ((3.18)-(3.20)). However, this method may result in the values of **ξ** that may not be the same as required to solve the integrals in Equation (3.7) using quadrature rules. To alleviate this problem, a more appropriate approach is to first formulate a PC expansion of the parameters **θ** as a function of the variables **ξ** using transformations Equations ((3.18)-(3.20)); then, select values of independent random variables **ξ** at the required collocation points, calculate the corresponding values of model parameters **θ** from the formulated PC expansions and finally, calculate the corresponding objective function f by solving the nonlinear dynamic model equations for each model parameter value.

For one parameter case, the PC formulation is straightforward where the coefficients can be calculated by Equation (3.7) using the map between model parameter θ and independent random variable ξ, obtained from Equation (3.18). The resulted PC expansion is as follows:

$$\theta = \sum_{i}^{O} \theta_i \Psi_i(\xi) \tag{3.21}$$

For the case of two parameters, first the mapping between the model parameter θ_1 and the corresponding independent random variable ξ_1, given by Equation (3.19) is used to formulate the PC expansion for θ_1 as follows:

$$\theta_1 = \sum_{i}^{O} \theta_{1_i} \Psi_i(\xi_1) \tag{3.22}$$

Similarly, the mapping given by Equation (3.20) is used to formulate the PC expansion for θ_2 as follows:

$$\theta_2(\theta_1) = \sum_{i}^{O} \theta_{2_i}(\theta_1) \Psi_i(\xi_2) \tag{3.23}$$

Here, it should be noted that the above PC expansion for θ_2 is conditional on a particular value of θ_1 and therefore, the different values of θ_1 will give different PC expansions. In order to explicitly incorporate the effect of θ_1 in the above expansion, each of the coefficients θ_{2_i} are further expanded in terms of the independent random variable ξ_1 as follows:

$$\theta_{2_i} = \sum_{k}^{P} \theta_{2_{ik}} \Psi_k(\xi_1) \tag{3.24}$$

Substitution of Equation (2.24) into Equation (3.23) results in one PC expansion for θ_2 which depends on both random variables ξ_1 and ξ_2 as follows:

$$\theta_2 = \sum_{i}^{O} \left(\sum_{k}^{P} \theta_{2_{ik}} \Psi_k(\xi_1) \right) \Psi_i(\xi_2) \tag{3.25}$$

Finally, using the map between f and $\boldsymbol{\xi}$, the PC expansion of the objective function f with respect to the independent random variables is obtained as follows:

$$f = \sum_{i}^{O} \hat{a}_i \Psi_i(\boldsymbol{\xi}) \tag{3.26}$$

The expected value and the variance of the objective function f can then be calculated using the following analytical expressions:

$$E(f) = \hat{a}_0 \tag{3.27}$$

$$V(f) = \sum_{i=1}^{P} \hat{a}_i^2 < \Psi_i^2 > \tag{3.28}$$

Where, $\langle \cdot \rangle$ represents inner product with respect to $\boldsymbol{\xi}$. After substituting the expected value and variance of the objective function *f*, as calculated above, in the cost function of Equation (3.1), the search for the optimum can be performed using appropriate optimization technique. It should be noted that within the optimization problem (Equation 3.1), the methodology given above to represent the objective function with respect to the random variables can also be used to describe any other variable for which constraints have to be satisfied. The overall methodology, to be executed at each function evaluation in the search for the optimum, is summarized for clarity in a stepwise procedure as follows:

1. Obtain the approximation of model outputs as a function of model parameters $\boldsymbol{\theta}$ using multi-resolution approach, as outlined in the stepwise procedure in section 3.2.2.1

2. Calculate the posterior distribution of the parameters $\boldsymbol{\theta}$ from the given data using Equation (3.8) where the likelihood function is calculated using above approximation

3. Transform the model parameters $\boldsymbol{\theta}$ into a set of independent random variables $\boldsymbol{\xi}$ using Equations ((3.18)-(3.20)) and formulate their PC expansions, as given by Equation ((3.21)-(3.25))

4. Select the values of independent random variables $\boldsymbol{\xi}$ at the specific points as required by the quadrature rule to be used

5. Calculate for these $\boldsymbol{\xi}$, the corresponding values of parameters $\boldsymbol{\theta}$ using PC expansions (Equations (3.21)-(3.25))

6. Calculate the values of objective function f for the $\boldsymbol{\theta}$ obtained in the previous step using model equations

7. Formulate a PC expansion for the objective function f and calculate the expected value and the variance using Equations ((3.27)-(3.28)).

3.3 Case Study: Fed-batch bioprocess

3.3.1 Problem formulation

The proposed methodology is applied to a fed-batch process for penicillin production. To simulate the actual process, a model based on the governing equations proposed by Birol et al. (2002) is considered. Assuming temperature and pH to be constant and, also, that the oxygen is available in excess, the following set of modified Equations ((3.29)-(3.32)) is used to describe the process:

$$\frac{dX}{dt} = \left(\frac{\mu_X S X}{K_X X + S}\right) - \frac{X}{V}\frac{dV}{dt} \tag{3.29}$$

$$\frac{dP}{dt} = \left(\frac{\mu_P S X}{K_P + S + \frac{S^2}{K_I}}\right) - K_H P - \frac{P}{V}\frac{dV}{dt} \tag{3.30}$$

$$\frac{dS}{dt} = -\frac{1}{Y_{X/S}}\left(\frac{\mu_X S X}{K_X X + S}\right) - \frac{1}{Y_{P/S}}\left(\frac{\mu_P S X}{K_P + S + \frac{S^2}{K_I}}\right) - m_X X + \frac{F s_f}{V} - \frac{S}{V}\frac{dV}{dt} \tag{3.31}$$

$$\frac{dV}{dt} = F - 6.226 * 10^{-4} V \tag{3.32}$$

where, X is the concentration of biomass, μ_X is the specific growth rate, K_X and $K_P$are saturation constants, S is the concentration of substrate, V is the volume of the culture medium, P is the concentration of penicillin, μ_P is the specific rate of penicillin production, K_I is an inhibition constant, $Y_{X/S}$ is the yield of biomass per unit mass of substrate, $Y_{P/S}$ is the yield of penicillin per unit mass of substrate, m_X is the maintenance constant, F is feed-rate and s_f is the concentration of substrate in the feed.

To formulate an uncertain model, it is assumed that the knowledge about the consumption of penicillin by hydrolysis is not available to the user. Therefore, the rate of change in penicillin, as available to the user, is as follows:

$$\frac{dP}{dt} = \left(\frac{\mu_P S X}{K_P + S + \frac{S^2}{K_I}} \right) - \frac{P}{V}\frac{dV}{dt} \tag{3.33}$$

The set of Equations (3.29) and ((3.31)-(3.33)) then represents an uncertain model of the process which, hereafter, will be referred to as "process model" or simply as "model". To estimate the model parameters, the simulated data is obtained by running the simulator (Equations ((3.29)-(3.32)) with the initial concentrations and the inlet feed profile as listed in Table 3.1. The parameters used in the simulator Equations ((3.29)-(3.32)) are listed in Table 3.2.

Table 3-1: Initial concentrations and input feed profile for the simulated data

Biomass Conc. (X_0)	0.1 (g/l)
Substrate Conc. (S_0)	0.1 (g/l)
Product Conc. (P_0)	0 (g/l)
Initial Culture Volume (V_0)	100 (L)
Input Feed (F)	0.04 (L/hr)

Table 3-2: Parameters' values for process simulator (Equations (3.29)-(3.32))

μ_x	K_x	μ_P	K_P	K_I	K_H	$Y_{X/S}$	$Y_{P/S}$	m_X	s_f
0.092	0.15	0.005	0.0002	0.1	0.04	0.45	0.9	0.014	600

To introduce uncertainty in the measurements, it is further assumed that the culture volume cannot be measured and the measurements of the other states; biomass, penicillin and substrate, are corrupted with Gaussian noise.

Once the parameters are estimated with their associated uncertainty, as per the methods described in the previous section, the objective is to maximize the amount of penicillin, at the end of batch, with minimum variability and the culture volume not exceeding 120 L. With initial substrate concentration (S_o) and inlet feed rate (F) as manipulated variables, the robust optimization problem is formulated as follows:

$$\min_{S_o,F} \; -\left[E\left(P(\boldsymbol{x},\boldsymbol{\theta},S_o,F,t_f)\right) - Var\left(P(\boldsymbol{x},\boldsymbol{\theta},S_o,F,t_f)\right)\right]$$

$$\text{s.t.} \quad (3.29) \text{and } (3.31-3.33)$$

$$\left\|V(\boldsymbol{x},\boldsymbol{\theta},S_o,F,t_f)\right\|_\infty \leq 120 \tag{3.34}$$

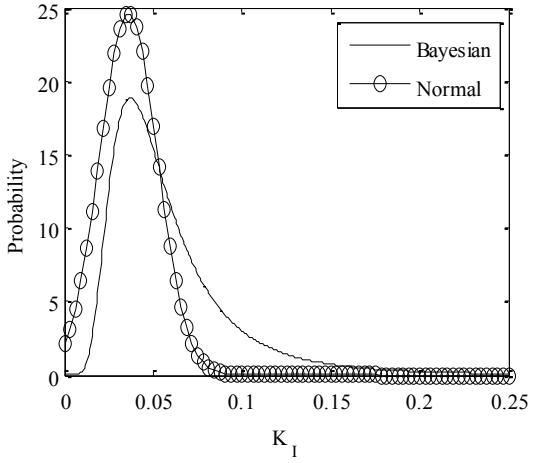
The signs of the expected value and the variance of the objective function are introduced to implicitly maximize the lower bound of the penicillin production, i.e. the mean value minus the variance, at the end of the batch (t_f) or alternatively to minimize the negative value of this bound. The bound on the volume is introduced to avoid overflowing of the vessel at all times.

3.3.2 Results and discussion

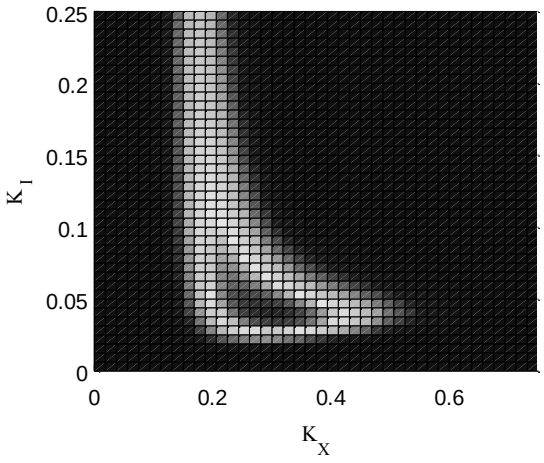
To illustrate the methodology, we considered four examples where only a subset of parameters is considered to be uncertain. The remaining parameters, in all the examples, were fixed at their nominal values, estimated by the standard least squares' method. The subset of parameters is selected following a preliminary study where the parameters were ranked according to decreasing order of their uncertainty. The uncertainty description for this purpose was obtained using an approach based on linearization of the model around the nominal parameter values (Bard, 1974). The three most uncertain parameters in the ranking were K_X, K_I and K_P. The individual effect of these parameters is studied in three different examples and finally, the fourth example considered the combined effect of uncertainty in K_X and K_I. One of the key points of this study is to investigate how the Bayesian description for parametric uncertainty affects the solution of the robust optimization problem when compared to simpler descriptions, as often used. For this purpose, in all the examples, the uncertainty description in the parameters is obtained using both the Bayesian as well as the linear approximation approach (Bard, 1974). For normally distributed errors, where the Bayesian approach results in a more realistic description of the uncertainty, the linear approximation results in normally distributed parameters. The mean and variance of the parameters, as a result of these two descriptions, is summarized in Table 3.3. Although, in each example, the Bayesian description of the uncertainty differs from its normal counterpart, the most significant differences are in the third and fourth example that considered uncertainty only in K_I and joint uncertainty in K_X and K_I respectively.

Table 3-3: List of means and variances, comparing Bayesian and Normal distributions for each example

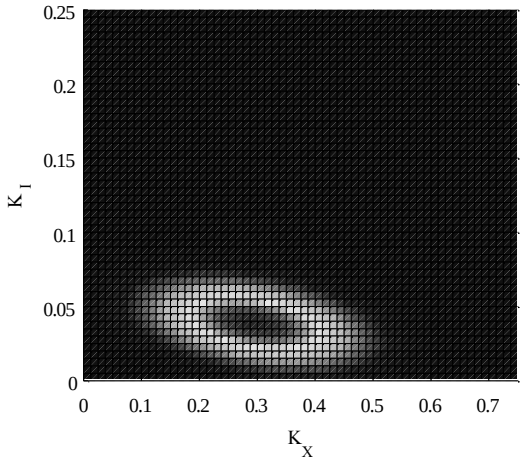
$\boldsymbol{\theta}$	K_X		K_P		K_I		K_X & K_I	
	Normal	Bayesian	Normal	Bayesian	Normal	Bayesian	Normal	Bayesian
E($\boldsymbol{\theta}$)	0.2892	0.3253	0.9986	1.2076	0.036	0.0589	[0.2892 0.036]	[0.2498 0.0955]
V($\boldsymbol{\theta}$)	0.0107	0.0098	0.0807	0.1605	0.000263	0.0012	[0.0122 0.0003]	[0.0089 0.004]



(a)



(b)



(c)

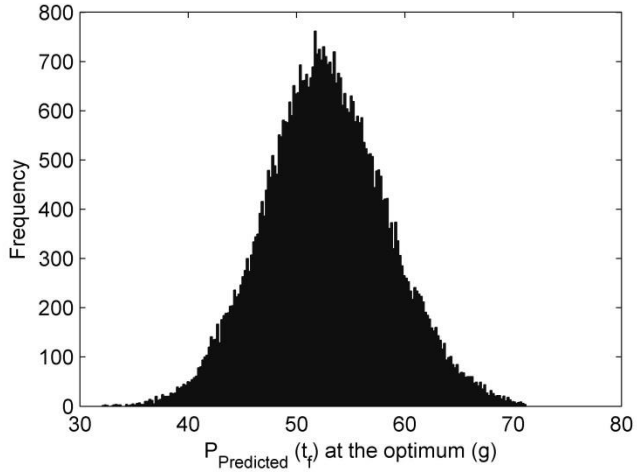
Figure 3-1: (a) Comparison of Bayesian and Normal uncertainty for K_I; (b) 2-D representation of joint Bayesian uncertainty for K_X and K_I (c) 2-D representation of joint normal uncertainty for K_X and K_I

In the third example, both the expected value and the variance of K_I are larger for the Bayesian description. It is further clear from Figure 3.1a, that the Bayesian uncertainty favors the larger values of K_I as compared to the normal uncertainty description. For example, the region with K_I larger than ~0.075 is more probable when the Bayesian uncertainty is considered, whereas, the region with K_I below ~0.01, is less probable with this description. In the fourth example, as shown in Figure 3.1b, the Bayesian uncertainty shows very strong correlation between K_X and K_I. The region, with K_X ranging from 0.15-0.25 and K_I ranging from 0.075-0.25 that has very low probability according to the normal description, is significantly favored by the Bayesian based uncertainty description. It should be noted that this region also favors the larger values of K_I. The parameter K_I basically represents the extent of substrate inhibition in the penicillin production. The larger the value of K_I the lesser will be the inhibition effect. From the above discussion, it is expected that the Bayesian based description of uncertainty will favor higher amounts of penicillin in both the examples.

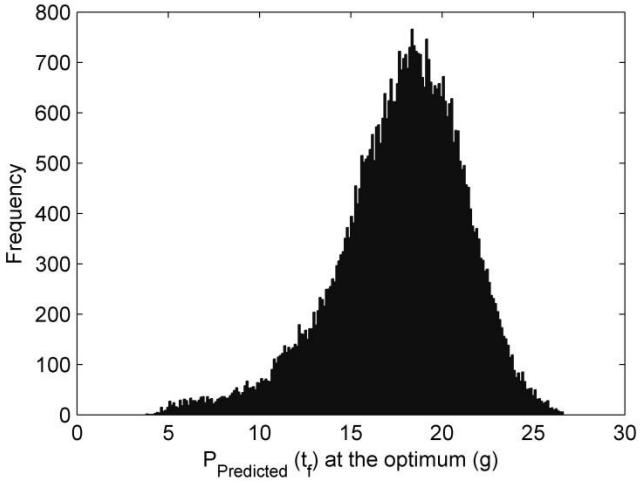
The next step was to investigate how the differences in the uncertainty descriptions in each example, i.e. Bayesian versus Normal, affect the corresponding solution of the robust optimization problem (3.34). The search was performed using *fmincon* function in the MATLAB Optimization Toolbox. To ensure that the optimal solution is not a local minima, the search was repeated several times with different guesses of the initial operating conditions. The mean and variance of the objective function f at the optimal solution are summarized in Table 3.4. The results clearly shows that the robust optimal solution obtained using the Bayesian description of the parametric uncertainty differs significantly from the one corresponds to the normal, especially for the two parameter's case in the fourth example. The distribution of the objective function f at the optimum based on the joint uncertainty in K_X sand K_I is shown in Figure 3.2a and 3.2b.

Table 3-4: Comparison of optimal solution based on Bayesian and Normal description of uncertainty

$\boldsymbol{\theta}$	$p(\boldsymbol{\theta})$	S_o	F	$E\left(P(t_f)\right)$	$\mathrm{Var}\left(P(t_f)\right)$	$P_{measured}(t_f)$
K_X	Normal	12.54	0.078	53.01	28.83	164.12
	Bayesian	38.04	0.150	75.95	13.76	328.02
K_P	Normal	31.74	0.277	86.58	11.80	484.45
	Bayesian	29.38	0.277	80.59	12.12	479.06
K_I	Normal	26.35	0.0797	30.55	14.09	191.55
	Bayesian	22.91	0.0799	46.93	22.145	191.83
K_X & K_I	Normal	26.87	0.0734	17.65	12.58	173.10
	Bayesian	29.16	0.1009	52.75	32.73	235.87



(a)



(b)

Figure 3-2: Distribution of the penicillin at the end of batch, predicted at the optimum, based on the (a) Bayesian uncertainty and (b) Normal uncertainty

It should be noted here that the model has structural uncertainty and because of this, the optimal solution in all the four examples, might not corresponds to the true process optimum. For that, an iterative procedure has to be applied that slowly converges to the true process optimum (Chen et al., 1987; Chachuat et al., 2009; Mandur et al., 2013b), which is beyond the scope of this study. However, it is relevant to check the effect of the normal and the Bayesian uncertainty descriptions on the measured penicillin at the end of batch, simulated at the corresponding optimal solutions. From Table 3.4, it appears that for the second and the third example, both the descriptions resulted in nearly the same optimal solutions. However, for the first example, the optimal solution corresponding to the Bayesian uncertainty results in nearly twice the measured penicillin as compared to normal uncertainty. Similarly, for the fourth example, this increase in the measured penicillin is nearly 36%. In other words, for the first and the fourth examples, the optimal solutions are closer to the true process optimum when the Bayesian description of uncertainty is used. From these results, it is also observed that the optimal solution is very sensitive to the uncertainty in K_X whereas sensitivity to the uncertainty in K_I is negligible.

In practical situations, it is common that the measurements may not be available as frequent as expected. To study the effect of frequency of the measurements, an additional simulation study is conducted where the measurements are assumed to be available every 12 hours instead of every 6 hours, as is the case in previous examples. For this comparative study, only two parameters, K_X and K_I are re-estimated. As expected, fewer measurements resulted in increase in the parametric uncertainty which further results in more conservative optimal solutions. For normal uncertainty, the amount of penicillin at the end of batch is observed to be decreased by 80.5 % and for Bayesian uncertainty, a decrease of 69.8% is observed.

3.3.3 Computational Efficiency

The computational time of the proposed methodology was compared with the most widely used Monte Carlo approach. In the latter, the expected value and the variability of the objective function are evaluated using Monte Carlo integration which basically involves repeated evaluations of the model for different parameter values, selected randomly from their distribution. When the distribution has to be obtained by the Bayesian Inference, as in this study, there is an added computational load in the Monte Carlo approach. Since the posterior distribution has no analytical expression, the parameter samples have to be obtained using Markov Chain Monte Carlo (MCMC) approach (Robert et al., 2004). In MCMC, the aim is to construct a Markov chain that converges to a desired posterior distribution after a large number of samples. Since in the beginning, the chain may be far off from the convergence, few initial samples are generally rejected as *burn-in* (Robert et al., 2004). In the comparison, we constructed the Markov Chain with 50,000 samples and rejected the first 5000 samples as a *burn-in*. The remaining samples were then used to calculate the expected value and the variance of the objective function. The search for the optimum was completed in 50-60 iterations and takes approximately 60-70 hours on average on a Quad-Core 3GHz Core-i7 Workstation. On the other hand, the proposed methodology based on PC expansions completed the search in approximately 4-5 min on average for the three examples considering uncertainty in single parameter whereas, for the fourth example which considered uncertainty in two parameters the computations required 30-40 min. The difference in computational load between the one uncertain parameter and two uncertain parameters cases is that for the latter the number of quadrature points required to solve the two-dimensional integrals in the computation of PC coefficients is higher as compared to the number needed for the one-dimensional integrals in the remaining examples. The above significant reduction in the computational time as compared to standard Monte Carlo approach is primarily due to the reduction in model runs required for each function evaluation. Whereas, in the Monte Carlo approach, the model has to be solved 45,000 times to evaluate the

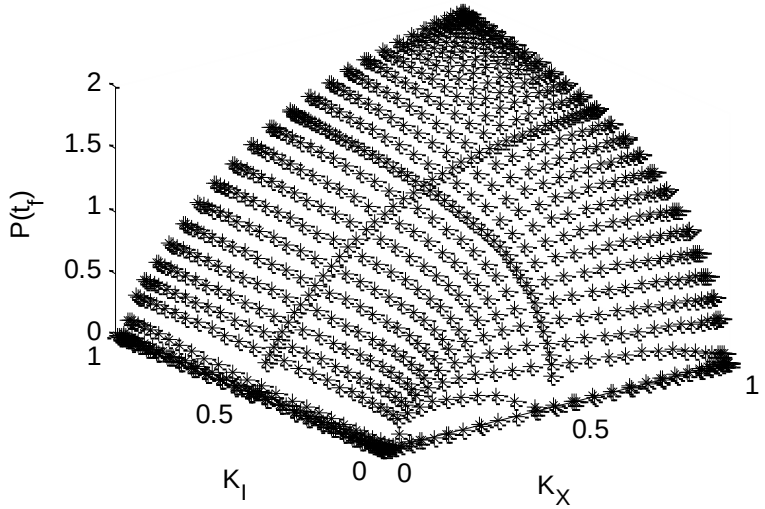
expected value and the variance in the objective function, in the proposed approach only the PC coefficients have to be evaluated which requires few tens to hundred model runs. This computational advantage further makes the proposed methodology suitable for solving online robust optimization problems using Bayesian description of uncertainty, where the standard Monte Carlo based approach might fails to provide a solution within reasonable time before the next measurement.

Another major saving is achieved by using an approximation of the original nonlinear model (Equations (3.29) and ((3.31)-(3.33))) in the likelihood function. However, it should be noted that the model used in this case study is not highly nonlinear in parameters. As a result, the proposed iterative approach involving the refinement of the approximation in parameter regions of higher probability did not provide any computational advantage. In fact, in all the four examples presented above, the approximation of the model was obtained using polynomials from the base resolution subspace $V_0^{N_o}$.

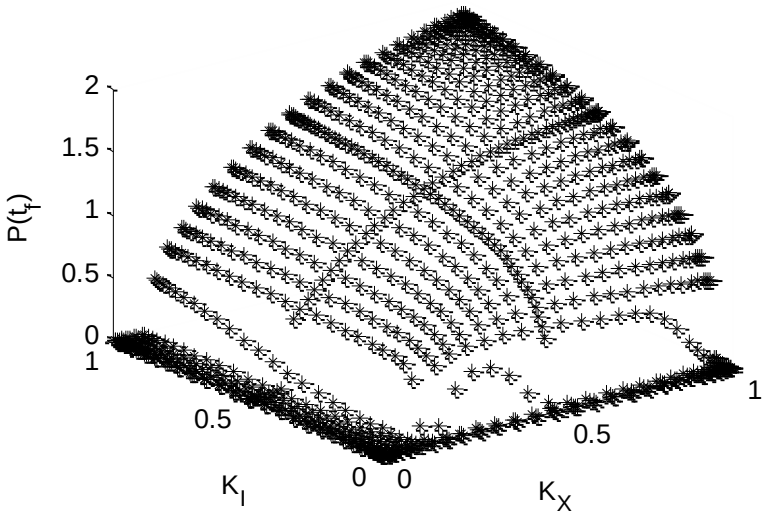
It was hypothesized that the multi-resolution approach with adaptive refinement would significantly contribute to computational efficiency when the dependency of the objective function f would be highly nonlinear or even discontinuous with respect to the uncertain parameters as occurring in problems with bifurcations or high parametric sensitivity. To illustrate the efficiency of the proposed approach, we considered a particular situation where there is a discontinuity in the measurement of penicillin due to a minimal threshold for detection in that measurement. This situation is common in chromatographic based measurements where the peaks related to penicillin or other products are measured with respect to minimal baseline and below that baseline the measurement is assumed as zero (Nakashima et al., 1993). A straightforward way to estimate the parameters in such situation, while not altering the assumption of Gaussian noise, is to model the process and device related threshold together. In that case, the model output for penicillin will be zero below the threshold and the value produced by the Equations ((3.29) and ((3.31)-(3.33))) when

the value is above the threshold. Following this assumption and considering the case of uncertainty in two parameters as for the fourth example above, the model output as a function of parameters is a discontinuous function as shown in Figure 3.3.

Such situation cannot be approximated by using polynomials only from the base subspace $V_0^{N_o}$. Figure 3.4 compares the joint probability distribution of K_X and K_I as obtained by two different approximations; one using the polynomials only from $V_0^{N_o}$ and another using the adaptive approach as outlined in this work. In the first approximation, the maximum order of polynomials used in each dimension was 19 whereas in the adaptive approach, only the first four Legendre polynomials in the base resolution and a maximum of four resolutions were used. As it can be seen from Figure 3.4a, where the threshold was considered to be 0.2, both the approximations results in nearly same distribution. On the other hand, when the threshold was considered to be 0.5, the first approximation results in oscillatory profile and because of this an erroneous multimodal distribution is obtained as shown in Figure 3.4b. Thus, the discontinuity in the measurements because of 0.2 threshold has very local and overall little effect in the remaining parameter region whereas the effect of a 0.5 threshold spreads over the entire region. Accordingly, if only polynomials from the base resolution are used, the model cannot be approximated accurately over the entire region thus justifying the need for a multi-resolution approximation.



(a)



(b)

Figure 3-3: (a) Penicillin, at end of the batch, as a function of K_X and K_I for (a) 0.2 threshold and (b) 0.5 threshold

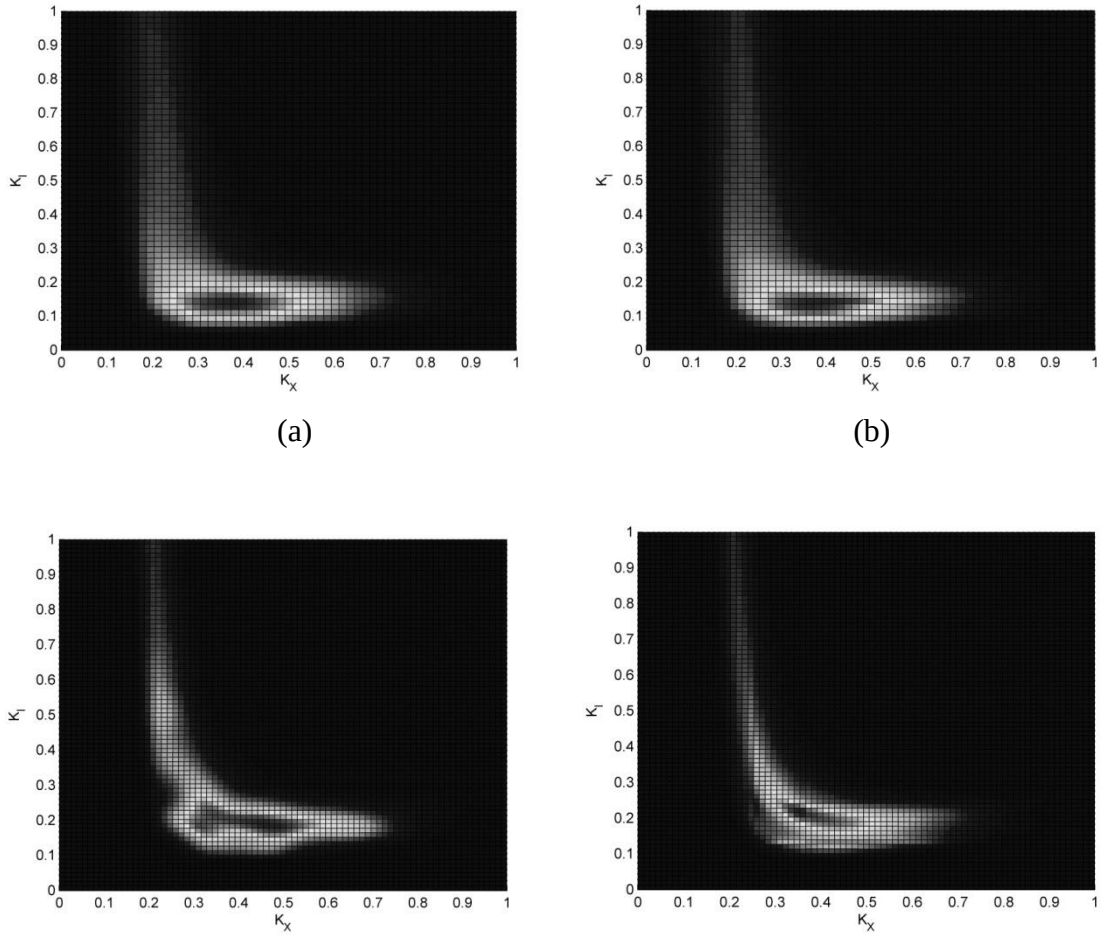


Figure 3-4: 2-D representation of the joint probability distribution of K_X and K_I obtained by (a) PC approximation for 0.2 threshold, (b) Multi-resolution approximation for 0.2 threshold (c) PC approximation for 0.5 threshold and (d) Multi-resolution approximation for 0.5 threshold

3.4 Conclusions

A robust optimization problem is solved when the uncertainty in model parameters is obtained by using the Bayes' Theorem. At each function evaluation, PC expansions are used to propagate the parametric uncertainty into the objective function thus allowing for the calculation of the expected value and the variance of the objective function by analytical expressions. The use of a model approximation for the calculation of the likelihood function reduces the computational time further when compared to solving the full nonlinear model repeatedly. The multi-resolution based model approximation proposed in the work, proved to be especially useful when approximating the functions that exhibit sharp gradients with respect to the uncertain parameters. The overall methodology has been shown to be much more efficient as compared to conventional Monte Carlo approach thus making it attractive for both off-line optimizations as well as for potential application in on-line problems. When compared with the optimization results based on the normal description of parametric uncertainty, the Bayesian description results in significantly different uncertainty descriptions and correspondingly in significantly different optimal solutions.

Chapter 4
Simultaneous model identification and optimization in presence of model-plant mismatch

(Submitted to Chemical Engineering Science)

4.1 Introduction

Mathematical models have become an integral part of the process development and subsequent production environment. Besides providing novel insights into the underlying process, they are also used in various model-based optimization and optimal control strategies. When a model is an exact representation of the actual process and is calibrated against noise-free process data, optimizing the model is identical to optimizing the process itself. In such case, the optimal policies derived from model-based optimization can be applied in an "open loop" fashion. However, the above conditions are extremely difficult to meet in practical situations. In the presence of any model uncertainty resulting from either incorrect model structure or measurement noise, the model-based optimization algorithms will result in sub-optimal policies or, in a worst case scenario, may also result in violation of process constraints. To tackle this problem, one possible solution is to search for optimal policies that are robust to model uncertainties (Beyer et al., 2007; Samsatli et al., 1998; Diwekar et al, 1996; Nagy et al., 2004; Ruppen et al., 1995; Terwiesch et al., 1994). Although this approach can ensure feasibility within *a priori* known bounds of uncertainty, the optimal policies are often conservative and, in some processes, may lead to significant loss in economic objectives. As an alternative, another possibility is to use an iterative approach where the model is updated using new measurements at

previously calculated optimal policy and the updated model is, then, re-optimized for the next optimal policy (Ruppen et al., 1998; Eaton et al., 1990; Chen et al., 1987; Marlin et al., 1997). This process is referred to as a "two-step" approach and is repeated until a convergence is achieved.

This chapter deals with the application of latter approach to batch/fed-batch processes. Assuming the process data is available only at the end of batch, the problem is solved in a run-to-run optimization framework. However, there is no restriction on applying the proposed algorithm to online optimization problems if measurements are available online.

The convergence of the standard two-step approach is governed by; (1) whether the sub-optimal policies provide enough excitation to update all the parameters and (2) how close the model can describe the actual process. The first condition can be addressed, to a certain extent, by incorporating design of experiments in the optimization objectives (Martinez et al., 2013). In this way, a trade-off between the optimal policies and the policies that generate more informative process data can be achieved. Then, if the model is a true of representation of the process, the two-step approach will converge to the actual process optimum, where the total number of iterations needed for convergence, will depend on measurement noise and the extent of excitation.

Regarding the second condition, mentioned above, model-plant mismatch is inevitable in almost all practical applications. In an attempt to capture the process behavior accurately, models often become too complex and computationally demanding rendering them unsuitable for optimization or control purposes. Also, with a limited number of measurable states and in the presence of measurement noise, it is not possible to estimate all the parameters accurately and, therefore, model reduction techniques are often used to reduce the number of parameters that can be identified from the given process data. Because of these reasons, one has to generally rely on simpler but inaccurate model structures for optimization and control. If the structural

inaccuracy is not considered explicitly in the model, calibrating the model over different operating conditions may result in significantly different parameter estimates in order to compensate for the model error around different operating points. As a result of this parametric variability, it is possible that the optimization objectives may get compromised. The change in parameter estimates may be of such an extent that the predicted gradients of the optimization objectives no longer coincide with the gradients measured from the process or, in a worst case scenario, they may even get reversed thus leading to premature convergence to sub-optimal operating policies.

For convergence to the process optimum, it is necessary that the model accurately predicts the optimality conditions of the process as given by the first-order Karush-Kuhn-Tucker (KKT) conditions. Following this idea, a class of algorithms has been developed where the optimization objectives and the constraints are corrected for the bias as well as the difference between their predicted and measured gradients (Roberts et al., 1979; Tatjewski, 2002; Gao et al., 2005; Chachuat et al., 2009; Marchetti et al., 2009; Costello et al., 2011). These algorithms differ in the way the model is updated and on how the modifications to the objective function and constraints are implemented. For instance, in their pioneer work, Roberts et al. (1979) modified only the optimization objective to account for the difference between predicted and measured output derivatives assuming the constraints are process independent and are known. The modification term was based on the Lagrange multipliers where the Lagrangian function was obtained by integrating the identification and optimization objectives. In subsequent studies, Tatjewski (2002) and Gao et al. (2005) replaced the parameter estimation step by introducing a linear term in the outputs that corrects for the difference between the predicted and measured outputs. In a more recent version of these algorithms, referred to as modifier adaptation (Chachuat et al., 2009; Marchetti et al., 2009 and Costello et al., 2011), the authors eliminated the model update step altogether and updated only the optimization objectives based on differences between the gradients in the optimality

condition. Since the convergence to a process optimum was driven solely by the correction in the optimization gradients, the final model-based optimal solution remained unaffected by the elimination of the model-update step. However, this approach results in a bias between the predicted and measured outputs and as result, the algorithm can no longer be applied to the problems where prediction accuracy is required. One such case, as recently pointed out by Costello et al. (2011), is when the optimal input profiles are implemented within a closed-loop control to ensure that the process is operated to meet safety and environmental constraints. Here, the model is required to provide accurate reference trajectories for low-level controllers. The prediction accuracy of the model is also very relevant for biotechnological processes where it is important to predict the evolution of toxic by-products along a batch culture. Thus, to address a broad range of problems, it is very important to satisfy both identification and optimization objectives at the optimum. One of the major bottlenecks in implementation of this class of algorithms is their sensitivity to the noise in measured gradients (Marchetti et al., 2009). To avoid too much aggressiveness in the corrections and to achieve a smoother convergence, the corrections have to be filtered using an empirical filter.

In another class of algorithm, Srinivasan et al. (2002) proposed an alternate approach where the identification objective is modified to account for the difference between predicted and measured optimality conditions. By this modification, the parameter estimates can be obtained so as to provide a trade-off between the identification and optimization objectives based on preselected weights.

In this work, we propose a linear correction to the model outputs in a way that the updated model parameters not only minimize the bias between the predicted and measured outputs, but it also correct for the optimization objectives. The corrections made over the previous iterations are progressively integrated and by implementing this progressive correction in the model, the conflict between the identification and optimization objectives is reduced significantly. Another key advantage of this

approach is that it provides a model-based filter that is shown to outperform the external exponential filter, used in the previous studies, in terms of the rate at which convergence can be achieved. A preliminary discussion of this methodology has been presented in Mandur et al. (2013b).

The contents of this chapter are organized as follows: Section 4.2 provides a brief background on two-step approach and modifier adaptation algorithms and also discusses the motivation in detail. Section 4.3 presents the methodology and theory behind the proposed model correction. The methodology is then illustrated with a case study in Section 4.4 and finally Section 4.5 concludes the chapter.

4.2 Preliminaries

Let us consider a process model, described by a set of differential equations as follows:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \boldsymbol{\theta}, \mathbf{u}, \mathbf{t}) + \boldsymbol{\upsilon}$$

$$\mathbf{y_m} = h(\mathbf{x}) + \boldsymbol{\eta} \qquad \text{(P.4.1)}$$

Where, $\mathbf{x} \in \mathbb{R}^{n_x}$ is the vector of model states, $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}$ is the vector of model parameters, $\mathbf{u} \in \mathbb{R}^{n_u}$ is the vector of inputs, $\mathbf{y_m} \in \mathbb{R}^{n_y}$ is the vector of measured output variables, $f \in \mathbb{R}^{n_x}$ is a set of differential equations based on mass and energy balances, $h \in \mathbb{R}^{n_y}$is a mapping between the model states and predicted outputs and $\boldsymbol{\upsilon}$ and $\boldsymbol{\eta}$ are the vectors of uncertainties representing modelling and measurement errors respectively.

The standard two-step optimization approach starts with a model identification step where the model is calibrated using the process measurements at some initial input conditions. The identification objective is generally based on the minimization of the errors between predicted and measured outputs.

For example, the standard least squares estimation problem can be formulated as follows:

$$\boldsymbol{\theta_k} = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} \|\mathbf{y_m}(\mathbf{u_k}) - \mathbf{y}(\mathbf{u_k}, \boldsymbol{\theta})\|^2$$

$$\mathbf{s.t.} \quad \dot{\mathbf{x}} = f(\mathbf{x}, \boldsymbol{\theta}, \mathbf{u_k}, \mathbf{t})$$

$$\mathbf{y} = h(\mathbf{x}) \tag{P.4.2}$$

Where, $\mathbf{y} \in \mathbb{R}^{n_y}$ is the vector of predicted outputs, N is the number of time points and subscript **k** is the current iteration.

The identification is then followed by an optimization step, formulated as follows:

$$\mathbf{u_{k+1}} = \arg\min_{\mathbf{u}} \phi(\mathbf{y}, \mathbf{u}, \boldsymbol{\theta_k})$$

$$\text{s.t.} \quad \dot{\mathbf{x}} = f(\mathbf{x}, \boldsymbol{\theta_k}, \mathbf{u}, \mathbf{t})$$

$$\mathbf{y} = h(\mathbf{x})$$

$$\mathbf{g}(\mathbf{y}, \mathbf{u}, \boldsymbol{\theta_k}) \leq \mathbf{0} \tag{P.4.3}$$

Where, ϕ is the objective function to be minimized and $\mathbf{g} \in \mathbb{R}^{n_g}$ is a vector of equalities or inequalities.

Let the functions ϕ and g_j be continuously differentiable at a set of input conditions $\mathbf{u}^*$. If $\mathbf{u}^*$ is a process optimum, then there exists a unique vector, $\boldsymbol{\mu} \in \mathbb{R}^{n_g}$ such that:

$$\frac{\partial \phi(\mathbf{y_m}, \mathbf{u}^*)}{\partial \mathbf{u}} + \boldsymbol{\mu}^T \frac{\partial \mathbf{g}(\mathbf{y_m}, \mathbf{u}^*)}{\partial \mathbf{u}} = 0 \tag{4.1a}$$

$$\boldsymbol{\mu}^{\mathbf{T}} \mathbf{g}(\mathbf{y_m}, \mathbf{u}^*) = \mathbf{0} \tag{4.1b}$$

$$\boldsymbol{\mu} \geq \mathbf{0} \tag{4.1c}$$

$$\mathbf{g}(\mathbf{y_m}, \mathbf{u}^*) \leq \mathbf{0} \tag{4.1d}$$

These conditions are collectively known as Karush-Kuhn-Tucker (KKT) conditions, where $\boldsymbol{\mu}$ is a vector of KKT multipliers. For the model-based optimal solution to converge to $\mathbf{u}^*$, it is necessary that the model predicts the KKT above conditions accurately. Since $\mathbf{u}^*$ is not known a priori, this can be only guaranteed if the identification step (P.4.2) results in a unique set of model parameters ($\boldsymbol{\theta_k}$) such that the model satisfies the following conditions for all set of values of $\mathbf{u} \in \mathbb{R}^{n_u}$:

$$\frac{\partial \phi(\mathbf{y}, \mathbf{u}, \boldsymbol{\theta_k})}{\partial \mathrm{u_i}} = \frac{\partial \phi(\mathbf{y_m}, \mathbf{u})}{\partial \mathrm{u_i}} \tag{4.2a}$$

$$\frac{\partial g_j(\mathbf{y}, \mathbf{u}, \boldsymbol{\theta_k})}{\partial \mathrm{u_i}} = \frac{\partial g_j(\mathbf{y_m}, \mathbf{u})}{\partial \mathrm{u_i}} \tag{4.2b}$$

When there is only measurement noise (i.e. $\boldsymbol{\upsilon} = \mathbf{0}$), the above conditions can be satisfied over a finite number of iterations where:

$$\frac{\sum_{k=1}^{N} \boldsymbol{\theta_k}}{N} \sim \boldsymbol{\theta}^* \tag{4.3}$$

Where, $\boldsymbol{\theta}^* \in \mathbb{R}^{n_\theta}$ is the set of parameter values satisfying the conditions (4.2a) and (4.2b)

However, in the presence of model structure error ($\boldsymbol{\upsilon} \neq \mathbf{0}$), $\boldsymbol{\theta}^*$ does not exist. Since the error term $\boldsymbol{\upsilon}$ represents the unmodelled dynamics of the process, it is a time varying function of model states $(\mathbf{x})$ and inputs $(\mathbf{u})$. Then, when the inaccurate model, given by $\dot{\mathbf{x}} = f(\mathbf{x}, \boldsymbol{\theta}, \mathbf{u_k}, \mathbf{t})$, is calibrated over different input conditions, the parameter estimates change so as to compensate for the modelling error which varies with respect to the input conditions. Consequently, there is no unique set of parameter estimates that can satisfy the identification objective (P.4.2) over the entire space of input conditions. For any particular set of parameter estimates $(\boldsymbol{\theta_k})$, the model is accurate only in the neighbourhood of the corresponding input values $(\mathbf{u_k})$ whereas away from this region, the prediction accuracy of the model continues to decrease as the distance increases. As a result, the model may not predict the gradients of the optimization objective and constraints accurately. In a worst case scenario, it is also possible that the change in model parameters as the input conditions change is of such an extent that the predicted and measured gradients have opposite signs, in which case the model-based optimization can no longer drive the changes in the inputs in the direction of process optimum. This implies:

$$\phi(\mathbf{y_m}, \mathbf{u_{k+1}}) > \phi(\mathbf{y_m}, \mathbf{u_k}) \tag{4.4}$$

In this case, the two-step approach will converge to a non-optimal solution where the measured gradients are still non-zero, or in other words, the predicted KKT conditions do not match with those measured from the process. Therefore, to ensure convergence of the algorithm to a process optimum, the differences between the predicted and measured gradients of the optimization problem must be eliminated or at least minimized at each intermediate input condition.

As stated in the Introduction, *modifier adaptation* algorithms enforce the matching conditions (Equations (4.2a) and (4.2b)) by adding correction terms directly to the corresponding optimization quantities.

Accordingly, the modified optimization problem is as follows:

$$\mathbf{u_{k+1}} = \arg\min_{\mathbf{u}} \phi(\mathbf{y}, \mathbf{u}, \boldsymbol{\theta}) + \boldsymbol{\lambda}_{\phi k}^{\mathbf{T}}\mathbf{u}$$

$$\text{s.t.} \quad \dot{\mathbf{x}} = f(\mathbf{x}, \boldsymbol{\theta}, \mathbf{u}, \mathbf{t})$$

$$\mathbf{y} = h(\mathbf{x})$$

$$\mathbf{g}(\mathbf{y}, \mathbf{u}, \boldsymbol{\theta}) + \boldsymbol{\varepsilon}_{gk} + \boldsymbol{\lambda}_{gk}^{\mathbf{T}}(\mathbf{u} - \mathbf{u_k}) \leq \mathbf{0} \qquad \text{(P.4.4)}$$

Where, $\boldsymbol{\lambda}_{\phi}$ and $\boldsymbol{\lambda}_{g}$ are referred to as modifiers that are used to correct for the gradients of objective function and constraints respectively and $\boldsymbol{\varepsilon}_{g}$ is a modifier introduced to correct for the bias in predicted and measured constraints. The corrections are calculated at the $\boldsymbol{k}^{th}$ iteration as follows:

$$\lambda_{\phi k_i} = \frac{\partial \phi(\mathbf{y_m}, \mathbf{u_k})}{\partial u_i} - \frac{\partial \phi(\mathbf{y}, \mathbf{u_k}, \boldsymbol{\theta})}{\partial u_i} \qquad \text{(4.5a)}$$

$$\lambda_{gk_{ij}} = \frac{\partial g_j(\mathbf{y_m}, \mathbf{u_k})}{\partial u_i} - \frac{\partial g_j(\mathbf{y}, \mathbf{u_k}, \boldsymbol{\theta})}{\partial u_i} \qquad \text{(4.5b)}$$

$$\varepsilon_{gk_j} = g_j(\mathbf{y_m}, \mathbf{u_k}) - g_j(\mathbf{y}, \mathbf{u_k}, \boldsymbol{\theta}) \qquad \text{(4.5c)}$$

To avoid excessive corrective actions and to reduce the sensitivity to measurement noise, these corrections are filtered before implemented in (P.4.4) as follows:

$$\boldsymbol{\Lambda}_k = \mathbf{K}\boldsymbol{\Lambda}'_k + (1 - \mathbf{K})\boldsymbol{\Lambda}_{k-1} \qquad \text{(4.6)}$$

Where, $\boldsymbol{\Lambda}$ represent the vector of modifiers defined as: $\boldsymbol{\Lambda} = [\boldsymbol{\lambda}_{\phi}, \boldsymbol{\lambda}_{g}, \boldsymbol{\varepsilon}_{g}]$ and $\mathbf{K}$ represents the filter gain.

It should be noted that whether the model is updated or not, the corrected optimization objective does not depend on the model. Therefore, the aggressive changes in the inputs, resulting from inaccurate predictions of the modified objective

function or constraints, have to be controlled by the filter gain which may further reduce the speed of convergence. Also, there is no systematic way to choose a priori the filter gain. In the case study presented later, it is observed that when the model is used to correct for the errors in gradients as proposed in the current study, it results in more accurate predictions during the search for optimal solution thus leading to faster convergence with less oscillatory behaviour.

4.3 Proposed Methodology

The basic idea in the proposed methodology is to search for model parameters such that the differences between the predicted and measured gradients of optimization problem given in Equations (4.2a) and (4.2b) are minimized along with the minimum prediction error from identification problem (P.4.2). However, since the identification and optimization objectives are independent of each other, in the presence of model structure error, the values of model parameters that satisfy both objectives do not exist. In other words, the parameter estimates that minimize the difference in gradients may not minimize the prediction error at the same time. To this end, a linear correction term is added to the model outputs such that the conflicting objectives can be reconciled.

Let ϵ_k be the minimum sum of squared errors between the predicted and measured outputs in problem (P.4.2), corresponding to the parameter estimates $\boldsymbol{\theta_k}$ as follows:

$$\epsilon_k = \sum_{i=1}^{N} \|\mathbf{y_m}(\mathbf{u_k}) - \mathbf{y}(\mathbf{u_k}, \boldsymbol{\theta_k})\|^2 \tag{4.7}$$

Let $\Delta\boldsymbol{\theta_k}$ be the change in parameter estimates, with respect to $\boldsymbol{\theta_k}$, required to minimize the difference between the predicted and measured gradients (Equations (4.2a) and (4.2b)) at $\boldsymbol{k}^{th}$ iteration. The updated model parameters i.e. $\boldsymbol{\theta_k} + \Delta\boldsymbol{\theta_k}$, then, no longer minimizes the updated sum of squared errors. This is schematically

illustrated in Figure 4.1 which shows a probability density function of a model parameter centered on the estimate $\theta_{\mathbf{k}}$, calculated by least squares.

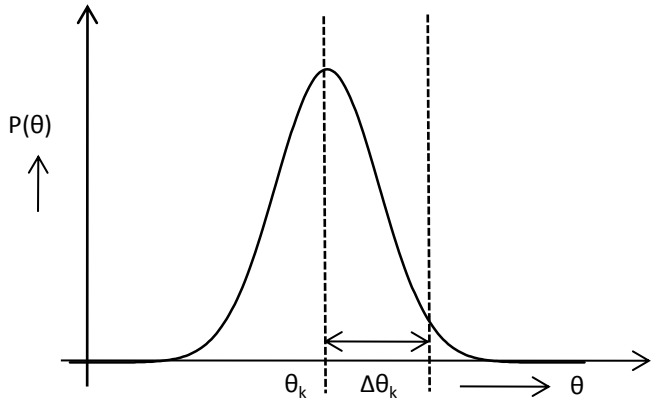


Figure 4-1: Illustration of lack of fit

The corresponding sum of squared errors using the perturbed parameter value $\boldsymbol{\theta_k} + \Delta\boldsymbol{\theta_k}$ can be represented as:

$$\epsilon_k' = \sum_{i=1}^{N} \|\mathbf{y_m}(\mathbf{u_k}) - \mathbf{y}(\mathbf{u_k}, \boldsymbol{\theta_k} + \Delta\boldsymbol{\theta_k})\|^2 \tag{4.8}$$

Let us introduce a vector of corrections $-\mathbf{c_k}$ to the model outputs such that the sum of squared errors, ϵ_k' with the corrected model remains equal to the original value of ϵ_k (given by Equation (4.7)):

$$\begin{aligned} \sum_{i=1}^{N} \|\mathbf{y_m}(\mathbf{u_k}) - (\mathbf{y}(\mathbf{u_k}, \boldsymbol{\theta_k} + \Delta\boldsymbol{\theta_k}) - \mathbf{c_k})\|^2 \\ = \sum_{i=1}^{N} \|\mathbf{y_m}(\mathbf{u_k}) - \mathbf{y}(\mathbf{u_k}, \boldsymbol{\theta_k})\|^2 \end{aligned} \tag{4.9}$$

To satisfy (4.9), the equality can be satisfied term by term as follows:

$$\mathbf{y_m}(\mathbf{u_k}) - \mathbf{y}(\mathbf{u_k}, \boldsymbol{\theta}_\mathbf{k} + \Delta\boldsymbol{\theta}_\mathbf{k}) + \mathbf{c_k} = \mathbf{y_m}(\mathbf{u_k}) - \mathbf{y}(\mathbf{u_k}, \boldsymbol{\theta}_\mathbf{k}) \tag{4.10}$$

The correction term can then be solved from (4.10) by:

$$\mathbf{c_k} = \mathbf{y}(\mathbf{u_k}, \boldsymbol{\theta}_\mathbf{k} + \Delta\boldsymbol{\theta}_\mathbf{k}) - \mathbf{y}(\mathbf{u_k}, \boldsymbol{\theta}_\mathbf{k}) \tag{4.11}$$

Let the model outputs $\mathbf{y}(\mathbf{u_k}, \boldsymbol{\theta}_\mathbf{k} + \Delta\boldsymbol{\theta}_\mathbf{k})$ be approximated around $\boldsymbol{\theta}_\mathbf{k}$ using Taylor Series Expansions. This will result in the following expression:

$$\mathbf{y}(\mathbf{u_k}, \boldsymbol{\theta}_\mathbf{k} + \Delta\boldsymbol{\theta}_\mathbf{k}) = \mathbf{y}(\mathbf{u_k}, \boldsymbol{\theta}_\mathbf{k}) + D\mathbf{y}(\boldsymbol{\theta}_\mathbf{k})\, \Delta\boldsymbol{\theta}_\mathbf{k} + \cdots \tag{4.12}$$

Where, D is the Jacobian matrix of output derivatives with respect to model parameters. After substituting Equation (4.12) into (4.11), the correction term $\mathbf{c_k}$ can be expressed as:

$$\mathbf{c_k} = D\mathbf{y}(\boldsymbol{\theta}_\mathbf{k})\, \Delta\boldsymbol{\theta}_\mathbf{k} + \cdots \tag{4.13}$$

Assuming the model to be linear in the neighborhood of $\boldsymbol{\theta}_\mathbf{k}$, the correction term $\mathbf{c_k}$ is approximated by the first-order derivative as follows:

$$\mathbf{c_k} = D\mathbf{y}(\boldsymbol{\theta}_\mathbf{k})\, \Delta\boldsymbol{\theta}_\mathbf{k} \tag{4.14}$$

Since the linear approximation of the model is generally valid only within a certain region around $\boldsymbol{\theta}_\mathbf{k}$, for $\Delta\boldsymbol{\theta}_\mathbf{k}$ outside this region, it would not be possible to restore the prediction error with the updated model to its minimum as the LHS of the equality, given by Equation (9), may have significant error. Therefore, to enforce the approximate validity of the linear approximation, a constraint on $\Delta\boldsymbol{\theta}_\mathbf{k}$ is imposed which is based on a relative truncation error, defined as follows:

$$\boldsymbol{\epsilon^{T}} = \frac{\mathbf{y}(\mathbf{u_k}, \boldsymbol{\theta_k} + \Delta\boldsymbol{\theta_k}) - D\mathbf{y}(\boldsymbol{\theta_k})\,\Delta\boldsymbol{\theta_k}}{\mathbf{y}(\mathbf{u_k}, \boldsymbol{\theta_k})} \tag{4.15}$$

Based on its definition, the calculation of $\Delta\boldsymbol{\theta_k}$ is calculated using an optimization problem as follows:

$$\Delta\boldsymbol{\theta_k} = \arg\min_{\Delta\boldsymbol{\theta}} \left(\mathbf{w}_{\phi} \left| \frac{\partial \phi(\mathbf{y_m}, \mathbf{u_k})}{\partial \mathbf{u}} - \frac{\partial \phi(\mathbf{y}, \mathbf{u_k}, \boldsymbol{\theta_k} + \Delta\boldsymbol{\theta})}{\partial \mathbf{u}} \right| + \mathbf{w}_{g} \left| \frac{\partial \mathbf{g}(\mathbf{y_m}, \mathbf{u_k})}{\partial \mathbf{u}} - \frac{\partial \mathbf{g}(\mathbf{y}, \mathbf{u_k}, \boldsymbol{\theta_k} + \Delta\boldsymbol{\theta})}{\partial \mathbf{u}} \right| \right)$$

$$\text{s.t.} \quad \dot{\mathbf{x}} = f(\mathbf{x}, \boldsymbol{\theta_k} + \Delta\boldsymbol{\theta}, \mathbf{u}, \mathbf{t})$$

$$\mathbf{y} = h(\mathbf{x}) - \boldsymbol{D}\mathbf{y}(\boldsymbol{\theta_k})\,\Delta\boldsymbol{\theta}$$

$$\boldsymbol{\epsilon^{T}} \leq \epsilon_{max}^{T} \tag{P.4.5}$$

Where, $\mathbf{w}_{\phi}$ and $\mathbf{w}_{g}$ are vectors of normalizing weights for the objective function and constraints gradients respectively and ϵ_{max}^{T} is the constraint or limit on truncation error that is imposed to ensure the approximated validity of the linear approximation of the correction term.

It is important to note here that the above constraint on $\Delta\boldsymbol{\theta_k}$ is somewhat equivalent to the filter gain in modifier adaptation algorithms as the restriction on $\Delta\boldsymbol{\theta_k}$ also restricts the ability of the model to predict gradients of the optimization problem exactly which is very critical when the gradients are associated with significant level of noise. On the other hand, in contrast with the filter gain in modifier adaptation algorithms, $\Delta\boldsymbol{\theta_k}$ is based on a physical rationale since it represents an allowable model prediction error.

To summarize the procedure, the estimation of model parameters is divided into two steps:

Step 1: The parameters are updated to minimize the error between the outputs as predicted by the previously corrected model and those measured from the process. Let us define this update in parameter estimates by $\Delta\boldsymbol{\theta}_{\mathbf{iden}_{\boldsymbol{k}}}$.

Step 2: The change in model parameters $\Delta\boldsymbol{\theta}_{\mathbf{k}}$ and the corresponding model correction is, then, calculated such that the updated model predict the gradients of the optimization problem at current input conditions and at the same time, to adjust the prediction error to the same value obtained in the previous step.

The overall update step can be written as:

$$\boldsymbol{\theta}'_{\mathbf{k}} = \boldsymbol{\theta}'_{\mathbf{k-1}} + \Delta\boldsymbol{\theta}_{\mathbf{iden}_{\boldsymbol{k}}} + \Delta\boldsymbol{\theta}_{\mathbf{k}} \tag{4.16}$$

The corrected model with the updated parameter estimates $\boldsymbol{\theta}'_{\mathbf{k}}$ is then optimized for the next iteration. It should be noted, here, that the model correction term is being carried forward into the next iteration and as a result, it has a cumulative effect. The prediction inaccuracies of the model continue to decrease as the model is corrected progressively towards the process optimum. Finally, at the optimum, the corrected model simultaneously satisfies both the identification and objective objectives.

4.3.1 Conditions for Convergence

At a given set of input conditions, let us define a bounded space for model parameters such that $\forall\ \boldsymbol{\theta} \in [\boldsymbol{\theta}_{\mathbf{lb}}, \boldsymbol{\theta}_{\mathbf{ub}}]$:

The corrected model is stable (4.17a)

$\nabla^2\phi(\mathbf{u_k}, \boldsymbol{\theta}'_{\mathbf{k}}) > 0$ (Positive definite) (4.17b)

Then if the bound on truncation error ϵ_{max}^{T} is such that $\forall\ \mathbf{u}_k$:

$$\left|\frac{\partial\phi(\mathbf{y_m}, \mathbf{u_k})}{\partial \mathbf{u}} - \frac{\partial\phi(\mathbf{y}, \mathbf{u_k}, \boldsymbol{\theta}'_{\mathbf{k}})}{\partial \mathbf{u}}\right| < \varepsilon \tag{C.4.1}$$

Where, ε is the tolerance with which the above differences between the measured and predicted gradients of the cost function are minimized, then, the algorithm has a guaranteed convergence towards the process optimum.

Since the update in input conditions $\mathbf{u}_k$ is based on model-based optimization, the algorithm will converge only if:

$$\frac{\partial\phi(\mathbf{y}, \mathbf{u_k}, \boldsymbol{\theta}'_{\mathbf{k}})}{\partial \mathbf{u}} = 0 \tag{4.18}$$

From condition (C.4.1), since the predicted gradients are always matches to the ones measured from the process, the Equation (4.18) holds only when:

$$\left|\frac{\partial\phi(\mathbf{y_m}, \mathbf{u_k})}{\partial \mathbf{u}}\right| < \varepsilon \tag{4.19}$$

4.3.2 Termination Criteria

Let the algorithm converges to a stationary point $\mathbf{u}_{\mathbf{k}}^{*}$. Then, at $\mathbf{u}_{\mathbf{k}}^{*}$:

$$\Delta\boldsymbol{\theta}_{\mathbf{iden}_k} = \mathbf{0} \tag{4.20a}$$

$$\Delta\boldsymbol{\theta}_{\mathbf{k}} = \mathbf{0} \tag{4.20b}$$

Since at convergence $\mathbf{u}_{\mathbf{k}}^{*} = \mathbf{u}_{\mathbf{k-1}}$, the parameter estimates $\boldsymbol{\theta}'_{\mathbf{k-1}}$ minimizing the prediction error at $\mathbf{u}_{\mathbf{k-1}}$ also minimize the prediction error at $\mathbf{u}_{\mathbf{k}}$ and, therefore, the parameter change in **Step 1** is zero. Similarly no further corrections are required for the gradients as they have already been corrected in the previous iteration. Hence, the update in **Step 2** is also zero.

Equations (4.20a) and (4.20b) can then be used to define termination criteria. However, in the presence of measurement noise, the above criteria cannot be exactly achieved. To this end, the convergence of the algorithm can be evaluated in terms of convergence in the probability distribution of model parameters. The difference between the distributions in two successive iterations is calculated using the Kullback-Leibler (K-L) divergence (Cover et al., 1991). If $\mathbf{P_k}$ and $\mathbf{P'_{k-1}}$ are the distributions of parameters $\boldsymbol{\theta'_{k-1}} + \boldsymbol{\Delta\theta_{k}}_{iden}$ and $\boldsymbol{\theta'_{k-1}}$ respectively, the K-L divergence between $\mathbf{P_k}$ and $\mathbf{P'_{k-1}}$ is given by;

$$d(\mathbf{P'_{k-1}} || \mathbf{P_k}) = \int \mathbf{P'_{k-1}}(\boldsymbol{\theta}) \log \frac{\mathbf{P'_{k-1}}(\boldsymbol{\theta})}{\mathbf{P_k}(\boldsymbol{\theta})} \mathrm{d}\boldsymbol{\theta} \tag{4.21}$$

The condition based on Equation (4.20a) is then formulated as:

$$d(\mathbf{P'_{k-1}} || \mathbf{P_k}) \leq \varepsilon_1 \tag{C.4.2}$$

Similarly, for Equation (4.20b), the difference between the distributions of $\boldsymbol{\theta'_{k-1}} + \boldsymbol{\Delta\theta_{k}}_{iden}$ and $\boldsymbol{\theta'_k}$ is measured as $d(\mathbf{P_k} || \mathbf{P'_k})$, where, P'_k is the distribution of $\boldsymbol{\theta'_k}$

$$d(\mathbf{P_k} || \mathbf{P'_k}) \leq \varepsilon_2 \tag{C.4.3}$$

4.3.3 Summary of algorithm

Figure 4.2 presents the flowchart of the algorithm. The algorithm begins with the identification step using the initial inaccurate model structure. For estimation, the problem posed in (P.4.2) is used. Then, the update in model parameters and corresponding model correction is calculated to correct for the model for both identification and optimization objectives by using the minimization problem posed in (P.4.5). The updated model is then optimized for the next operating conditions where the above steps are repeated with the updated model. The procedure is repeated until a termination criteria based on either (4.20a & 4.20b) or (C.4.2 & C.4.3) are satisfied.

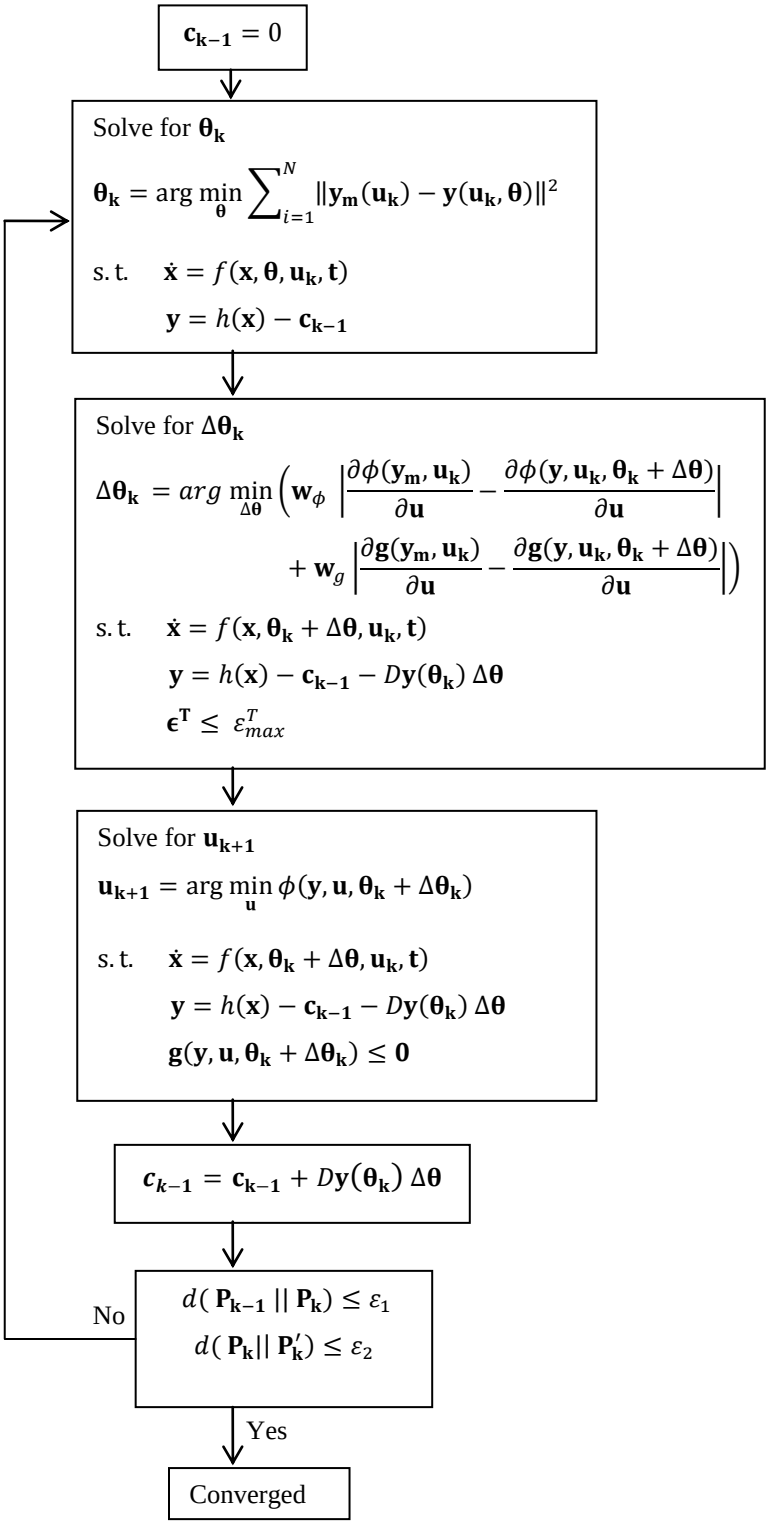


Figure 4-2: Proposed Algorithm with linear model corrections

4.4 Case Study

4.4.1 Problem formulation

The proposed optimization algorithm is applied to a penicillin production process where the goal is to maximize the amount of penicillin at the end of batch. To generate the experimental data for model identification and correcting the model for optimization, the *in-silico* experiments are conducted using a process simulator based on the following set of ordinary differential equations (Bajpai and Reuss, 1980; Birol et al. 2002):

$$\frac{dX}{dt} = \left(\frac{\mu_X SX}{K_X X + S}\right) - \frac{X}{V}\frac{dV}{dt} \tag{4.22}$$

$$\frac{dP}{dt} = \left(\frac{\mu_P SX}{K_P + S + \frac{S^2}{K_I}}\right) - K_H P - \frac{P}{V}\frac{dV}{dt} \tag{4.23}$$

$$\frac{dS}{dt} = -\frac{1}{Y_{X/S}}\left(\frac{\mu_X SX}{K_X X + S}\right) - \frac{1}{Y_{P/S}}\left(\frac{\mu_P SX}{K_P + S + \frac{S^2}{K_I}}\right) - m_X X + \frac{Fs_f}{V} - \frac{S}{V}\frac{dV}{dt} \tag{4.24}$$

$$\frac{dV}{dt} = F - 6.226 * 10^{-4} V \tag{4.25}$$

The set of Equations (4.22)-(4.24) describes the rate of change in the concentrations of biomass (X), penicillin (P) and substrate (S) respectively and Equation (4.25) describes the rate of change in the culture volume (V). The constants in these equations are defined as follows; μ_X is the specific growth rate of biomass, μ_P is the specific rate of penicillin production, K_X and K_P are saturation constants, K_I is a substrate inhibition constant, K_H is a constant representing the rate of consumption of

penicillin by hydrolysis, $Y_{X/S}$ and $Y_{P/S}$ are the yields per unit mass of substrate for the biomass and penicillin respectively, m_X represents the consumption rate of substrate needed for maintaining the biomass and s_f is the concentration of substrate in the feed. The values used for these constants are listed in Table 4.1.

To formulate a model with structural inaccuracy, it is assumed that the user does not have prior knowledge about the consumption of penicillin by hydrolysis and, as a result, the rate of change in the penicillin concentration is modelled as:

$$\frac{dP}{dt} = \left(\frac{\mu_p SX}{K_P + S + \frac{S^2}{K_I}} \right) - \frac{P}{V}\frac{dV}{dt} \tag{4.26}$$

Assuming the dynamics for the other states to be known accurately, the uncertain model is then given by the set of Equations (4.22) and (4.24-4.26). To simplify the numerical calculations, it is further assumed that only two model parameters K_X and K_I will be updated in the algorithm whereas the rest of the model parameters are fixed at their nominal values, estimated at initial input conditions as listed in Table 4.2. The choice of these two parameters as the uncertain ones was based on a preliminary sensitivity analysis.

Table 4-1: Parameters' values for process simulator (Equations 4.22-4.25)

μ_x	K_x	μ_P	K_P	K_I	K_H	$Y_{X/S}$	$Y_{P/S}$	m_X	s_f
0.092	0.15	0.005	0.0002	0.1	0.04	0.45	0.9	0.014	600

Table 4-2: Initial set of input conditions used to estimate the parameters in uncertain model

Biomass Conc. (X_0)	0.1 (g/l)
Substrate Conc. (S_0)	0.1 (g/l)
Product Conc. (P_0)	0 (g/l)
Initial Culture Volume (V_0)	100 (L)
Input Feed (F)	0.04 (L/hr)

The uncertain model is, then, optimized iteratively as per the procedure summarized in Figure 4.2, where the final objective is to maximize the amount of penicillin at the end of batch, subject to a terminal constraint on the culture volume. The initial substrate concentration S_o and the input feed rate F are selected as the decision variables whereas the rest of the input variables are fixed at their initial values listed in Table 4.2. Accordingly, the optimization problem is formulated as follows:

$$\min_{S_o,F} \quad -P\left(\mathbf{x}, \boldsymbol{\theta}, S_o, F, t_f\right)$$

$$s.t. \quad (4.22) \text{ and } (4.24) - (4.26)$$

$$V\left(\mathbf{x}, \boldsymbol{\theta}, S_o, F, t_f\right) \leq V_{max} \qquad \text{(P.4.6)}$$

For reference, the process optimum corresponds to $S_o = 55$ g/l and $F = 0.1728$ L with the final penicillin measured to be $= 592$g

4.4.2 Results and discussion

In the first part of discussion, the convergence properties of the algorithm will be discussed. The performance of the algorithm is evaluated in terms of (1) the rate of convergence and (2) the final converged solution.

The bound on truncation error ϵ_{max}^{T} is the major factor that affects the rate of convergence. Let us recall the calculation of $\Delta\boldsymbol{\theta}_{\mathbf{k}}$. The parameter estimates $\boldsymbol{\theta}_{\mathbf{k}}$ that minimize the prediction error at $\boldsymbol{k}^{th}$ iteration may not predict the gradients of the optimization problem correctly for which a change in estimates $\Delta\boldsymbol{\theta}_{\mathbf{k}}$ is required. Then, in order to ensure that the prediction error is also minimized with the updated parameters, the model outputs are corrected with a term $\mathbf{c}_{\mathbf{k}}$. The larger the change in parameters $\Delta\boldsymbol{\theta}_{\mathbf{k}}$, the more accurately the model predicts the measured gradients of the optimization problem. On the other hand selecting a large $\Delta\boldsymbol{\theta}_{\mathbf{k}}$ will have a negative effect on the model correction, $\mathbf{c}_{\mathbf{k}}$ (Equation (4.14)). Since $\mathbf{c}_{\mathbf{k}}$ is based on the linear approximation of the model around $\boldsymbol{\theta}_{\mathbf{k}}$, as $\Delta\boldsymbol{\theta}_{\mathbf{k}}$ increases, the validity of the linear approximation decreases. Thus, as explained in the previous section, to control the accuracy of the linear approximation, a constraint on $\Delta\boldsymbol{\theta}_{\mathbf{k}}$ is imposed by bounding the truncation error ϵ_{max}^{T}. To summarize, the larger values of ϵ_{max}^{T} will allow for large moves in $\Delta\boldsymbol{\theta}_{\mathbf{k}}$ which favours a faster convergence towards the process optimum but this might increase the prediction error incurred by the model. On the other hand, the smaller values will restrict the moves in $\Delta\boldsymbol{\theta}_{\mathbf{k}}$ to generate better predictions but the model may not be able to correct for the optimization gradients accurately, making the algorithm more sensitive to the modelling error. To illustrate this relative effect, the algorithm is solved for $\epsilon_{max}^{T} = 1\%$ and 5%. The convergence in the optimal S_o for these two scenarios is compared in Figure 4.3. In all the iterations, the input feed rate F converged to the same optimal value ($\sim$0.1728 L) so as to satisfy the volume constraint and, therefore, it is not considered further in the discussion. From Figure 4.3, it is evident that regardless of the choice of ϵ_{max}^{T}, the algorithm eventually converges to the process optimum.

However, the rate of convergence is significantly different for the two cases. In some of the intermediate iterations for $\epsilon_{max}^{T} = 1\%$, the values of $\Delta\boldsymbol{\theta}_{\mathbf{k}}$ are not sufficiently large for the model to predict the optimization gradients in the correct direction, resulting in an oscillatory and much slower convergence. Whereas, for $\epsilon_{max}^{T} = 5\%$, the corrections for the optimization are more accurate as a result of which the algorithm converges much faster. However, this improved convergence is at the cost of prediction accuracy. On comparing the total prediction error for all iterations, it was found that on average, this value is nearly 2.5 times higher for $\epsilon_{max}^{T} = 5\%$.

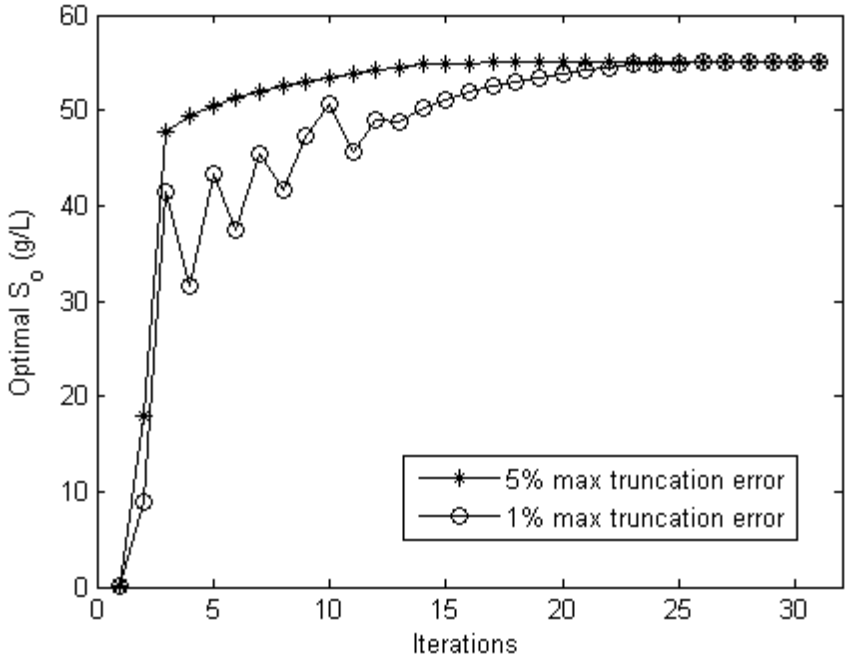


Figure 4-3: Comparison of the effect of ϵ_{max}^{T} on convergence of optimalS_o

The key feature of this algorithm, as discussed in previous sections, is not only the convergence to a process optimum but also that the final corrected model predicts the process behaviour accurately and this is corroborated from Figure 4.4 where the model is used to predict the process variables around the optimum. As can be seen, the predictions are in close agreement with the measurements.

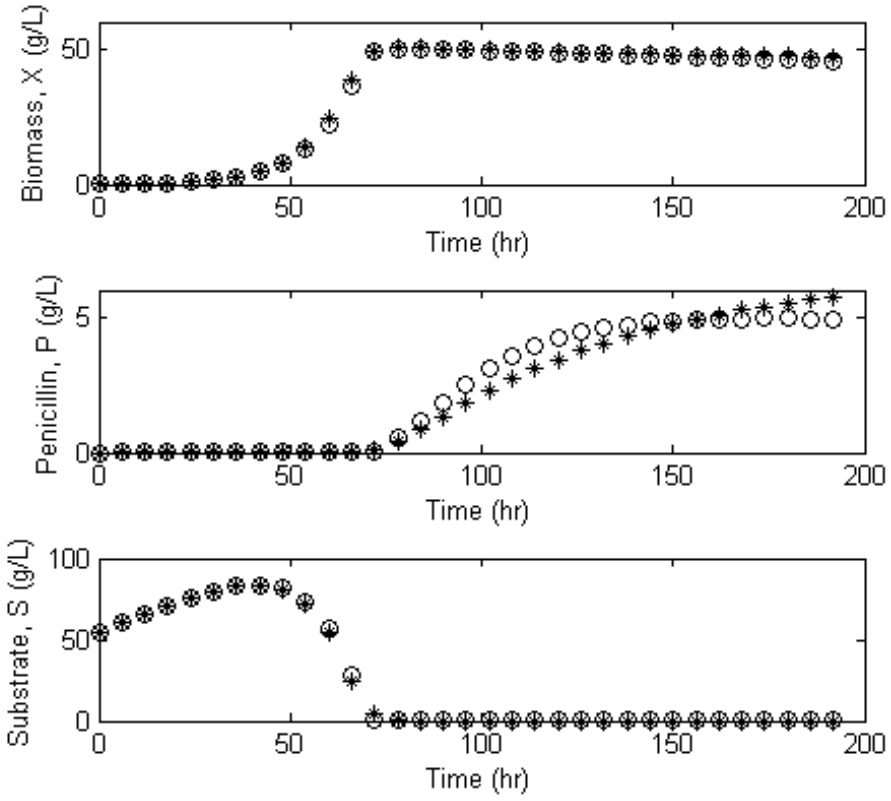


Figure 4-4: Predictions of corrected model at converged optimal solution

(− ∗ − Predictions; −o − Noise-free measurements)

In the next section, the performance of the proposed algorithm is compared with (1) the standard two-step approach and (2) the modifier adaptation algorithms.

4.4.2.1 Comparison with Standard "two-step" approach

The convergence in the optimal S_o corresponding to both "two-step" and proposed methodology is shown in Figure 4.5. Based on these results, where the proposed algorithm converges to the process optimum $S_o \sim 55$ g/l, the two-step approach converges prematurely to $S_o \sim 34$ g/l. The measured penicillin at the end of batch at $S_o \sim 34$ g/l is ~ 445 g which is nearly 25% less as compared to the ~ 592g measured at the true process optimum.

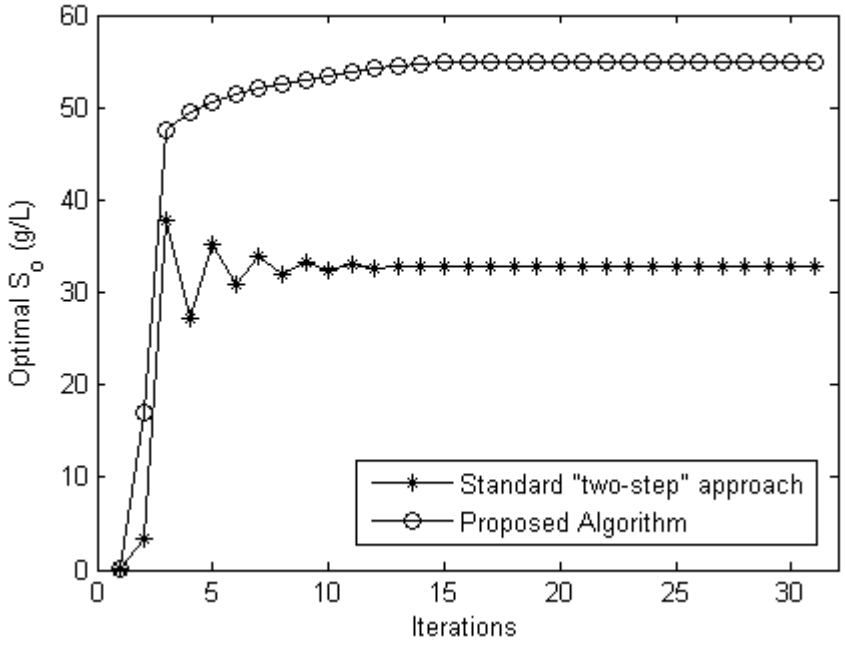


Figure 4-5: Convergence of optimal S_o

The reason for this premature convergence is that the model is inadequate for predicting the true process optimum, or in other words, it cannot satisfy the Equations (4.2a) and (4.2b). For all $S_o > 34$ g/l, the change in model parameters, to compensate for the model structure error, is of such an extent that the predicted gradients no longer drive the optimization objective in the direction of its true optimum. On the other hand, in the proposed algorithm, the prediction error and the differences between the predicted and measured gradients of the optimization problem were both used to update the model, thus correcting for the structural uncertainty along the iterations.

4.4.2.2 Comparison with modifier adaptation algorithms

When compared to the class of modifier adaptation algorithms, the proposed correction in this work offers an added advantage in terms of the rate of convergence. As discussed in previous section, in the modifier adaptation algorithms, the optimization objective and the constraints are corrected by adding the differences between the predicted and measured gradients directly to their respective equations (Problem P.4.4). Since the model is not updated explicitly, the corrected optimization problem may have significant errors in predictions and, generally, this is controlled by filtering the corrections using an empirical filter. For the comparative study, we used first-order exponential filters with three different values for the gain $K = 0.65, 0.5$ and 0.35. Figure 4.6, then, compares the convergence in the optimal S_o for the two algorithms for noise free case. In these results, the optimal S_o corresponding to $K = 0.65$ is highly oscillatory and it is quite clear that, without enough filtering, the gradient corrections are more aggressive, leading to significant prediction errors in the input space. However, as the filter gain is decreased, the convergence is much smoother but at the cost of decreased rate of convergence. On the other hand, the proposed algorithm converges much faster and smoothly to the true optimum. The reason is that the model itself is updated to correct for the gradients in the optimization problem, which provides a model based filtering that have superior

prediction capabilities as compared to the exponential filter in modifier adaptation algorithms.

Finally, the convergence of these algorithms is compared in the presence of measurement noise. The modifier adaptation and the proposed algorithms are each solved 10 times with different realizations of the noise and the performance is evaluated in terms of integral absolute error (IAE) and standard deviation in the optimal S_o, as summarized in Table 4.3. The convergence in the average optimal S_o is shown in Figure 4.7 in the form of error plots. From these results, it is evident that the proposed algorithm is more robust to model errors in the presence of noise. The filter gain that provided smooth convergence in the noise-free situation cannot filter the noise as efficiently as the truncation error (ϵ_{max}^{T}) in the proposed algorithm. When the corrections are added directly to the optimization problem, the effect of noise in gradients on the optimization objective is additive. Whereas, in the proposed algorithm, the noise in gradients affects the parameter estimates $\boldsymbol{\theta_k} + \Delta\boldsymbol{\theta_k}$ but since the optimization problem is not linear with parameters, this effect is not additive.

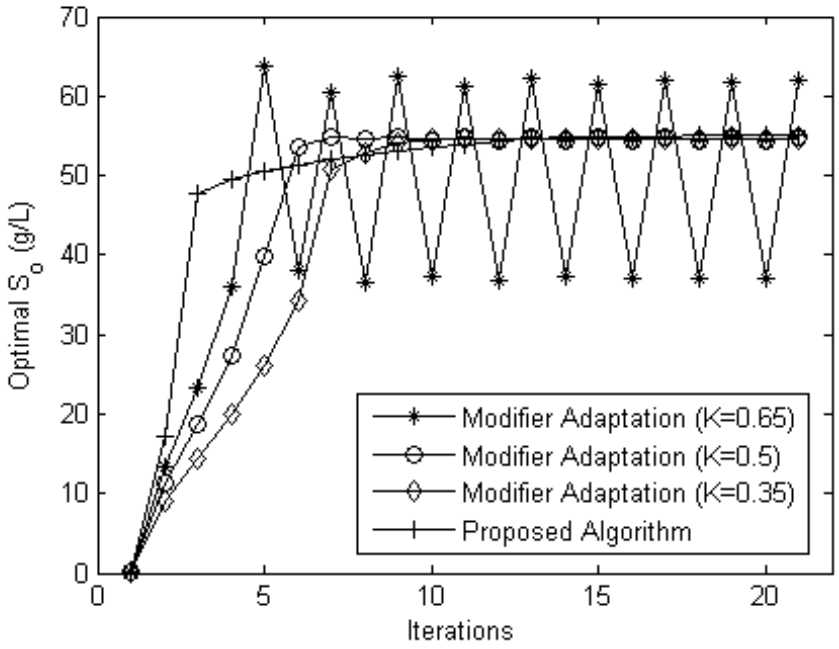
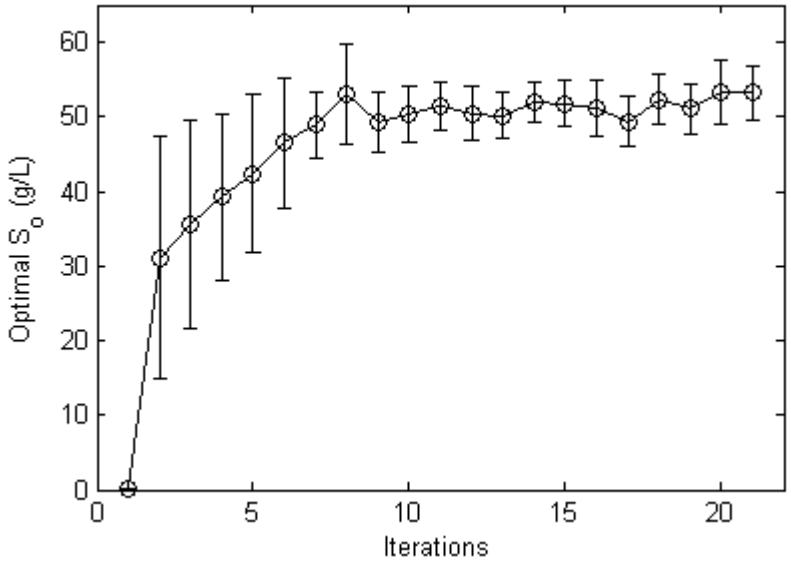
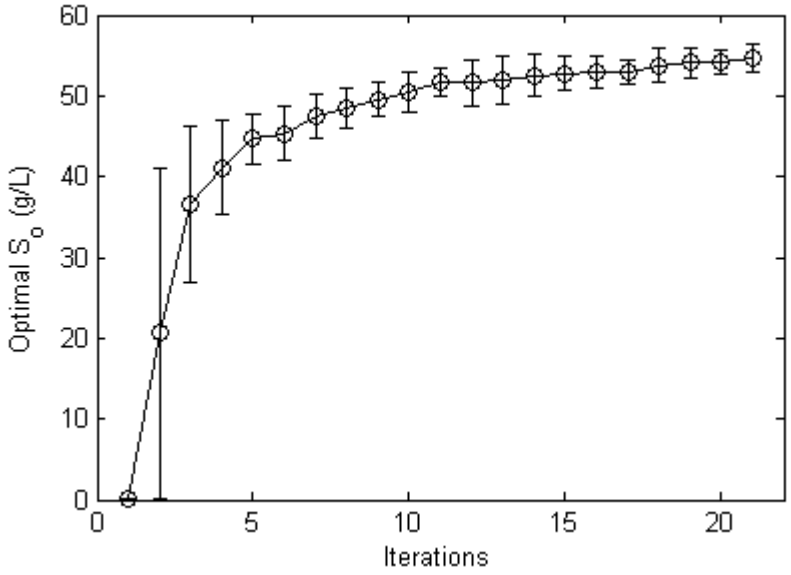


Figure 4-6: Comparison of proposed and modifier adaption algorithms on convergence of optimal S_o



(a)



(b)

Figure 4-7: Average convergence of optimal S_o for (a) Modifier Adaptation algorithm and (b) Proposed algorithm

Table 4-3: Comparison of convergence properties for the proposed algorithm vs modifier adaptation

	IAE	Std. deviation σ
Proposed Algorithm	8.8597	3.5563
Modifier Adaptation	9.5525	5.5805

From Figure 4.7 and Table 4.3, it can be seen that there is a significant amount of variability in the transient phase for the modifier adaptation algorithm. This is partly related to the fact that the model parameters are never updated, in which case the initial uncertainty in their estimates is propagated throughout the iterations. As a result, for each noise realization, the algorithm may have a significantly different search path if the filter gain is low enough to allow for smaller corrections, as seems to be the case in this example. Increasing the gain decreases this variability in transient but it increases the sensitivity of the algorithm to the noise in gradients, resulting in larger oscillations around the optimum as already shown in the noise free case (Figure 4.6).

4.5 Conclusions

An iterative optimization algorithm has been proposed where the process models are corrected iteratively for model-plant mismatch in order to guarantee the convergence to the process optimum. The correction is based on linear approximation of the model and is added in a way that upon convergence, the model not only predicts the process behaviour accurately but also satisfies the process optimality conditions. To achieve this goal, the parameter estimation is performed in two sequential steps where a standard estimation problem to minimize the prediction errors is solved first. Then, in the second step, the differences between the predicted and measured gradients of the optimization problem are minimized. The key advantage of this approach is that it provides a model-based filter which has been shown to outperform the exponential filter needed in previously reported algorithms where the gain is selected ad-hoc. The efficiency of the algorithm is illustrated using a fed-batch bioprocess. The rate of convergence depends on the truncation error, used to validate the linear approximation of the correction. For nonlinear models, this approximation may only be valid over a smaller region, therefore limiting the rate of convergence. To this end, an improved approximation of the model has to be used that will be considered in a future study.

Chapter 5
On improving the convergence of simultaneous model identification and optimization algorithm

(To be submitted)

5.1 Introduction

Model-based optimization provides a systematic framework to improve the profitability of the process while respecting various process constraints, especially those related to safety and environmental regulations. However, its successful implementation rely on the accuracy of underlying process models in terms of three major factors; (1) how close the model represents the actual process, (2) at what operating conditions and how the experiments are to be carried out and (3) the quality of the measurements involved. The standard modelling approach is usually an iterative procedure that begins with the estimation of model parameters using an initial set of experiments. The calibrated model is, then, validated over different operating conditions and if the prediction error is significant, the parameters are updated using additional experimental data. This process is repeated until the model with required prediction accuracy is obtained. The basic idea behind the aforementioned procedure is to learn more precisely about the model parameters from different experiments and to obtain the estimates with minimum possible uncertainty.

If the model structure captures the true dynamics of the process, the above approach may converge to a unique set of parameter values that can predict the process behavior accurately over an entire range of possible operating conditions. This is theoretically possible if a sufficient number of experiments have been performed to accurately identify all the model parameters. However, in most practical

situations, there is usually a restriction on the number of experiments than can be performed, either because of the duration of experiments or budget constraints. Then, if the main goal is to use the model for process optimization, it is desirable to perform a selected limited number of experiments along the direction of process optimum. To this end, a "two-step" approach can be implemented where the operating conditions for the next experiment are, basically, obtained by optimizing the model identified in the previous step. In other words; first, a model is identified around particular operating conditions and, then in the subsequent step, it is used to calculate the optimal operating conditions for the next iteration where the model is re-calibrated using the new measurements. The updated model is re-optimized again and the procedure is repeated until a convergence is obtained (Chen et al., 1987). However, the optimal operating conditions along the search may not provide sufficient excitation required for accurately identifying the model parameters. A possible solution to this problem is to use a combined approach where the next operating conditions are obtained by defining a trade-off between process optimality and a required excitation for proper model identification.

On the other hand, when there is mismatch between the proposed model structure and the actual process, the calibrated model will be accurate only around the region of operating conditions where the parameters are estimated. The reason behind this discrepancy is that, for models with incorrect structure, the reduced set of parameters has to compensate for the unmodelled dynamics. Since these dynamics are basically a varying function of the operating conditions, calibrating the inaccurate model over different operating conditions will result in different parameter estimates. Often, this variability in parameter estimates may be of such an extent that the predicted gradients of the optimization problem no longer coincides with those measured from the process or, in a worst case, they may be of an opposite sign thus driving the optimization away from the actual process optimum. Overall, in these situations, the aforementioned standard "two-step" approach will converge to sub-optimal operating conditions.

To ensure convergence towards the process optimum, it is necessary that the model should predict the optimality conditions of the process accurately (Beigler et al., 1985). To achieve this goal, a class of algorithms has been proposed in literature (Roberts et al., 1979; Tatjewski, 2002; Gao et al., 2005; Chachuat et al., 2009; Marchetti et al., 2009; Costello et al., 2011) where the optimization objective and constraints are adjusted for the differences between their predicted and measured gradients. Since these corrections are added directly to the corresponding functions in the optimization problem, they have to be filtered in order to avoid aggressive corrections so as to obtain a smoother convergence. However, if the process data is very noisy, this may slow down the rate of convergence significantly. In another study (Srinivasan et al., 2002), unlike the standard estimation approach where the parameters are estimated only to correct for the prediction error, the authors also included the difference between the predicted and measured objective functions' gradients as a feedback and, as a result, the parameters were updated to obtain a trade-off between the identification and optimization objectives. If the mismatch between these objectives is significant around the optimum, one of the two objectives has to be compromised.

In Chapter 4, we proposed an alternative approach where a correction is added to the model outputs in a way that, upon convergence, the updated set of model parameters not only minimizes the prediction error but also predicts the gradients of the optimization problem with a minimum error thus simultaneously satisfying the identification and optimization objectives. At a given iteration, the proposed correction was based on the linearization of model outputs around the parameter estimates that minimizes the prediction error. Thus, the correction was given by a linear function of the change in model parameters that also minimizes the difference between predicted and measured gradients of the optimization problem for that iteration. However, this linear approximation limited the change in parameter estimates only to a region where the assumption of linearity is valid. For nonlinear problems it was found that this region could be very small thus limiting the ability to

correct for the gradients of the optimization problem, resulting in slower convergence to the optimum.

In this chapter, the previous study is expanded to improve upon the convergence of the algorithm in two ways; (1) the correction term is considered to be a quadratic function of the model parameters and (2) a robust formulation is considered for the optimization problem where a weighted sum of the nominal performance and its variability due to model uncertainties is minimized. An additional novel contribution of this work is related to the description of parametric uncertainty used in the robust optimization problem. The conventional method for estimating uncertainty in parameter estimates is based on linearization of the model which, for normally distributed measurement errors, results in normally distributed uncertainty in parameter estimates. However, the assumption of linearization may not be valid for the given level of measurement noise, especially in nonlinear problems. Instead, a more accurate approach based on the Bayes Theorem is adopted in this work where linearization assumptions are not required. To this end, a comparative study is also presented where the effect of Bayesian and normal description of parametric uncertainty is compared with respect to the convergence of the algorithm.

The organization of this chapter is as follows. Section 5.2 & 5.3 covers the theory for the proposed modifications with the overall optimization methodology summarized in Section 5.4. The methodology is, then, illustrated using a penicillin production process as a case study in Section 5.5 and finally we conclude the chapter with the future work in section 5.6.

5.2 Model Correction

5.2.1 Preliminaries

The uncertainties in the model, either because of insufficient excitation from the measurements or incorrect model structure, often results in non-optimal operating policies. The standard two-step approach aims to tackle this problem by continuously updating the model parameters at the sub-optimal operating conditions, calculated using the last calibrated model. The first step in this approach is the model identification, where the model parameters are, in general, estimated by minimizing the sum of squared error between the predicted and the measured variables at some given operating conditions.

Mathematically, this can be expressed as follows:

$$\boldsymbol{\theta}_{\mathbf{k}} = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} \|\mathbf{y}_{\mathbf{m}}(\mathbf{u}_{\mathbf{k}}) - \mathbf{y}(\mathbf{u}_{\mathbf{k}}, \boldsymbol{\theta})\|^2$$

$$s.t. \quad \dot{\mathbf{x}} = f(\mathbf{x}, \boldsymbol{\theta}, \mathbf{u}_{\mathbf{k}}, \mathbf{t})$$

$$\mathbf{y} = h(\mathbf{x}) \qquad \text{(P.5.1)}$$

Where, $\mathbf{x} \in \mathbb{R}^{n_x}$ is the vector of model states, $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}$ is the vector of model parameters, $\mathbf{u} \in \mathbb{R}^{n_u}$ is the vector of process input variables, $\mathbf{y} \in \mathbb{R}^{n_y}$ and $\mathbf{y}_{\boldsymbol{m}} \in \mathbb{R}^{n_y}$ are the vectors of predicted and measured output variables, N is the number of time points, $f \in \mathbb{R}^{n_x}$ is a set of differential equations representing the correlation between the model states and the input variables and, finally, $h \in \mathbb{R}^{n_y}$ is a mapping between the model states and the predicted outputs.

Once the parameter estimates are obtained, the model is optimized as follows:

$$\mathbf{u_{k+1}} = \arg\min_{\mathbf{u}} \phi(\mathbf{y}, \mathbf{u}, \boldsymbol{\theta}_\mathbf{k})$$

$$\text{s.t.} \quad \dot{\mathbf{x}} = f(\mathbf{x}, \boldsymbol{\theta}_\mathbf{k}, \mathbf{u}, \mathbf{t})$$

$$\mathbf{y} = h(\mathbf{x})$$

$$\mathbf{g}(\mathbf{y}, \mathbf{u}, \boldsymbol{\theta}_\mathbf{k}) \leq \mathbf{0} \qquad \text{(P.5.2)}$$

Where, ϕ is the objective function (or the cost) to be minimized and $\mathbf{g} \in \mathbb{R}^{n_g}$ is a vector of equality or inequality constraints.

However, when there is a mismatch between the model structure and the actual process, i.e. the set of model equations given by $f(\mathbf{x}, \boldsymbol{\theta}, \mathbf{u_k}, \mathbf{t})$ does not represent the actual process behaviour, the convergence of the two-step approach cannot be guaranteed. Since the unmodelled dynamics are, in general, a varying function of the model states and the inputs, when such model is calibrated against given experimental data, the parameter estimates have to compensate for this variable error. As a result, there is no unique set of parameter values that can satisfy the identification objective (P.5.1) for all the possible realizations of input conditions. It is also possible that the set of parameter estimates that minimizes the prediction error at a given set of operating conditions $\mathbf{u_k}$, may not minimize this error in the neighbourhood of $\mathbf{u_k}$. Therefore, the model may predict incorrectly the gradients of the optimization objective function, i.e.:

$$\frac{\partial \phi(\mathbf{y}, \mathbf{u_k}, \boldsymbol{\theta}_\mathbf{k})}{\partial u_i} \neq \frac{\partial \phi(\mathbf{y_m}, \mathbf{u_k})}{\partial u_i} \qquad (5.1)$$

This is also true for the constraints. In a worst case scenario, there is a possibility that the set of above predicted gradients drives the model-based optimization search away from the process optimum and as a result of that:

$$\phi(\mathbf{y_m}, \mathbf{u_{k+1}}) > \phi(\mathbf{y_m}, \mathbf{u_k}) \tag{5.2}$$

To address these problems, if it is desired to estimate the parameters to predict these gradients correctly, the minimization of the prediction error, solved by problem P.5.1, has to be relaxed to a certain extent. To reduce this contradiction between the identification and optimization objectives, in Chapter 4, we proposed a linear correction term in the model outputs such that it attempts to correct for the modelling error as the algorithm proceeds towards the process optimum. Upon convergence, the updated model parameters not only minimize the difference between the predicted and measured gradients but also the prediction error. In the next sub-section, the theory behind this model correction is summarized and a new correction based on quadratic approximation is proposed which is shown to improve the rate of convergence significantly over the previously proposed linear correction.

5.2.2 Quadratic Model Correction

The standard identification problem (P.5.1) results in the parameter estimates that minimize the sum of squared error between the predicted and measured outputs. Let ϵ_k represents the minimum sum of squared errors corresponding to the parameter estimates $\boldsymbol{\theta_k}$, as follows:

$$\epsilon_k = \sum_{i=1}^{N} \|\mathbf{y_m}(\mathbf{u_k}) - \mathbf{y}(\mathbf{u_k}, \boldsymbol{\theta_k})\|^2 \tag{5.3}$$

Let $\Delta\boldsymbol{\theta_k}$ be the change in parameter estimates $\boldsymbol{\theta_k}$, required to minimize the difference between the predicted and measured gradients of the optimization objective function and the constraints at the current operating conditions. However, with $\boldsymbol{\theta_k} + \Delta\boldsymbol{\theta_k}$, the updated sum of squared errors between the predicted and measured outputs is no longer at its minimum and can be expressed as follows:

$$\epsilon_k' = \sum_{i=1}^{N} \|\mathbf{y_m}(\mathbf{u_k}) - \mathbf{y}(\mathbf{u_k}, \boldsymbol{\theta}_\mathbf{k} + \Delta\boldsymbol{\theta}_\mathbf{k})\|^2 \tag{5.4}$$

To minimize this error, or in other words, to have $\epsilon_k' = \epsilon_k$, a vector of constant corrections $-\mathbf{c_k}$ is introduced to the model outputs in Equation (5.4), such that:

$$\begin{aligned} &\sum_{i=1}^{N} \|\mathbf{y_m}(\mathbf{u_k}) - (\mathbf{y}(\mathbf{u_k}, \boldsymbol{\theta}_\mathbf{k} + \Delta\boldsymbol{\theta}_\mathbf{k}) - \mathbf{c_k})\|^2 \\ &\qquad = \sum_{i=1}^{N} \|\mathbf{y_m}(\mathbf{u_k}) - \mathbf{y}(\mathbf{u_k}, \boldsymbol{\theta}_\mathbf{k})\|^2 \end{aligned} \tag{5.5}$$

To satisfy Equation (5.5), the corresponding error terms on the both sides of the equality should satisfy:

$$\mathbf{y_m}(\mathbf{u_k}) - \mathbf{y}(\mathbf{u_k}, \boldsymbol{\theta}_\mathbf{k} + \Delta\boldsymbol{\theta}_\mathbf{k}) + \mathbf{c_k} = \mathbf{y_m}(\mathbf{u_k}) - \mathbf{y}(\mathbf{u_k}, \boldsymbol{\theta}_\mathbf{k}) \tag{5.6}$$

Following a rearrangement, the correction term can be expressed as:

$$\mathbf{c_k} = \mathbf{y}(\mathbf{u_k}, \boldsymbol{\theta}_\mathbf{k} + \Delta\boldsymbol{\theta}_\mathbf{k}) - \mathbf{y}(\mathbf{u_k}, \boldsymbol{\theta}_\mathbf{k}) \tag{5.7}$$

Using a Taylor Series Expansion, the model outputs with the updated model parameters, $\mathbf{y}(\mathbf{u_k}, \boldsymbol{\theta}_\mathbf{k} + \Delta\boldsymbol{\theta}_\mathbf{k})$ can be expanded around $\boldsymbol{\theta}_\mathbf{k}$ as follows:

$$\begin{aligned} &\mathbf{y}(\mathbf{u_k}, \boldsymbol{\theta}_\mathbf{k} + \Delta\boldsymbol{\theta}_\mathbf{k}) \\ &\qquad = \mathbf{y}(\mathbf{u_k}, \boldsymbol{\theta}_\mathbf{k}) + D\mathbf{y}(\boldsymbol{\theta}_\mathbf{k})\, \Delta\boldsymbol{\theta}_\mathbf{k} + \frac{1}{2}\, \Delta\boldsymbol{\theta}_\mathbf{k}^{\mathrm{T}} H\mathbf{y}(\boldsymbol{\theta}_\mathbf{k})\, \Delta\boldsymbol{\theta}_\mathbf{k} + \cdots \end{aligned} \tag{5.8}$$

Where, D is the Jacobian whose elements are the derivatives of outputs with respect to model parameters and H is the corresponding Hessian

Combining Equations (5.7) and (5.8), the correction term is, then, given by:

$$\mathbf{c_k} = D\mathbf{y}(\boldsymbol{\theta}_\mathbf{k})\, \Delta\boldsymbol{\theta}_\mathbf{k} + \frac{1}{2}\, \Delta\boldsymbol{\theta}_\mathbf{k}^{\mathrm{T}} H\mathbf{y}(\boldsymbol{\theta}_\mathbf{k})\, \Delta\boldsymbol{\theta}_\mathbf{k} + \cdots \tag{5.9}$$

Neglecting terms higher than second order, the above correction can be approximated by the quadratic expansion around $\boldsymbol{\theta_k}$ as follows:

$$\mathbf{c_k} = D\mathbf{y}(\boldsymbol{\theta_k})\,\Delta\boldsymbol{\theta_k} + \frac{1}{2}\,\Delta\boldsymbol{\theta_k}^{\mathrm{T}} H\mathbf{y}(\boldsymbol{\theta_k})\,\Delta\boldsymbol{\theta_k} \tag{5.10}$$

The region where the above quadratic expansion is valid can be calculated by using the relative truncation error as follows:

$$\boldsymbol{\epsilon^T} = \frac{\mathbf{y}(\mathbf{u_k}, \boldsymbol{\theta_k} + \Delta\boldsymbol{\theta_k}) - D\mathbf{y}(\boldsymbol{\theta_k})\,\Delta\boldsymbol{\theta_k} - \frac{1}{2}\,\Delta\boldsymbol{\theta_k}^{\mathrm{T}} H\mathbf{y}(\boldsymbol{\theta_k})\,\Delta\boldsymbol{\theta_k} - \mathbf{y}(\mathbf{u_k}, \boldsymbol{\theta_k})}{\mathbf{y}(\mathbf{u_k}, \boldsymbol{\theta_k})} \tag{5.11}$$

The calculation of the change in model parameter, $\Delta\boldsymbol{\theta_k}$ is, then, calculated from an optimization problem as follows:

$$\Delta\boldsymbol{\theta_k} = \arg\min_{\Delta\boldsymbol{\theta}} \left(\mathbf{w}_\phi \left| \frac{\partial\phi(\mathbf{y_m}, \mathbf{u_k})}{\partial\mathbf{u}} - \frac{\partial\phi(\mathbf{y}, \mathbf{u_k}, \boldsymbol{\theta_k} + \Delta\boldsymbol{\theta})}{\partial\mathbf{u}} \right| + \mathbf{w}_g \left| \frac{\partial\mathbf{g}(\mathbf{y_m}, \mathbf{u_k})}{\partial\mathbf{u}} - \frac{\partial\mathbf{g}(\mathbf{y}, \mathbf{u_k}, \boldsymbol{\theta_k} + \Delta\boldsymbol{\theta})}{\partial\mathbf{u}} \right| \right)$$

$$\text{s.t.} \quad \dot{\mathbf{x}} = f(\mathbf{x}, \boldsymbol{\theta_k} + \Delta\boldsymbol{\theta}, \mathbf{u}, \mathbf{t})$$

$$\mathbf{y} = h(\mathbf{x}) - \boldsymbol{D}\mathbf{y}(\boldsymbol{\theta_k})\,\Delta\boldsymbol{\theta} - \frac{1}{2}\,\Delta\boldsymbol{\theta_k}^{\mathbf{T}} H\mathbf{y}(\boldsymbol{\theta_k})\,\Delta\boldsymbol{\theta_k}$$

$$\boldsymbol{\epsilon}^{\mathbf{T}} \leq \epsilon_{max}^{T} \tag{P.5.3}$$

Where, $\mathbf{w}_\phi$ and $\mathbf{w}_g$ are weights for the objective function and the constraints' gradients respectively and ϵ_{max}^{T} is a limit on the relative truncation error representing the required degree of accuracy for the quadratic approximation of the model.

To summarize, the estimation of parameters procedure is divided into two steps. In the first step, the estimates are obtained independent of the optimization objectives to minimize the prediction error at the given operating conditions. In the subsequent step, the change in these estimates is computed such that the model predicts the measured gradients of the optimization problem with a minimum possible error and, at the same time, a quadratic correction is added to the model so as to maintain the prediction error to its minimum value from the first step. The updated model with the parameter estimates $\boldsymbol{\theta}_{\mathbf{k}}'$ is, then, optimized for the optimal operating conditions where the model is re-calibrated with the new correction, followed by re-optimization. This procedure is repeated until convergence to the process optimum is achieved. The termination criteria and the conditions for guaranteed convergence are the same as presented in Chapter 4.

5.3 Robust Optimization

The rate at which the algorithm will converge highly depends on the change in parameter estimates $\Delta\boldsymbol{\theta}_{\mathbf{k}}$ along the iterations. For faster convergence, it is desirable to have larger values for $\Delta\boldsymbol{\theta}_{\mathbf{k}}$ but this is restricted by the relative truncation error ϵ_{max}^{T} which restricts the values of $\Delta\boldsymbol{\theta}_{\mathbf{k}}$ to a region where the proposed model correction is assumed to be valid. The new correction, as proposed above, addresses this problem by increases this region but it does not explicitly address the robustness in the presence of noise. To this end, a robust optimization problem is proposed where a weighted sum of the nominal objective and its variability is minimized.

5.3.1 Parametric Uncertainty

The effect of modelling error and the measurement noise is usually expressed by computing the confidence region for the parameter estimates. In principle, this region represents the probabilities by which the different parameter estimates predict the given experimental data. The conventional method to compute this region is based on

linearization of model around the parameter estimates. Then, for normally distributed measurement errors, the resulted parametric uncertainty is also normal and can be expressed by the following hyper-ellipsoid (Beck et al., 1977):

$$E_\theta = \{\theta: (\boldsymbol{\theta} - \boldsymbol{\theta}_\mathbf{k})^T \boldsymbol{V}_{\boldsymbol{\theta}}^{-\mathbf{1}} (\boldsymbol{\theta} - \boldsymbol{\theta}_\mathbf{k}) \leq \chi^2_{n_\theta}(\alpha) \tag{5.12}$$

Where, $\boldsymbol{V}_{\boldsymbol{\theta}} \in \mathbb{R}^{n_\theta \, \mathrm{x} \, n_\theta}$ is a covariance matrix of the measurement errors and $\chi^2_{n_\theta}(\alpha)$ is the chi-square distribution with n_θ degrees of freedom and α confidence level. This description is, however, valid only if either the model is linear or the uncertain parameter region is small enough that the underlying assumption of linearity is valid. For nonlinear problems, a more generic approach is required. In this study, it is proposed to compute the parametric distribution using the Bayes' theorem as follows:

$$\mathbf{p}(\boldsymbol{\theta}|D) = \frac{L(\boldsymbol{\theta}|D)\, \mathbf{p}(\boldsymbol{\theta})}{\int L(\boldsymbol{\theta}|D)\, \mathbf{p}(\boldsymbol{\theta}) d\boldsymbol{\theta}} \tag{5.13}$$

Here, $\mathbf{p}(\boldsymbol{\theta}|D)$ is the posterior probability of the parameters, conditional on the given set of measurements *D*, $\mathbf{p}(\boldsymbol{\theta})$ is the prior probability of parameters, representing any information about the parameters available *a priori* and $L(.)$ represents the likelihood of parameters which basically defines the error distribution.

Assuming the errors between the output measurements and predictions to be independent and normally distributed, the likelihood function is given by a n_θ-dimensional multivariate normal distribution as follows:

$$L(\boldsymbol{\theta}|D) = \frac{1}{(2\pi)^{\mathrm{k}/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}\left(\mathbf{y_m} - \mathbf{y}(\boldsymbol{\theta})\right)^T \Sigma^{-1} \left(\mathbf{y_m} - \mathbf{y}(\boldsymbol{\theta})\right)\right) \tag{5.14}$$

In this study, the effect of both the normal and the Bayesian parametric distributions is investigated. Since the procedure requires the likelihood function to be solved multiple times over the parameter space, using a full nonlinear model is usually computationally expensive. Instead, an approach based on Polynomial Chaos

(PC) expansions is used. Since the Bayesian Inference considers the model parameters as well as the model outputs as random variables, the PC expansions can be used to build a surrogate representation of the model outputs as a function of parameters by propagating the prior parametric distribution into the model outputs, as shown in chapter 3. Once the uncertainty description is obtained, the PC expansions are used to propagate this uncertainty into the optimization objective function to compute its variability.

5.3.2 Polynomial Chaos Expansions

In this sub-section, a brief background for PC-based uncertainty propagation is presented.

Let us define a probability space $(\Omega, \mathcal{F}, P)$, where Ω is the sample space, $\mathcal{F}$ is the σ-algebra over Ω and P is a probability measure on $\mathcal{F}$. If $\{\xi_i(\omega)\}_{i=1}^{\infty}$ is a set of independent random variables with a standard probability distribution, then, the PC expansion of any random variable *X* with a finite variance will be as follows:

$$X(\omega) = a_0\Gamma_o + \sum_{i_1=1}^{\infty} a_{i_1}\Gamma_1(\xi_{i_1}) + \sum_{i_1=1}^{\infty}\sum_{i_2=1}^{i_1} a_{i_1 i_2}\Gamma_2(\xi_{i_1}, \xi_{i_2}) + \sum_{i_1=1}^{\infty}\sum_{i_2=1}^{i_1}\sum_{i_3=1}^{i_2} a_{i_1 i_2 i_3}\Gamma_3(\xi_{i_1}, \xi_{i_2}, \xi_{i_3}) + \cdots \quad (5.15)$$

Where, Γ_p is the basis function of the order p (Ghanem et al., 1991), ω is the random event and $a_{(.)}$ is the corresponding deterministic coefficient. The expansion can be expressed in more compact form (Ghanem et al., 1991) as:

$$X(\omega) = \sum_{k=0}^{\infty} \hat{a}_k \Psi_k(\xi_1, \xi_2, \xi_3, \dots) \quad (5.16)$$

The fundamental property of a PC expansion is that all basis functions are orthogonal to each other, i.e.

$$\langle \Psi_i \Psi_j \rangle = \int \Psi_i(\boldsymbol{\xi}) \Psi_j(\boldsymbol{\xi}) p(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} = \delta_{ij} \langle \Psi_i^2 \rangle \tag{5.17}$$

The coefficients in the expansion can, then, easily be computed using Galerkin projections as follows:

$$\hat{a}_k = \frac{\langle X \, \Psi_k \rangle}{\langle \, \Psi_k^2 \rangle} \tag{5.18}$$

To propagate the effect of parametric uncertainty into the desired variable, the first step is to formulate the PC expansions for the uncertain model parameters. To compute the expansion coefficients, a one-to-one mapping between the set of parameters $\boldsymbol{\theta}$ and the set of independent random variables $\boldsymbol{\xi}$ is needed. For the case of uncorrelated parameters, this can be achieved by an inverse transformation as follows:

$$\xi_i = F^{-1}\left(\int_0^{\theta} p(\theta_i) d\theta \right) \tag{5.19}$$

Where, F^{-1} is the inverse of the cumulative density function for the independent random variable ξ_i and $p(\theta_i)$ is the probability of the model parameter θ_i.

For the case of correlated parameters, the following transformations based on conditional and marginal probabilities can be applied:

$$\xi_1 = {F_1}^{-1}\left(\int_0^{\theta_1} p(\theta_1 | D) d\theta_1 \right) \tag{5.20}$$

$$\xi_2 = {F_2}^{-1}\left(\int_0^{\theta_2} p(\theta_2 | \theta_1, D) \, d\theta_2 \right) \tag{5.21}$$

The formulation of PC expansion for uncorrelated parameters is straightforward where the mapping by Equation (5.19) can be used to compute the coefficients of the respective expansions. However, for the case of correlated parameters, the mapping for the parameters based on conditional probabilities will result in PC expansions that are conditional on the other parameters. For such situation, the approach presented in our recent work (Mandur et al. 2013b) can be applied where the case of two correlated parameters is illustrated. Once the PC expansions for the uncertain parameters are formulated, to propagate this uncertainty into the desired variable, e.g. the optimization cost, a map is built between the desired variable and the independent random variables as follows:

1. Select the value of independent random variables $\boldsymbol{\xi}$ at the required collocation points

2. Calculate the set of parameter values from their PC expansions

3. Solve the nonlinear model with this set of parameter values for the desired output.

Using this map, the coefficients in the PC expansion of the desired variable can be calculated from Equation (5.18).

Due to the orthogonality of basis functions (equation), the variability of the output variable can be calculated by an analytical expression as follows:

$$Var(X) = \sum_{i=1}^{P} \hat{a}_i^2 < \Psi_i^2 > \tag{5.22}$$

5.4 Proposed optimization methodology

The overall methodology is summarized by the flowchart in Figure 5.1. The algorithm begins with the identification step with the objective function as posed in P.5.1 and the initial model correction term as zero. The change in model parameters and the new model correction is, then, calculated using the problem P.5.3. With the updated model parameters, the nominal or robust optimization problem is solved. Then, the procedure is repeated with the updated model corrections at new optimal operating conditions until it converges to a steady state.

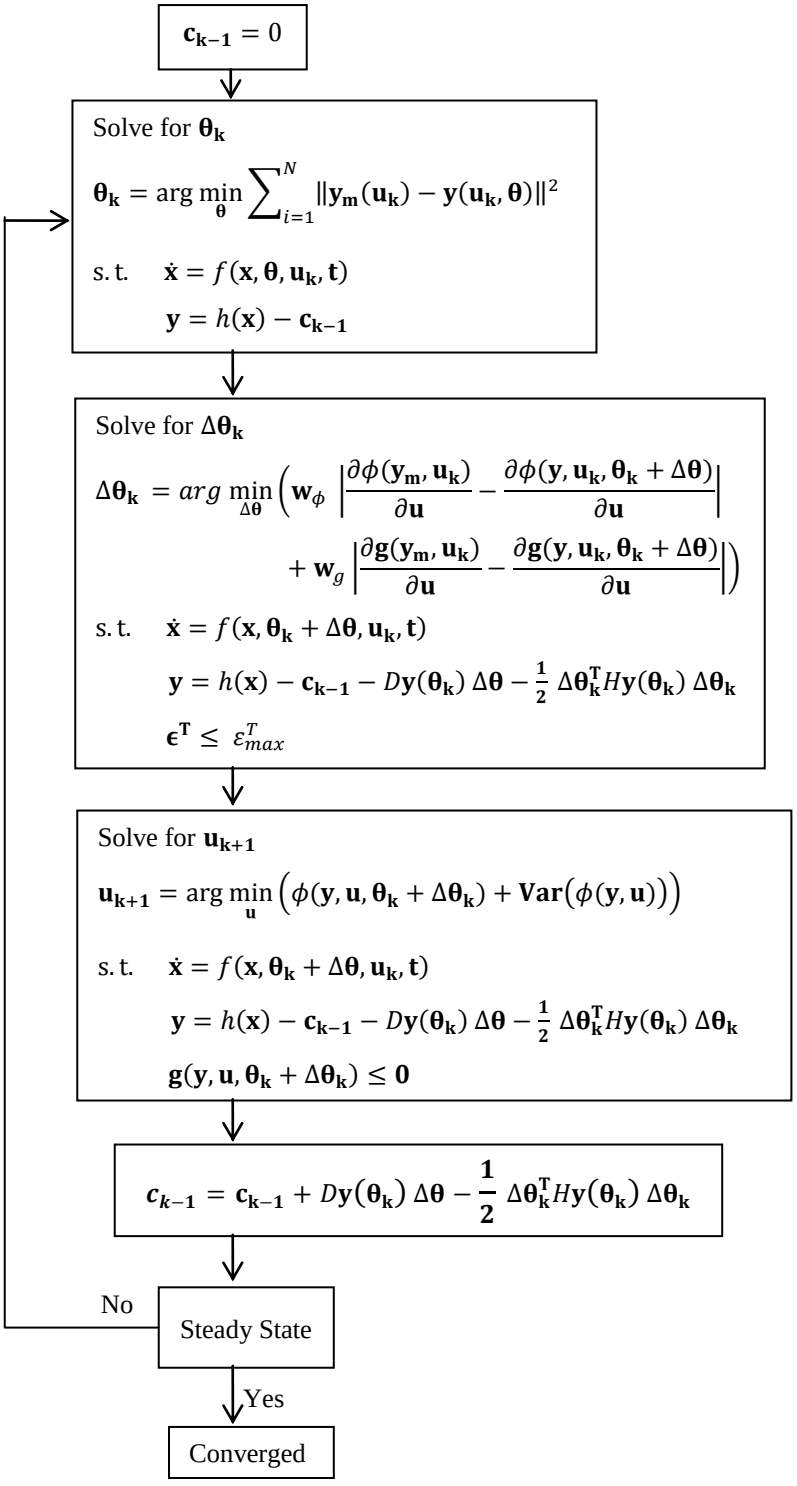


Figure 5-1: Proposed Algorithm with quadratic correction and robust formulation

5.5 Case Study

The proposed algorithm is illustrated on a fed-batch bioprocess for penicillin production. The actual process is considered to be described by a system of differential equations ((5.23)-(5.26)) as follows (Bajpai and Reuss, 1980; Birol et al. 2002):

$$\frac{dX}{dt} = \left(\frac{\mu_X SX}{K_X X + S}\right) - \frac{X}{V}\frac{dV}{dt} \tag{5.23}$$

$$\frac{dP}{dt} = \left(\frac{\mu_P SX}{K_P + S + \frac{S^2}{K_I}}\right) - K_H P - \frac{P}{V}\frac{dV}{dt} \tag{5.24}$$

$$\frac{dS}{dt} = -\frac{1}{Y_{X/S}}\left(\frac{\mu_X SX}{K_X X + S}\right) - \frac{1}{Y_{P/S}}\left(\frac{\mu_P SX}{K_P + S + \frac{S^2}{K_I}}\right) - m_X X + \frac{F s_f}{V} - \frac{S}{V}\frac{dV}{dt} \tag{5.25}$$

$$\frac{dV}{dt} = F - 6.226 * 10^{-4} V \tag{5.26}$$

Where, X, P and S represent the concentrations of biomass, penicillin and substrate respectively and V is the culture volume. The rate constants and other parameters are defined as follows: μ_X and μ_P are specific growth rates for the biomass and penicillin respectively with K_X and K_P as respective saturation constants, K_I represents the substrate inhibition in the growth kinetics of the penicillin, K_H accounts for the consumption of penicillin by hydrolysis, $Y_{X/S}$ and $Y_{P/S}$ are the yields per unit mass of substrate for the biomass and penicillin respectively, m_X accounts for the consumption of substrate in maintaining the biomass and, finally, s_f is the concentration of substrate in the feed.

The above Equations ((5.23)-(5.26)) are used to generate *in silico* data for the output variables as well as for the gradient of the cost function in the optimization problem.

For the process model, the consumption of penicillin by hydrolysis occurring in the actual process is assumed to be unknown to the user. Accordingly, the rate of change in the penicillin concentration is inaccurately modelled, as:

$$\frac{dP}{dt} = \left(\frac{\mu_p S X}{K_P + S + \frac{S^2}{K_I}} \right) - \frac{P}{V}\frac{dV}{dt} \tag{5.27}$$

Assuming the dynamics of the other states to be known accurately, the set of Equations (5.23) and (5.25)-(5.27), then, represents the inaccurate model of the process to be used in the proposed algorithm.

The optimization objective is to maximize the amount of penicillin at the end of batch by manipulating the initial substrate concentration S_o and the input feed rate F while ensuring that the culture volume does not exceed a maximal volume of 120L.

Mathematically, the problem is formulated as:

$$\begin{aligned} \min_{S_o, F} \quad & -P\left(\mathbf{x}, \boldsymbol{\theta}, \mathbf{u}, S_o, F, t_f\right) \\ s.t. \quad & (5.23)\ and\ (5.25 - 5.27) \\ & V\left(\mathbf{x}, \boldsymbol{\theta}, \mathbf{u}, S_o, F, t_f\right) \leq V_{max} \end{aligned} \tag{P.5.4}$$

The algorithm starts with a parameter estimation step at initial input conditions listed in Table 5.1. Here, only two parameters K_X and K_I are updated in the algorithm whereas the remaining parameters are kept at their initial values, that were estimated in the first iteration with the incorrect model. The reasons for selecting a subset of parameters are; (1) to reduce the sensitivity to noise and (2) to reduce the

computational load in the robust optimization problem. It has been shown that as the number of parameters increase, the propagation of parametric uncertainty using PC expansions needs more model runs, thus increasing the overall computational time since this propagation step has to be repeated several times in optimization framework (Mandur et al., 2013b). In situations where the optimal inputs provide insufficient excitation to estimate all the parameters, updating only a subset of parameters becomes even more relevant. However, this is beyond the scope of this study.

Table 5-1: Set of initial input conditions

Biomass Conc. (X_0)	0.1 (g/l)
Substrate Conc. (S_0)	0.1 (g/l)
Product Conc. (P_0)	0 (g/l)
Initial Culture Volume (V_0)	100 (L)
Input Feed (F)	0.04 (L/hr)

5.5.1 Results and discussion

As mentioned in the introduction, the key motivation behind this work was to improve the convergence of the previously proposed algorithm (c.f. Chapter 4) in two directions: (1) by using a new model correction term based on quadratic expansion of the model and (2) by addressing robustness in the optimization problem to reduce the effect of modelling errors and the noise.

The effect of the model correction is discussed first. It is worth mentioning, here, that in all optimizations the optimal feed rate F always converged to a value of 0.1728 L/hr satisfying the constraint on culture volume and, therefore, it will not be considered in the following discussion. At any iteration, the trade-off between the identification and optimization objectives is obtained by the proposed model correction and the maximum truncation error (ϵ_{max}^{T}), for which this correction is assumed to be valid. It was hypothesized that, for highly nonlinear models, the linear approximation used in our previous work may be valid only in a small neighbourhood of the operating point. Thus, when using linear corrections, a small value of the truncation error bound ϵ_{max}^{T} had to be used to improve the predictive accuracy in the transients which limited the extent to which correction for the optimization gradients can be made. To test the effect of a quadratic correction, the algorithm is solved using both quadratic and linear corrections for $\epsilon_{max}^{T} = 1\%$ and the corresponding convergence of the optimal S_o is compared in Figure 5.2. It is clear from this figure that with the linear correction, the algorithm is not even able to match the sign of the predicted and measured gradients for some of the iterations, thus, leading to an oscillatory profile during the transient. On the other hand, the quadratic approximation of the given model allows for much larger changes in the parameter estimates in all the iterations, increasing the ability to fit the measured optimization gradients more accurately. As a result, the new correction resulted in a smooth and much faster convergence towards the process optimum. In these results, the

measurements were assumed to be noise free so as to evaluate the performance of the algorithm when only the modelling error is present.

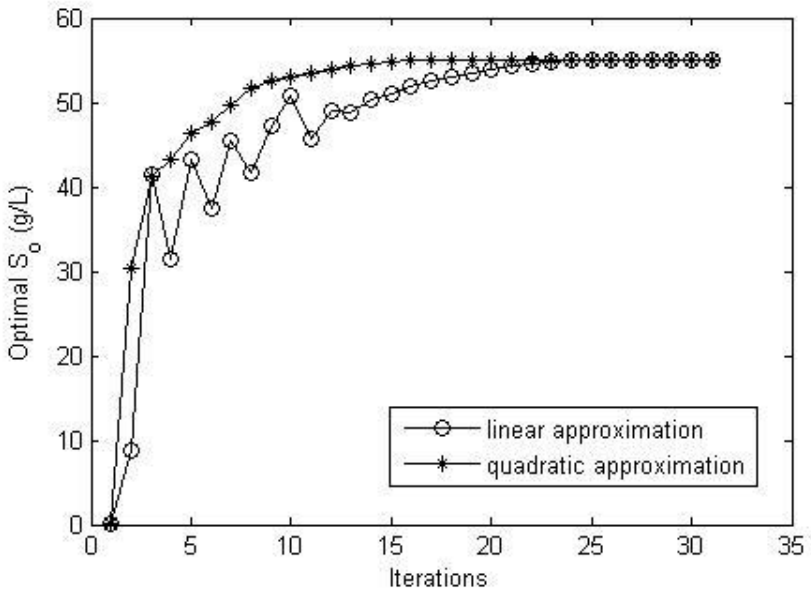
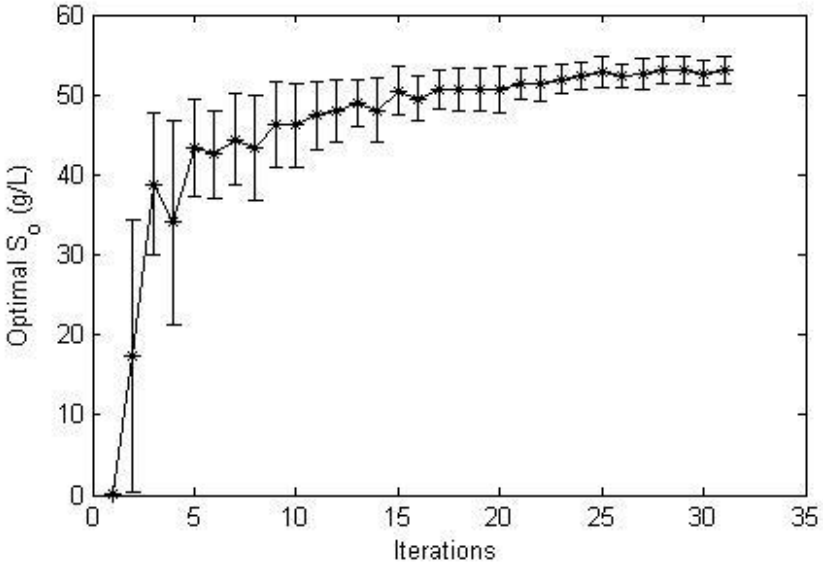


Figure 5-2: Comparing the effect of linear vs. quadratic model correction on the convergence of optimal S_o

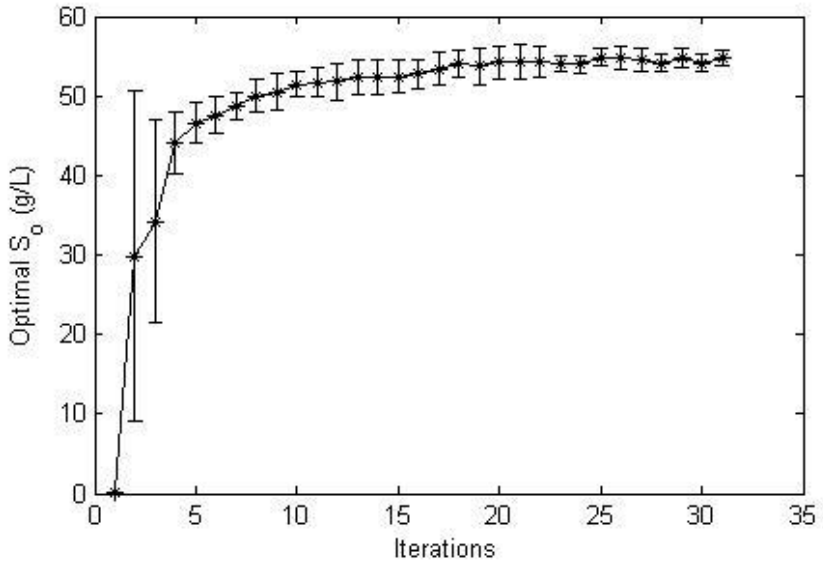
In the next comparison, the measured outputs as well as the gradients of the optimization problem are assumed to be corrupted by Gaussian noise. Comparisons are conducted for both types of corrections, i.e. linear and quadratic, and also for two different levels of truncation error. Since the noise realizations are random, the algorithm is solved for each case 10 times with different realizations of the noise. Figure 5.3 shows the convergence of the average optimal value of the inlet substrate concentration S_o along with the associated variability in the search path in each case. The effectiveness of the algorithm is, then, evaluated in terms of (1) the integral absolute error (IAE) between the predicted and the actual optimal S_o and (2) the total variability in the predicted optimal S_o, summarized in Table 5.2.

Table 5-2: Comparison of model correction based on linear vs quadratic approximation

	$\epsilon_{max}^{T} = 1\%$		$\epsilon_{max}^{T} = 5\%$	
	IAE	Std. deviation σ	IAE	Std. deviation σ
Linear approximation	8.7292	4.0109	6.3651	2.8862
Quadratic approximation	5.7419	2.7338	4.8090	2.3256

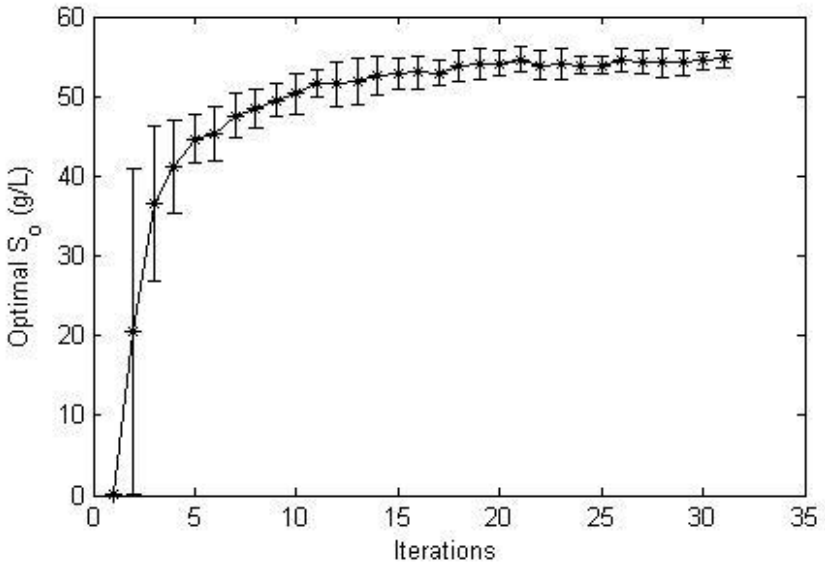


(a)

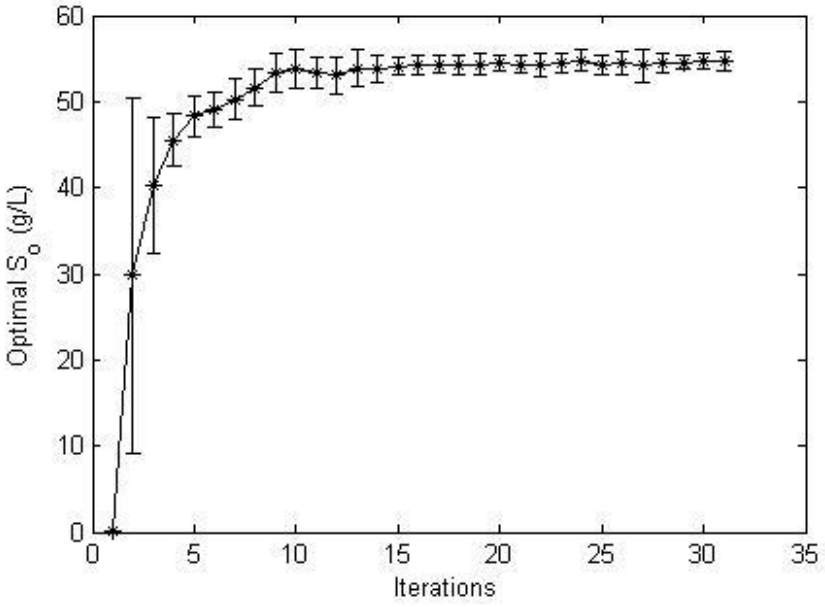


(b)

Figure 5-3: Convergence of optimal S_o based on (a) linear model correction and $\epsilon_{max}^T = 1\%$, (b) quadratic model correction and $\epsilon_{max}^T = 1\%$, (c) linear model correction and $\epsilon_{max}^T = 5\%$ and (d) quadratic model correction and $\epsilon_{max}^T = 5\%$



(c)



(d)

Figure 5-3: Convergence of optimal S_o based on (a) linear model correction and $\epsilon_{max}^T = 1\%$, (b) quadratic model correction and $\epsilon_{max}^T = 1\%$, (c) linear model correction and $\epsilon_{max}^T = 5\%$ and (d) quadratic model correction and $\epsilon_{max}^T = 5\%$

Compared to the linear approximation of the model, the proposed correction based on the quadratic approximation resulted in a significant reduction in both IAE and variability of the optimal S_o in all cases. For $\epsilon_{max}^{T} = 5\%$, these reductions are nearly 24.45 % and 19.42% respectively whereas for $\epsilon_{max}^{T} = 1\%$, the numbers are much higher with the IAE reduced by nearly 34.22 % and the variability by nearly 31.84 %. These results are critical since, in a practical situation, a smaller IAE and variability will provide higher confidence to plant personnel that the algorithm is actually converging to a process optimum rather than changing at random due to process variability and measurement noise.

To better understand how the measurement noise affects the convergence let us recall the parameter estimation procedure and the role of the truncation error. The estimation, basically, involves two sequential steps. In the first step, the parameters' and states' estimates are obtained by minimizing the prediction error with the model from previous iteration. The uncertainty in these estimates originates from a combination of modelling errors and the noise in measured outputs used in the identification objective. Then, in the second step, a change in these estimates is calculated such that the model can predict the optimization gradients more accurately. To ensure that the model still minimizes the prediction error, a correction to the model is added which is based on an approximation around the previous estimates. The change in the estimates is limited to a region where this approximation is valid. Here, it can be observed that allowing for a larger change will make the estimates more sensitive to the noise in gradients whereas allowing for a smaller change will make them more sensitive to the modelling errors and the noise in measured outputs.

Out of the four cases, the correction based on linear approximation and 1% truncation error allows for the smallest change and, as a result, the estimates are highly affected by both modelling errors and the noise in measured outputs. With the correction based on quadratic approximation, the change is of such an extent that the effect of modelling errors is reduced significantly, giving much higher reduction in

the variability of optimal S_o. From these results, it can be concluded that, for the same level of prediction accuracy, the proposed quadratic correction provides much better robustness to noise and modelling error as compared to a linear correction Although, the new correction term has an added computational load because of the calculation of Hessian matrix, this increase has a marginal effect on the overall computational time which is dominated by the optimization step.

In the second approach, the sensitivity of the algorithm to noise and modelling error is reduced by explicitly adding a measure of robustness to the optimization objective. For the given case study, a robust optimization problem is formulated as follows:

$$\begin{aligned} \min_{S_o,F} \quad & -\left[P\left(\mathbf{x},\mathbf{u},S_o,F,t_f\right) - w\ \mathrm{Var}\left(P\left(\mathbf{x},\mathbf{u},S_o,F,t_f\right)\right)\right] \\ s.t. \quad & (5.23)\ and\ (5.25-5.27) \\ & \left\|V\left(\mathbf{x},\boldsymbol{\theta},\mathbf{u},S_o,F,t_f\right)\right\|_\infty \leq V_{max} \end{aligned} \tag{P.5.5}$$

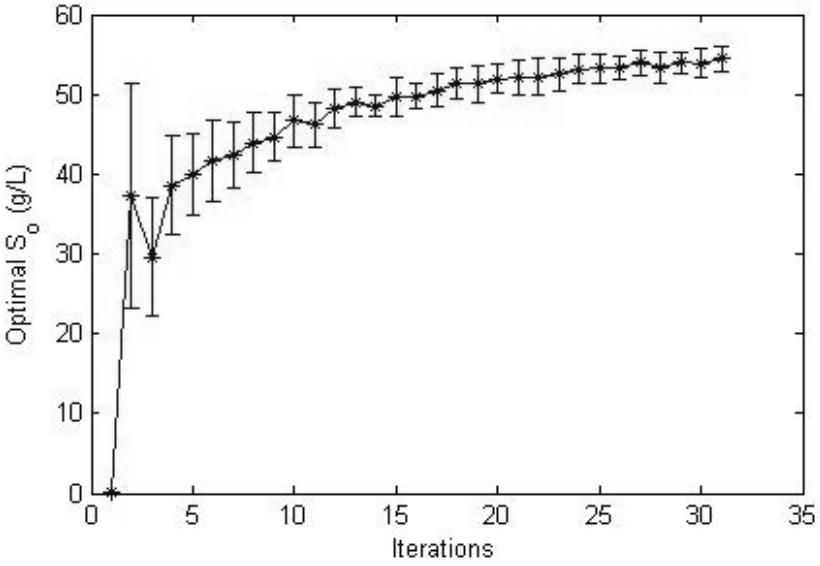
Where, Var represents the variability in the amount of penicillin at final time averaged over the uncertain parameter region and *w* is the weight on the robustness. Since, in the parameter estimation step, the parameter estimates are obtained such that the model can predict the measured gradient of the nominal objective function, the above formulation will promote a trade-off between the corresponding nominal performance and its variability.

The results using a normal distribution for parametric uncertainty will be discussed first. Since this uncertainty description is much easier to compute, it is the most commonly used description in probabilistic uncertainty analysis. The robust optimization problem is, then, solved 10 times with different realizations of noise for both 1% and 5% truncation error and $w = 1\%$. Since one of the motivations was also to compare this approach with the updated model correction, the above problem is solved with the previously proposed linear correction. The resulting convergence of

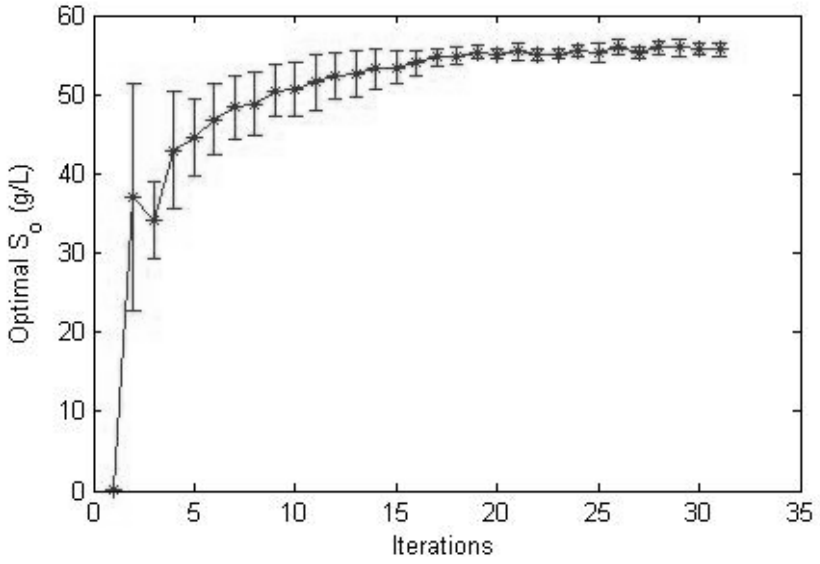
the average optimal S_o in successive runs is shown in Figure 5.4 with the corresponding IAE and variability in the optimal S_o listed in Table 5.3. It is evident from these results that the variability is significantly reduced in both cases with nearly 25.83% for $\epsilon_{max}^{T} = 1\%$ and 10.8% for $\epsilon_{max}^{T} = 5\%$.

Table 5-3: Comparison of nominal vs. robust iterative optimization (Linear model correction)

	$\epsilon_{max}^{T} = 1\%$		$\epsilon_{max}^{T} = 5\%$	
	IAE	Std. deviation σ	IAE	Std. deviation σ
Nominal Optimization	8.7292	4.0109	6.3651	2.8862
Robust Optimization	8.1793	2.9748	5.6249	2.5745



(a)



(b)

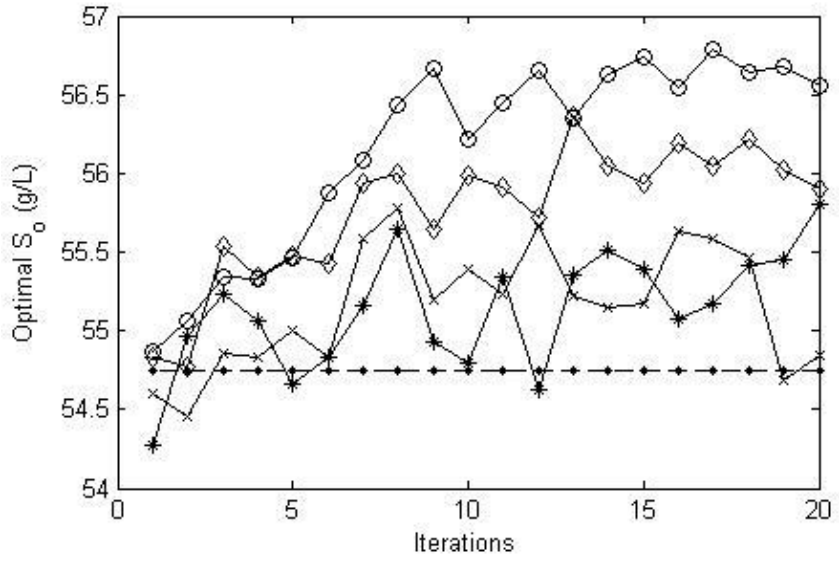
Figure 5-4: Robust convergence of optimal S_o for linear model correction and (a) $\epsilon_{max}^{T} = 1\%$ and (b) $\epsilon_{max}^{T} = 5\%$

Based on these results, the proposed quadratic correction provides better improvements with respect to the overall variability (see Table 5.2). This difference is attributed to the way the two approaches handle the model uncertainties. By allowing a larger change in parameter estimates, the quadratic correction aims to reduce the effect of modelling error and the noise in measured outputs but, at the time, it increases the sensitivity of the algorithm towards the noise in gradients. On the other hand, the robust approach aims to reduce the variability irrespective of its source. For this, we compared the performance of both approaches when the algorithm reaches the steady state and it was observed that, in this region, for $\epsilon_{max}^{T} = 5\%$, the robust formulation results in nearly 27.4% of reduction in the variability as compared to the quadratic correction based solution whereas for $\epsilon_{max}^{T} = 1\%$, the reduction is nearly the same for both cases. Thus, these results show that by explicitly taking the variability into account, the sensitivity to the overall noise can be reduced to a significant level. So, which method works better depends on the degree of nonlinearity in the problem and the level of noise in the measurements. However, the robust approach can always be used along with the quadratic correction which will reduce the overall variability to a greater extent.

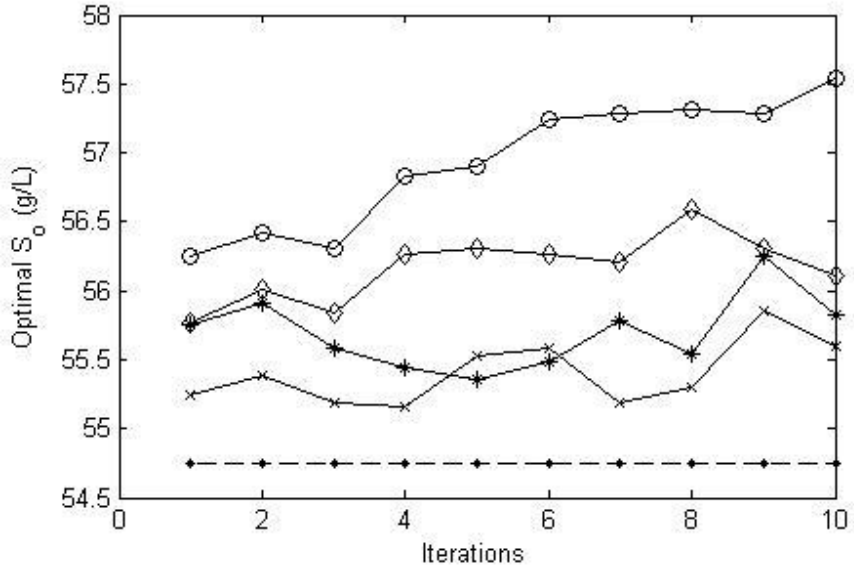
One of the key limitations of the robust formulations is the possibility of an offset which is also true for this problem. This offset is due to the presence of variability term in the cost function which forces the algorithm to converge to a cost that is different from the nominal cost. For a maximal truncation error of $\epsilon_{max}^{T} = 5\%$, the average optimal S_o converged to ~55.75g/L with an offset of ~1g/L. With the increase in weight on variability, this offset increases further. One option to eliminate the offset is to switch the algorithm to the nominal mode, corresponding to a cost function that does not include the variability, when approaching steady state. However, the effectiveness of this solution is dependent on when the switching between robust to nominal optimization is done and it may result in higher variability after the switching. Another possible solution to reduce the offset, as observed in this

study, could be to use more realistic descriptions of model uncertainty such as the Bayesian based distributions as discussed below.

For comparison, using the final optimal solutions from the above results as initial conditions, we solved the algorithm for both normal and Bayesian uncertainty descriptions with $w = 1\% \text{ and } 2\%$. The results are summarised in Figure 5.5. It is interesting to observe that for $w = 2\%$, the Bayesian uncertainty resulted in a steady state closer to the process optimum as compared to the normal counterpart. This is related to the fact that in the Bayesian uncertainty approach linearization is not required, unlike the normal description which is based on linearization of the model. Figure 5.6 compares the Bayesian and normal description for one of the initial conditions and it is evident that the Bayesian uncertainty is significantly different from its normal counterpart and as we increase the weight on the variability term in the objective function, the difference between the two uncertainty descriptions becomes more evident.



(a)



(b)

Figure 5-5: Comparing the effect of Bayesian vs normal parametric uncertainty on the robust convergence of optimal S_o for (a) $\epsilon_{max}^{T} = 1\%$ and (b) $\epsilon_{max}^{T} = 5\%$; ($-\mathrm{o}-$ Normal with $w = 2\%$; $-\diamond-$ Bayesian with $w = 2\%$; $-*-$ Normal with $w = 1\%$; $-\mathrm{x}-$ Bayesian with $w = 1\%$)

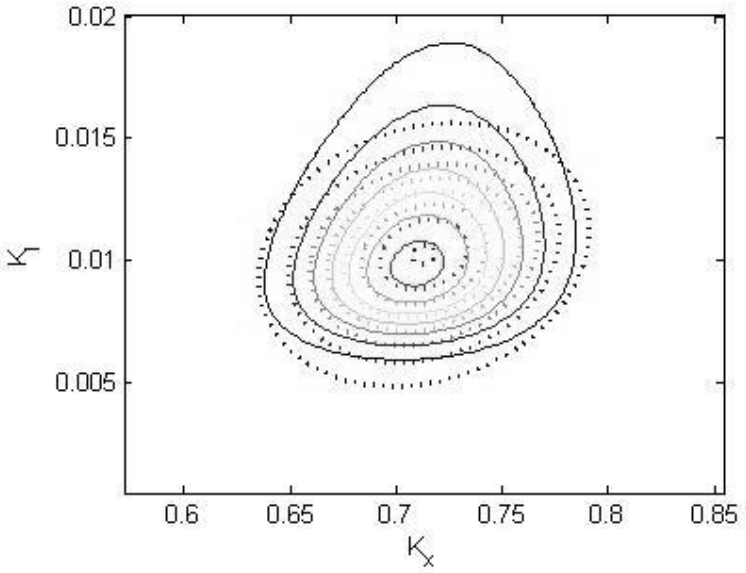


Figure 5-6: Comparing normal (dotted) and Bayesian (solid) descriptions of parametric uncertainty in K_X and K_I.

5.6 Conclusion

A new correction term has been proposed to correct the model for structural inaccuracies as the model is optimized progressively towards the process optimum. The correction is based on a quadratic approximation of the model. Compared to a previously proposed formulation that was based on a linear correction, the proposed quadratic correction allows for more accurate predictions of the optimization gradient along the iterations, thus, achieving much faster convergence. In addition, a significant reduction in variability of the search path has been observed. In the second approach, the variability is reduced by explicitly adding a measure of variability in the optimization objective. Although, for the problem in this study, the first approach performs better in reducing the overall variability, the robust formulation has been shown to perform better when the algorithm reaches the steady state and is very sensitive to the noise in gradients. In a comparative study, it is also shown that the description of parametric uncertainty has a significant effect on the convergence of the algorithm. When compared to the Bayesian description, the normal description of uncertainty resulted in more conservative solutions and also in a higher offset of the final converged solution with respect to the true optimum.

Chapter 6
Conclusions and Future Work

This thesis presented new optimization methodologies that address and mitigate the effect of model uncertainties on the optimal solution. It was assumed that the uncertainty in a given model is due to inaccurate model structure and measurement noise. When an optimization problem is solved without considering the effect of these uncertainties, the resulting optimal solution may have a significant variability corresponding to different realizations of the uncertain parameters and, in case that the model-plant mismatch is quite significant, this solution will be far away from the actual process optimum. Therefore, the central theme of this research was to reduce both variability and bias in the optimal solution. These two objectives were accomplished independently in Chapters 4 and 5 and then, the respective methodologies were combined in Chapter 6 to formulate a robust iterative algorithm that has a guaranteed convergence to an actual process optimum with a minimum variability. Compared to the previous studies that have addressed the model uncertainties, the proposed algorithms resulted in much faster and smoother convergence and are less conservative. In the following sections, a summary of the key contributions and conclusions is presented.

6.1 Robust Optimization based on Bayesian parametric uncertainty

The first contribution is in the context of robust optimization where the goal is to search for an optimal solution such that the corresponding objective function has a minimum variability. This class of algorithms involves two major steps: (1) the quantification of model uncertainties, expressed in terms of the uncertainty in

parameter estimates and (2) the propagation of this parametric uncertainty onto the optimization objective and constraints, if any.

Regarding the first step, the description of parametric uncertainty was obtained using the Bayesian approach. Since this approach requires repetitive simulations of the model, generally an order of $10^4 - 10^5$, an approximation of the model was used to speed up the calculations. An adaptive procedure, based on Multi-Resolution analysis and Polynomial Chaos (PC) expansions, was developed to formulate the approximation with a higher accuracy in the regions of parameter space where the posterior distribution is higher. It was shown that this approach has a significant advantage when dealing with the models that are highly nonlinear in parameters, especially in the case of model discontinuities.

Then, for the second step, a procedure based on PC expansions was developed to propagate the Bayesian description of parametric uncertainty onto the optimization objective. When compared to the conventional approach based on Monte-Carlo sampling, the proposed PC-based approach reduced the computational time drastically from 60-70 hr to 4-5 min in the case of one uncertain parameter and to 25-30 min in the case of two uncertain parameters. This opens up the possibility of applying this algorithm to online problems. Although the number of simulations in this approach grows rapidly with the number of uncertain parameters, it has been shown that, for a moderate number of parameters, this number is still much smaller than the one required for Monte Carlo sampling.

In another comparative study, the optimization problem was solved for both Bayesian and normally distributed parametric uncertainty. It was shown in two of the examples that the optimal solutions based on the Bayesian parametric uncertainty are much less conservative with the measured values of the objective function improved by approximately 36-50%. For the other two examples, the optimal solutions were observed to be less sensitive to the choice of uncertainty descriptions.

6.2 Model correction for optimization

The second contribution is an iterative optimization algorithm where the model is corrected for structural error as the algorithm progresses towards the process optimum. This algorithm is motivated by the fact that, in the presence of model structure error, the parameter estimates that minimize the prediction error in the outputs may not predict the measured gradients of the optimization objective accurately. To this end, corrections were added to the model outputs such that, with the updated parameter estimates, the corrected model achieves a better tradeoff between the identification and optimization objectives. The proposed procedure involves two sequential steps. In the first step, the parameter estimates $\boldsymbol{\theta}_k$ were obtained that minimizes the standard identification objective for the inaccurate model. Then, in the second step, a change in these estimates $\Delta\boldsymbol{\theta}_k$ was computed such that the difference between predicted and measured gradient of the optimization objective is minimized and at the same time, the model outputs were corrected to restore the prediction error to its minimum, corresponding to the value in the first step. In the initial algorithm, developed in Chapter 4, each correction term was based on the linearization of the corresponding model output with respect to the parameter estimates in the first step.

A trade-off between the identification and optimization objectives depends on the value of $\Delta\boldsymbol{\theta}_k$. The larger values allow the model to predict the measured optimization gradients more accurately but at the expense of accuracy in predictions. The value of $\Delta\boldsymbol{\theta}_k$ was controlled by using a bound on a relative truncation error ϵ_k^T in the approximation. In a comparative study, the algorithm was solved for two different values of ϵ_k^T and it was observed that for smaller ϵ_k^T, the convergence towards the process optimum was slower as the model was not able to predict the optimization gradients accurately. On the other hand, as expected, the convergence was relatively faster for the larger value of ϵ_k^T but at the cost of a higher prediction error. When compared to the previous reported studies, the proposed methodology converges

much faster to the process optimum and exhibits less variability along the search path. One of the reasons is that the proposed model updating strategy provides a model-based filter that outperforms the ad-hoc filtering needed in these studies.

Later in Chapter 5, an updated model correction based on quadratic approximations was proposed. In a comparative study, it was observed that for the same values of ϵ_k^T, the IAE in the optimal solution was reduced by approximately 30% when the model was corrected using quadratic corrections. Similarly, the variability in the average optimal solution was reduced by approximately 25%. As expected for nonlinear models, the linear approximation required the use of much smaller values for $\Delta\boldsymbol{\theta}_k$ thus making the convergence much slower whereas the quadratic approximation was able to capture nonlinearity very well which, for the same ϵ_k^T, allowed the algorithm to select larger values of $\Delta\boldsymbol{\theta}_k$.

It was also shown that, regardless of the value of truncation error and the type of approximation in the model correction, the algorithm eventually converges to the process optimum. Moreover, upon convergence, both identification and optimization objectives were satisfied.

6.3 Robust run-to-run optimization

The third and final contribution is a robust run-to-run optimization algorithm developed by combining the first two contributions presented above. In the iterative algorithm, discussed in the previous section (6.2), the main objective was to eliminate the bias between the model-based optimal solution and the actual process optimum. However, depending on the value of truncation error, the algorithm can be made less sensitive to either modeling error and the noise in measured outputs or the noise in measured gradients. To address this trade-off, the effect of model uncertainties was accounted for explicitly by adding a measure for robustness in the optimization objective. Since the idea behind this approach was to show the relative importance of

addressing model uncertainties explicitly, it is presented as a part of Chapter 6 where it was also compared with the quadratic model correction. Although, the quadratic correction increased the robustness of algorithm towards modelling error and the noise in measured outputs, it makes the algorithm more sensitive to the noise in measured gradients. On the other hand, the robust iterative approach reduced sensitivity with respect to all three sources of uncertainties. In one comparison, the results were specifically investigated in the neighborhood of the optimum. For one of the examples where the nominal algorithm is more sensitive to the noise in measured gradients, it was observed that the robust approach reduces the variability by as much as 27%.

In another comparative study, the algorithm was solved for both Bayesian and normal descriptions of parametric uncertainty. In terms of the final optimal solution, it was observed that the algorithm converges closer to the process optimum when the Bayesian description was implemented. It is well known that the robust algorithms cannot eliminate the bias as the objective function that is minimized corresponds to a worst-case scenario based on the measured model uncertainties. To this end, the common approach is to switch to a nominal optimization when the algorithm is close to convergence but this makes the algorithm more sensitive to model uncertainties around the optimum. Based on the comparative results in this thesis, it is concluded that using more accurate uncertainty description, as it was the case with Bayesian approach, can reduce the bias while increasing robustness with respect to uncertainties.

6.4 Future Work

Based on the findings in this thesis, the following directions were identified for future research:

1. One important conclusion in this thesis is that there are significant advantages in using Bayesian uncertainty in optimization framework especially when optimizing nonlinear models with model error. However, its application in online optimization problems such as robust RTO and robust MPC is somewhat limited, perhaps, due to very high computational time associated with its quantification and subsequent propagation into the objective function. To this end, when using PC expansions, the overall computational time is observed to be of the order of few minutes which opens up the possibility for solving Bayesian based robust optimization in online applications. One possible research direction is to reduce the computational time further by using an intrusive approach in calculating the PC expansions. To this end, the challenges involved in reformulating the model equations have to be addressed.

2. In the iterative optimization algorithm (Chapters 4 & 5), the proposed model correction is a constant term, calculated at the specific set of operating conditions during the iterative search. The convergence of the algorithm can be further improved by using past corrections to formulate a new correction term that is dependent on the operating conditions, i.e. both a function of the parameters and the decision variables. In this way, the model will be corrected for structural uncertainty over a larger region of operating conditions.

3. In this thesis, only a subset of parameters was updated in the iterative framework. This was somewhat motivated by the fact that the optimal operating conditions may not provide enough excitation to estimate all the parameters. For simplicity, only two parameters were updated in the proposed algorithm. However, the selection of these parameters was fixed from run to run. To this end, it would be more appropriate to select different set of parameters every time the model is updated and this could be based on information about the sensitivity of the identification and optimization

objectives with respect to parameters. Another possibility is to incorporate the design of experiments in the optimization objectives and in this way, it could be possible to obtain a solution that resolves a trade-off between sufficient excitation for model identification and optimality.

4. One of the applications where the proposed methodologies can be applied is the design of growth media for the cell cultures. The cell growth models are generally associated with very high uncertainty. First, the complete knowledge of intracellular mechanisms is still unknown and it is an active of research. Secondly, the measurements of biomass and different metabolites are very uncertain, mostly because of inherit variability in the cellular mechanisms. There is a strong interest from the biopharmaceutical industry to design growth media with optimal concentrations of amino acids and free of animal derived compounds. Since the cells continue to secrete toxins, it very important to predict the toxins level at the optimal solution before implementing this solution to the processes. Since the proposed methodology allows for accurate predictions around the optimum, it is quite suitable for this application. Moreover, the robust framework will provide an added advantage in filtering out the effect of uncertainties.

Copyright Permissions

ELSEVIER LICENSE
TERMS AND CONDITIONS

Jun 16, 2014

This is a License Agreement between Jasdeep Mandur ("You") and Elsevier ("Elsevier") provided by Copyright Clearance Center ("CCC"). The license consists of your order details, the terms and conditions provided by Elsevier, and the payment terms and conditions.

All payments must be made in full to CCC. For payment instructions, please see information listed at the bottom of this form.

Supplier	Elsevier Limited The Boulevard,Langford Lane Kidlington,Oxford,OX5 1GB,UK
Registered Company Number	1982084
Customer name	Jasdeep Mandur
Customer address	200 University Avenue W Waterloo, ON N2L3G1
License number	3370961239953
License date	Apr 16, 2014
Licensed content publisher	Elsevier
Licensed content publication	Journal of Process Control
Licensed content title	Robust optimization of chemical processes using Bayesian description of parametric uncertainty
Licensed content author	Jasdeep Mandur,Hector Budman
Licensed content date	February 2014
Licensed content volume number	24
Licensed content issue number	2
Number of pages	9
Start Page	422
End Page	430
Type of Use	reuse in a thesis/dissertation
Portion	full article
Format	both print and electronic
Are you the author of this Elsevier article?	Yes
Will you be translating?	No
Title of your thesis/dissertation	Robust algorithm for Optimization of Chemical Processes in the presence of model-plant mismatch
Expected completion date	May 2014
Estimated size (number of pages)	150
Elsevier VAT number	GB 494 6272 12

Bibliography

Astrom, K.J. (1970). *Introduction to Stochastic Control Theory*. Academic Press, New York

Bajpai, R. K., Reuss, M. (1980). A Mechanistic Model for Penicillin Production. Journal of Chemical Technology and Biotechnology, 30, 332-344

Balakrishnan, S., Roy, A., Ierapetritou, M. G., Flach, G. P., & Georgopoulos, P. G. (2003). Uncertainty reduction and characterization for complex environmental fate and transport models: An empirical Bayesian framework incorporating the stochastic response surface method. *Water Resources Research*, 39(12), 1350

Bard, Y. (1974). *Nonlinear Parameter Estimation* Academic Press, New York

Beck, J. V., Arnold, K. J. (1977). *Parameter Estimation in Engineering and Science*. John Wiley & Sons

Beigler, L. T., Grossmann, I. E., Westerberg, A. W. (1985). A Note on Approximation Techniques Used for Process Optimization. *Computers and Chemical Engineering*, 9, 201-206

Beyer, H., Sendhoff, B. (2007). Robust optimization - A comprehensive survey. *Computer Methods in Applied Mechanics and Engineering*, 196(33-34), 3190-3218

Birol, G., Undey, C., Cinar A. (2002). A modular simulation package for fed-batch fermentation: penicillin production. *Computers and Chemical Engineering*, 26(11), 1553-1565

Chachuat B., B. Srinivasan and D. Bonvin (2009). Adaptation strategies for real-time optimization. *Computers and Chemical Engineering*, **33(10)**, 1557-1567.

Chen, C. Y., Joseph, B. (1987). On-line Optimization Using a Two-Phase Approach: An Application Study. *Industrial and Engineering Chemistry Research*, 26(9), 1924-1930

Costello S., G, Francois, B. Srinivasan and D. Bonvin (2011). Modifier Adaptation for Run-to-Run Optimization of Dynamic Processes. *Proceedings of the 18th IFAC World Congress*, 11471-11476, Milano

Cover T.M and J.A. Thomas (1991). *Elements of Information theory*, John Wiley and Sons, New York

Diwekar, U. M., Kalagnanam, J.R. (1996). Robust Design using an Efficient Sampling Technique. *Computers and Chemical Engineering*, 20, S389-S394

Eaton, J.W., Rawlings, J.B. (1990). Feedback control of nonlinear processes using on-line optimization techniques. *Computers and Chemical Engineering*, 14, 469-479

Forbes, J. F., Marlin, T. E., MacGregor, J. F. (1994). Model adequacy requirements for optimizing plant operations. *Computers and Chemical Engineering*, 18(6), 497-510

Forbes, J. F., Marlin, T. E. (1996). Design cost: A systematic approach to technology selection for model-based real-time optimization systems. *Computers and Chemical Engineering*, 20(6), 717-734

Gao, W. and S. Engell (2005). Iterative set-point optimization of batch chromatography. *Computers and Chemical Engineering*, **29**, 1401-1409

Ghanem, R. G., Spanos, P. (1991). *Stochastic Finite Elements: A Spectral Approach*, Springer, Berlin

Ghanem, R. (1998a). Hybrid stochastic finite elements and generalized Monte Carlo simulations. *ASME Journal of Applied Mechanics*, 65, 1004-9

Ghanem, R. (1998b). Probabilistic characterization of transport in heterogeneous media. *Computational Methods in Applied Mechanics*, 158, 199-220

Ghanem, R., Dham, S. (1998). Stochastic finite element analysis for multiphase flow in heterogeneous porous media. *Transport in Porous Media*, 32(3), 239-262

Gilks, W. R., Richardson, S., Spiegelhalter, D. J. (1996). *Markov Chain Monte Carlo in Practice*. Chapman & Hall, London

Gustafson, P. (1998). A guided walk Metropolis algorithm. *Statistics and Computing*, 8(4), 357-364

Hastings, W.K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1), 97-109

Hosder, S., Walters, R. W., Perez, R. (2006). A Non-Intrusive Polynomial Chaos Methods For Uncertainty Propagation in CFD Simulations. *44th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada

Isukapalli, S. S. (1999). Uncertainty analysis of transport-transformation models. PhD Thesis, Rutgers University, Piscataway

Knio, O. M., Le Maître, O. P. (2006). Uncertainty propagation in CFD using polynomial chaos decomposition. *Fluid Dynamics Research*, 38(9), 616-640.

Le Maître, O. P., Ghanem, R., Knio, O. M., Najm, H. (2004a). Uncertainty propagation using Wiener-Haar expansions. *Journal of Computational Physics*, 197, 28-57.

Le Maître, O. P., Knio, O. M. (2007). A stochastic particle-mesh scheme for uncertainty propagation in vortical flows. *Journal of Computational Physics*, 226, 645-71. 57

Mandur J. and H. Budman (2012). A Polynomial-Chaos based Algorithm for Robust optimization in the presence of Bayesian Uncertainty. *Preprints of the 8th IFAC Symposium on Advanced Control of Chemical Processes*, 549-554, Singapore

Mandur J. and H. Budman (2013a). Robust Optimization of Chemical Processes using Bayesian description of parametric uncertainty, *Journal of Process Control*, 24, 2, 422-430

Mandur J. and H. Budman (2013b). A Robust algorithm for Run-to-run Optimization of Batch Processes, In Proceedings of *10th International Symposium on Dynamics and Control of Process Systems*, 541-546, Mumbai

Marchetti A., Chachuat, B., Bonvin, D. (2009). Modifier-adaptation methodology for real–time optimization, *Industrial and Engineering Chemistry Research*, 48 (13), 6022-6033.

Marchetti A., B. Chachuat and D. Bonvin (2010). A dual modifier-adaptation approach for real–time optimization. *Journal of Process Control*, **20(9)**, 1027-1037

Marlin, T. E., Hrymak, A. N. (1997). Real-time operations optimization of continuous processes. *In AIChE Symp Ser – CPC – V*, Vol. 93, 156-164

Martinez, E.C., Cristaldi, M.D., Grau, R.J. (2013). Dynamic optimization of bioreactors using probabilistic tendency models and Bayesian active learning. *Computers and Chemical Engineering*, 49, 37-49

Marzouk, Y. M., Najm, H. N., & Rahn, L. A. (2007). Stochastic spectral methods for efficient Bayesian solution of inverse problems. *Journal of Computational Physic*s, 224, 560-586.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., Teller, E. (1953). Equations of state calculations by fast computing machine. *The Journal of Chemical Physics*, 21(6), 1087-1092

Nagy, Z. K., Braatz, R. D. (2004). Open-loop and closed-loop robust optimal control of batch processes using distributional and worst-case analysis. *Journal of Process Control*, 14(4), 411-422

Nagy, Z. K., Braatz, R. D. (2007). Distributional uncertainty analysis using power series and polynomial chaos expansions. *Journal of Process Control*, 17, 229-240.

Najm, H. N. (2009). Uncertainty Quantification and Polynomial Chaos Techniques in Computational Fluid Dynamics. *Annual Review of Fluid Mechanics*, 41, 35-52.

Pan, W., Tatang, M. A., McRae, G. J., Prinn, R. G. (1997). Uncertainty analysis of direct radiative forcing by anthropogenic sulfate aerosols. *Journal of Geophysical Research*, 102, 21915-21924.

Roberts, P. D. (1979). An algorithm for Steady-State System Optimization and Parameter Estimation. *International Journal of Systems Science*, 10, 719-734.

Ruppen, D., Benthack, C., Bonvin, D. (1995). Optimization of batch reactor operation under parametric uncertainty-computational aspects. *Journal of Process control*, 5(4), 235-240.

Ruppen, D., Bonvin, D., Rippin, D.W.T. (1998). Implementation of adaptive optimal operation for a semi-batch reactor system. *Computers and Chemical Engineering*, 22(1-2), 185-199

Samsatli, N. J., Papageorgiou, L. G., Shah, N. (1998). Robustness metrics for dynamic optimization models under parameter uncertainty. *American Institute of Chemical Engineering Journal*, 44(9), 1993-2006

Smolyak. S. (1963). Quadrature and interpolation formulas for tensor products of certain classes of functions. *Soviet Math. Dokl.*, 4, 240-243

Srinivasan, B., Bonvin, D. (2002). Interplay between identification and optimization in run-to-run optimization schemes. In *Proceedings of American Control Conference*, Anchorage AK (2174-2179)

Srinivasan, B., Bonvin, D. (2003). Convergence analysis of iterative identification and optimization schemes. *Proceedings of American Control Conference*, Denver, Colorado (1956-1961)

Tatang, M. A., Pan, W., Prinn, R. G., McRae, G. J. (1997). An efficient method for parametric uncertainty analysis of numerical geophysical models. *Journal of Geophysical Research*, 102, 21925-21932.

Tatjewski, P (2002). Iterative optimizing set point control-The basic principle redesigned. *In Proceedings of the 15th Triennial IFAC World Congress*

Terwiesch, P., Agarwal, M., Rippin, D.W.T. (1994). Batch unit optimization with imperfect modelling-a survey. *Journal of Process Control*, 4, 238-258

Tierney, L. (1994). Markov chains for exploring posterior distributions. *The Annals of Statistics*, 22(4), 1701-1762

Vajda, S., Rabitz, H., Walter, E., Lecourtier, Y. (1989). Qualitative and quantitative identifiability analysis of non-linear chemical kinetic models. *Chemical Engineering Communications*, 83(2), 191-219

Wan, X., Karniadakis, G. E. (2005). An adaptive multi-element generalized polynomial chaos method for stochastic differential equations. *Journal of Computational Physics*, 209, 617-42.

Watts, D. G. (1994). Estimating parameters in nonlinear rate equations, *The Canadian Journal of Chemical Engineering*, 72(4), 701-710

Xiu, D. (2007). Efficient Collocational Approach for Parametric Uncertainty Analysis. *Communications in Computational Physics*, 2(2), 293-309.

Xiu, D., Hesthaven, J. S. (2005). High-Order Collocation Methods for Differential Equations with Random Inputs. *SIAM Journal on Scientific Computing*, 27(3), 1118-1139.

Xiu, D., Karniadakis, G. E. (2002a). Modeling uncertainty in steady state diffusion problems via generalized polynomial chaos. *Computer Methods in Applied Mechanics and Engineering*, 191, 4927-4948.

Xiu, D., Karniadakis, G. E. (2003). Modeling uncertainty in flow simulations via generalized polynomial chaos. *Journal of Computational Physics*, 187, 137-167.

Xiu, D., Karniadakis, G. E. (2002b). The Weiner-Askey polynomial chaos for stochastic differential equations. *SIAM Journal of Scientific Computing*, 24(2), 619-644 60

Yao, K. Z., Shaw, B. M., Kou, B., McAuley, K. B., Bacon, D. W. (2003). Modeling Ethylene/Butene Copolymerization with Multi-site Catalysts: Parameter Estimability and Experimental Design. *Polymer Reaction Engineering*, 11(3), 563-588

Zhang, Y., Monder, D., Forbes, J. F. (2002). Real-time optimization under parametric uncertainty: a probability constrained approach. *Journal of Process Control*, 12, 373-389

Appendix A

Matlab Codes

Algorithm to solve iterative procedure

```
function robust_loop_fmincon2P

clear all; close all; clc; tic;

epsis = [sym('epsi1','real') sym('epsi2','real')];

pci_epsi1=pcepoly1d('legendre',epsis(1),20);
pci_epsi2=pcepoly1d('legendre',epsis(2),20);

for z=1:1

U0=[0.1 0.04];

pci_func=2D_pce('epsi1','epsi2');
pci_den=2D_pce_den('epsi1','epsi2');

% Define deviations in input for calculating gradients

dev_1=0.5;
dev_2=0.5;

opt=odeset('NonNegative',[1,2,3,4],'RelTol',1e-6,'AbsTol',1e-8);
[exp_data, plantgrd] = pen_sim(U0,dev_1,dev_2); toc

K=dlmread('fmincon_parameters_LCLS.txt');
K0=[K(2) K(5)]; prev_corr=0;

[Kout, Pfinal,cov_data_prev] = par_estim(U0,exp_data,prev_corr,K0);
toc
[Kout_prime, Pfinal_prime,c,gradpred,cov_data] =
delta_theta(U0,prev_corr,plantgrd,K,Kout,Kout); toc
[theta1_PCE, theta2_PCE] =
twoP_legendre_sse_normal_shiftedmeans(Kout_prime,cov_data,pci_epsi1,
pci_epsi2); toc

new_corr=prev_corr+c;
disp(plantgrd)
[x,f,~,~] =
runobjconstrfunc(U0,new_corr,K,theta1_PCE,theta2_PCE,pci_func,pci_de
n,Kout_prime); toc
```

```
output=[Kout Pfinal plantgrd Kout_prime Pfinal_prime gradpred x -f
cov_data_prev cov_data];
dlmwrite(['outputdata_12mar_RBTB_trunc1_noise_both_grad_',num2str(z)
,'.txt'], output, 'delimiter', '\t','precision',  '%.12f');
dlmwrite(['xdata_12mar_RBTB_trunc1_noise_both_grad_',num2str(z),'.tx
t'], x, 'delimiter', '\t','precision',  '%.4f');

prev_corr=new_corr;
U0=x;

correc(:,:,1)=new_corr;
open(['xdata_12mar_RBTB_trunc1_noise_both_grad_',num2str(z),'.txt'])
open(['outputdata_12mar_RBTB_trunc1_noise_both_grad_',num2str(z),'.t
xt'])

for i=2:30

opt=odeset('NonNegative',[1,2,3,4],'RelTol',1e-6,'AbsTol',1e-8);
[exp_data, plantgrd] = pen_sim(U0,dev_1,dev_2);

K=dlmread('fmincon_parameters_LCLS.txt');
[Kout, Pfinal,cov_data_prev] =
par_estim(U0,exp_data,prev_corr,Kout_prime); toc
[Kout_prime, Pfinal_prime,c,gradpred,cov_data] =
delta_theta(U0,prev_corr,plantgrd,K,Kout,Kout_prime); toc
[theta1_PCE, theta2_PCE] =
twoP_legendre_sse_normal_shiftedmeans(Kout_prime,cov_data,pci_epsi1,
pci_epsi2); toc

new_corr=prev_corr+c;
disp(plantgrd)
[x,f,~,~] =
runobjconstrfunc(U0,new_corr,K,theta1_PCE,theta2_PCE,pci_func,pci_de
n,Kout_prime);

prev_corr=new_corr;
U0=x;

output=[Kout Pfinal plantgrd Kout_prime Pfinal_prime gradpred x -f
cov_data_prev cov_data];
dlmwrite(['outputdata_12mar_RBTB_trunc1_noise_both_grad_',num2str(z)
,'.txt'], output,'-append', 'delimiter', '\t','precision',
'%.12f');
dlmwrite(['xdata_12mar_RBTB_trunc1_noise_both_grad_',num2str(z),'.tx
t'], x, '-append', 'delimiter', '\t','precision',  '%.4f');

correc(:,:,i)=new_corr;
save(['correc_12mar_RBTB_trunc1_noise_both_grad_',num2str(z),'.mat']
,'correc');

end
end
toc
```

Process simulator

```
function [exp_data, plantgrd] = pen_sim(U0,dev_1,dev_2)

U=U0; Y0=[0.1 0 U(1) 100];
[~, Y] = ode15s(@model,[0:6:8*24], Y0, opt);
disp([Y(end,2)*Y(end,4) Y(end,4) Y(end,2)])

U=[U0(1)+dev_1 U0(2)]; Y0=[0.1 0 U(1) 100];
[~, Y1] = ode15s(@model,[0:6:8*24], Y0, opt);
U=[U0(1) U0(2)+dev_2]; Y0=[0.1 0 U(1) 100];
[~, Y2] = ode15s(@model,[0:6:8*24], Y0, opt);

% Generating Exp. Data

std_data=sqrt([4 0.4 4]);
exp_data=[normrnd(Y(:,1),std_data(1),size(Y,1),1)
normrnd(Y(:,2),std_data(2),size(Y,1),1)
normrnd(Y(:,3),std_data(3),size(Y,1),1)];

% Calculating plant gradients

dphidu_1=(Y1(end,2)*Y1(end,4)-Y(end,2)*Y(end,4))/dev_1;
dphidu_2=(Y2(end,2)*Y2(end,4)-Y(end,2)*Y(end,4))/dev_2;
dphidu_1=normrnd(dphidu_1,5);
dphidu_2=normrnd(dphidu_2,5);

plantgrd=[dphidu_1 dphidu_2];

function dY = model(~, Y)

dY=zeros(4,1);

X=Y(1);
P=Y(2);
S=Y(3);
V=Y(4);

% Input Feed rate
F=U(2);

% Substrate conc. in the feed
sf=600; % g/l

% Model Parameters
K=[0.092 0.15 0.005 0.0002 0.1 0.04 0.45 0.9 0.014]';

% Biomass
mux=K(1);
Kx=K(2);
% Product
mup=K(3);
Kp=K(4);
```

```
KI=K(5);
Kh=K(6);
% Substrate
Yxs=K(7);
Yps=K(8);
mx=K(9);

% Loss in the culture volume due to evaporation
Floss=V*2.5*10^-4*(exp(5*(298-273)/(373-273))-1);

% ODEs
dY(1)=mux*S*X/(Kx*X+S)-(X/V)*(F-Floss);
dY(2)=mup*S*X/(Kp+S*(1+S/KI))-Kh*P-(P/V)*(F-Floss);
dY(3)=-mux*S*X/(Kx*X+S)/Yxs-mup*S*X/(Kp+S*(1+S/KI))/Yps-mx*X+F*sf/V-
(S/V)*(F-Floss);
dY(4)=F-Floss;

end
end
```

Function to estimate model parameters

```
function [Kout, Pfinal, cov_data_prev] =
par_estim(U,exp_data,prev_corr,K0)

Y0=[0.1 0 U(1) 100];

lb=1e-6*ones(1,2);
ub=5*ones(1,2);

options=psoptimset('Display','iter','UseParallel','always',...
    'TolX',1e-6,'TolFun',1e-6,'TolCon',1e-8,...
    'CompletePoll','on','CompleteSearch','on');
[x,~] = fmincon(@sseobj,K0,[],[],[],[],lb,ub,[],options);
Kout=x; sseobj(x); Pfinal=Y(end,2)*Y(end,4);

K(2)=x(1);
K(5)=x(2);
[T,~,DXDP] = sens_sys('altered_model_jac',[0:6:8*24],[0.1 0 U(1)
100],opt,K',[],[],U);

V=diag([4 0.4 4]);

Jac=0;
for m=1:length(T)
    J=[DXDP(m,1:3,2)' DXDP(m,1:3,5)'];
    Jac=Jac+J'*inv(V)*J;
end
param_var=inv(Jac);
cov_data_prev=[param_var(1,1) param_var(2,2) param_var(1,2)];

function F = sseobj(x)
```

```
[~,Y] = ode15s(@process_model_inhibit,[0:6:8*24],Y0,opt,K,U,x);
Y=Y-prev_corr;

sseX=(Y(:,1)-exp_data(:,1)).^2/4; %/var(itr,1);
sseP=(Y(:,2)-exp_data(:,2)).^2/0.4; %/var(itr,2);
sseS=(Y(:,3)-exp_data(:,3)).^2/4; %/var(itr,3);
F=sum(sseX)+sum(sseP)+sum(sseS);
end
end
```

Function to calculate delta_theta

```
function [Kout_prime,Pfinal_prime,correc,gradpred,cov_data] =
delta_theta(U0,prev_corr,plantgrd,K,K0,Kini)

K(2)=K0(1);
K(5)=K0(2);
[~,DX,DXDP] = sens_sys('altered_model_jac',[0:6:8*24],[0.1 0 U0(1)
100],opt,K',[],[],U0);

xLast = []; % Last place computeall was called
myf = []; % Use for objective at xLast
myc = []; % Use for nonlinear inequality constraint
myceq = []; % Use for nonlinear equality constraint

lb=1e-6*ones(1,2)-K0;
ub=5*ones(1,2)-K0;

iniK=Kini-K0;

options=optimset('Display','iter','UseParallel','always',...
    'TolX',1e-6,'TolFun',1e-6,'TolCon',1e-8,'MaxFunEvals',400);
[x,~] =
fmincon(@dt_obj,iniK,[],[],[],[],lb,ub,@dt_constr,options,...
    U0,prev_corr,plantgrd,DX,DXDP,K,K0);

[~,~,~,Y,correc,gradpred]=grad_calc(x,U0,prev_corr,plantgrd,DX,DXDP,
K,K0);
Kout_prime=K0+x; Pfinal_prime=Y(end,2)*Y(end,4);

K(2)=x(1)+K0(1);
K(5)=x(2)+K0(2);
[T,~,DXDP] = sens_sys('altered_model_jac',[0:6:8*24],[0.1 0 U0(1)
100],opt,K',[],[],U0);

V=diag([4 0.4 4]);

Jac=0;
for m=1:length(T)
    J=[DXDP(m,1:3,2)' DXDP(m,1:3,5)'];
    Jac=Jac+J'*inv(V)*J;
end
```

```
param_var=inv(Jac);
cov_data=[param_var(1,1) param_var(2,2) param_var(1,2)];

function y = dt_obj(x,U0,prev_corr,plantgrd,DX,DXDP,K,K0)
    if ~isequal(x,xLast) % Check if computation is necessary
        [myf,myc,myceq,~,~,~] =
grad_calc(x,U0,prev_corr,plantgrd,DX,DXDP,K,K0);
        xLast = x;
    end
    % Now compute objective function
    y = myf;
%     disp(K0+x)
end

function [c,ceq] = dt_constr(x,U0,prev_corr,plantgrd,DX,DXDP,K,K0)
    if ~isequal(x,xLast) % Check if computation is necessary
        [myf,myc,myceq,~,~,~] =
grad_calc(x,U0,prev_corr,plantgrd,DX,DXDP,K,K0);
        xLast = x;
    end
%     Now compute constraint functions
    c = myc; % In this case, the computation is trivial
    ceq = myceq;
end

end
```

Function to calculate objective function and constraint for delta_theta optimization procedure

```
function [f1,c1,ceq1,Y,c,gradpred,trunc_err] =
grad_calc(x,U0,prev_corr,plantgrd,DX,DXDP,K,K0)

devpred_1=0.5; devpred_2=0.5;
opt=odeset('NonNegative',[1,2,3,4],'RelTol',1e-6,'AbsTol',1e-8);

[~,Y] = ode15s(@process_model_inhibit,[0:6:8*24],[0.1 0 U0(1)
100],opt,K,[U0(1) U0(2)],K0+x);
[~,Y1] = ode15s(@process_model_inhibit,[0:6:8*24],[0.1 0
U0(1)+devpred_1 100],opt,K,[U0(1)+devpred_1 U0(2)],K0+x);
[~,Y2] = ode15s(@process_model_inhibit,[0:6:8*24],[0.1 0 U0(1)
100],opt,K,[U0(1) U0(2)+devpred_2],K0+x);

c=DXDP(:,:,2)*(x(1))+DXDP(:,:,5)*(x(2));
Y(:,1:4)=Y(:,1:4)-c-prev_corr;
Y1(:,1:4)=Y1(:,1:4)-c-prev_corr;
Y2(:,1:4)=Y2(:,1:4)-c-prev_corr;
gradpred1=(Y1(end,2)*Y1(end,4)-Y(end,2)*Y(end,4))/devpred_1;
gradpred2=(Y2(end,2)*Y2(end,4)-Y(end,2)*Y(end,4))/devpred_2;
gradpred=[gradpred1 gradpred2];
```

```
f1=abs(gradpred1-plantgrd(1))+0.01*abs(gradpred2-plantgrd(2));

Y_prime=DX-prev_corr;
trunc_err=Y-Y_prime;
trunc_e=abs(Y./Y_prime-1);
c1=max(trunc_e(end,:))-0.01;
ceq1=[];

end
```

Function to formulate PCE expansion for normal distribution

```
function [theta1_PCE, theta2_PCE] =
twoP_legendre_sse_normal_shiftedmeans(Kout_prime,cov_data,pci_epsi1,
pci_epsi2)

K=[Kout_prime cov_data];

gleq=[1     -0.997263861849     0.00701814576495
2   -0.985611511545     0.0162774265831
3   -0.964762255588     0.0253910098329
4   -0.934906075938     0.0342745478477
5   -0.896321155766     0.0428359896785
6   -0.849367613733     0.0509978738117
7   -0.794483795968     0.0586839394615
8   -0.73218211874  0.0658220603578
9   -0.66304426693  0.0723456094297
10  -0.587715757241     0.078193695762
11  -0.506899908932     0.083311711103
12  -0.421351276131     0.0876518688047
13  -0.331868602282     0.0911736454878
14  -0.239287362252     0.0938441590423
15  -0.144471961583     0.0956384754512
16  -0.0483076656877    0.0965398415811
17  0.0483076656877     0.0965398415811
18  0.144471961583  0.0956384754512
19  0.239287362252  0.0938441590423
20  0.331868602282  0.0911736454878
21  0.421351276131  0.0876518688047
22  0.506899908932  0.083311711103
23  0.587715757241  0.078193695762
24  0.66304426693   0.0723456094297
25  0.73218211874   0.0658220603578
26  0.794483795968  0.0586839394615
27  0.849367613733  0.0509978738117
28  0.896321155766  0.0428359896785
29  0.934906075938  0.0342745478477
30  0.964762255588  0.0253910098329
31  0.985611511545  0.0162774265831
32  0.997263861849  0.00701814576495];
```

```
mu = [K(1) K(2)];
Sigma = [K(3) K(5); K(5) K(4)];

chi2val=chi2inv(0.99,2);
rect_coord=[mu(1)-sqrt(chi2val*Sigma(1,1)), mu(2)-
sqrt(chi2val*Sigma(2,2)), 2*sqrt(chi2val*Sigma(1,1)),
2*sqrt(chi2val*Sigma(2,2))];
x_1=rect_coord(1)+rect_coord(3);
x_2=rect_coord(2)+rect_coord(4);

if rect_coord(1)<0
    rect_coord(1)=0;
end
if rect_coord(2)<0
    rect_coord(2)=0;
end
epsi_1 = linspace(rect_coord(1), x_1,100);
epsi_2 = linspace(rect_coord(2), x_2,100);

[X1,X2] = meshgrid(epsi_1,epsi_2);
F = mvnpdf([X1(:) X2(:)],mu,Sigma);
F = reshape(F,length(epsi_2),length(epsi_1));

% contour(epsi_1,epsi_2,F); hold on
theta1=X1; theta2=X2; p_theta=F;

mp_theta1=[];
for i=1:length(epsi_1)
    mp_theta1(i)=trapz(theta2(:,i),p_theta(:,i));
end

cum_p=trapz(theta1(1,:),mp_theta1);
p_theta=p_theta/cum_p;

mp_theta1=[];
for i=1:length(epsi_1)
    mp_theta1(i)=trapz(theta2(:,i),p_theta(:,i));
end

cp_theta2=p_theta./repmat(mp_theta1,size(p_theta,1),1);

cump_theta1=[]; cump_theta1(1)=0;
for j=2:length(mp_theta1)
    cump_theta1(j)=trapz(theta1(1,1:j),mp_theta1(1:j));
end

epsi1=gleq(:,2);

theta1_i=theta1(1,cump_theta1<=0.9999);
cump_theta1_i=cump_theta1(cump_theta1<=0.9999);
theta1=interp1(cump_theta1_i',theta1_i',unifcdf(epsi1,-
1,1),'linear','extrap');
```

```
pci=pci_epsi1;

c=[]; err=[];
c(1,1)=(gleq(:,3)'*0.5*theta1)/sum(gleq(:,3)*0.5);
err(1,1)=sum((theta1-repmat(c(1,1),length(theta1),1)).^2);

for i=2:20
c(i,1)=sum(gleq(:,3).*subs(pci(i)).*theta1*0.5)/sum(gleq(:,3).*subs(
pci(i)^2)*0.5);
err(i,1)=sum((theta1-repmat(c(1,1),length(theta1),1)-
subs(pci(2:i))*c(2:i,1)).^2);

if(err(i,:)<1e-6)
    pci=pci(1:i);
    break;
end
end

theta1=pci*c; % theta1=c;


cump_theta2=zeros(size(p_theta,1),size(p_theta,2));

for i=1:size(cp_theta2,2)
    cump_theta2(1,i)=0;
for j=2:size(cp_theta2,1)
    cump_theta2(j,i)=trapz(theta2(1:j,i),cp_theta2(1:j,i));
end
end

epsi2=gleq(:,2);
theta2_new=[];

for i=1:size(cp_theta2,2)
    theta2_i=theta2(cump_theta2(:,i)<=0.999,i);
    cump_theta2_i=cump_theta2(cump_theta2(:,i)<=0.999,i);
    theta2_new(:,i)=interp1(cump_theta2_i,theta2_i,unifcdf(epsi2,-
1,1),'linear','extrap');
end
theta2=theta2_new;

pci=pci_epsi2;

a=[]; err=[];
a(1,:)=(gleq(:,3)'*0.5*theta2(:,:))/sum(gleq(:,3)*0.5);
err(1,:)=sum((theta2(:,:)-repmat(a(1,:),size(theta2,1),1)).^2);

for i=2:20

a(i,:)=((gleq(:,3).*subs(pci(i)))'*theta2(:,:)*0.5)/sum(gleq(:,3).*s
ubs(pci(i)^2)*0.5);
    err(i,:)=sum((theta2(:,:)-repmat(a(1,:),size(theta2,1),1)-
subs(pci(2:i))*a(2:i,:)).^2);
```

```
if(max(err(i,:))<1e-8)
    pci=pci(1:i);
    break;

end
end

epsi1=gleq(:,2);
a_new=[];

for i=1:size(a,1)
    a_i=a(i,cump_theta1<=0.9999);
    cump_theta1_i=cump_theta1(cump_theta1<=0.9999);
    a_new(:,i)=interp1(cump_theta1_i',a_i',unifcdf(epsi1,-
1,1),'linear','extrap');
end

a=a_new';

pci_2=pci_epsi1;


b=[]; err=[];
b(1,:)=(gleq(:,3)'*0.5*a')/sum(gleq(:,3)*0.5);
err(1,:)=sum((a'-repmat(b(1,:),length(a'),1)).^2);

for i=2:20

b(i,:)=((gleq(:,3).*subs(pci_2(i)))'*a'*0.5)/sum(gleq(:,3).*subs(pci
_2(i)^2)*0.5);
    err(i,:)=sum((a'-repmat(b(1,:),length(a'),1)-
subs(pci_2(2:i))*b(2:i,:)).^2);

if(max(err(i,:))<1e-6)
    pci_2=pci_2(1:i);
    break;
elseif(max(err(i,:))-max(err(i-1,:))>1)
    pci_2=pci_2(1:i-1);
    b=b(1:i-1,:);
    err=err(1:i-1,:);
    break;
end
end

theta2=pci*(pci_2*b)'; % theta2=b;

theta1_PCE=theta1;
theta2_PCE=theta2;

end
```

Function to calculate robust process optimum

```
function [x,f,eflag,outpt] =
runobjconstrfunc(U0,new_corr,K,theta1_PCE,theta2_PCE,pci_func,pci_de
n,Kout_prime)

% if nargin == 1 % No options supplied
%     opts = [];
% end

xLast = []; % Last place computeall was called
myf = []; % Use for objective at xLast
myc = []; % Use for nonlinear inequality constraint
myceq = []; % Use for nonlinear equality constraint

fun = @objfun; % the objective function, nested below
cfun = @constr; % the constraint function, nested below

gleq=[1 -0.978228658146 0.0556685671162
2   -0.887062599768 0.125580369465
3   -0.730152005574 0.186290210928
4   -0.519096129207 0.233193764592
5   -0.269543155952 0.26280454451
6   0   0.272925086778
7   0.269543155952  0.26280454451
8   0.519096129207  0.233193764592
9   0.730152005574  0.186290210928
10  0.887062599768  0.125580369465
11  0.978228658146  0.0556685671162];

epsi_1=gleq(:,2); epsi_2=gleq(:,2);
epsi1=[]; epsi2=[]; w=[];

for j=1:length(epsi_1)
    epsi1=[epsi1; ones(length(epsi_2),1)*epsi_1(j)];
    epsi2=[epsi2; gleq(1:end,2)];
    w=[w; gleq(j,3)*gleq(:,3)];
end
theta1=subs(theta1_PCE); theta2=subs(theta2_PCE);

% Call fmincon

lb=[0.001 0.001];
ub=[100 100];

options=optimset('Display','iter',...
    'TolX',1e-6,'TolFun',1e-6,'TolCon',1e-6);
[x,f,eflag,outpt] =
fmincon(fun,U0,[],[],[],[],lb,ub,cfun,options,new_corr,K,theta1,thet
a2,epsi1,epsi2,w,pci_func,pci_den,Kout_prime);
```

```
function y =
objfun(x,new_corr,K,theta1,theta2,epsi1,epsi2,w,pci_func,pci_den,Kou
t_prime)
    if ~isequal(x,xLast) % Check if computation is necessary
        [myf,myc,myceq] =
compute_robust_loop2P(x,new_corr,K,theta1,theta2,epsi1,epsi2,w,pci_f
unc,pci_den,Kout_prime);
        xLast = x;
    end
    % Now compute objective function
    y = myf;
end

function [c,ceq] =
constr(x,new_corr,K,theta1,theta2,epsi1,epsi2,w,pci_func,pci_den,Kou
t_prime)
    if ~isequal(x,xLast) % Check if computation is necessary
        [myf,myc,myceq] =
compute_robust_loop2P(x,new_corr,K,theta1,theta2,epsi1,epsi2,w,pci_f
unc,pci_den,Kout_prime);
        xLast = x;
    end
    % Now compute constraint functions
    c = myc; % In this case, the computation is trivial
    ceq = myceq;
end

end
end
```

Function to calculate robust objective function and constraints in process optimization procedure

```
function [f1,c1,ceq1,S,variance] =
compute_robust_loop2P(U,new_corr,K,theta1,theta2,epsi1,epsi2,w,pci_f
unc,pci_den,Kout_prime)

nsamples=length(theta1);
spmd
    if (labindex==4)
        ai=(labindex-1)*floor(nsamples/numlabs)+1;
        if (rem(nsamples,numlabs)>0)
            bi=nsamples;
        else
            bi=labindex*floor(nsamples/numlabs);
        end
    else
        ai=(labindex-1)*floor(nsamples/numlabs)+1;
        bi=labindex*floor(nsamples/numlabs);
    end
end
```

```
opt=odeset('NonNegative',[1,2,3,4],'RelTol',1e-6,'AbsTol',1e-8);

numofgridpts=length(theta1);
S=zeros(length(theta1),1);
constr=zeros(length(theta1),1);
Khat=K;
spmd
    for i=ai:bi
        Khat(2)=theta1(i);
        Khat(5)=theta2(i);
        [~,dX] = ode15s(@process_opt_model_inhibit,[0:6:8*24],[0.1 0
U(1) 100],opt,U,Khat);
        dX(:,:)=dX(:,:)-new_corr;
        S(i,:)=dX(end,2)*dX(end,4);
        constr(i,:)=dX(end,4);
    end
end
Shat=zeros(length(theta1),1);
chat=zeros(length(theta1),1);
for i=1:4
    Shat=Shat+S{i};
    chat=chat+constr{i};
end
S=Shat;
constr=chat;

a=(w'*0.25*S)/sum(w*0.25);
numofterms=length(pci_den)+1;

coeffs=(w'*(0.25*pci_func(epsi1,epsi2).*repmat(S,1,numofterms-
1)))./pci_den;
error=sum((repmat(S,1,numofterms)-
cumsum([repmat(a(1,1),numofgridpts,1)
pci_func(epsi1,epsi2).*repmat(coeffs,numofgridpts,1)],2)).^2);
lim=find(error==min(error));
a=[a(1,1) coeffs(1:lim-1)];

% mean=a(1);
variance=sum((a(2:end).^2).*pci_den(1:lim-1));
% f1=-mean+0*variance;
c1=max(constr)-120;
ceq1=[];

opt=odeset('NonNegative',[1,2,3,4],'RelTol',1e-6,'AbsTol',1e-8);
K(2)=Kout_prime(1); K(5)=Kout_prime(2);
[~,dX] = ode15s(@process_opt_model_inhibit,[0:6:8*24],[0.1 0 U(1)
100],opt,U,K);
dX(:,:)=dX(:,:)-new_corr;
S=dX(end,2)*dX(end,4);

f1=-S+0.02*variance;

end
```

Uncertain Model

```
function dY = process_model_inhibit(t, Y, K, U ,UP)

dY=zeros(4,1);

X=Y(1);
P=Y(2);
S=Y(3);
V=Y(4);

% Input Feed rate
F=U(2); % l/h [LCLS]

% Substrate conc. in the feed
sf=600; % g/l

% Biomass
mux=K(1);
Kx=UP(1);
% Product
mup=K(3);
Kp=K(4);
KI=UP(2);

% Substrate
Yxs=K(6);
Yps=K(7);
mx=K(8);

% Loss in the culture volume due to evaporation
Floss=V*2.5*10^-4*(exp(5*(298-273)/(373-273))-1);

% ODEs
dY(1)=mux*S*X/(Kx*X+S)-(X/V)*(F-Floss);
dY(2)=mup*S*X/(Kp+S*(1+S/KI))-(P/V)*(F-Floss);
dY(3)=-mux*S*X/(Kx*X+S)/Yxs-mup*S*X/(Kp+S*(1+S/KI))/Yps-mx*X+F*sf/V-
(S/V)*(F-Floss);
dY(4)=F-Floss;

end
```