Parameter Estimation of Stochastic Nonlinear Dynamic Processes using Multiple Experimental Data Sets

with Biological Applications

by

Seunghee Shelly Jang

Chemical Engineering B.S., The University of Washington, 2007
Mathematics & Chemistry Minor, The University of Washington, 2007

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

The Faculty of Graduate Studies

(Chemical and Biological Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

April, 2009

Abstract

The dynamic behavior of many chemical and biological processes is defined by a set of nonlinear differential equations that constitute a model. These models typically contain parameters that need to be estimated using experimental data. A number of factors such as sampling intervals, number of measurements and noise level characterize the quality of data, and have a direct effect on the quality of estimated parameters. The quality of experimental data is rather poor in many processes due to instrument limitations or other physical and economical constraints. Traditional parameter estimation methods either yield inaccurate results or are not applicable when applied to such data. Despite this, it is common practice to apply them on a merged data set obtained by pooling together data from multiple experiments. Considering the difficulties in maintaining consistent experimental conditions, straightforward integration of multiple data sets will not provide the best estimates of parameters.

In this thesis, a new approach to estimate parameters of nonlinear dynamic models using multiple experimental data is proposed. The approach uses Bayesian inference, and sequentially updates prior probability distribution of parameters for systematic integration of multiple data sets. An expression for posterior probability distribution of parameters conditional on all experimental data sets is derived. This expression is often analytically intractable; therefore two instances of numerical approximation method called Markov Chain Monte Carlo - Metropolis-Hastings (MH) algorithm and Gibbs sampler (GS) - are implemented. The two algorithms form inner and outer levels of iterations, where the MH algorithm is used in the inner level to estimate conditional probability distributions of individual parameters, which is used in the outer level in conjunction with the GS to estimate joint probability distributions of the parameters.

The proposed method is applied to three nonlinear biological processes to estimate probability distribution of parameters with a small number of irregular samples. The approximated probability distribution provides a straightforward tool to calculate confidence interval of parameter estimates and is robust to initial guess of parameter value. Correlation among model parameters, quality of each model, and the approach taken to optimize the high cost of MCMC sampling are discussed.

Table of Contents

Appendices

List of Tables

List of Figures

Acknowledgements

My sincerest gratitude is due toward my supervisor, Dr. R. Bhushan Gopaluni, who guided and inspired me, and provided me with an incredible opportunity. His support and generosity during my first step in graduate school sparked my interest and opened up my eyes to the wonderful potential that awaits me. And most of all, he has shown me how to carry myself with grace and kindness through this challenging process. I would like to acknowledge Dr. Ryozo Nagamune for demonstrating the wit and senses required to keep up my courage. He was never hesitant to provide me with his insights when I was struggling with my direction. Dr. Amos Ben-Zvi from University of Alberta deserves a special mention for preparing me for some of the difficult obstacles that I will face in the future.

Dr. Ezra Kwok and Dr. James Piret have been the most gracious members of my research committee and I am very thankful for their valuable advice. I am also grateful for the members of the UBC Control reading group, Drs. Oishi, Nagamune, Gopaluni, and Sassani and their students for their feedback, insights and advice. I would also like to thank the faculty and staff of Chemical and Biological Engineering Department for making up such a great environment for me to learn and grow, and for being my mentors.

My appreciation goes out to my friends - Jackie, for being the truest friend a person can hope for; Roger, Nancy and Kathy for showing me that I'm never alone and being there for all the personal hardships; Adriana for always being the calming presence in my life; Denis for somehow convincing me to join his team; Brenda, Shelly and David, who make up my fondest memories of Vancouver; and Stefan for all the challenges and support that made me stronger everyday.

Lastly, I would like to express my deepest appreciation and respect for my family. I thank my grandparents for raising me with great discipline and integrity so that I can be true to my work. All the privileges and opportunities I've been given in this life would not be possible without them. Words cannot express the respect I have for my parents, for setting the standard of strength and brilliance in all that I must accomplish. Mom, I would count myself lucky if I can match the kind of passion and dedication you have for your work. Dad, whatever path I choose, I know I have your support and I am ever grateful for it. A big thank you to my big brother, Wonsik, and his wife, Sarah, for paving the way for their little sister. And lastly, I would like to thank my uncle, aunt and cousins, Grace and Carol, for being the home away from home.

Dedication

To my parents.

Chapter 1

Introduction to Modeling Nonlinear Dynamical Systems

A variety of aspects of parameter estimation for nonlinear stochastic dynamical systems are defined in this chapter. Major components of dynamic modeling such as model structure, model identification, and model validation are discussed. The success of model identification and validation steps is largely influenced by the quality of experimental data, and hence aspects of data quality that have adverse effect on estimation accuracy are examined. Some commonly used modeling frameworks are presented. This is followed by a description of three nonlinear biological systems that are used throughout this thesis to illustrate the proposed algorithms. The chapter concludes with an overview of the thesis.

1.1 Modeling Framework

The dynamic behavior of many chemical and biological processes is defined by a set of mathematical equations that constitute a model. A typical model consists of four major components *viz.,* model structure, independent variables, dependent variables and model parameters. The structure refers to the way variables and model parameters are related to each other by mathematical operators. The independent variables, often called inputs, are those that affect other process variables but are not affected by them. Some input variables can be directly manipulated and are used to control the output of the system. The dependent variables can further be classified into states and outputs. The state variables are required to describe the internal dynamics of a process, however, they are usually not directly measured. The output variables are directly

measured and are generally functions of state and input variables. These output variables can be predicted given the model structure, measurements of input variables, and model parameters. The model parameters are usually constants that relate independent and dependent variables through the model structure. Once these major components - model structure, independent variables, dependent variables and model parameters - are defined (or estimated), the model can be used to infer the dynamic behavior of the corresponding process, and to develop algorithms for control, fault detection and process monitoring.

Modeling of dynamic processes is an iterative approach that consists of i) identifying independent and dependent variables, ii) selecting a proper model structure, iii) estimating model parameters using experimental data, and iv) validating and revising the structure and model parameters based on certain model quality criteria. It is straightforward to identify the independent and dependent variables based on process information. The structure of a model can either be determined through physical laws, such as mass and energy balance equations, or through empirical approximations based on basis functions [35]. The emphasis in this thesis is on structures based on physical laws as they are very common in biotechnology and biomedical industries. Such structures provide intuition into the process behavior and its various measurements, and give physical meaning to its parameters. For instance, a simple batch fermentation reaction can be described by the following coupled ordinary differential equations,

$$\frac{dC_X}{dt} = \frac{\mu_m C_S}{k_s + C_S} C_X - k_d C_X, \tag{1.1}$$

$$\frac{dC_S}{dt} = -\frac{\mu_m C_S}{(k_s + C_S) Y_{xs}} C_X, \tag{1.2}$$

$$\hat{C}_X = C_X + \eta_X, \tag{1.3}$$

$$\hat{C}_Y = C_Y + \eta_Y. \tag{1.4}$$

where C_X and C_S denote the concentration of biomass and substrate respectively, $\hat{C}_X$ and $\hat{C}_Y$ denote measured values of the concentrations corrupted with noise sequences η_X and η_Y. The structure of the model refers to the manner in which these two concentrations are related through

the above equations. This model is based on mass balance equations and hence provides physical meaning to various constants in it. The model parameters, $\boldsymbol{\theta} = [\mu_m, k_d, k_s, Y_{xs}]$, are the constants. They are estimated using experimental measurements of biomass and substrate concentrations. Once the parameters are estimated, the process can be quantitatively represented with the values of the parameters. In this case, the parameters are meaningful physical quantities in that μ_m is the maximum growth rate of biomass, k_d is the decay rate of biomass, k_s is the Monod constant which is equal to the substrate concentration at which the biomass growth rate reaches half of its maximum growth rate, (μ_m), and Y_{xs} is the stoichiometric yield coefficient of biomass to substrate.

Once a model structure is in place, the parameters are chosen such that the model predictions are, in some sense, close to the actual process measurements. The accuracy of the estimated parameters depends on the quality of the process measurements. A detailed description of various aspects of the data that affect the parameter estimation are described in a latter section. Once the parameters are estimated, the model is tested for its accuracy against a new set of process measurements to verify the validity of the model on measurements not used in the estimation step. This step is called model validation. If a model fails the validation step, experiments are repeated to collect more data.

Mathematical models can be broadly classified as linear or nonlinear, static or dynamic, deterministic or stochastic. A linear model is one where the variables of the system are related through linear differential equations. Within most biological and chemical processes, there are complex bio/chemical reactions that cannot be expressed using linear differential equations, thus many models developed for such processes are nonlinear in either variables or parameters and sometimes both. A dynamic model accounts for the rate of change of process variables while a static model assumes that the variables are constant. A deterministic model assumes that if multiple experiments were conducted with identical initial conditions and experimental variables, the observed time series of output variables will always be the same. However, this is

not the case in reality due to process and measurement noise. Multiple observations made of a system under identical experimental conditions vary with some probability distribution. Thus, a stochastic view that considers the process variables and parameters as random variables with probabilistic qualities is more appropriate for biological systems. Furthermore, such stochastic models are suitable for applications where the parameters are time varying [5]. The fermentation model described earlier is a stochastic nonlinear model with constant parameters. The focus of this thesis is on estimating model parameters for a generic nonlinear stochastic model.

1.2 Previous Work

Modeling has been an active area of research for over a century and it is beyond the scope of this thesis to provide an exhaustive literature survey. However, the research in this area can be broadly classified into two areas - linear and nonlinear modeling. A number of textbooks have been written on the topic of linear modeling [12, 16, 35, 50], and many theoretical results on the quality of estimated parameters have been derived. Similarly, a number of approaches have been developed for modeling nonlinear dynamic processes [4, 19, 31, 35, 45, 46]. The most popular among them are black-box techniques such as neural network models, state space techniques such as hidden-Markov models, and continuous time modeling techniques such as ordinary differential equation (ODE) models. While an exhaustive literature survey of these techniques is beyond the scope of this thesis, applications of these approaches are briefly discussed below.

The neural network models use the learning principles of mammalian brain. The relationship between input, output, and state variables is developed by training the neural network models with large amounts of process data. Some notable works using neural networks for model identification include Bohr *et al.*'s work on predicting 3-D structure of protein backbones [7]; Cai *et al.*'s work on predicting the content of protein 2-D structures [9]; Fukushima's proposal on synaptic network between neurons [18]; Gupta and Achenie identified metabolic pathways, complex genetic disease and toxicology analysis through neural network modeling [25]; Laursena *et al.*

used gray-box model approach to identify the complex behavior of bacteria after induction [34]; Maraziotis and colleagues' development of complex causal relationships among genes from microarray experimental data is based on novel neural fuzzy recurrent network [37]; and Zhang's work on developing mechanistic model for batch process is based on using stacked neural network models [56]. Complex nonlinear relationships between various process variables can be implicitly detected using neural network models. However, there are a few drawbacks to using neural network models. Neural network models are known to require heavy computational effort, prone to over-fitting, require large amounts of data, and the developed models suffer from the disadvantages of black-box models [53].

Hidden-Markov models (HMM) were initially developed in the field of speech recognition in 1970s and later became popular in other disciplines because of their characteristically rich mathematical structure [43]. Major components of a HMM are states, observations, transition probabilities corresponding to each state, emission probabilities and initial state distribution. HMM is often portrayed with a diagram, in which the states are denoted with network of nodes and the transition and emission probabilities are denoted with arrows connecting the nodes. If some initial probability distribution of state is given, HMM can compute the probability distribution function of output variables. Since initial probability distribution of the states can be specified by the user, it is possible to incorporate prior knowledge regarding the probability of states when using HMM. Other advantages include ease of interpretability (because of the intuitive nodal network) and its modular nature, making it easy to combine several models to create a larger one [28]. The disadvantages are that the computational cost is very large and that there are several strong assumptions made regarding the process, such as Gaussian distribution of process variables, and Markov property of emission and transition probabilities. A number of applications of HMM framework can be found in computational biology field. Some notable works include Kimmel and Shamir's development of a novel HMM for identifying haplotype and genotype generation [29]; Wu and Xie proposed a HMM of transcription factor binding sites and cis-regulatory modules identification [54]; Siepel and Haussler combined phylogenetic model with

HMMs to explore the genome substitution that occurs through evolution [48]; Krogh *et al.*'s statistical model and multiple sequence alignment of protein families and protein domains is also based on HMM framework [32].

Ordinary differential equation (ODE) models are used in a wide range of scientific disciplines to describe biochemical systems, fluid mechanics, financial markets, etc. A generic ODE model can be represented by the following equations,

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\theta}) + \mathbf{v}(t) \tag{1.5}$$

$$\mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\theta}) + \boldsymbol{\eta}(t) \tag{1.6}$$

where $\mathbf{x}(t)$ is the vector of state variables, $\mathbf{u}(t)$ is the vector of exogenous input variables, $\boldsymbol{\theta}$ is the vector of model parameters and $\mathbf{y}(t)$ is the vector of output variables. The process noise term, $\mathbf{v}(t)$, and measurement noise term, $\boldsymbol{\eta}(t)$, are included in the model to account for the stochastic nature of the process; without these terms, the model is deterministic. The exact values of these noise variables are unknown, however, assumptions can be made regarding the probability distribution of these variables (e.g. $\boldsymbol{\eta}$ is usually assumed to be Normally distributed with zero mean and some variance $\boldsymbol{\sigma}_\eta^2$, such that $\eta \sim \mathcal{N}(0, \boldsymbol{\sigma}_\eta^2)$). $\mathbf{f}$ and $\mathbf{h}$ are vectors of functions that form the model structure and they determine whether the model is linear or nonlinear. If the model equations are nonlinear, it is often impossible to obtain analytical expressions for the states and outputs, and one has to approximate the model equations through numerical analysis. The ODE models are typically developed using physical laws, and hence various parameters in these models have physical significance. In this thesis, the focus is on estimating model parameters in a stochastic nonlinear ODE model described above. The parameters can be easily estimated through nonlinear least squares if there is no noise in the state dynamic equations. However, in presence of state noise, parameter estimation is a difficult problem. There are a few maximum likelihood approaches, that are based on expectation maximization, to estimate parameters in presence of state noise. These approaches require large amounts of data for good estimates.

Moreover, maximum likelihood approaches do not automatically provide confidence intervals on the estimated parameters.

In this thesis, a Bayesian approach is proposed to estimate parameters in a set of stochastic nonlinear differential equations. The proposed approach does not require large data sets, it can easily handle multi-rate measurements, and provides confidence intervals on estimated parameters. This approach is also extended to handle data from multiple experiments.

1.3 Motivation for a Bayesian Approach to Parameter Estimation

Once the structure of a model is defined, the parameters are estimated using the data obtained from the process. The quality of the data will therefore influence the quality of the estimated parameters. In this thesis, the goal is to develop a Bayesian approach to parameter estimation that accounts for three important features of data *viz.*, noise, scarcity and irregular samples.

Most experimental data are corrupt with measurement and process noise, making it difficult to obtain the precise values of model parameters. Thus, the estimated values contain some uncertainty, which is usually quantified with a confidence interval. The size of the confidence intervals of estimated parameters is an indication of the effectiveness of the proposed model in accurately representing the process. The accuracy of estimation and its reliability are affected by the level of noise present in the experimental data, and therefore a robust method to obtain as much information as possible from the noisy data is required. If the estimated parameters suffer from large levels of uncertainty (e.g. wide confidence interval) then the model is deemed to be a poor representation of the process. Therefore, in such a case, the proposed model needs to be examined for revision. This is done so that the revised model yields estimated parameters, using a new set of experimental data, with smaller level of uncertainty. Thus, the iterative process of proposing

a model - estimating the model parameters and revising the model - is required to obtain a model that is able to capture the key components of the process and is not excessively complex.

The frequency of process measurements, in many biological processes, is determined by various physical and economical constraints. These constraints often lead to data sets that are too small and contain irregular samples. The parameters estimated from such data sets usually have large confidence intervals. Therefore, point estimates of the parameters obtained through methods such as maximum likelihood and nonlinear least squares, have to be qualified with a description of their confidence intervals.

Various parameter estimation methods are adept at addressing poor quality of data - sparsity, irregularity and large amount of noise. For instance, Expectation Maximization (EM) algorithm, an instance of maximum likelihood estimation (MLE), has been shown to handle irregular process data well enough to obtain reliable parameter estimates. However, it still requires a large number of samples in order to overcome the loss in information due to irregularity in sampling intervals. It is known that MLE (or EM) yields asymptotically unbiased and minimum variance parameter estimates as the number of data points reaches infinity. However, MLE is prone to biased estimation when not enough data is provided. An alternative approach to address the problem of sample sparsity is to conduct multiple experiments and pool the data points together to create a large data set. Though this approach seems straightforward, it is difficult to maintain consistent process conditions during different experimental runs. Hence, when a nonlinear least-squares or MLE approach is applied on such pooled data the differences in experimental conditions are obfuscated.

There are other challenges presented by least-squares (LS) approaches and frequentist methods, which refer to statistical point of view where probability of a given random event is obtained through large amount of observations. When estimating process parameters, the accuracy of the estimation can be improved by exploiting all of the available information. Aside from the

obvious choice of experimental data, *a priori* information can also be included in this information database. *A priori* information may refer to any knowledge regarding the parameters that is available before conducting experiments, such as constraints on the parameter value available through physical laws and other means. Unfortunately, in the currently used techniques for parameter estimation, these types of information are applied in a somewhat limited fashion. For instance, one can provide the upper and lower bounds of the parameter value to an optimization problem that solves nonlinear least-squares or MLE. However, if there's a probabilistic *a priori* information, such that x is normally distributed with some mean μ and variance σ^2, it is difficult to incorporate this information into LS or frequentist estimation methods. Lastly, for computational convenience, these methods make an assumption that state variables and parameter vector have a fixed time invariant distribution (e.g. Gaussian). However, it has been shown that this assumption may be inaccurate in many nonlinear processes. In the work by Chen et al., simulated posterior distribution of a CSTR concentration was shown to exhibit time-variant behavior with multimodal, asymmetric distribution [13].

Bayesian approaches are naturally suited to handle not only *a priori* information but also scarce, irregularly sampled noisy data. In this thesis, a parameter estimation method that resolves the above mentioned challenges by incorporating *a priori* information is developed. The proposed algorithms are illustrated through three ordinary differential equation models: i) a batch fermentation reaction, ii) a genetic regulatory network and iii) a signal transduction pathway. These models are referred to throughout this work to demonstrate the proposed algorithm and the theories behind it.

1.4 Motivational Examples

1.4.1 Batch Fermentation Reaction

First example is the simplified version of Michaelis-Menten reaction model in a batch reactor. Following set of coupled ordinary differential equations is used to describe the uninhibited growth of biomass and increasing concentration of its by-products (e.g. alcohol), as well as depletion of substrate present in a batch fermentor.

$$\frac{dC_X(t)}{dt} = \mu C_X(t) + v_1(t), \tag{1.7}$$

$$\frac{dC_S(t)}{dt} = -\frac{\mu}{Y_{XS}} C_X(t) + v_2(t), \tag{1.8}$$

$$\frac{dC_P(t)}{dt} = \mu Y_{PX} C_X(t) + v_3(t). \tag{1.9}$$

The state variables C_X, C_S and C_P are concentrations of biomass, substrate and by-products, respectively. The stochastic process noise is indicated by v_1, v_2 and v_3. The specific growth rate of biomass, μ, is defined using Michaelis-Menten kinetics as follows.

$$\mu = \frac{\mu_m C_S(t)}{k_s + C_S(t)} \left(1 - \frac{C_P(t)}{k_P}\right) \tag{1.10}$$

The model is nonlinear with respect to the five parameters: (i) μ_m, upper limit to the growth rate of biomass, (ii) k_s, Monod constant, (iii) k_p, product inhibition term, (iv) Y_{XS}, yield ratio of biomass concentration to the substrate uptake and (v) Y_{PX}, yield ratio of by-product to biomass. Thus, the model parameter vector is defined as $\theta = [\mu_m, k_s, k_p, Y_{XS}, Y_{PX}]$.

The details of simulation data sets for this stochastic dynamic system are provided in Appendix A. For a numerically stable implementation of the algorithm, two of the parameters k_p and Y_{PX} were normalized using the following convention and the normalized values are shown in

Table 1.1 along with their pre-normalized values.

$$k'_P = 10/k_p, \tag{1.11}$$

$$Y'_{PX} = Y_{PX}/100. \tag{1.12}$$

A total of six independently simulated data sets ($\mathbf{D} = \{D_1, D_2, D_3, D_4, D_5, D_6\}$), each data set consisting of $N = 15$ irregularly sampled data points spanning the time interval of $[0, 24]$ hours, are collected. A single set of experimental data is shown in Figure 1.1. The true trajectories of the state variables, without measurement noise, are indicated with solid lines and the measured sample points are indicated with x's. It is ensured that the sampling intervals are irregular and the measurements are corrupt with noise.

1.4.2 Feed-Forward Loop : Genetic Regulation Network

Gene expression is the foundation of regulatory biological functions [1]. Within cells, transcription factors are triggered by various environmental changes or self-serving signals. Through the recent development in molecular biology technology, it has become possible to closely study the dynamic behavior of networks formed by a group of transcription factors present in living organisms. Recently, a number of predominantly recurring wiring patterns within genetic networks, called *network motifs*, were identified in bacterium *Escherichia coli* and yeast *Saccharomyces cerevisiae*. Their observed abundance is assumed to be due to their significant role in the transcription network. One of the identified motifs is called the Feed-Forward Loop (FFL) [36, 47] and it shows similarities in its regulation action to a feed-forward action in process control.

Two different regulating actions are present in a general form of FFL, where they involve two transcription factors, X and Y, and a gene Z. The first action is regulation of Y expression by X and the second action is regulation of Z expression by both X and Y. A graphical illustration of FFL mechanism is shown in Figure 1.2, where S_X and S_Y are the inducers of X and Y, respectively. The inducers are either saturating stimulus or absent, and they trigger the transcription

factors. There are three transcription interactions (denoted with the three arrows in the figure) which can either act as an activator or a repressor, resulting in eight possible configurations of FFL. The following set of coupled ordinary differential equations is the general form of FFL.

$$\frac{dY(t)}{dt} = -\alpha_y Y(t) + \beta_y f(X(t), K_{XY}) + v_1(t), \tag{1.13}$$

$$\frac{dZ(t)}{dt} = -\alpha_z Z(t) + \beta_z g(X(t), Y(t), K_{XZ}, K_{YZ}) + v_2(t), \tag{1.14}$$

where $X(t), Y(t)$ and $Z(t)$ denote the gene expression rate of transcription factor X, Y and gene Z at time t. In all FFL configurations, the expression of X is assumed to be constitutive, which means that it is continuously produced within the organism regardless of the cell's need. For each different configuration of FFL, the functional forms of f and g have different expressions. The regulation function f, when it is an activator, is defined as follows.

$$f(\chi, K_{ij}) = (\chi/K_{ij})^H/(1 + (\chi/K_{ij})^H), \qquad \chi = X(t), Y(t). \tag{1.15}$$

where coefficient H indicates the steepness of the activation function f and K_{ij} is the activation or repression coefficient of gene j by transcription factor gene i. The gate function g can either be an AND-gate or an OR-gate, where in an AND-gate it is assumed that X and Y regulate Z independently and not compete with each other, making the activation function of X and Y to Z equal to the product of two activation functions, $f(X(t), K_{XZ})$ and $f(Y(t), K_{YZ})$. On the other hand, for an OR-gate, the transcription factors X and Y compete for the binding site in a promoter, and the gate function is expressed as a linear combination of the two activation functions.

There are eight possible configurations of FFL, four coherent types and four incoherent types. When a given FFL's indirect causal regulation of X to Z through Y agrees with its direct causal regulation of X to Z (e.g. both regulations activate or both regulations repress), then it is a coherent type of FFL, otherwise it is an incoherent type of FFL. In this thesis, coherent type 1 FFL is considered. In it, transcription factor X activates both gene Z and transcription factor

Table 1.1: The true and normalized parameter values used to simulate the batch fermentation reaction data, and to implement the estimation algorithm.

Process Parameter	μ_m	k_s	k_P	Y_{XS}	Y_{PX}
Value	0.15 [1/hr]	0.5 [g/L]	40 [g/L]	0.25 [g/g]	20 [g/g]
Normalized Value	0.15 [1/hr]	0.5 [g/L]	0.25 [L/g]	0.25 [g/g]	0.2 [g/g]

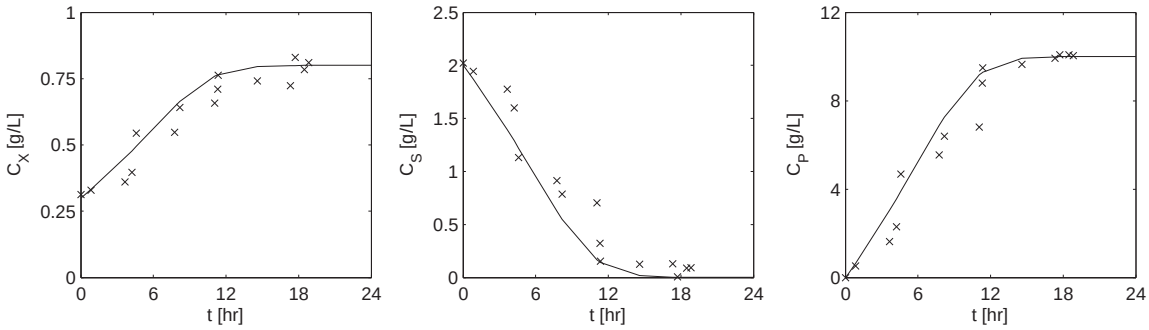

Figure 1.1: Batch fermentation reaction from $t = 0\,\text{hr}$ to $t = 24\,\text{hr}$. From left, each panel corresponds to the concentration of biomass (C_X), substrate (C_S) and by-product (C_P). Measured data points are denoted with 'x' and the solid curve denotes the 'true' trajectory of each state variables.

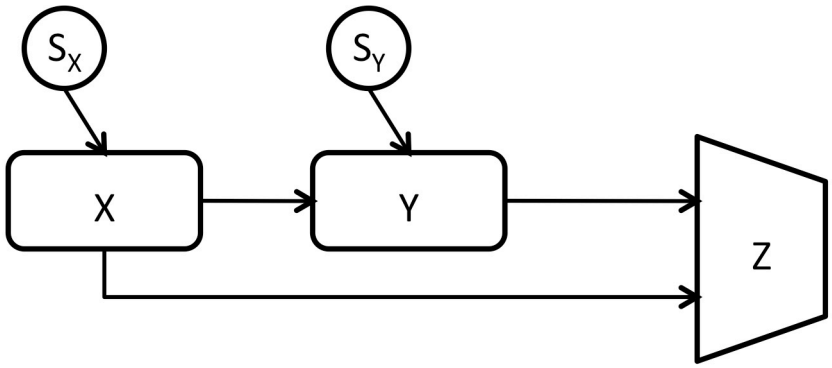

Figure 1.2: Feed-Forward Loop model where transcription factor X regulates the expression of Y and both X and Y regulate the expression of Z. S_X and S_Y are inducers of X and Y, respectively [36].

Y, which in turn activates the expression of gene Z. The inducers of X and Y, S_X and S_Y, both have a strong effect in the expression level of Z in coherent type 1, unlike in types 3 and 4. This is suggested to be a possible explanation for the dominance of coherent type 1 in the evolutionary process of transcription networks. Observations show that coherent type 1 FFL is the most frequently observed FFL type in *E. coli* and *S. cerevisiae* with 28 occurrence out of 42 identified FFLs and 26 occurrence out of 56 identified FFLs, respectively [36].

In [20], the expression levels of yeast *S. cerevisiae* gene were identified under a number of stimulations, including heat shock, toxicity level, and substrate concentration, in order to study the gene expression patterns. There were around 6,200 genes that were identified and about 900 among them showed similar pattern that can be further studied to identify the role of each genomic response. Since FFL is a recurring pattern involving a group of three genes X-Y-Z, a number of different FFLs can be identified within a single organism, each with a different group of X-Y-Z. One of the identified coherent type 1 FFL in *S. cerevisiae* is GCN4-LEU3-ILV5 group and in Figure 1.3 the expression level time series of these genes published in [20] are shown. The expression levels of GCN4, LEU3 and ILV5 are denoted with $X(t), Y(t)$ and $Z(t)$, respectively. The expression levels were measured at $t = 5, 10, 15, 20, 30, 40, 60, 80$ minutes ($N = 8$ samples) where at $t = 0$ min, the environmental temperature was raised from $25°$C to $37°$C.

The FFL model has a total of eight process parameters, which are $\boldsymbol{\theta} = [\beta_y, \beta_z, \alpha_y, \alpha_z, K_{XY}, K_{XZ}, K_{YZ}, H]$. The parameters, α_y, α_z, represent the sum of degradation rate and dilution rate of Y and Z, respectively. Using these parameters, the half-time of Y and Z decay ($t_{1/2}$) is obtained as $\log(2)/\alpha_y$ and $\log(2)/\alpha_z$. In order to obtain simplified version of the FFL interactions, a few parameters are assigned approximate values based on *a priori* information ($\beta_y = 1$, $\beta_z = 1$ and $H = 2$).

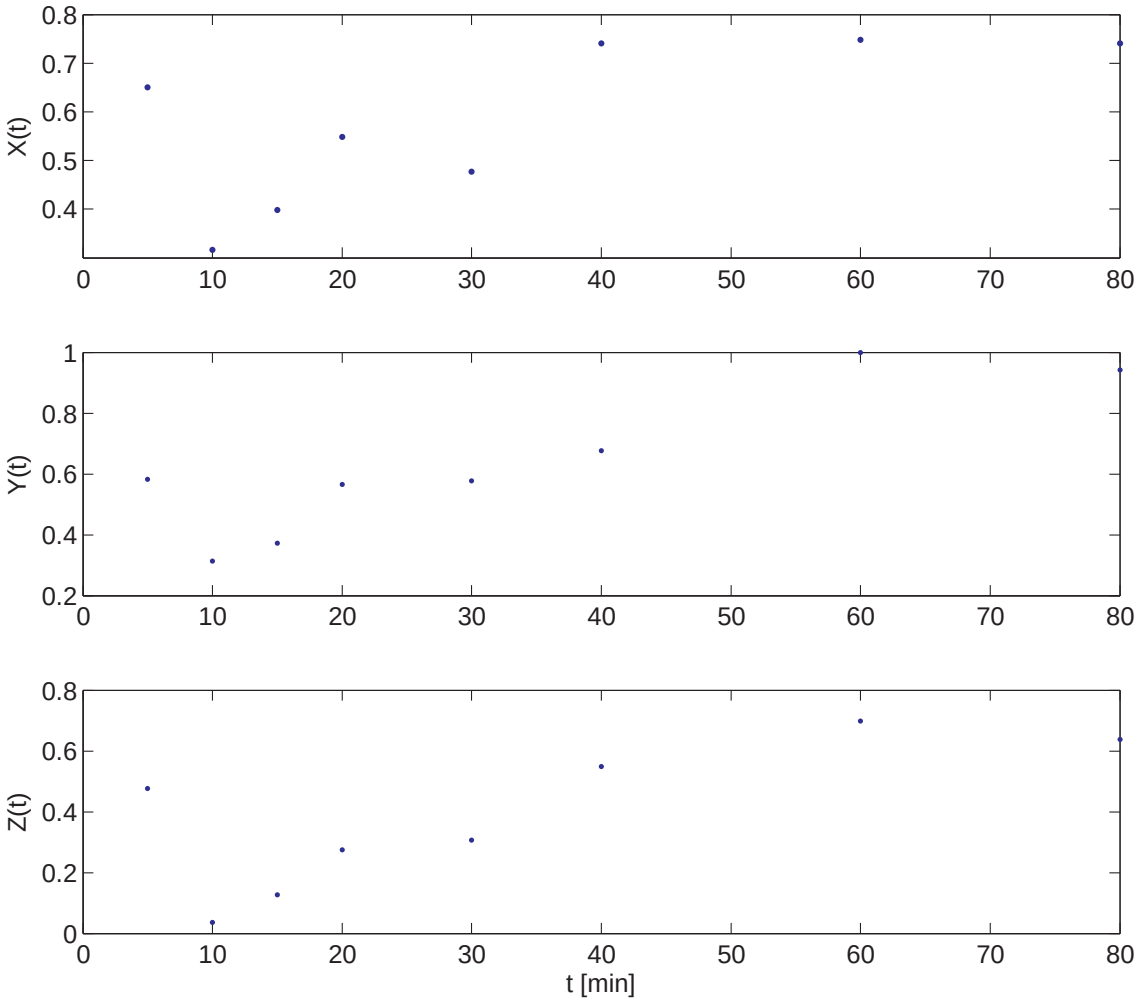

Figure 1.3: Experimental time series data of coherent type 1 FFL gene expression level where $X(t)$ denotes the expression level of GCN4, $Y(t)$ denotes the expression level of LEU3 and $Z(t)$ denotes the expression level of ILV5. [20].

1.4.3 JAK-STAT : Signal Transduction Pathway

The third example is that of signal transduction pathways. The Janus family of kinases (JAK) - signal transducer and activator of transcription (STAT) pathway describes the series of reactions taking place across cytoplasm and nucleus to trigger transcription of key genes. The signaling pathway occurs through multiple cell surface receptors, one of them being the erythropoietin receptor (EpoR). EpoR plays an important role in the proliferation and differentiation of erythroid progenitor cells [51], which refer to cells that are able to grow into a specific type of cell - in this case, red blood cell - through cell-division [33]. Figure 1.4 shows the diagram of the JAK-STAT signal transduction pathway. Through a series of reactions, EpoR creates docking sites for STAT5, a latent transcription factor. The key actions taken by STAT5 are phosphorylation (x_1 to x_2, in Figure 1.4), formation of dimers (x_2 to x_3, in Figure 1.4), and migration from cytoplasm into nucleus (x_3 to x_4, in Figure 1.4). Once present in the nucleus, STAT5 is able to trigger the transcription of target genes. There are several hypotheses for the termination mechanism of JAK-STAT pathway, including degradation of STAT5 within the nucleus and migration of STAT5 from nucleus back to cytoplasm.

The mathematical model of JAK-STAT signaling pathway was originally developed in [51]. There are four state variables which represent the concentrations of unphosphorylated STAT5 (x_1), tyrosine phosphorylated monomeric STAT5 (x_2), tyrosine phosphorylated dimeric STAT5 (x_3) and STAT5 within the nucleus (x_4). The exogenous input variable of the model, $u(t)$, is the concentration of EpoR. The original model was adapted following the suggestion in [42, 57] and expressed as a set of four coupled ordinary differential equations as follows.

$$\frac{dx_1(t)}{dt} = -a_1 x_1(t) u(t) + 2a_4 x_4(t) I_{\{t \geq \tau\}}, \tag{1.16}$$

$$\frac{dx_2(t)}{dt} = a_1 x_1(t) u(t) - 2a_4 x_2^2(t), \tag{1.17}$$

$$\frac{dx_3(t)}{dt} = -a_3 x_3(t) + x_2^2(t), \tag{1.18}$$

$$\frac{dx_4(t)}{dt} = a_3 x_3(t) - a_4 x_4(t) I_{\{t \geq \tau\}}, \tag{1.19}$$

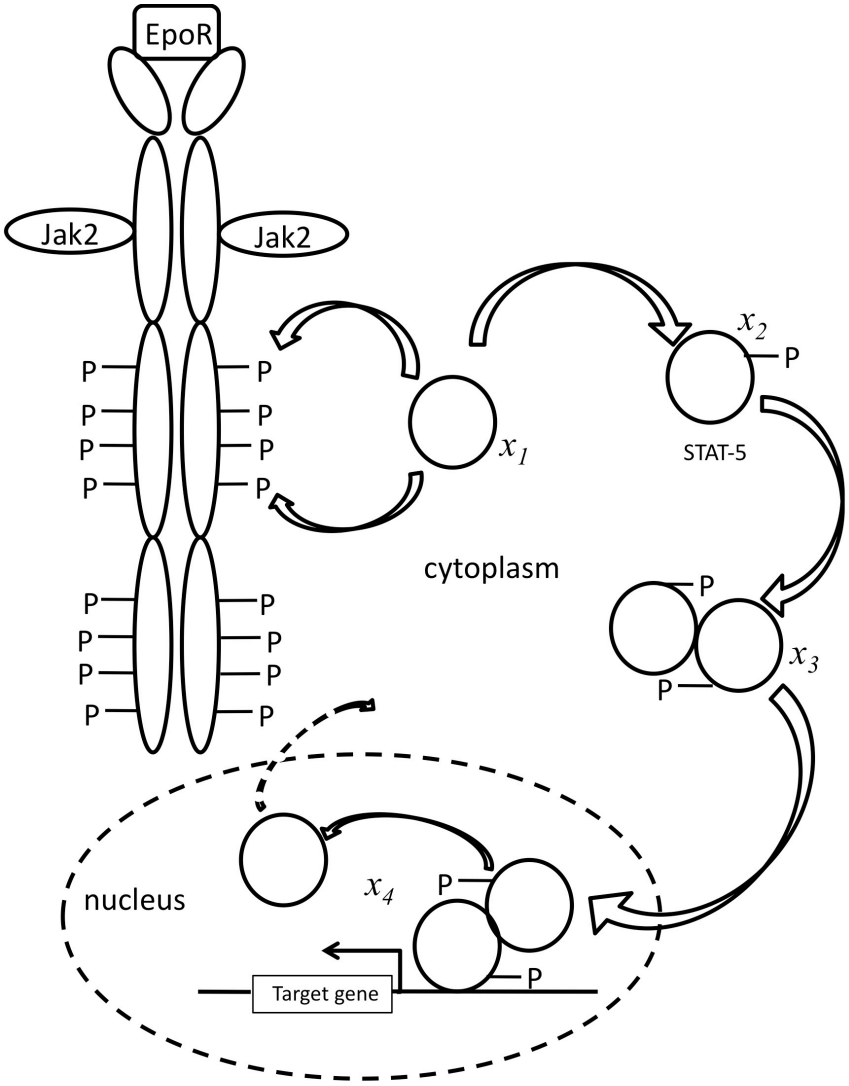

Figure 1.4: JAK-STAT signal transduction pathway diagram [51]. Initially, EpoR creates docking sites for STAT5. This triggers a series of STAT5 reaction where unphosphorylated monomeric STAT5 (x_1) becomes phosphorylated monomeric STAT5 (x_2), which in turn forms phosphorylated dimeric STAT5 (x_3) that migrates into the nucleus. Once inside the nucleus, phosphorylated dimeric STAT5 (x_4) triggers the expression of target gene. The signal transduction pathway terminates, by migration of STAT5 from nucleus back to cytoplasm

where $I_{\{t\geq\tau\}}$ is an indicator function that is equal to zero when $t < \tau$ and is equal to one when $t \geq \tau$. Since there is a time delay from the initial addition of EpoR into the system, triggering the activation of STAT5 signal transduction pathway, to the time STAT5 actually migrates into nucleus, τ is present in the model to account for such time delay. Though this model is suggested to be a simplified version of the original model developed in [51], one aspect of the model seems to result in over-lumping of the parameters in order to reduce the dimension of the parameter space. This aspect is regarding the reduction of phosphorylated STAT5 (x_2) due to the formation of phosphorylated dimeric STAT5 (x_3). These are allegedly represented with the second terms in (1.17) and (1.18). However, the lack of $2a_4$ term in (1.18) seems inconsistent with the physical explanation of the process, and is a worthwhile problem to be addressed in the context of model validation.[1] It is proven to be very difficult to monitor the population of individual types of STAT5 within the process. Thus, the following two output variables were measured, instead of direct measurement of each state variable.

$$\begin{aligned} y_1(t) &= x_2(t) + 2x_3(t), \\ y_2(t) &= x_1(t) + x_2(t) + 2x_3(t), \end{aligned} \tag{1.20}$$

where y_1 denotes the amount of tyrosine phosphorylated STAT5 in cytoplasm and y_2 denotes the total amount of STAT5 in cytoplasm. These output variables are expressed as linear combination of the state variables that depict their stoichiometric relationship. The JAK-STAT model is more complex compared to the FFL model introduced in the previous section, because the direct measurements of state variables are not available. Furthermore, x_4 is not accounted for in the output measurements at all, but only present in the state equations as a functional component of rate of change for x_1 and x_4. This is because making measurements of population of STAT5 within the nucleus is difficult.

[1] One speculation is that when the simplified model was devised, the modelers accounted for degradation of x_2 in the rate of $-(2a_4 - 1)x_2^2$, which would account for the difference in the decrease in x_2 population and the increase in the x_3 population. This is just one conjecture among many, though following this logic would place a constraint on the parameter such that $a_4 \geq 0.5$.

An experimental data set published by [51] is shown in Figure 1.5, where the output variables y_1 and y_2 are measured at $t = 0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 25, 30, 40, 60$ minutes. At $t = 0$ minutes, the population of phosphorylated STAT5 is 0 ($y_2(t = 0) = 0$). This is because phosphorylation of STAT5 is triggered by the addition of EpoR into the system which occurs at $t = 0$ minute.

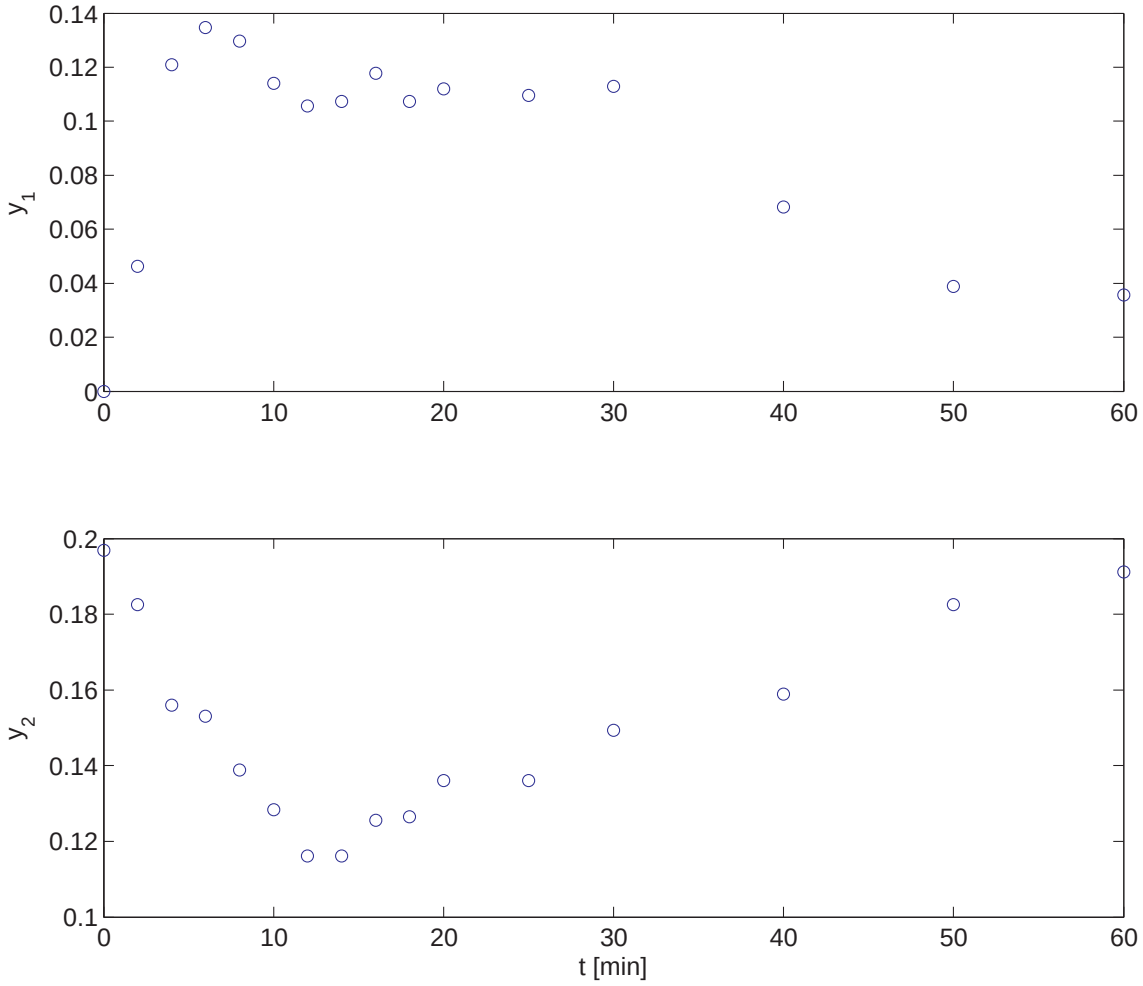

Figure 1.5: Experimental time series data of JAK-STAT signal transduction pathway experimental data. $y_1(t)$ and $y_2(t)$ were measured at t = [0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 25, 30, 40, 60] minutes.

Certain constraints can be readily imposed on the parameters a_1, a_3 and a_4. It is straightforward to place the following constraints by relating the state equations, (1.16) - (1.19), and the reactions taking place in the process. Upon injecting EpoR, unphosphorylated STAT5 becomes phospho-

rylated, thereby reducing the initial population of x_1 from its initial value, $x_1(t_0)$ (a non-negative value), and increasing the x_2 population. This reaction is described by the first term in (1.16) with a negative sign, which is mirrored by the positive sign in the first term of (1.17). In order for this reaction to be feasible, $a_1 \geq 0$, needs to be satisfied. The second term in (1.16) indicates the increasing population in x_1; this term is present under the assumption that STAT5 that traveled to nucleus (x_4) becomes monomeric, and migrates back out to cytoplasm, thus adding to the x_1 population. This observation is mirrored in the second term of (1.19), indicating the decrease in x_4 population. Also, the indicator function conveys that until $t = \tau$, not enough x_4 will have formed within the nucleus to migrate back out to the cytoplasm. Thus $a_4 \geq 0$ needs to be satisfied for physical feasibility. Lastly, the first terms in (1.18) and (1.19) show the mirroring effect of decreasing x_3 and x_4. Therefore, the lower limit on $a_3 \geq 0$ needs to be satisfied for physical coherence of the model.

1.5 Thesis Overview

An overview of the two most widely-used parameter estimation methods for ordinary differential equation models, Maximum Likelihood Estimator (MLE) and Bayesian Inference based method, is presented in Chapter 2. The basic theory, advantages and disadvantages of each method are discussed along with their performances of handling irregularly sampled data sets.

Chapter 3 discusses the issue of handling multiple experimental data sets. For some biological processes, multiple experiments are conducted in order to create a larger merged data set so that traditional estimation methods that require larger data sets can yield reliable results. However, due to varying experimental conditions between multiple runs, the question of how the data sets can be systematically merged arises. This challenge is addressed with sequential Bayesian inference approach which is illustrated through examples.

In Chapter 4, the methodology of Markov Chain Monte Carlo (MCMC) approximation is dis-

cussed. Two instances of MCMC are explored in this thesis, and they are Metropolis-Hastings algorithm and Gibbs sampler. The two methods form two separate levels of iterative estimation in order to approximate the asymmetric probability distribution of nonlinear parameter vector. Chapter 5 presents the case study results, along with the analysis of the quality of each model used in the case study. The last chapter of the thesis presents some conclusions and recommendation for future research.

1.5.1 Problem Formulation

Consider a nonlinear process model as follows.

$$\frac{\mathbf{x}(t)}{dt} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\theta}) + \boldsymbol{v}(t), \tag{1.21}$$

$$\mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\theta}) + \boldsymbol{\eta}(t) \tag{1.22}$$

where $\boldsymbol{\theta} = [\theta_1, \cdots, \theta_m]^T$ is an m-dimensional vector of model parameters; $\mathbf{x} = [x_1, \cdots, x_p]^T$ is a p-dimensional vector of state variables; $\dot{\mathbf{x}}(t)$ is a time derivative vector of state variables; $\mathbf{u} = [u_1, \cdots, u_r]^T$ is an r-dimensional vector of input variables which are pre-determined by the experimentalist or are measured precisely and the numerical values are known; $\mathbf{y} = [y_1, \cdots, y_q]^T$ is a q-dimensional vector of output variables, i.e. the set of variables that are measured experimentally; $\mathbf{f} = [f_1, \cdots, f_p]$ and $\mathbf{h} = [h_1, \cdots, h_q]$ are a p-dimensional vector and a q-dimensional vector of functions and the form of each function is known; $\boldsymbol{v} = [v_1, \cdots, v_p]^T$ is a p-dimensional vector of process noise variables; and $\boldsymbol{\eta} = [\eta_1, \cdots, \eta_q]$ is a q-dimensional vector of measurement noise variables. The objective of parameter estimation problem is to estimate $\boldsymbol{\theta}$ from experimental data, $\mathbf{y}(t)$, which is related to the state variables, $\mathbf{x}(t)$, corrupted with noise, $\boldsymbol{v}(t)$ through some functions $\mathbf{h}$. In this thesis, the noise variables are assumed to have Gaussian distribution with zero mean and standard deviation, σ, i.e. $\mathbf{v} \sim \mathcal{N}(0, \sigma^2)$.

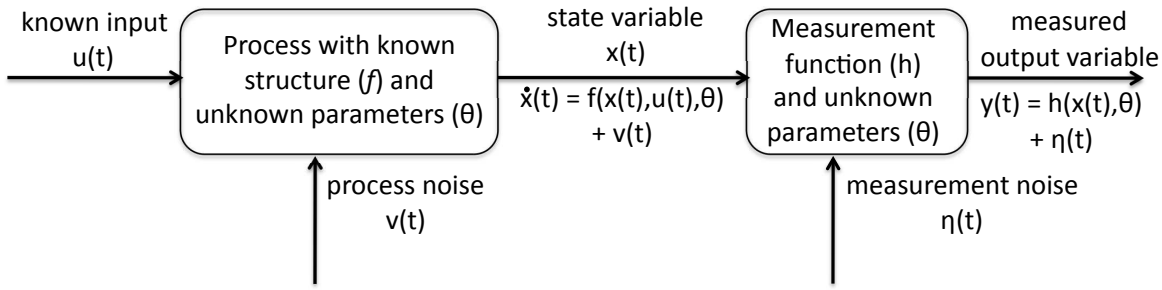

Figure 1.6: Illustration of standard parameter estimation problem. The first block represents the process and the second block represents the measurement device. Two different types of noise, process and measurement, affect the two separate blocks. The first and second blocks are related by the state variables. The observed experimental data is the output of measurement device block.

Chapter 2

Parameter Estimation for Ordinary Differential Equation models

A brief description of probability distribution function is presented, followed by two of the most widely used statistical parameter estimation methods, Maximum Likelihood Estimator (MLE) and Bayes estimator. Though straightforward to implement, MLE is prone to 'getting stuck' in local minima, which appear frequently in nonlinear systems. Bayes estimator, instead of calculating a point estimate like MLE, estimates the full probability distribution of parameters using *a priori* information. From the estimated probability distribution function, the mean, the mode and the posterior interval of the parameters can be obtained.

2.1 Probability Density Function

In statistics, the likelihood of observing some random variable is described by their corresponding probability distribution function. To explain the concept of random variable and probability distribution function, consider a game of hundred coin tosses, where a player wins if 50 or more heads (H) are observed out of the hundred tosses. The chance of observing k number of H in a series of n coin tosses is described by the Binomial distribution, $p(k) = \binom{n}{k} m^k (1-m)^{n-k}$, where $m \in [0, 1]$ is the probability of observing H in a single coin toss [24]. Here k is a random variable and $p(k)$ is the probability distribution function corresponding to k. For a fair coin that has equal chance of H or T (tail), the chance of observing H in a single coin toss is 50% ($m = 0.5$). Consider a trick coin that is unfairly weighted so that tails (T) will be observed seven out of ten tosses, then the chance of observing H is 30% ($m = 0.3$). The two different proba-

bility distributions are plotted in Figure 2.1. For instance, if a player is given the trick coin, the probability of observing 50 H's or more out of one hundred coin tosses is close to zero, and the player will most certainly lose the game. And if the unlucky player continues to play the game with the trick coin, and record the number of H he observes in each game, the average value will be 30. This is because the probability distribution is maximized at $k = 30$. In other words,

$$\arg\max_k p_2(k) = 30. \tag{2.1}$$

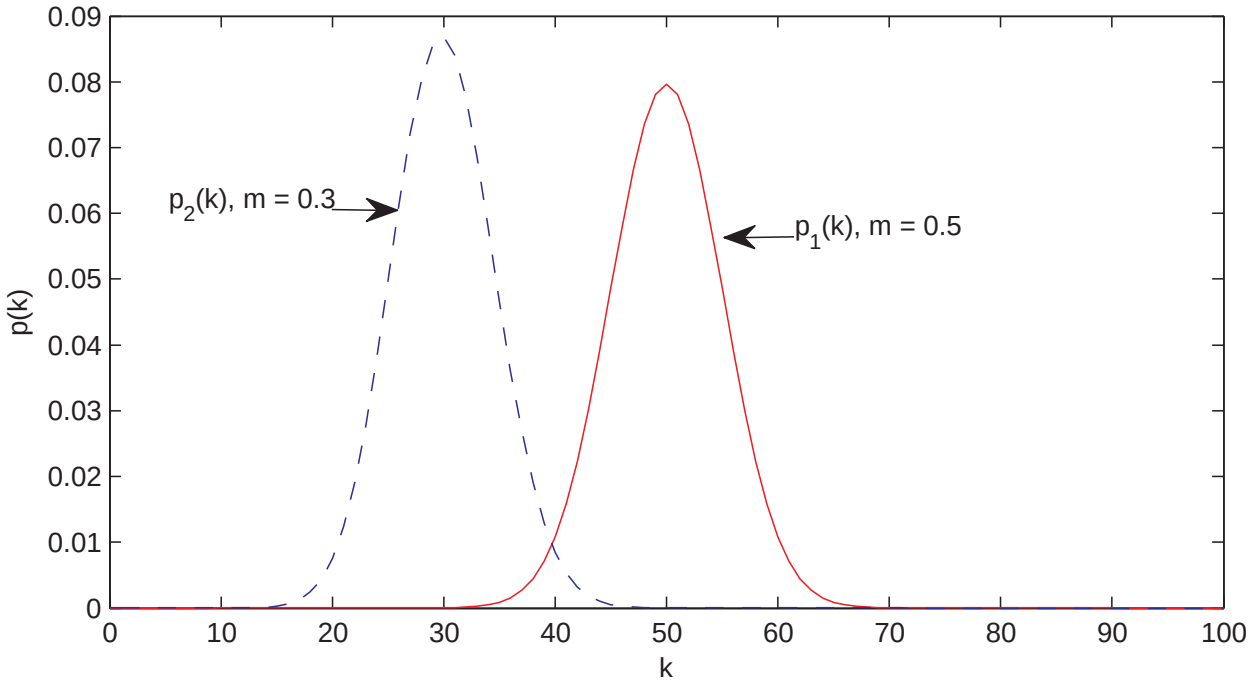

Figure 2.1: Probability distribution functions, $p_1(k)$ and $p_2(k)$, of a game of hundred coin tosses with a fair coin ($m = 0.5$) and a trick coin ($m = 0.3$), respectively. m is the chance of observing H in a single toss and k is the number of H observed during a single game.

From a probability distribution function, such as the one shown in Figure 2.1, several statistical properties, such as expected value, variance and maximum *a posteriori* estimate, of the corresponding random variable can be obtained. Thus, in statistical parameter estimation methods, the stochastic nature of measurements and process parameters is explored in order to account for the inevitable uncertainty associated with any 'real' system. For instance, in maximum likelihood

estimation (MLE), the measurements are considered as random variables, and the parameter value that maximizes the likelihood of observed data is chosen as the optimal parameter estimate. On the other hand, in Bayesian inference based approaches, the process parameters are also considered as random variables and the corresponding probability distribution is estimated rather than a point estimate. In the next section, these two methods are discussed with numerical examples.

2.2 Maximum Likelihood Estimator

Maximum Likelihood Estimator (MLE)[2] is a popular statistical estimation method and it is often applied to parameter estimation problems. In this framework, process parameters are assumed to be fixed while the process data are assumed to be stochastic variables with associated probability distribution functions. The likelihood function, central to MLE, is defined as

$$L(\theta \mid D) \quad = \quad p(D \mid \theta) \tag{2.2}$$

where θ is the process parameter vector and D is the vector of observed output and input variables, $\{y_1(t), \cdots, y_q(t), u_1(t), \cdots, u_r(t)\}$[3]. Notice that the likelihood function is equal to the conditional distribution function of D, conditional to some value $\theta = \bar{\theta}$. If two different values of the likelihood function is computed, such that $L(\theta_1 \mid D) > L(\theta_2 \mid D)$, then it can be concluded that the observation D was more likely to have occurred when $\theta = \theta_1$ [11]. Therefore, by evaluating the likelihood function in the parameter space, $\mathcal{S}$, where $\theta \in \mathcal{S} \subseteq \Re^m$, the value of θ that maximizes the likelihood function can be obtained. The MLE of the parameter vector is obtained as,

$$\theta^* \quad = \quad \arg\max_{\theta} L(\theta \mid D). \tag{2.3}$$

[2]MLE is also an acronym for maximum likelihood estimate when referred to as estimated result

[3]$t = t_0, \cdots, t_{N-1}$

MLE is easy to implement if the likelihood function is differentiable in $\mathcal{S}$, such that the optimum can be located by finding the values of θ such that $\frac{\partial}{\partial\theta}L(\theta \mid D) = 0$ and verifying that the optimum is global maximum. However, in nonlinear processes, the likelihood function is usually too complex, and to derive an analytical expression for the θ derivative is difficult. Hence, derivative free optimization methods are often required to implement MLE [6]. It is well-known that derivative free optimization methods are slow and require a very good initial guess of the parameters. These optimization methods lead to poor parameter estimates if the dimensionality of the parameter space is large or if a good initial guess is not provided. Another disadvantage of MLE is that the sensitivity of the parameters to the likelihood function dictates the accuracy of the estimates. For instance, if the change in likelihood function is negligible for relatively large changes in a particular parameter, then that parameter is difficult to estimate. Moreover, MLE is prone to a large bias in the estimate parameters when the size of the data set is small. However, as the data size approaches infinity, the MLE asymptotically approaches the true parameter values with no bias and minimum variance [39].

The likelihood function of dynamic processes is dependent on the assumptions made about the probabilistic distribution of noise (and hence the process measurements) in the process. For instance, if the probability distribution of observations $D = [y_1(t_0, \theta), \cdots, y_1(t_{N-1}, \theta)]$[4] is Gaussian, independent and identically distributed with some mean (μ) and variance (σ^2), then the likelihood function is defined as follows,

$$\begin{aligned} L(\theta \mid D) &= \prod_{i=1}^{N-1} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-(y_1(t_i,\theta)-\mu)^2}{2\sigma^2}\right) \\ &= \frac{1}{(2\pi)^{(N-1)/2}\sigma^{N-1}} \exp\left(\frac{-1}{2\sigma^2}\sum_{i=1}^{N-1}(y_1(t_i,\theta)-\mu)^2\right) \end{aligned} \tag{2.4}$$

[4]consider a single output variable case. t_0 to t_{N-1} are the sample times. Note that the parameter, θ, dependence of the observations is explicitly shown. In the rest of this thesis, the dependence of observations on parameters is not explicitly mentioned.

Then, maximizing the likelihood function reduces to (assuming the variance is known),

$$\max \log[L(\theta \mid D)] = \max \sum_{i=1}^{N-1} -\left(y_1(t_i, \theta) - \mu\right)^2 \tag{2.5}$$

$$= \min \sum_{i=1}^{N-1} \left(y_1(t_i, \theta) - \mu\right)^2 \tag{2.6}$$

which is precisely equal to the least-squares estimator. Thus, in the special case where the observations are Normally distributed and their variance is known, the MLE and LSE become equivalent.

2.3 Bayesian Parameter Estimation

Bayesian statistics is a departure from the more generally practiced frequency statistics, where the probability of a random event is defined as the percentage of its occurrence in a large number of trials. Therefore, if a frequentist were to derive the absolute probability of a given event, theoretically that person would require an infinite number of trials. However, in Bayesian statistics, a prior distribution of the event is updated with every new observation and the posterior distribution of the random event, conditional on the observations, is calculated. This is mathematically expressed as follows (also called Bayes Rule),

$$p(\theta \mid D) = \frac{L(\theta \mid D)\, p(\theta)}{\int p(D \mid \theta)\, p(\theta) d\theta} \tag{2.7}$$

where D is the set of observations; θ is the random event (process parameter in this work); $p(\theta)$ is the prior probability distribution; $L(\theta \mid D)$ is the previously mentioned likelihood function; and $p(\theta \mid D)$ is the posterior probability distribution. To illustrate the difference between frequency statistics and Bayesian statistics, consider the example of a tossing game where a fair coin and a trick coin are used as in section 2.1. Assume that a player randomly chooses one of the two coins, plays the game of hundred coin tosses, and observes 38 heads. A frequentist, given the

following information,

- the trick coin yields 3 H out of 10 tosses and there was a 50-50 chance of the player choosing fair coin or the trick coin (This information is *a priori*, since it was known before the observation is made);
- 38 H out of a hundred tosses (observation)

would not be able to present a precise numerical value for the probability that the chosen coin is the trick coin. This is due to the limited number of observations. This frequentist needs sufficient number of hundred-toss games in order to determine the type of coin used. A Bayesian statistician can use the same information and perform the following computations,

- The prior knowledge on the events of selecting a trick coin or a fair coin can be assigned a probability as $p(Tr) = 0.5, p(Fa) = 0.5$, where Tr and Fa correspond to the respective events.
- The probability of the chosen coin being the trick coin, given 38 H out of hundred toss, denoted by $p(Tr \mid k = 38)$, can be expressed as follows (using Bayes rule):

$$p(Tr \mid k = 38) \;=\; \frac{p(k = 38 \mid Tr)\, p(Tr)}{p(k = 38 \mid Tr)\, p(Tr) + p(k = 38 \mid Fa)\, p(Fa)}, \tag{2.8}$$

- $p(k = 38 \mid Tr) = \binom{100}{38} 0.3^{38} 0.7^{62}$,
- $p(k = 38 \mid Fa) = \binom{100}{38} 0.5^{38} 0.5^{62}$.

Substituting the prior information and the last two conditional distribution values in (2.8), the Bayesian statistician would conclude that there is a 81% chance that the chosen coin is the trick coin and that there is a 19% chance of it being the fair coin. The main advantage of Bayesian statistics, as illustrated in this example, is its ability to compute the probability of a given event with a limited number of observations, in this case a single game.

Two basic approaches exist for Bayesian parameter estimation. One is to find the parameter value that maximizes *a posteriori* (MAP) distribution as follows

$$\theta^* = \arg\max_{\theta} p(\theta \mid D) \tag{2.9}$$

and the other is to obtain the expected value of the *a posteriori* distribution as follows.

$$E[\theta] = \int p(\theta \mid D)\, p(\theta) d\theta \tag{2.10}$$

The MAP value indicates the most probable value of the parameter, however the expected value is a more appropriate representation of a parameter if its distribution is skewed [49]. The difference between the (2.3) and (2.9), though they both seek the probability maximizing value, is that MLE maximizes the likelihood of observing the experimental data, whereas MAP maximizes the probability of the parameter conditional on the observations.

Bayesian inference uses both *a priori* information and experimental data to compute full *a posteriori* distribution. This presents Bayesian inference with two major advantages over the traditional frequentist methods such as Nonlinear Least-Squares Regression or Maximum Likelihood Estimation. The first advantage is that by incorporating *a priori* information, we can make full use of all of the available information in the estimation process. This can become extremely helpful, especially when there is limited amount of experimental data available. Physical constraints arising from theory, and heuristic knowledge are examples of *a priori* information. By not using *a priori* information, frequentist methods usually yield inaccurate estimates, and this is demonstrated in Section 3.1. The second advantage of Bayesian inference is that by computing full probability distribution of the parameter, a statistically meaningful confidence intervals of the estimated parameters can be obtained. However, in previously developed Bayesian inference based approaches, this advantage had not been exploited to its full potential. While there have been many non-Bayesian studies that deal with multiple experimental data, Bayesian inference

was only used to compute *a posteriori* parameter distribution from each experimental data individually.

For point-estimation methods such as MLE, the probability distribution of the parameter is altogether disregarded or considered to be Gaussian by default. And, this practice does not translate into nonlinear processes, as it has been discussed in previous chapter that nonlinear process parameters probability distribution have more complex shape. Therefore, Bayesian parameter estimation method that calculates the full probability distribution of parameters is an advantageous tool for evaluating nonlinear processes. The main challenges of using Bayesian parameters arise from the fact that there is no analytical solution to the posterior distribution, because of the complex integral present in the denominator of the right hand side term in (2.7). This term serves as a normalizing constant that ensures the posterior distribution to integrate to unity [6]. In the absence of analytical solution, the posterior can be approximated numerically by random sampling method called Markov Chain Monte Carlo (MCMC). The details of this method are discussed in Chapter 4.

Chapter 3

Multiple Experimental Data Sets for Parameter Estimation

This chapter discusses the problem arising from pooling data sets from multiple experiments. The commonly employed method of straightforward data merging is shown to result in loss of information and has difficulty yielding statistically sound confidence intervals. An alternative approach that sequentially updates the *a priori* distribution function of the parameters based on multiple experimental data is developed in this chapter.

3.1 Merging Multiple Experimental Data Sets

Most parameter estimation methods require a large number of data points in order to obtain unbiased estimates with small confidence intervals. However, it is common in many experiments to have only a limited number of data points (N). Hence to reduce bias in the estimated parameters, it is common to conduct multiple experiments (k) and obtain Nk number of sample points. While such an approach may reduce the bias, the question of how the data sets from k different experiments, $D_1, \cdots, D_k$, can be systematically integrated is not clear addressed. Theoretically speaking, one can only merge data sets from multiple experimental runs if the experimental conditions and the noise characteristics are the same during different runs. However, it can be difficult to maintain the same experimental conditions through multiple runs. Therefore, creating a large merged data set from multiple experimental data sets can lead to poor parameter estimates. Another, more obvious problem is that there may be more than one value corresponding to some sampling time t_j. In Figure 3.1, $Y(t)$ measurements of FFL gene reg-

ulatory network process from three independent experiments, D_1, D_2 and D_3, are shown. The measurements are made at irregular time intervals and at $t = 30$ min, all three experiments have made measurement of $Y(t)$. If these three data sets were merged straightforwardly, then the time series of the merged data set $\boldsymbol{D}$, is denoted by the heavy black line shown in the figure, where the sample points of D_1, D_2 and D_3 are connected in sequential order of the corresponding measurement time. However, at $t = 30$ min, it is unclear as to which of the data points, $Y|_{D_1}(t = 30\,\text{min}), Y|_{D_2}(t = 30\,\text{min}), Y|_{D_3}(t = 30\,\text{min})$ (denoted by the three different paths of grey dotted lines), is the best representation of the true system.

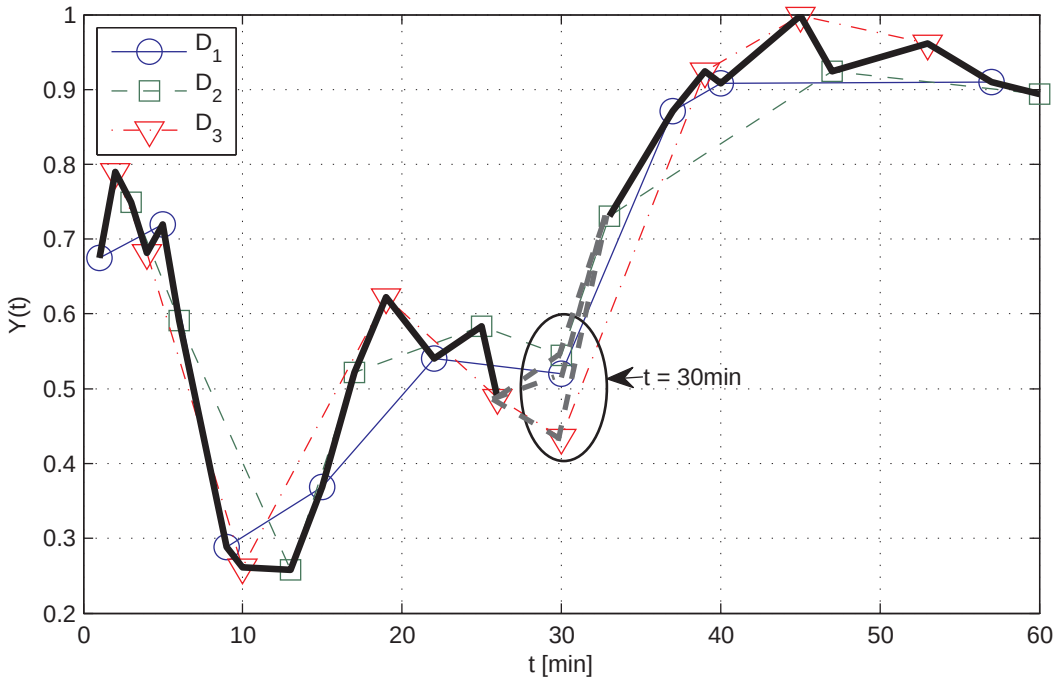

Figure 3.1: Time series data of $Y(t)$ obtained from three independent experiments D_1, D_2 and D_3, of Feed-Forward Loop genetic regulatory network process. The heavy black line denotes the $Y(t)$ trajectory using larger data set created from merging the three data sets. The line is broken at $t = 30$ min, because there are three different measurements corresponding to this time, and the three possible trajectories are denoted with grey dotted lines.

An alternative approach to analyzing multiple experimental data sets is to obtain a parameter estimate from each data set and compute the mean of the estimates, such that if the individual estimates are $\hat{\boldsymbol{\theta}}_1, \cdots, \hat{\boldsymbol{\theta}}_k$, and then the mean is $\hat{\boldsymbol{\theta}} = (\sum_{i=1}^{k} \hat{\boldsymbol{\theta}}_i)/k$. Consider the JAK-STAT signal

pathway process, where six simulated data sets are available[5]. From the data sets, using a type of point-estimation method introduced in [42], estimated values of a_1, a_3 and a_4 are obtained. These values are shown in Table 3.1. The mean and variance of the estimation results from six data sets is equal to $\bar{\boldsymbol{\theta}} = [0.0386, 2.393, 0.168]$ and $\sigma_{\hat{\theta}} = [0.0134, 2.412, 0.114]$. Due to the small number of data points from each experiment and the presence of noise, the estimates show large variances. The common practice of computing this average and the standard deviation, by default, assumes Gaussian distribution of the parameter estimates, and the normalized Gaussian distributions obtained from the average estimate and the standard deviations are shown in Figure 3.2. The true values used to simulate the data sets, $\boldsymbol{\theta} = [0.0515,\ 3.39,\ 0.35]$, are shown as vertical dotted lines. If Gaussian distribution of the estimated parameters is assumed, the probability of the parameters may show a significant positive value even in the obviously infeasible regions of the parameter space. For instance, the estimated distributions of a_3 and a_4 have the left-tail well into the negative region, which contradicts the constraints on the parameter that they must have positive values (discussed in Section 1.4.3). This example also demonstrates the case where even though average estimates satisfy the constraints, the corresponding confidence intervals does not.

Table 3.1: Point-estimation method is applied to the six simulated data of JAK-STAT signal pathway process, $D_1, \cdots, D_6$. The individual estimation result for the parameters a_1, a_3 and a_4 are shown.

	$\hat{a_1}$	$\hat{a_3}$	$\hat{a_4}$
D_1	0.0557	5.993	0.253
D_2	0.0385	1.652	0.247
D_3	0.0249	0.392	0.025
D_4	0.0488	4.803	0.273
D_5	0.0423	1.215	0.183
D_6	0.0212	0.303	0.026

The challenge, of obtaining an estimate that accurately represents all of the data sets, is easily

[5]Simulated data is obtained by the method shown in Appendix A

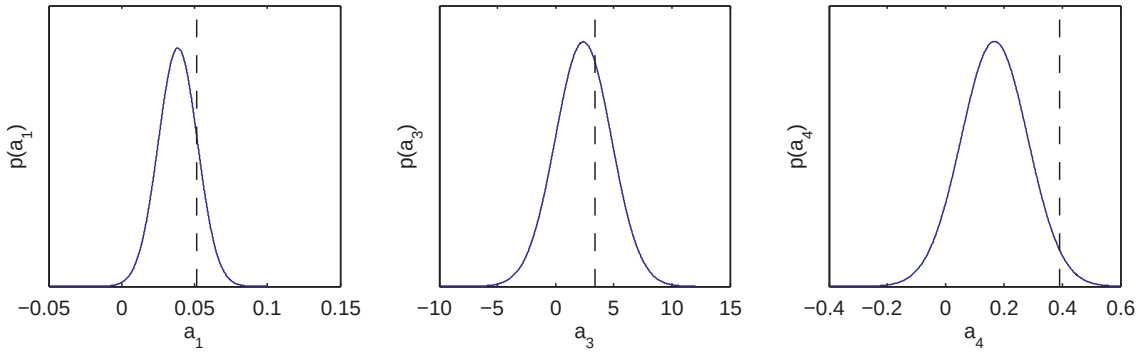

Figure 3.2: Applying point-estimation method, six individual parameter estimate are obtained from each simulated data, $D_1, \cdots, D_6$. The average and standard deviation of the results shown in Table 3.1 is as follows $\bar{\boldsymbol{\theta}} = [0.0386, 2.393, 0.168]$ and $\sigma_{\hat{\theta}} = [0.0134, 2.412, 0.114]$. This result is used to compute the Gaussian probability distributions that the general 'average $\pm$ 1.96$\times$standard deviation' confidence interval computation assumes. The vertical dotted lines denote the true values of the parameters used to simulate the data sets $\boldsymbol{\theta} = [0.0515,\ 3.39,\ 0.35]$.

handled by Bayesian parameter estimation methods as they can account for *a priori* information and estimate the full probability distribution. In the work by Coleman and Block [15], parameters of fermentation process model are estimated using informative prior probability distribution. The authors use several experimental data sets and obtain the full probability distributions of the parameters and proceed to report the data in the 'average $\pm$ 1.96 $\times$ standard deviation' (95% Confidence Interval) format. Such an approach fails to take advantage of estimating the full probability distribution as it does not carry information gained from one experiment to the next. To illustrate this point, the JAK-STAT process is considered again. Using the same sets of simulated data, the probability distribution functions, $\hat{p}(a_3 \,|\, a_1 = 0.0515, a_4 = 0.39, D_i)$ are estimated where $i = 1, \cdots, 6$.[6,7] The approximated probability distribution functions are shown in Figure 3.3 (a)-(f). The asymmetric distributions demonstrated in the figure is a common characteristic of probability distributions of nonlinear parameters. From these probability distribution functions, the expected values are computed as $\bar{a_3} = E(a_3) = \int a_3 \hat{p}(a_3)\, da_3$, which are $[0.0349, 0.0700, 0.0448, 0.0728, 0.0316, 0.0594]$. Then finally the average parameter estimation

[6]The approximated probability distribution shown in Figure 3.3 is obtained using the method introduced in Chapter 4

[7]The estimation of a_3 is conditional to the true value of a_1 and a_4 which are assumed to be known.

is reported with 95 % confidence interval as '$\bar{a_3} \pm 1.96\sigma_{a_3}$'[8] and the corresponding Gaussian probability distribution is shown in the center panel of Figure 3.3. When studying probability distribution, it should be noted that the shape and the variance of the distribution can be considered as 'information', in the sense that they are clues to the relative likelihood of a parameter taking up some value. From studying the panels (a)-(f), it is clear that some of the information contained in each probability distribution is eliminated from the average result because of the Gaussian assumption. For example, Panels (b) and (d) show probability distributions that are heavily negative-skewed, and this fact is not properly conveyed in the center panel.

It is mentioned in the previous section that nonlinear process parameters often exhibit asymmetric probability distribution, sometimes even multi-modal distributions. Therefore, when dealing with nonlinear processes, it is important to avoid Gaussian approximation of complex probability distributions. For instance, consider the two different types of distributions shown in Figure 3.4. The distribution shown in left is a bimodal distribution where $\arg\max_\alpha p(\alpha)$ are 3 and 6. The distribution shown in the right is a Gaussian distribution with $\arg\max_\beta p(\beta)$ is 4.50. These two distributions, though different in shape, have the same expected mean and standard deviation. The formulas used to calculate the expected mean and standard deviation are $E(x) = \left(\sum_{i=1}^{N} x_i\right)/(N)$ and standard deviation is $\sigma_x = \left[\left(\sum_{i=1}^{N}(x_i - \bar{x})^2\right)/(N-1)\right]^{1/2}$, where $x = \alpha, \beta$. It is easy to notice from these figures that if $p(\alpha)$ is approximated with a Gaussian distribution, the mid-point of the bimodal distribution, which has a very low probability, will be reported as the 'most likely' value.

[8]The average is $\bar{a_3} = \frac{\sum_{i=1}^{6} \hat{a_{3i}}}{6}$ and standard deviation is $\sigma_{a_3} = \left(\frac{1}{k-1}\sum_{i=1}^{k}(\hat{a_{3i}} - \bar{a_3})^2\right)^{1/2}$

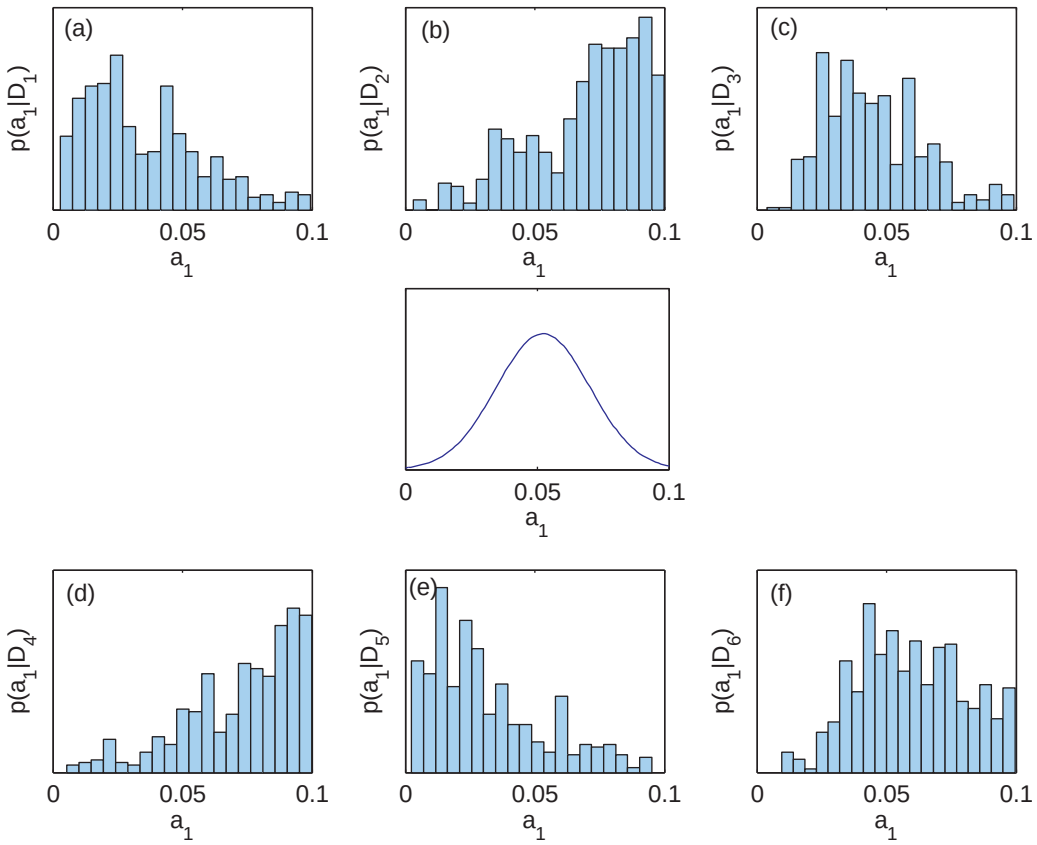

Figure 3.3: Six independently simulated data sets of JAK-STAT signal pathway process are used to estimate a_1 assuming that the two other parameter values are known. The individually estimated probability distributions are shown in Panels (a)-(f). The expected mean from each of the distribution is calculated and using these expected means, the overall average of the $\hat{a_1}$ is computed along with its standard deviation. Using these values, a Gaussian distribution is plotted in the center panel.

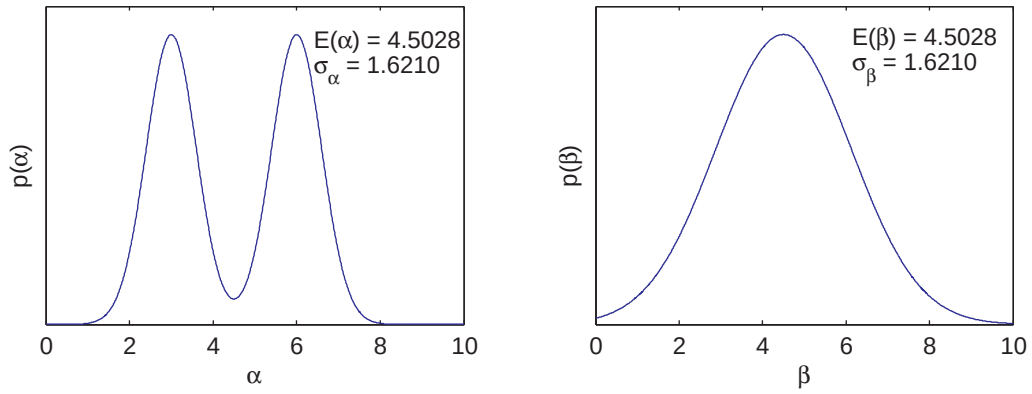

Figure 3.4: Two probability distributions with identical expected mean, $E(\alpha) = E(\beta) = 4.5028$, and standard deviation $\sigma_\alpha = \sigma_\beta = 1.6210$. However, for the probability distribution on the left, the *max a posteriori* is $\arg\max_\alpha p(\alpha) = 3 = 6$, where as the probability distribution on the right, the *max a posteriori* is $\arg\max_\beta = p(\beta) = 4.5028$.

3.2 *A priori* and Timeline Shift

In this thesis, a parameter estimation approach that makes use of the two major advantages of Bayesian inference, discussed in the previous chapter, is developed. The approach aims to derive a single *posterior* probability distribution, conditional on all of the available experimental data by iteratively updating the *prior* probability distribution in the Bayes Rule. By doing so, the information available from all of the databases is propagated through the estimated probability distribution functions.

Suppose that the FFL model parameter vector, θ, needs to be estimated and that k independent experimental data sets are available, $D_1, \cdots, D_k$. First, the *a priori* probability distribution function is defined from heuristic information and other constraining conditions obtained from theoretical considerations (3.5 (A)). This function is denoted with $p_0(\theta)$. In order to obtain *a posteriori* probability distribution function using Bayes Rule, the likelihood function, $L(D \mid \theta)$, is also required. The procedure for deriving $L(D \mid \theta)$ is discussed in Appendix B. Using these two components and the first experimental data set D_1, the expression for the first posterior

distribution function is computed as follows.

$$p_1(\theta \mid D_1) = \frac{L(\theta \mid D_1)\, p_0(\theta)}{\int p(D_1 \mid \theta)\, p(\theta)\, d\theta} \tag{3.1}$$

This newly calculated posterior probability distribution is conditional on D_1. This step of obtaining the first *posterior* is denoted with 'E1'. Following E1, present time is shifted forward and the information regarding the parameters obtained using D_1 is now considered past knowledge. The *posterior* information from E1 becomes *a priori* information for all the future experiments (Figure 3.5 (B)). Thus, the prior distribution function needs to be updated to account for this fact. Therefore, the next prior probability distribution function is set equal to the posterior distribution obtained in E1 as follows.

$$p_1(\theta) := p_1(\theta \mid D_1) \tag{3.2}$$

Using this new prior distribution and the likelihood function, the second posterior probability distribution is calculated as follows.

$$p_2(\theta \mid D_1,\, D_2) = \frac{L(\theta \mid D_2)\, p_1(\theta)}{\int p(D_2 \mid \theta)\, p(\theta)\, d\theta} \tag{3.3}$$

$$= \frac{L(\theta \mid D_2)\, p_1(\theta \mid D_1)}{\int p(D_2 \mid \theta)\, p(\theta)\, d\theta} \tag{3.4}$$

This new posterior probability distribution, $p_2(\theta \mid D_1, D_2)$ is conditional on both D_1 and D_2. The process of obtaining the second posterior is denoted with 'E2'. Following E2, the prior distribution needs to be updated again as the present time has shifted forward and as it now needs to include the information about the parameter obtained from analyzing D_2. The new prior is then set equal to

$$p_2(\theta) := p_2(\theta \mid D_1, D_2) \tag{3.5}$$

This sequential update of the prior distribution with the newly calculated posterior distribution

is repeated until all of the experimental data sets have been evaluated. Then, the final posterior distribution that is conditional on all of the information is equal to,

$$p_k(\theta \mid D_1, D_2, \cdots, D_k) \quad = \quad \frac{L(\theta \mid D_k)\, p_{k-1}(\theta \mid D_1, \cdots, D_{k-1})}{\int p(D_k \mid \theta)\, p(\theta)\, d\theta} \tag{3.6}$$

The process of evaluating the kth and last *posterior* is denoted with 'E_k' (Figure 3.5 (C)). This iterative approach offers a systematic procedure to integrating information from multiple experimental runs by computing a series of probability distribution functions. Figure 3.6 shows the typical behavior expected from a series of evolving posterior distributions obtained using the sequential Bayesian estimation method. The top left panel is shown with a flat uniform prior distribution of the parameter vector restricted to some lower and upper bounds. In some cases, the prior distribution is more informative and in such cases the *a priori* distribution will take on shape than the uniform distribution shown. The top right panel shows the posterior probability distribution obtained after applying the first experimental data (after E1). The shape of the distribution starts to form a plateau near the region of higher probability - in this case, near the center - conveyed by D_1. The posterior distribution function shown in the bottom left panel now shows more distinct peak near the value 0, from which it can be inferred that at $\theta \approx 0$, the probability distribution function is maximized (after E2). Finally, after all of $D_1, \cdots, D_k$ are applied to the sequential method, the resulting final posterior distribution of process parameter is a result of systematic integration of multiple experimental data sets.

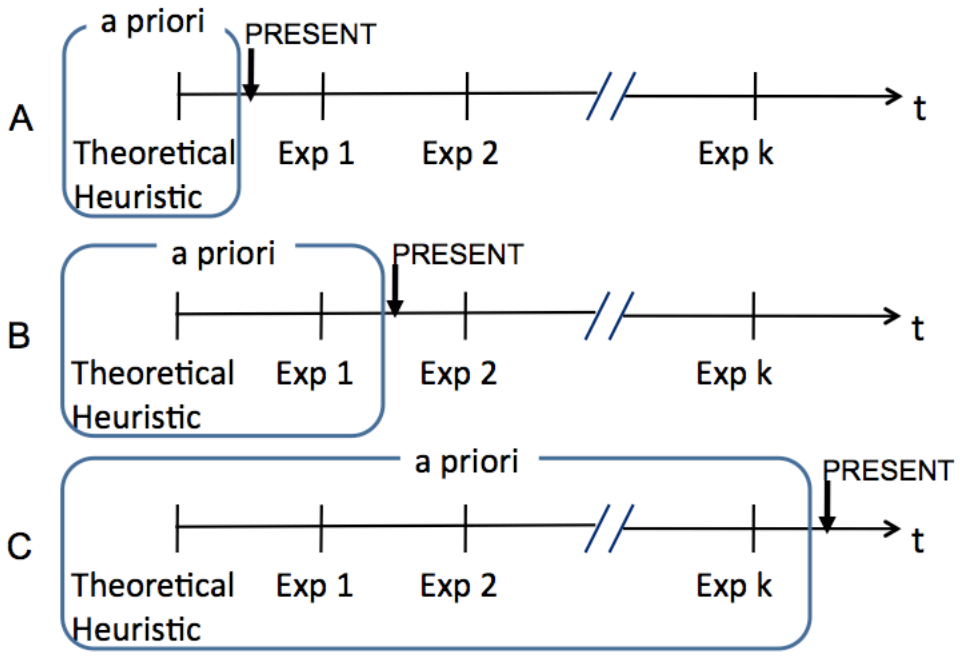

Figure 3.5: Three different snapshots of a linear timeline with multiple experiments and the information database; the present time is indicated by the vertical arrow in each panel. The top panel (A) shows the present time before any of the experiments have been conducted/analyzed. At this time, *a priori* consists of theoretical and heuristic information. The middle panel (B) shows the present time, after Experiment 1 has been conducted/analyzed and the experimental data has been evaluated for parameter estimation. At this time, *a priori* consists of theoretical and heuristic information as well as the information gathered from the first experiment. The bottom panel (C) portrays the present time, after k experiments have been conducted and the multiple experimental data sets have been evaluated for parameter estimation. At this time *a priori* consists of theoretical and heuristic information along with the information gathered from all of the experiments.

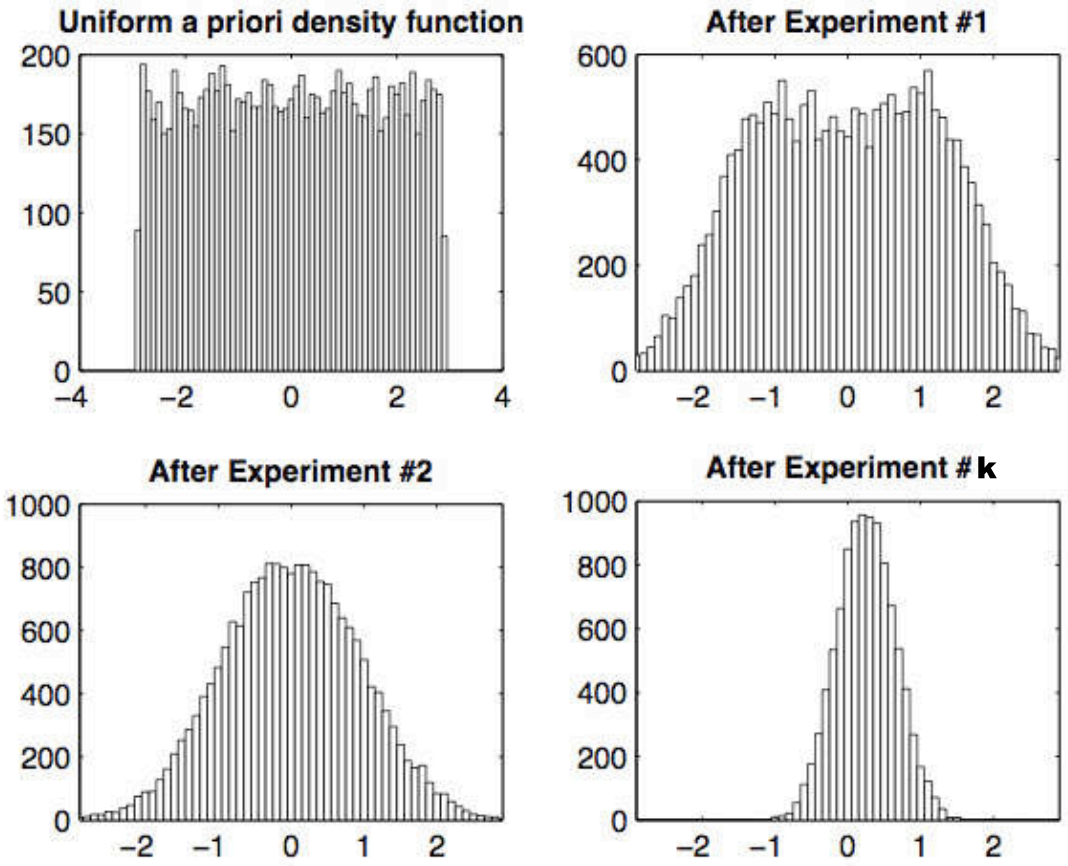

Figure 3.6: The four panels portray the evolving series of posterior distribution functions obtained from multiple experimental data sets applied to sequential Bayesian estimation method. The top left panel shows a uniform prior distribution before any experimental data has been analyzed. In some cases, the prior distribution may be more informative where it highlights the more probable region in the parameter space. The top right panel shows the posterior distribution computed from the first set of experimental data. The bottom left panel shows the posterior distribution of parameter after the second set of experimental data has been analyzed. The bottom right panel shows the posterior distribution of parameter obtained after all k experimental data sets have been applied to the algorithm.

Chapter 4

Markov Chain Monte Carlo (MCMC) for Approximating Probability Distribution Functions

Most posterior distributions computed using Bayes' Rule cannot be solved analytically, therefore a numerical approximation method called Markov Chain Monte Carlo is used to generate random samples from the distribution. These random samples are used to plot a histogram that is an approximation of the desired probability distribution. Two instances of MCMC, Metropolis-Hastings algorithm and Gibbs Sampler, are used in this work in a set of inner and outer level of iterations to approximate the probability distribution of the parameters. The large computational cost of these approximations is optimized using a novel multi-phase approach of Gibbs sampler.

4.1 Markov Chain Monte Carlo (MCMC)

Markov Chain Monte Carlo method was developed in order to solve optimization and integration problems arising in high dimensional spaces where analytical solutions are difficult to obtain. The approach relies on drawing a number of independent and identically distributed samples, $\{\theta^{(i)}\}_{i=1}^{M}$, from a target distribution (the distribution that needs to be approximated), $p(\theta \mid D)$. Using these M samples, the distribution is approximated by the following empirical point-mass function [2].

$$\hat{p}_M(\theta \mid D) \quad = \quad \frac{1}{M} \sum_{i=1}^{M} \delta_{\theta}^{(i)}(\theta), \tag{4.1}$$

where $\delta_\theta(\theta^{(i)})$ is the delta-Dirac function at $\theta^{(i)}$. From this approximation, the maximum *a posteriori*, expected mean and High Probability Distribution (HPD) interval can be calculated. For example, the expected mean is equal to

$$E(\theta \mid \mathbf{D}) \approx \frac{1}{M} \sum_{i=1}^{M} \theta^{(i)}. \tag{4.2}$$

The samples $\{\theta^{(1)}, \cdots, \theta^{(M)}\}$ are drawn in such a way that the following relationship is satisfied,

$$p(\theta^{(M)} \mid \theta^{(M-1)}, \cdots, \theta^{(1)}) \quad = \quad p(\theta^{(M)} \mid \theta^{(M-1)}). \tag{4.3}$$

A random variable that satisfies the above relation is said to have the Markov property. Therefore, the sequence of draws of the parameter vector forms a Markov chain and this procedure of approximating density functions is called Markov Chain Monte Carlo approach.

There are many variations of MCMC algorithm, and the general idea behind them is to construct the Markov chain in such a way that it draws more samples in regions of high probability of the target distribution, and as M approaches ∞, the approximation of the distribution will asymptotically be equal to the target distribution. In the following sections, two instances of MCMC, Metropolis-Hastings Algorithm and Gibbs Sampler, used in this work to approximate the posterior probability distribution of nonlinear process parameters, are introduced.

4.2 Metropolis-Hastings Algorithm: Inner Level Estimation

The idea behind MCMC is to approximate the target distribution by generating random samples from a proposal distribution, which is chosen such that it has the same support as the target distribution. It is usually difficult to sample directly from the target distribution and hence a simple proposal distribution from which samples can be easily drawn is chosen. The samples from the proposal distribution are either accepted or rejected based on a criterion that allows acceptance

of samples from the target distribution with a high probability.

Metropolis-Hastings algorithm (M-H) is a type of MCMC that was first developed by Metropolis in 1953 [38], then generalized by Hastings in 1970 [26]. It employs the acceptance-rejection approach to generate random samples from the desired target distribution and determine whether to accept each sample by computing an acceptance criterion [17]. This criterion, also known as the acceptance probability (α), is computed through the use of a proposal probability distribution function, $q(\theta^{(i)} \mid \theta^{(i-1)})$. This expression is interpreted as 'if the current value of the chain is $\theta^{(i-1)}$, then the newly generated random sample is $\theta^{(i)}$'. There exist several different candidates for the proposal distribution. In this work, an approach called 'independence sampling', where the newly generated sample value does not depend on the current value of the chain, is used. This proposal distribution can be expressed as $q(\theta^{(i)})$ (notice that the conditional term is eliminated) [14]. The quality of the target approximation using M-H is strongly influenced by the choice of the proposal distribution [2].

The following shows the step-by-step guide for implementing M-H algorithm, to approximate the expression for posterior distribution derived in the previous section, $p(D \mid \theta)$.

1. Choose an initial sample, $\theta^{(0)}$, such that $p(\theta^{(0)} \mid D) > 0$.
2. Repeat the following steps for $\theta^{(i)}$, where $i = 1, \cdots, M$.
3. From the proposal distribution of choice, $q(\theta)$, generate a random sample $\theta^{(i)}$.
4. Calculate the acceptance probability, α such that

$$\begin{aligned} \alpha &= \min\left[1, \frac{p(\theta^{(i)} \mid D_1)}{q(\theta^{(i)})} \frac{q(\theta^{(i-1)})}{p(\theta^{(i-1)} \mid D_1)}\right] \\ &= \min\left[1, \frac{L(\theta^{(i)} \mid D)\, p_0(\theta^{(i)}) \,/\, \mathcal{Z}}{q(\theta^{(i)})} \frac{q(\theta^{(i-1)})}{L(\theta^{(i-1)} \mid D)\, p_0(\theta^{(i-1)}) \,/\, \mathcal{Z}}\right] \end{aligned}$$

where $\mathcal{Z} = \int p(D \mid \theta)p(\theta)d\theta$.

5. If $\alpha = 1$, then accept the $\theta^{(i)}$ value, update the count to $i = i+1$ and repeat from step 2. If $\alpha \neq 1$, then sample $\beta \sim \mathcal{U}(0,1)$ where $\mathcal{U}(0,1)$ is a uniform distribution between 0 and 1.

6. If $\beta < \alpha$, accept the $\theta^{(i)}$ value, update the count to $i = i + 1$ and repeat from step 2. If $\beta \geq \alpha$, discard $\theta^{(i)}$ and repeat from step 2.

Notice that when computing the acceptance probability, the normalizing constant involving complex integral term gets canceled, which is the major advantage of using M-H algorithm. As mentioned earlier, the choice of proposal distribution is very important for the success of the algorithm. Two different choices of *independence sampling* are examined in this thesis. The first one is a Gaussian distribution multiplied with some constant Q and the second one is the prior distribution corresponding to the Bayes Rule at each of the steps $E1, \cdots, Ek$. This discussion is presented in sections 4.2.1 and 4.2.2.

The mechanism behind M-H's acceptance probability is explained with a brief example shown in Figure 4.1. The target distribution that needs to be approximated is denoted with red curve and it is shown to be bimodal, asymmetric distribution. The proposal distribution is denoted with green curve in the figure. Let's assume that the initial value of the chain is equal to 3 ($\theta^{(0)} = 3$), and using the proposal distribution, a new random sample $\theta = 4.5$ is generated. Then the acceptance probability for this new sample is calculated to be 0.39, using the equation introduced previously. Since α is smaller than 1, another random variable β is sampled from a uniform distribution between 0 and 1. The reason behind sampling β in steps 5 and 6 can be explained as follows. Generally, a sample moving from a higher probability region to a lower probability region is undesired, but is not always avoided. Therefore, the larger the 'jump down' from higher to lower probability region, the smaller is the chance of accepting that sample. The current example, for instance, has 39% chance of being accepted and 61% chance of being discarded. Let's assume that β is equal to 0.78, then the sample 4.5 is discarded, and the current

value of θ is accepted as the next sample ($\theta^{(1)} = 3$), making up the following Gibbs sequence of length 2, $\{3, 3\}$. Another value is sampled from the proposal distribution, and this time around assume that it is 6. In this case, the acceptance probability is 1. This is because the random sample 'jumped up' in the distribution to a higher region, and the new sample is accepted. Now the Gibbs sequence has length 3, $\{3, 3, 6\}$. The current sample now is $\theta^{(2)} = 6$ and the same procedure can be repeated until a desired length is reached.

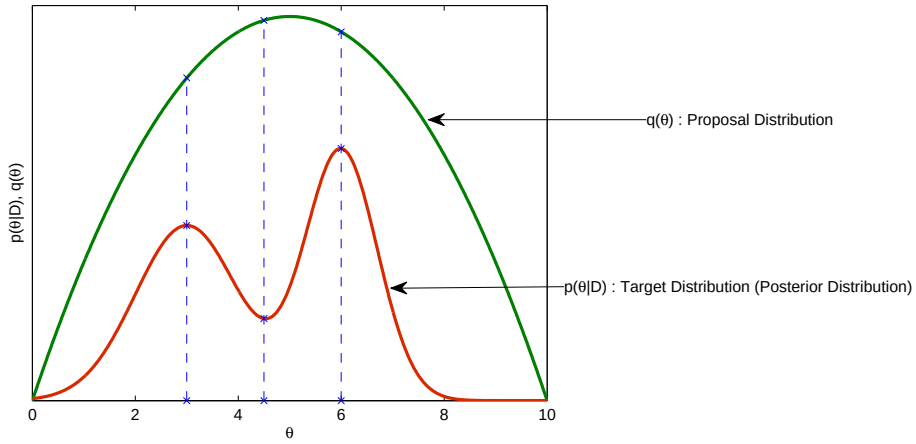

Figure 4.1: Metropolis-Hastings algorithm. The asymmetric red curve represents the target distribution that needs to be approximated (posterior distribution in this work). The green curve represents the proposal distribution chosen by the practitioner. The three points are the random samples of $\theta = 3, 4.5, 6$ with $p(\theta)/q(\theta)$ is equal to $0.5427, 0.2157, 0.6785$, respectively.

Another instance of MCMC that is recently popularized is the Simulated Annealing (SA) algorithm. The SA algorithm was developed by borrowing the concept used in metallurgy where the quality of the material is controlled by implementing a cooling schedule that adapts as time evolves. The users of SA agree that if the ultimate goal of approximating the probability distribution is to obtain the MAP, it is wasteful in terms of computational resources to sample from the lower probability regions. This is because high probability regions of a distribution are concentrated around its mode, and hence the computational resources can be concentrated on the high probability regions. The algorithm is explained in more detail in [23], [30], [41], [52]. Though this method is computationally more efficient, the approximated probability distribution is not

the true representation of the target distribution, but an altered form of the distribution of which the sole purpose is to obtain the MAP value, and this is illustrated in Figure 4.2. Both of the histograms shown in the figure are plotted from Markov Chain having length $M = 5000$, but SA generated histogram is clearly shown to favor the higher probability region and thus is very efficient for obtaining MAP. Due to this reason, SA is not suitable for application in the work developed in this thesis. In this work, the prior distribution is updated sequentially in order to systematically merge experimental data sets and hence a good approximation of the complete distribution, rather an approximation for MAP, is needed. The parameter estimation method in this work places high priority on as closely estimating the probability distributions under consideration as possible without the assumption of Gaussian distribution. This will allow proper representation of the information extracted from the experimental data. Therefore M-H algorithm was chosen for implementation even though there exists computationally more efficient choices of MCMC.

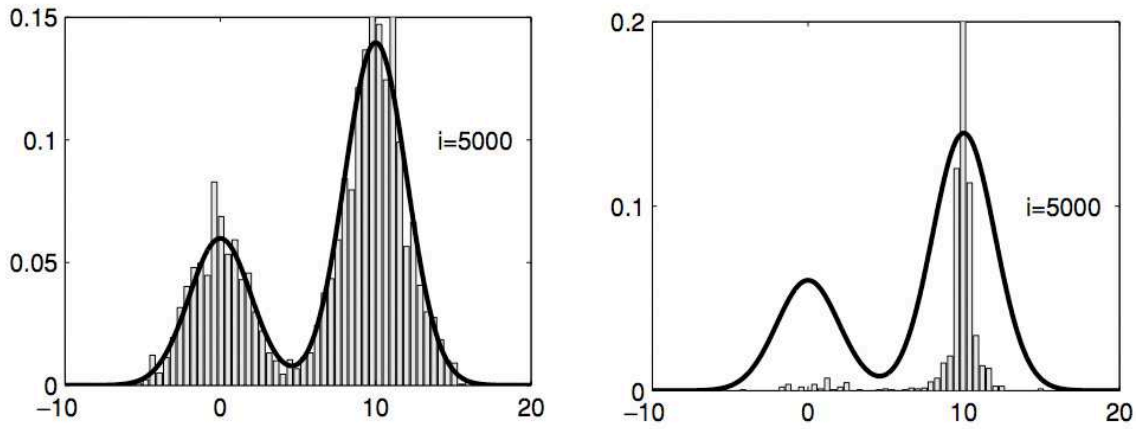

Figure 4.2: Metropolis-Hastings Algorithm vs. Simulated Annealing. The true target distribution form is denoted with a solid black curve and the histograms generated from the Markov Chain using the two MCMC methods are shown as well. The panel on the left show the Metropolis-Hastings Algorithm generated Markov Chain's histogram and the panel on the right show the Simulated-Annealing generated Markov Chain's histogram [2].

4.2.1 Proposal distribution I : Gaussian distribution

The proposal distribution in MH needs to cover the entire target distribution's probability space, such that $q(\theta) \geq p(\theta \mid D)$ for all θ. When using Gaussian distribution, this specification can be easily met by choosing the standard deviation to be wide enough and by multiplying the distribution with a constant Q. Using the Gaussian proposal distribution, consider approximating the first posterior distribution of α_y of FFL model, $p(\alpha_y \mid D_1)$, where the prior distribution is $\mathcal{U}(0,1)$ such that,

$$p_0(\theta) = \begin{cases} 1 & \text{if } 0 < \alpha_y \leq 1, \\ 0 & \text{otherwise} \end{cases} \tag{4.4}$$

In the first step, computing α is straight forward because the prior distribution term of the current sample and the future sample are equal to each other and are easily computed. However, for step E2, where the prior distribution is set equal to the approximation of the first posterior, $\hat{p}_1(D \mid \alpha_y)$, calculation of the acceptance probability is a bit more complicated. Calculation of acceptance probability requires the information regarding the histogram of the Markov chain, in particular the bin-index and the number of elements within the bin. A MATLAB® program that approximates the value of prior distribution function is shown below. A couple of approximations take

Program 4.1 MATLAB program that approximates the value of prior distribution $p_\kappa(\theta)$.

```
% approximating the posterior-prior distribution
[n,x] = hist(theta_v(:,1),20);
% generating new sample from proposal distribution
theta_new = Q*(0.5 + randn*sigma);
while x(i) >=theta_new
       % determining the index number where the new
       % sample belongs
       i = i+1;
end
prior(theta_new) = n(i);
```

place during the implementation of the M-H algorithm. Theoretically, the approximate distribution from M-H algorithm will converge to the target distribution only if the number of samples is infinite. However, in practice, only a finite number of samples are used. The second approxi-

mation is the one shown in the program above where the value of the probability distribution is numerically approximated using the `hist` function in MATLAB®.

4.2.2 Proposal distribution II : *a priori* distribution

The proposal distribution for M-H can be chosen to be the prior distribution. In other words, if the following posterior distribution needs to be approximated,

$$p_\kappa(\theta \mid D_\kappa) \;=\; \frac{L(\theta \mid D_\kappa)\, p_{\kappa-1}(\theta)}{\int p(D_\kappa \mid \theta)\, p(\theta)\, d\theta}, \qquad \kappa = 1, 2, \cdots, k \tag{4.5}$$

the proposal distribution is,

$$q(\theta) := p_{\kappa-1}(\theta) \tag{4.6}$$

By the above definition, the acceptance probability of random samples for approximation step E_κ is calculated as follows.

$$\begin{aligned}
\alpha &= \min\left[1, \frac{p_\kappa(\theta^{(i)} \mid D_\kappa)}{q(\theta^{(i)})}\, \frac{q(\theta^{(i-1)})}{p_\kappa(\theta^{(i-1)} \mid D_\kappa)}\right] \\
&= \min\left[1, \frac{L(\theta^{(i)} \mid D_\kappa)\, p_{\kappa-1}(\theta^{(i)})\, / \, \mathcal{Z}}{p_{\kappa-1}(\theta^{(i)})}\, \frac{p_{\kappa-1}(\theta^{(i-1)})}{L(\theta^{(i-1)} \mid D_\kappa)\, p_{\kappa-1}(\theta^{(i-1)})\, / \, \mathcal{Z}}\right] \\
&= \min\left[1, \frac{L(\theta^{(i)} \mid D_\kappa)}{L(\theta^{(i-1)} \mid D_\kappa)}\right]
\end{aligned} \tag{4.7}$$

where $\mathcal{Z} = \int p(D_\kappa \mid \theta)\, p(\theta)\, d\theta$. Notice that by using (4.6), the acceptance probability is only dependent on the likelihood values as a function of the current value of the proposed parameter value. The proposal distribution is updated along with the prior distribution after each step of evaluating the posterior, E1, $\cdots$, Ek.

4.3 Gibbs Sampler : Outer Level Estimation

In this part of the work, to approximate the multi-dimensional joint probability distribution of parameters of a nonlinear process, a variant of MCMC called Gibbs sampler is used. This particular MCMC approximation method is a special case of Metropolis-Hastings algorithm where the proposal distribution, $q(\theta)$, is defined specifically as the set of distributions of individual parameters conditional on the values of the rest of the parameters. Using the Gibbs sampler, the marginal distributions of individual parameters are calculated from which the maximum *a posteriori*, expected value and standard deviation of the distributions are calculated [40]. To illustrate this algorithm, consider the FFL genetic regulatory network model with the parameter vector, $\theta = [\alpha_y, \alpha_z, K_{xy}, K_{xz}, K_{yz}] \in \Re^5$. For example, the marginal probability distribution of α_y is computed by a series of integrations of the joint probability distribution as follows.

$$p(\alpha_y) = \int_{K_{yz}} \int_{K_{xz}} \int_{K_{xy}} \int_{\alpha_z} p(\alpha_y, \alpha_z, K_{xy}, K_{xz}, K_{yz})\, d\alpha_z\, dK_{xy}\, dK_{xz}\, dK_{yz} \tag{4.8}$$

The equation contains integrals that are often difficult to solve analytically. However, a Gibbs sequence with Markov property, $\{\alpha_y^{(0)}, \alpha_y^{(1)}, \cdots, \alpha_y^{(M)}\}$ (where M is the length of the Gibbs sequence) can be generated so that the marginal distribution and its statistical properties can be approximated as in (4.2).

In order to implement the Gibbs sampler, the following set of conditional probability distribution functions are needed,

$$\begin{aligned}
&p(\alpha_y \,|\, \alpha_z, K_{xy}, K_{xz}, K_{yz}),\\
&p(\alpha_z \,|\, \alpha_y, K_{xy}, K_{xz}, K_{yz}),\\
&p(K_{xy} \,|\, \alpha_y, \alpha_z, K_{xz}, K_{yz}),\\
&p(K_{xz} \,|\, \alpha_y, \alpha_z, K_{xy}, K_{yz}),\\
&p(K_{yz} \,|\, \alpha_y, \alpha_z, K_{xy}, K_{xz}).
\end{aligned} \tag{4.9}$$

With many dynamic process models, there is no general solution for calculating the conditional distributions of process parameters. Previous work on using Gibbs sampler assumed Gaussian distribution, with some heuristic values for its mean and standard deviation, as the conditional distribution of individual parameters [8, 21]. This assumption may apply to linear processes, but becomes questionable when dealing with nonlinear processes. However, these conditional distributions can be numerically approximated using Metropolis-Hastings algorithm, which forms the *inner* level of iterations of the algorithm proposed in the next section, and the Gibbs sampler forms the *outer* level of iterations. The following illustrate the steps of Gibbs sampler using FFL example.

1. Assign initial values $\alpha_z^{(0)}, K_{xy}^{(0)}, K_{xz}^{(0)}, K_{yz}^{(0)}$ for $\alpha_z, K_{xy}, K_{xz}, K_{yz}$.

2. Repeat the following steps for $i = 0, \cdots, M-1$

3. Approximate the conditional distribution of α_y by applying the experimental data sets to sequential M-H algorithm. Generate a value $\alpha_y^{(i+1)}$ by randomly sampling from the conditional distribution

$$\alpha_y^{(i+1)} \quad \sim \quad \hat{p}(\alpha_y \mid \alpha_z^{(i)}, K_{xy}^{(i)}, K_{xz}^{(i)}, K_{yz}^{(i)}, D_1, \cdots, D_k). \tag{4.10}$$

4. Approximate the conditional distribution of α_z by applying the experimental data sets to sequential M-H algorithm. Generate a value $\alpha_z^{(i+1)}$ by randomly sampling from the conditional distribution

$$\alpha_z^{(i+1)} \sim p(\alpha_z \mid \alpha_y^{(i+1)}, K_{xy}^{(i)}, K_{xz}^{(i)}, K_{yz}^{(i)}, D_1, \cdots, D_k). \tag{4.11}$$

5. Approximate the conditional distribution of K_{xy} by applying the experimental data sets to sequential M-H algorithm. Generate a value $K_{xy}^{(i+1)}$ by randomly sampling from the

conditional distribution

$$K_{xy}^{(i+1)} \sim p(K_{xy} \mid \alpha_y^{(i+1)}, \alpha_z^{(i+1)}, K_{xz}^{(i)}, K_{yz}^{(i)}, D_1, \cdots, D_k). \tag{4.12}$$

6. Approximate the conditional distribution of K_{xz} by applying the experimental data sets to sequential M-H algorithm. Generate a value $K_{xz}^{(i+1)}$ by randomly sampling from the conditional distribution

$$K_{xz}^{(i+1)} \sim p(K_{xz} \mid \alpha_y^{(i+1)}, \alpha_z^{(i+1)}, K_{xy}^{(i+1)}, K_{yz}^{(i)}, D_1, \cdots, D_k). \tag{4.13}$$

7. Approximate the conditional distribution of K_{yz} by applying the experimental data sets to sequential M-H algorithm. Generate a value $K_{yz}^{(i+1)}$ by randomly sampling from the conditional distribution

$$K_{yz}^{(i+1)} \sim p(K_{yz} \mid \alpha_y^{(i+1)}, \alpha_z^{(i+1)}, K_{xy}^{(i+1)}, K_{xz}^{(i+1)}, D_1, \cdots, D_k). \tag{4.14}$$

8. Update the count $i = i + 1$.

A single Gibbs *step* consists of updating the values of the random variables once (e.g. $\{x^{(0)}, y^{(0)}, z^{(0)}\} \rightarrow \{x^{(1)}, y^{(1)}, z^{(1)}\}$ is one step). Within a single Gibbs *step*, there are m number of Metropolis-Hastings approximations of posterior distribution, where m is the length of the process parameter vector, θ. Within a single M-H approximation of posterior distribution, k evolving posterior distributions are computed. Therefore, there is a large number of different instances of conditional distribution functions being calculated while using the Metropolis-Hastings & Gibbs sequence approach.

The experimental data sets are used more than once in these iterations. In each Metropolis-Hastings approximation step, the conditional distribution of the parameter values changes, and therefore the subsequent conditional distributions must account for this. For example, we can

compare three conditional distributions that are approximated as follows,

1. $f_1 = f(\alpha_y) = p(\alpha_y|\alpha_z = \phi_2, K_{xy} = \phi_3, K_{xz} = \phi_4, K_{yz} = \phi_5, D_1, \cdots, D_k)$

2. $f_2 = f(\alpha_z) = p(\alpha_z|\alpha_y = \varphi_2, K_{xy} = \varphi_3, K_{xz} = \varphi_4, K_{yz} = \varphi_5, D_1, \cdots, D_k)$

3. $f_3 = f(K_{xy}) = p(K_{xy}|\alpha_y = \varphi_1, \alpha_z = \varphi_2, K_{xz} = \varphi_4, K_{yz} = \varphi_5, D_1, \cdots, D_k)$

where $\phi \neq \varphi$. f_1 and f_2 are both functions of α_z, but computed using different sets of conditions. f_2 and f_3 have somewhat similar conditions, however they are functions of different parameters. Since the computational conditions are different in these functions, the experimental data set, $D_1, \cdots, D_k$, can be used repeatedly without over-exerting its 'information database'.

As mentioned previously, as M, the number of random samples generated, approaches ∞, the sequence can accurately approximate the desired target distribution. There are several ways to determine finite M where a sufficient accuracy of the approximation is reached. Similarly, there are several ways of detecting convergence of the approximation agreement to a steady-state distribution. Works by Gelfand *et al.*[21] and Gelfand and Smith [22] suggest monitoring distribution approximations from multiple independent Gibbs sequences, and choosing M to be the point where the distributions formed by the multiple chains with different initial conditions appear the same. An alternative method of choosing M is to implement the Raftery-Lewis criterion, also known as binary-control, which determines the value of M corresponding to the desired accuracy and avoids excessive sampling. The approach uses a two-state Markov chain model by analyzing a single run of Markov chain of output values. From the two-state Markov chain model, the length of the burn-in period is computed and the number of iterations required to meet the specified accuracy can be computed [44].

4.3.1 Multi-phase Gibbs sampler

When using Gibbs sampler, the rate of convergence of different parameters can vary significantly. Some parameters will approach their steady-state distribution very fast, and hence increasing the number of Gibbs iterations would result in negligible change in the distribution of the corresponding parameters. Since each Gibbs iteration is divided into multiple Metropolis-Hastings steps that require a considerable amount of computational cost, the parameters that reached their steady-state distribution can be removed from the individual Metropolis-Hastings approximations. This is achieved by fixing the distributions of converged parameters and only updating the distributions of the remaining parameters. For instance, if the distributions of parameters α_y and α_z reach their respective steady-states before the other three parameters after M Gibbs iterations were executed, then the approximated marginal probability distribution of $p_{(I)}(\alpha_y)$ and $p_{(I)}\alpha_z$ (using the M samples, (I) denotes *Phase I*) will have negligible difference between the marginal probability distribution obtained from executing G more Gibbs iteration, $p_{(II)}(\alpha_y)$ and $p_{(II)}(\alpha_z)$ (using $M+G$ samples, (II) denotes *Phase II*). If the other three parameters K_{xy}, K_{xz} and K_{yz} did not reach steady-state by the Mth Gibbs iteration, following additional steps to the original algorithm discussed in the previous section are required as follows.

1. Repeat the following steps for $i = M+1, \cdots, M+G$
2. Generate a value $\alpha_y^{(i+1)}$ by randomly sampling from the conditional distribution

$$\alpha_y^{(i+1)} \quad \sim \quad p_{(I)}(\alpha_y) \tag{4.15}$$

3. Generate a value $\alpha_z^{(i+1)}$ by randomly sampling from the conditional distribution

$$\alpha_z^{(i+1)} \sim p_{(I)}(\alpha_z) \tag{4.16}$$

4. Follow the Steps (4.12) to (4.14)
5. Update the count $i = i + 1$.

Using this semi-fixed distribution generated from previous phase is essentially the same as generating a new MH conditional distribution because those distributions, having reached their respective steady states, will not alter much with progressive random sampling. Each parameter eliminated from M-H step for computing their conditional distributions will result in $1/m \times 100\%$ reduction in computational cost after the Phase I. This approach can be adopted in several phases to further optimize the computational resources.

4.4 Sequential Metropolis-Hastings and Gibbs Algorithm

The previous two sections discussed the MCMC methods, Metropolis-Hastings algorithm and Gibbs sampler. These two random sampling techniques are integrated in order to approximate multi-dimensional, asymmetric, multi-modal probability distributions of parameters of nonlinear dynamic processes. The proposed method uses a sequential approach to approximate the full probability distributions of parameters without making any assumptions about the shape or variance of the probability distributions. By discarding the misleading assumptions that are only applicable to linear processes, the method preserves the information available from different experimental runs in the form of probability distributions of the process parameters. The following two figures summarize the sequential approach taken in this thesis. Figure 4.3 illustrates the outer level of iterative estimation where the multi-dimensionality of the probability distribution is handled. θ denotes the parameter vector of dimension $\Re^{p\times 1}$; superscript (i) denotes the Gibbs index in the chain of random samples generated by the algorithm; subscript j denotes the index of the parameter vector. Figure 4.4 illustrates the inner level or iterative estimation that approximates the univariate *posterior* distribution of the process parameters using multiple data sets. This process is embedded in the Gibbs Sampler as a means to compute the conditional distribution required in the algorithm. The input of this flowchart from the outer level of iterations is j (e.g. the jth element in the vector of θ that is currently being considered in the Gibbs sampling iteration). The prior distribution, $p_0(\theta)$ (before applying any experimental data), is user-defined. κ indicates the index of experimental data set and k is the total number of available sets. (h)

is the Metropolis-Hastings index in the current M-H Markov chain (not to be confused with the Gibbs index (i)).

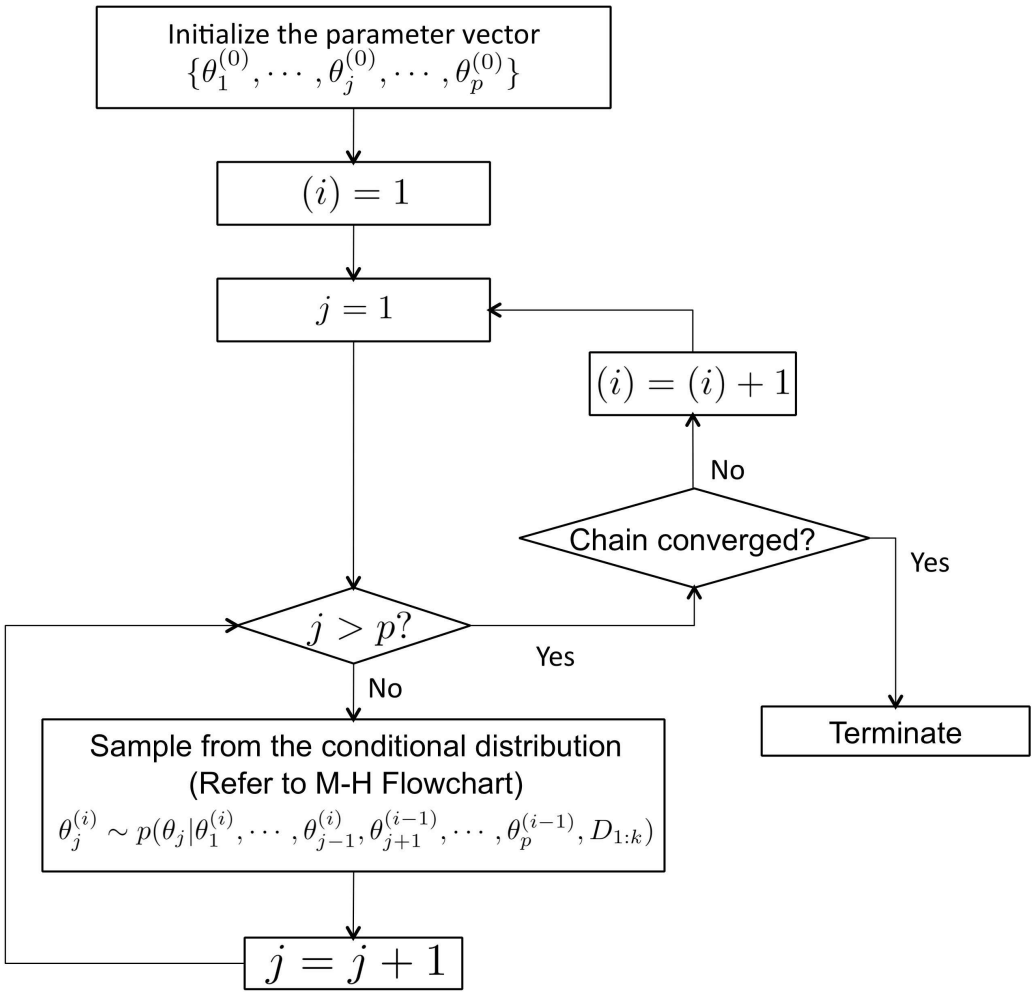

Figure 4.3: Flowchart of Gibbs Sampler for outer level of iterative estimation, where multi-dimensional probability distribution of nonlinear process parameters is approximated. θ denotes the parameter vector of dimension $\Re^{p\times 1}$; superscript (i) denotes the Gibbs index in the chain of random samples generated by the algorithm; subscript j denotes the index of the parameter vector.

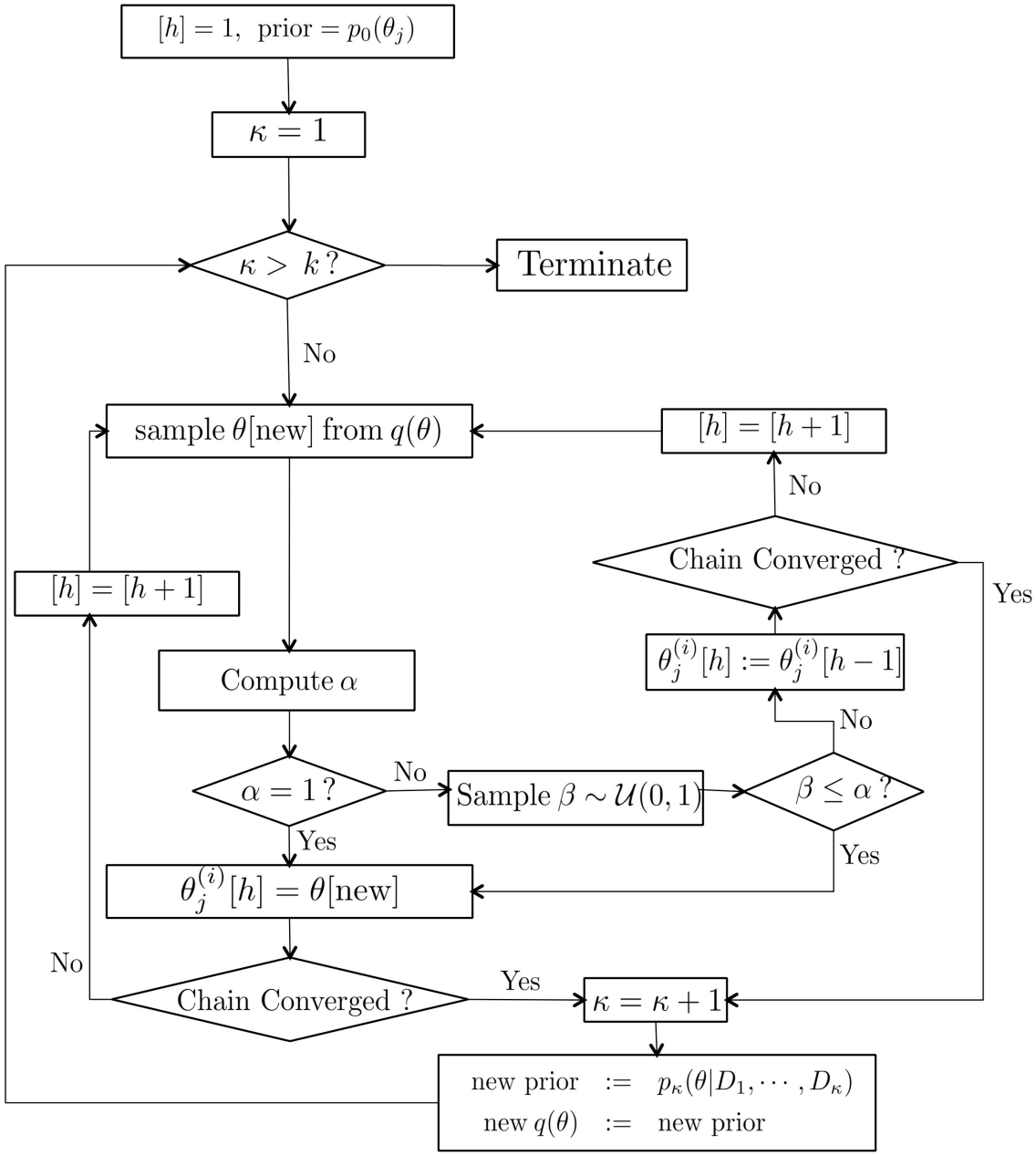

Figure 4.4: Metropolis-Hastings Algorithm for inner level of iterative estimation, where the univariate conditional distribution of *posterior* distribution is approximated. The conditional distribution is required for implementation of Gibbs Sampler. The input of this flowchart from the outer level of iteration is j (e.g. the element in the vector of θ is currently considered in the Gibbs Sampling). The prior distribution, $p_0(\theta$ (before applying any experimental data), is user-defined. κ indicates the index of experimental data among the multiple data sets, where k is the total number of available sets. $[h]$ is the Metropolis-Hastings index in the current MH Markov chain.

Chapter 5

Case Studies

The sequential Bayesian parameter estimation method discussed in the previous chapters is illustrated using three nonlinear biological systems - batch fermentation, Feed-Forward Loop genetic regulatory network and JAK-STAT signal pathway. Both published experimental data and simulated data are used. The estimated probability distributions of the process parameter vectors are analyzed for identifiability, correlation among the parameters and sensitivity of observations. It is argued that estimating the full probability distributions of nonlinear parameters allows improved confidence in parameters compared to point estimates.

5.1 Batch Fermentation Reaction

5.1.1 Single Parameter Estimation

In order to illustrate the inner-loop estimation method using Metropolis-Hastings algorithm, a single parameter, Y_{XS} is estimated using the following assumptions: i) All three state variables are measurable, and ii) the values of μ_m, k_s, k_p and Y_{PX} are known, reducing the parametric space from five dimensions to one-dimension. Thus, the parameter vector is $\theta' = \theta \,/\, \{\mu_m = 0.15, k_s = 0.50, k'_p = 0.25, Y'_{PX} = 0.20\} = \{Y_{XS}\}$.

The prior distribution, $p_0(Y_{XS})$, is assumed to be uniform. Previous literature and expert knowl-

edge indicates a lower and an upper bound on Y_{XS} of 0 and 1 respectively.

$$p_0(Y_{XS}) = \begin{cases} 1 & \text{if } 0 < Y_{XS} \leq 1, \\ 0 & \text{otherwise} \end{cases} \tag{5.1}$$

With this prior probability distribution and likelihood function (the derivation is explained in Appendix B), the M-H algorithm was applied to approximate the final probability distribution of Y_{XS} from six simulated data sets, D_1, D_2, D_3, D_4, D_5 and D_6.

$$p(Y_{XS} \,|\, \mu_m = 0.15, k_s = 0.5, k'_P = 0.25, Y'_{PX} = 0.2, D_1, \cdots, D_6) \tag{5.2}$$

The algorithm was run for 10000 iterations for each step of applying successive experimental data. The first 5000 samples in each sequence were discarded to avoid the 'burn-in' effect. The evolving behavior of posterior distributions is clearly illustrated in Figure 5.1. Starting from Panel (A), the first posterior probability distribution already indicates the Markov Chain's higher affinity towards the region near the true value. In Panel (F), the approximated posterior distribution is shown to form a sharp peak close to the true value, indicated by the solid vertical lines. The range of x-axes of the sub-figures in Figure 5.1 was kept constant in the $[0, 1]$ interval in order to illustrate the dramatic evolution of probability distributions.

The variance of the probability distributions cannot be calculated straightforwardly as in the case of Gaussian distributed random variables, because of the asymmetrical distribution of the parameters. Therefore, the approximated probability distributions are analyzed individually to obtain the 95% HPD interval. Table 5.1 summarizes the 95% HPD intervals of the estimated posterior probability distributions as well as their expected mean values. After each successive M-H step, the approximate posterior probability distribution shows higher confidence due to the decreasing trend in the 95% Highest Probability Density (HPD) interval.

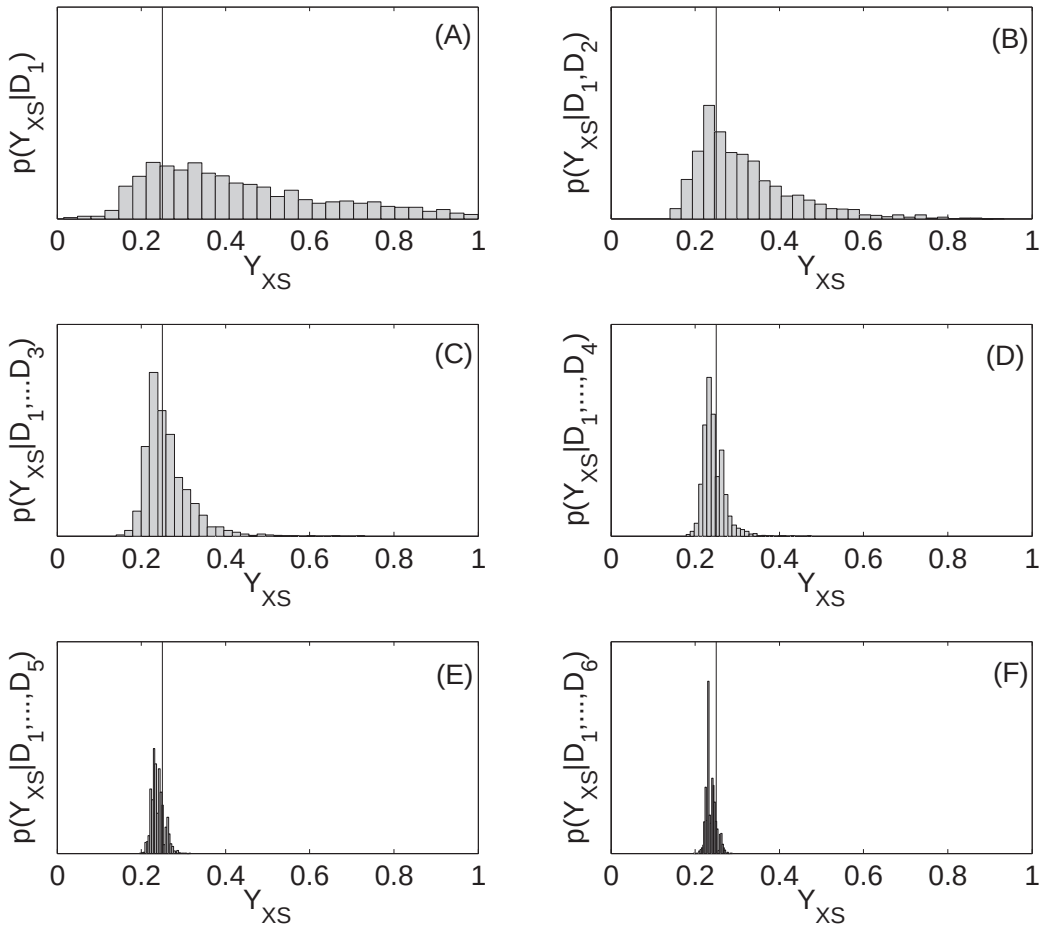

Figure 5.1: Each probability distribution is normalized so that $\int p(Y_{XS} \mid D_i)$ is equal to unity. The vertical line at $Y_{XS} = 0.25$ corresponds to the true value of the parameter. Panel (A) corresponds to the *posterior* probability distribution of Y_{XS} estimated using the first experimental data set, D_1
Panel (A) : $p(Y_{XS} \mid \mu_m = 0.15, k_s = 0.5, k'_P = 0.25, Y'_{PX} = 0.2, D_1)$,
Panel (B) : $p(Y_{XS} \mid \mu_m = 0.15, k_s = 0.5, k'_P = 0.25, Y'_{PX} = 0.2, D_1, D_2)$,
Panel (C) : $p(Y_{XS} \mid \mu_m = 0.15, k_s = 0.5, k'_P = 0.25, Y'_{PX} = 0.2, D_1, D_2, D_3)$,
Panel (D) : $p(Y_{XS} \mid \mu_m = 0.15, k_s = 0.5, k'_P = 0.25, Y'_{PX} = 0.2, D_1, \cdots, D_4)$,
Panel (E) : $p(Y_{XS} \mid \mu_m = 0.15, k_s = 0.5, k'_P = 0.25, Y'_{PX} = 0.2, D_1, \cdots, D_5)$,
Panel (F) : $p(Y_{XS} \mid \mu_m = 0.15, k_s = 0.5, k'_P = 0.25, Y'_{PX} = 0.2, D_1, \cdots, D_6)$

The rest of the parameters were estimated using an analogous approach. This was done to demonstrate that in the case of a single parameter process, the M-H algorithm is good enough to approximate the corresponding *posterior* probability distribution from multiple experimental data sets. Figure 5.2 shows the normalized distribution of each parameter. The expected mean values and 95% HPD intervals are shown in Table 5.2. It is shown that k_s and k_P have 95% HPD intervals of relatively larger magnitude compared to the three other parameters, μ_m, Y_{XS} and Y_{PX}. This discrepancy can be attributed to the innate property of the model. It has been shown in previous studies that the saturation constant parameters of the Michaelis-Menten kinetic model are theoretically identifiable in deterministic cases, but are difficult to estimate when the experimental data are corrupt with stochastic noise [3, 27]. Even so, M-H algorithm is successful in estimating the full probability distribution of these parameters from nonlinear stochastic time series data with a small bias.

An interesting aspect of the single parameter estimation using M-H algorithm is that it does not require an initial guess. A major disadvantage of traditional parameter estimation methods such as maximum likelihood estimator (MLE) or nonlinear least-squares (NLS) is that they require an initial guess in order to start the algorithm. If there are several local minima for the objective functions of MLE and NLS, then the choice of initial guess becomes critical to the estimation result. However, in M-H algorithm, it only requires that the initial prior probability distribution, p_0, is defined so that it contains the true value. In order to demonstrate this, following two different probability distributions of Y_{XS} were assigned to the initial prior probability distribution and M-H algorithm was executed.

$$\begin{aligned} p_0(Y_{XS}) &= \mathcal{U}(0, 10) \\ p_0(Y_{XS}) &= \mathcal{U}(-1, 1) \end{aligned}$$

where $\mathcal{U}(a, b)$ indicates a uniform distribution between a and b. For the approximation of each successive posterior probability distribution, 10000 iterations were executed and the first 5000

Table 5.1: The expected means and 95% Highest Probability Density (HPD) intervals of the estimated posterior probability distributions Y_{XS} using MH algorithm approximation.

	Expected Mean	95% HPD Intervals	
		Lower Bound	Upper Bound
$p(Y_{XS}\|\mu_m, k_s, k'_P, Y'_{PX}, D_1)$	0.338	0.035	0.757
$p(Y_{XS}\|\mu_m, k_s, k'_P, Y'_{PX}, D_1, D_2)$	0.278	0.086	0.466
$p(Y_{XS}\|\mu_m, k_s, k'_P, Y'_{PX}, D_1, \cdots, D_3)$	0.256	0.161	0.350
$p(Y_{XS}\|\mu_m, k_s, k'_P, Y'_{PX}, D_1, \cdots, D_4)$	0.251	0.223	0.290
$p(Y_{XS}\|\mu_m, k_s, k'_P, Y'_{PX}, D_1, \cdots, D_5)$	0.251	0.234	0.273
$p(Y_{XS}\|\mu_m, k_s, k'_P, Y'_{PX}, D_1, \cdots, D_6)$	0.250	0.237	0.269

Table 5.2: The expected means and 95% Highest Probability Density (HPD) intervals of the estimated posterior probability distributions of θ using MH algorithm approximation.

	Expected Mean	95% HPD Intervals	
		Lower Bound	Upper Bound
$p(\mu_m\|k_s, k'_P, Y_{XS}, Y'_{PX}, D_1, \cdots, D_6)$	0.152	0.134	0.170
$p(k_s\|\mu_m, k'_P, Y_{XS}, Y'_{PX}, D_1, \cdots, D_6)$	0.488	0.360	0.607
$p(k'_p\|\mu_m, k_s, Y_{XS}, Y'_{PX}, D_1, \cdots, D_6)$	0.220	0.056	0.399
$p(Y_{XS}\|\mu_m, k_s, k'_P, Y'_{PX}, D_1, \cdots, D_6)$	0.250	0.237	0.269
$p(Y'_{PX}\|\mu_m, k_s, k'_P, Y_{XS}, D_1, \cdots, D_6)$	0.201	0.190	0.213

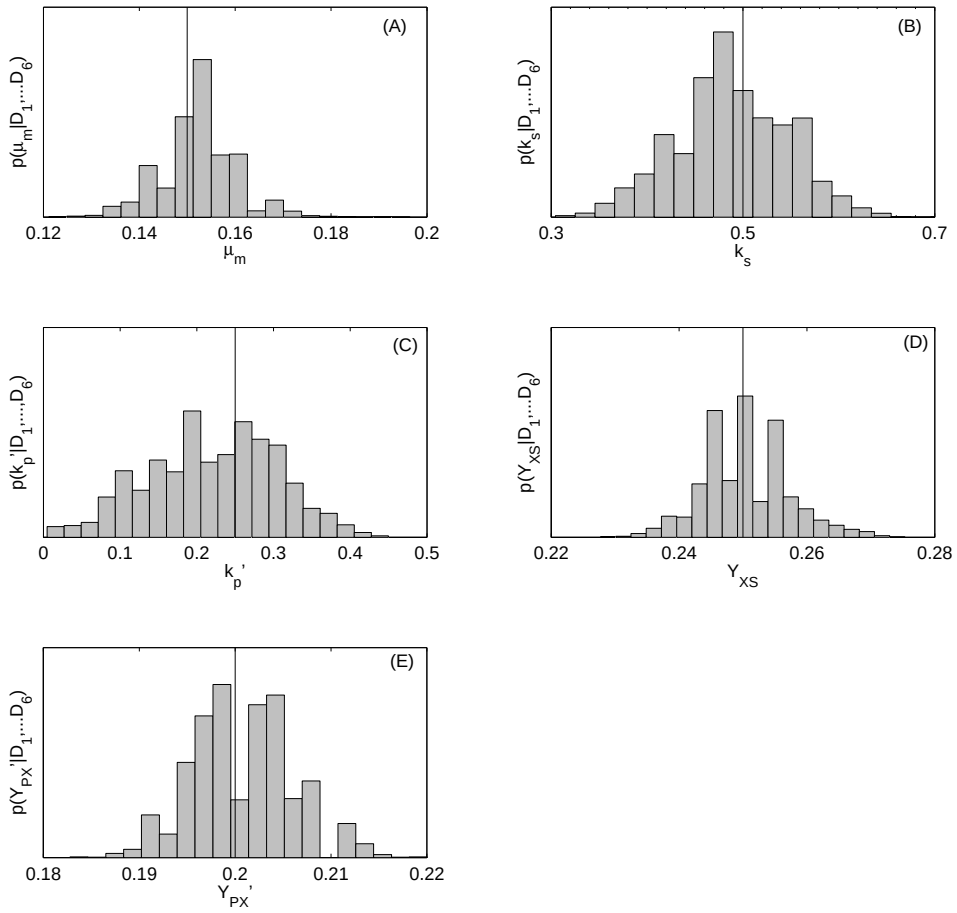

Figure 5.2: Normalized posterior distributions of the parameter vector θ, approximated using Metropolis-Hastings algorithm with 6 independently simulated batch reactor data sets while assuming that the true values of all other parameters are known.
Panel (A) : $p(\mu_m \mid k_s = 0.5, k'_P = 0.25, Y_{XS} = 0.25, Y'_{PX} = 0.2, D_1, \cdots, D_6)$
Panel (B) : $p(k_s \mid \mu_m = 0.15, k'_P = 0.25, Y_{XS} = 0.25, Y'_{PX} = 0.2, D_1, \cdots, D_6)$
Panel (C) : $p(k_P \mid \mu_m = 0.15, k_s = 0.5, Y_{XS} = 0.25, Y'_{PX} = 0.2, D_1, \cdots, D_6)$
Panel (D) : $p(Y_{XS} \mid \mu_m = 0.15, k_s = 0.5, k'_P = 0.25, Y'_{PX} = 0.2, D_1, \cdots, D_6)$
Panel (E) : $p(Y_{PX} \mid \mu_m = 0.15, k_s = 0.5, k'_P = 0.25, Y_{XS} = 0.25, D_1, \cdots, D_6)$

samples were discarded before using the remaining sequence to update the next prior probability distribution. The comparative results are shown in Figure 5.3. It is easy that as long as the uniform distribution contains the true value, the algorithm is successful. For the second row (in Figure 5.3), the uncertainty of the order of magnitude before the estimation was increased by one (i.e. interval (0,1) to interval (0,10)), and the approximated posterior probability distribution still pointed to an expected mean of 0.259, resulting in normalized error of 3.4%. Further simulations with an even higher level of *a priori* uncertainty were tried and the estimation was found to be equally good. For the third row, where the prior contains negative region, which is physically infeasible because the yield coefficient cannot be negative, the proposed method was still effective.

5.1.2 Multiple Parameter Estimation

After confirming that conditional probability distributions can be successfully approximated using M-H algorithm, Gibbs sampler is implemented in order to estimate the multi-dimensional probability distribution of θ. The order of parameters estimated in Gibbs sampler was arranged as follows

$$\mu_m \rightarrow k_s \rightarrow k_p \rightarrow Y_{XS} \rightarrow Y_{PX}. \tag{5.3}$$

The Gibbs sampler was run for 1000 iterations, with 3000 M-H algorithm iterations per parameter per Gibbs sampling step. The first 20% of the Markov sequences generated using M-H algorithm were discarded in order to eliminate the 'burn-in' effect.

Figure 5.4 shows the plot of 1000 Gibbs samples drawn for μ_m and the approximated marginal distribution, $p(\mu_m \mid D_1, \cdots, D_6)$. The expected mean is 0.177 with the 95% HPD interval of (0.124,0.291). There are two distinct deviations from the true value in the Gibbs sequence, shown near the 500th iteration and the 650th iteration. These deviations give rise to a skewed probability distribution for this parameter. This type of asymmetrical probability distribution is

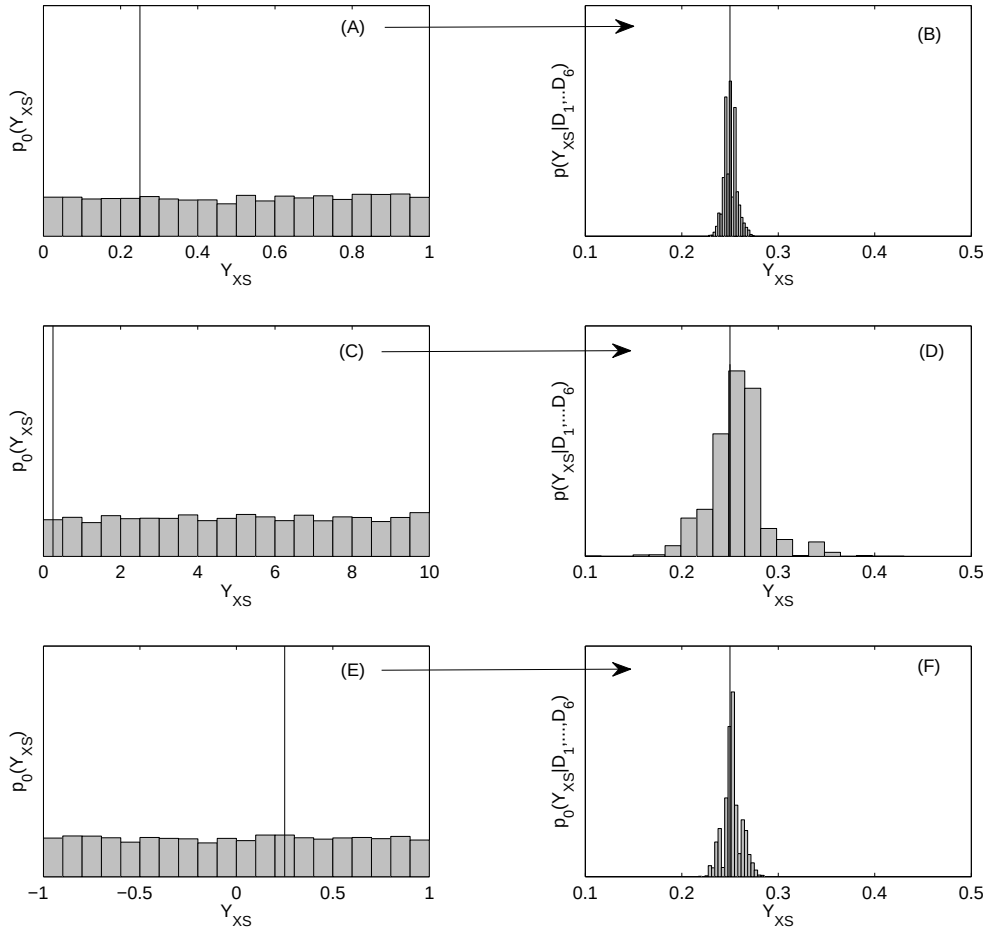

Figure 5.3: Normalized uniform prior distributions of Y_{XS} and the corresponding posterior probability distribution obtained using MH algorithm. The true value of Y_{XS} is indicated by solid vertical lines.

Panel (A) : $p_0(Y_{XS}) = \mathcal{U}(0,1)$ Panel (B) : $p(Y_{XS}|\mu_m, k_s, k'_P, Y'_{PX}, D_1, \cdots, D_6)$

Panel (C) : $p_0(Y_{XS}) = \mathcal{U}(0,10)$ Panel (D) : $p(Y_{XS}|\mu_m, k_s, k'_P, Y'_{PX}, D_1, \cdots, D_6)$

Panel (E) : $p_0(Y_{XS}) = \mathcal{U}(-1,1)$ Panel (F) : $p(Y_{XS}|\mu_m, k_s, k'_P, Y'_{PX}, D_1, \cdots, D_6)$

characteristic of parameters in nonlinear processes. This example illustrates that even though the added measurement noise is Gaussian, due to the nonlinearity of model, the resulting estimated parameters show non-Gaussian distribution.

When using MCMC sampling approach, it is important to determine how long the Markov chain has to be in order to obtain desired accuracy in approximation. There are several ways of doing this and monitoring the moving average of Markov chain is one of them. It helps to determine whether the sequence has reached steady-state distribution. Once the chain has reached its steady-state, it is generally accepted that the chain has converged to the target distribution with sufficient accuracy. The top panel in Figure 5.5 shows the moving average of μ_m Gibbs sequence and it is noted that there are slight initial fluctuations as the sequence explores the parameter space and gradually settles to a steady-state value. Another method to determine whether a sufficient length of the Markov chain has been generated is to monitor the behavior of the approximated distribution at different iterations. As the convergence is reached, the distributions approximated at increasing iterations will show negligible difference in their form. In Figure 5.5, panels (C), (D) and (E) look almost identical with positive skew, where panel (C) is the approximated marginal distribution of μ_m using the Gibbs sequence from 200th iteration to 600th iteration; panel (D) corresponds to the approximated distribution using the Gibbs sequence from 200th iteration to 800th iteration; and panel (E) corresponds to the approximated distribution using the Gibbs sequence from 200th iteration to the 1000th iteration (The first 200 samples of the Gibbs sequence were discarded to remove the 'burn-in' effect). The constant-shape trend demonstrates that the convergence of the chain is reached around 600th iteration.

The parameters, Y_{XS} and Y'_{PX}, demonstrated behavior similar to that of μ_m. However, for k_s and k'_P, as mentioned in the single parameter estimation case, it was difficult to obtain comparable accuracy of estimated values from the approximated marginal probability distributions. Thus, it was necessary to execute the Gibbs sampler for larger number of iterations in order to ensure that these two parameters reached their steady-state distribution. This, however, requires a larger

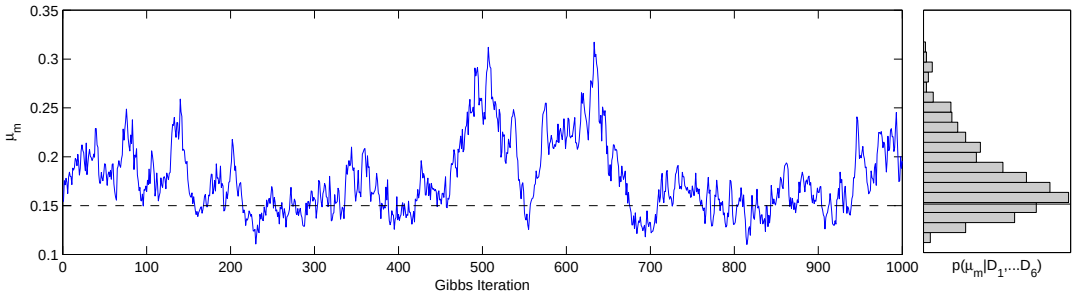

Figure 5.4: Plot of Gibbs sequence corresponding to μ_m for 1000 iterations and the approximated marginal distribution using the sequence. The dotted horizontal line corresponds to the true value.

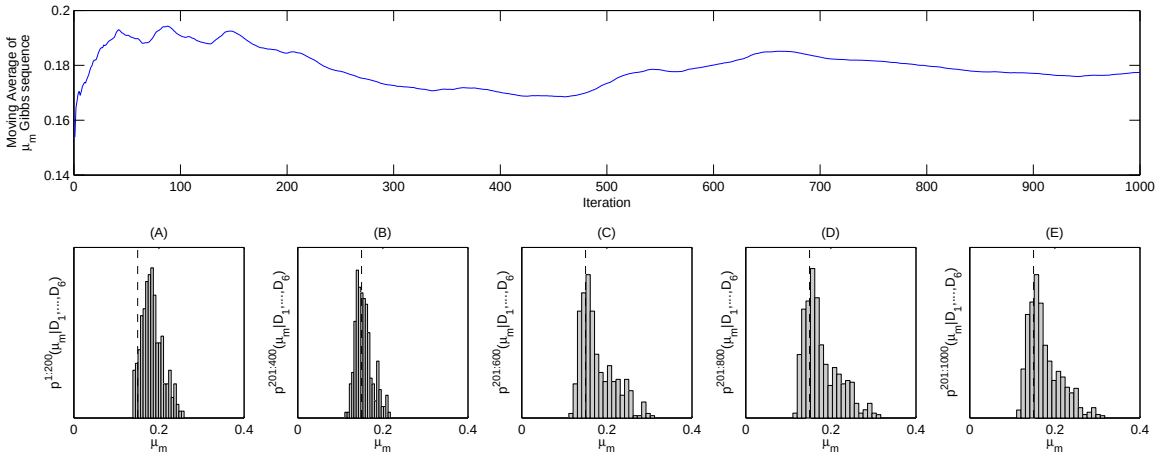

Figure 5.5: Moving average of Gibbs sequence corresponding to μ_m and the approximated marginal distributions computed using various portions of the sequence. Panel (A) corresponds to the marginal distribution approximated using the first 200 samples in the Gibbs sequence. This portion of the sequence is discarded from considering the convergence, in order to remove the residual effect of the initial point of the chain. Panel (B) corresponds to the marginal distribution of approximated using 201st to 400th samples in the Gibbs sequence. Panel (C) corresponds to the marginal distribution of approximated using 201st to 600th samples in the Gibbs sequence. Panel (D) corresponds to the marginal distribution approximated using 201st to 800th samples in the Gibbs sequence. Panel (E) corresponds to the marginal distribution approximated using 201st to 1000th samples in the Gibbs sequence.

amount of computational time. Therefore, in order to reduce the high computational cost, the Gibbs sequences corresponding to the parameters μ_m, Y_{XS} and Y'_{PX} were eliminated from the M-H algorithm steps that determine corresponding conditional distributions. In other words, the Gibbs sampler was paused once the steady-state convergence is confirmed for the three parameters and the resulting probability distributions were fixed as the full conditional probability distributions as follows,

$$p_C^{II}(\mu_m \,|\, k_s, k'_P, Y_{XS}, Y'_{PX}, D_1, \cdots, D_6) = p^{(1:1000)}(\mu_m \,|\, k_s, k'_P, Y_{XS}, Y'_{PX}, D_1, \cdots, D_6) \tag{5.4}$$

$$p_C^{II}(Y_{XS} \,|\, \mu_m, k_s, k'_P, Y'_{PX}, D_1, \cdots, D_6) = p^{(1:1000)}(Y_{XS} \,|\, \mu_m, k_s, k'_P, Y'_{PX}, D_1, \cdots, D_6) \tag{5.5}$$

$$p_C^{II}(Y'_{PX} \,|\, \mu_m, k_s, k'_P, Y_{XS}, D_1, \cdots, D_6) = p^{(1:1000)}(Y'_{PX} \,|\, \mu_m, k_s, k'_P, Y_{XS}, D_1, \cdots, D_6) \tag{5.6}$$

where p_{II} is the conditional distribution to be used in the Gibbs sampler Phase II and $p^{1:1000}$ denotes marginal probability distribution obtained from the first 1000 iterations of the Gibbs sampler, which are shown in Figure 5.6. In Phase II (i.e. Gibbs sequence iteration 1001st and onward), the M-H algorithm is no longer applied to μ_m, Y_{XS} and Y'_{PX} for individual approximation of the full conditional distribution, instead (5.4), (5.5) and (5.6) are used to randomly sample the next Gibbs sequence value. Thus, Phase I refers to the estimation process where all five of the parameters were actively analyzed for full conditional distributions using M-H algorithm and Phase II refers to the estimation process where only k_s and k'_P are actively analyzed for conditional distribution. An additional 1000 iterations of Gibbs sampler were executed in Phase II for the convergence of k_s.

Figure 5.7 shows the joint distribution contour plots of θ using the Gibbs sequence from both Phase I and Phase II. In order to visualize the five-dimensional probability space, the parameters

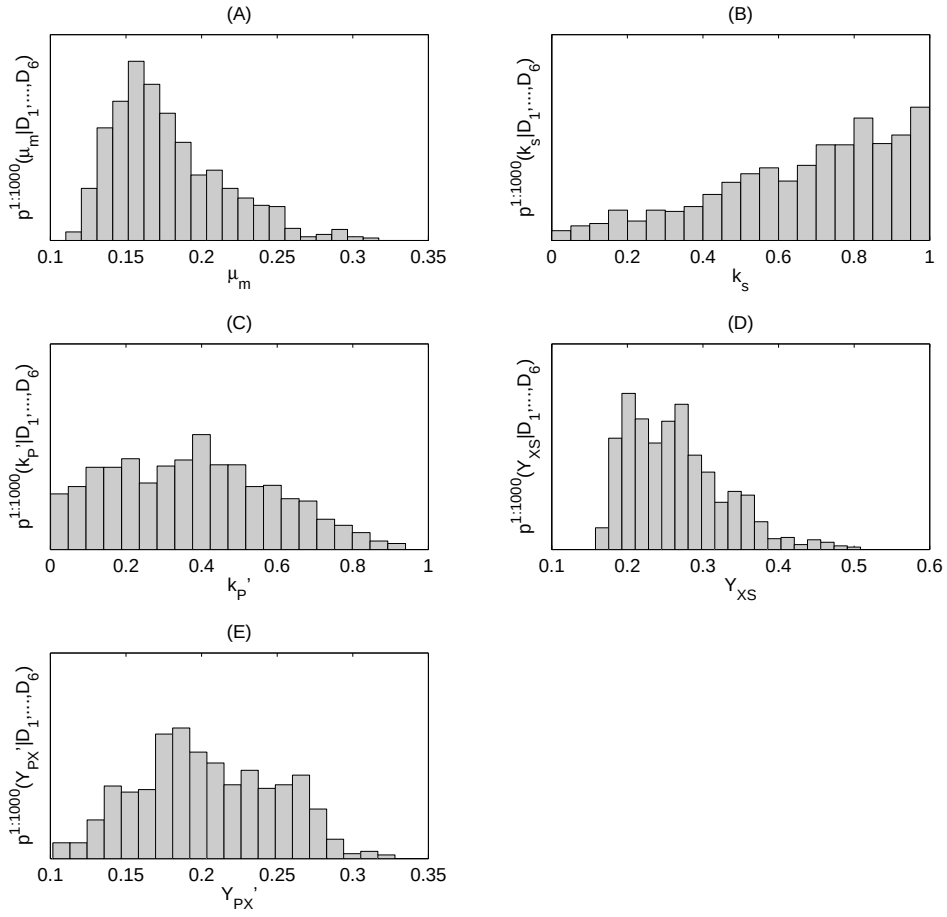

Figure 5.6: The approximated marginal distributions of θ using the first 1000 samples of the Gibbs sequence (i.e. Phase I). The distributions for k_s and k'_P have not reached convergence and have wider intervals compared to the other three parameters.
Panel (A) : $p^{1:1000}(\mu_m \mid D_1, \cdots, D_6)$
Panel (B) : $p^{1:1000}(k_s \mid D_1, \cdots, D_6)$
Panel (C) : $p^{1:1000}(k'_P \mid D_1, \cdots, D_6)$
Panel (D) : $p^{1:1000}(Y_{XS} \mid D_1, \cdots, D_6)$
Panel (E) : $p^{1:1000}(Y'_{PX} \mid D_1, \cdots, D_6)$

are paired up to portray two-dimensional joint distributions instead. The solid lines represent the true values of the process parameters. For panels (i) and (ii), the highest probability region is tightly clustered near the cross point of the solid lines. This indicates that the parameters μ_m, Y_{XS} and Y'_{PX}, were estimated with higher accuracy compared to the other parameters where larger distributions along the parameter space are demonstrated. For instance, in panel (x), the joint distribution of k_s and k'_P shows multiple high probability regions, agreeing with previous studies on the difficulties of estimating the Monod constants using stochastic experimental data. From studying panel (iii), it can be inferred that the yield coefficient parameters Y_{XS} and Y'_{PX} have an inversely proportional relationship, such that when the value of Y'_{PX} increases, Y_{XS} decreases. It may be possible to use parameters that have strong correlation with one another, such as this pair, in optimizing the structure of the model. Modeling a dynamic process is a compromise between accurate portrayal of the true process and simplicity of the model for computational purpose. Therefore, it is beneficial to conduct an iterative analysis between the true process and the proposed models in order to arrive at a parsimonious model. For instance, if a numerical function of Y_{XS} in terms of Y'_{PX} can be developed, the parameter space of the unstructured Michaelis-Menten model (1.9) can be reduced from five to four.

The maximum *a posteriori* estimate, the expected value and the 95% HPD interval of each marginal distribution is shown in Table 5.3. With asymmetric probability distributions shown in Figure 5.5, it is difficult to determine what is the best estimate of the parameter. In any case, as mentioned previously, it is difficult to summarize the shape into a couple of representative statistical values for asymmetric distributions.

5.2 Genetic Regulatory Network : Feed Forward Loop

In [10], four different candidates of FFLs were studied. The candidate genes were obtained from [20] where gene expression responses to twelve different environment changes in *Saccha-*

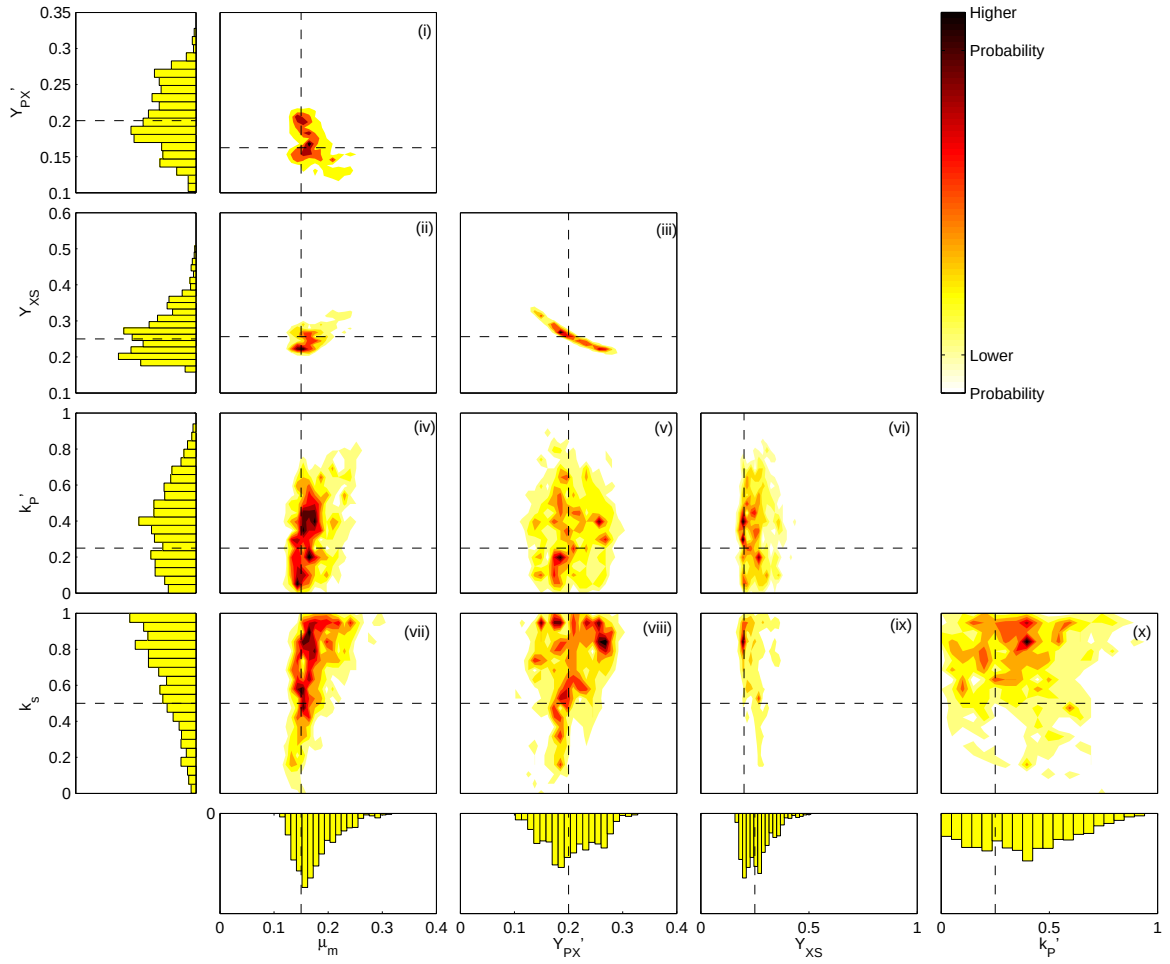

Figure 5.7: Joint distributions of batch fermentation process parameters approximated using Metropolis-Hastings algorithm and Gibbs sampler [Results from both Phase I and II were used]. Each panel corresponds to the pair of parameters indicated by the x-axis and the y-axis. The region corresponding to higher probability is indicated with black (the higher end of the color bar on the right) and the lowest probability region is indicated with white (the lower end of the color bar on the right). The intersection of dotted lines represent the coordinates of the true values of the parameter pairs.

romyces cerevisiae were studied. Among the four candidate FFLs, FFL1 with X: Gene GCN4, Y: Gene LEU3 and Z: Gene ILV5 performed the best in the test of the goodness of fit for dynamic models. In this case study, the same FFL is used to illustrate the algorithm.

The gene expression level of X is not a state variable in FFL model and therefore it is regarded as an input variable with a known sequence. To simulate $X(t)$, random noise variable, $w(t)$ with zero mean and variance, σ_w^2, was added to the experimental data from [20]. The initial conditions of Y and Z as well as the parameter values were obtained from [10], where three of the parameter values were heuristically determined such that $\beta_y = \beta_z = 1$ and $H = 2$. The rest of the parameter values are listed in Table 5.4. For the model (1.13) and (1.14), process noise terms, $v_1(t)$ and $v_2(t)$ were added to each equation as follows to simulate the stochastic nature of the dynamic process.

$$\frac{dY(t)}{dt} = -\alpha_y Y(t) + \beta_y f(X(t), K_{XY}) + v_1(t), \tag{5.7}$$

$$\frac{dZ(t)}{dt} = -\alpha_z Z(t) + \beta_z g(X(t), Y(t), K_{XZ}, K_{YZ}) + v_2(t), \tag{5.8}$$

The sampling times were set at $t = 5, 10, 15, 20, 30, 40, 60$ and 80 minutes ($N = 8$). The equations were solved using ode45 function in MATLAB® and to obtain the output variables, measurement noise variables were added as follows.

$$\begin{aligned} y_1(t) &= Y(t) + \eta_Y(t), \\ y_2(t) &= Z(t) + \eta_Z(t), \end{aligned}$$

where $\eta_Y(t)$ and $\eta_Z(t)$ are independent and Gaussian distributed variables with zero mean and variance σ_Y^2 and σ_Z^2, respectively. Using the parameter values in Table 5.4 and the simulated input sequences, a total of seven experimental data sets were collected. Each experimental data

set is defined as,

$$D_i = [Y(t_1), \cdots, Y(t_N), Z(t_1), \cdots, Z(t_N), X(t_1), \cdots, X(t_N)], \qquad i = 1, 2, 3, 4, 5, 6, 7 \tag{5.9}$$

Six of them, D_1, D_2, D_3, D_4, D_5 and D_6, were applied to the algorithm to estimate the probability distribution of process parameters and the last experimental data set, D_7, was used to validate the estimation result. Figure 5.8 shows a simulated FFL time series of the input variable $X(t)$ and the two state variables $Y(t)$ and $Z(t)$.

The initial priors for estimating full conditional probability distributions using M-H algorithm, $p_0(\theta)$, were set to a uniform probability distribution between 0 and 1, $\mathcal{U}(0, 1)$, for all five parameters. The likelihood function was derived in an analogous approach as the previous case study and is expressed as follows.

$$L(\theta \mid D_1) = \frac{1}{(2\pi)^{N/2}\sigma_Y^N \cdot \sigma_Z^N} \exp\left(\sum_{i=0}^{i=N-1} -\frac{(y_1(t_i) - \hat{Y}(t_i, \theta))^2}{2\sigma_Y^2} - \frac{(y_2(t_i) - \hat{Z}(t_i, \theta))^2}{2\sigma_Z^2} \right), \tag{5.10}$$

where $\hat{Y}(t_i)$ and $\hat{Z}(t_i)$ are predicted output variables as a function of θ. A total of 2000 iterations of Gibbs sampler with 3000 iterations of M-H algorithm per parameter per Gibbs iteration were executed. The resulting Gibbs sequences are shown in Figure 5.9. The left column shows the plots of the Gibbs sequences versus the iteration index and the right column shows the marginal probability distributions of process parameters approximated using the sequences shown on the left. The dotted lines denote the true parameter values. Table 5.5 shows the summary of maximum *a posteriori* estimate, expected mean and the 95% HPD interval obtained from the approximate marginal distributions.

From Figure 5.9, it can be noticed that K_{XZ} and K_{YZ} have negative and positive skew. Both

Table 5.3: The maximum *a posteriori*, the expected mean, the normalized error and the 95% confidence interval calculated from marginal distribution corresponding to each process parameter of batch fermentation reaction model.

Process Parameter	μ_m	k_s	k'_P	Y_{XS}	Y'_{PX}
Maximum *a posteriori*	0.153	0.591	0.005	0.195	0.189
↪Normalized Error	0.020	0.182	0.979	0.221	0.056
Expected Mean	0.177	0.593	0.460	0.264	0.204
↪Normalized Error	0.180	0.187	0.841	0.057	0.020
95% HPD Interval - Lower Bound	0.126	0.240	0.005	0.177	0.123
95% HPD Interval - Upper Bound	0.260	0.952	0.950	0.416	0.284

Table 5.4: The parameter vector value used in order to simulate the time series data of FFL genetic regulatory network.

Process Parameter	α_y	α_z	K_{XY}	K_{XZ}	K_{YZ}
Value	0.44	0.69	0.90	0.60	0.56

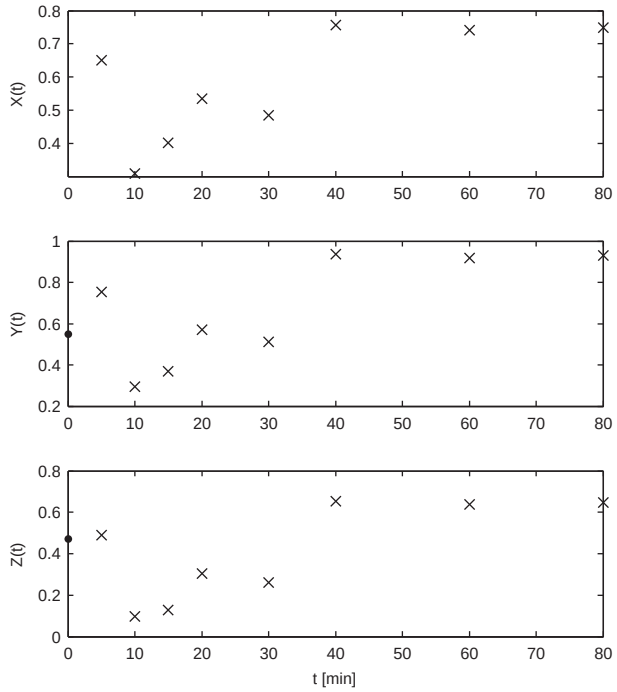

Figure 5.8: The simulated data of FFL genetic regulatory network model. There are eight measurements (denoted with x) of each state variable, $Y(t)$ and $Z(t)$. The initial values of the state variables (denoted with •) are assumed to be known from the estimation reported in [10]. $X(t)$ is regarded as an input variable and its initial value is not reported.

distributions are characterized by asymmetric bimodal distributions indicated by the presence of shorter peaks adjacent to the dominant ones. Maximum *a posteriori* values and the expected values are in good agreement with each other, as the marginal distributions have less asymmetry relative to the marginal distributions of batch fermentation process parameters. Also, it is observed that the normalized error of estimation result is relatively smaller, indicating that the algorithm was more successful for estimating the parameters of FFL model compared to the batch fermentation model.

Figure 5.10 shows the contour plots of process parameter joint distributions. Similar to Figure 5.7, the parameters were paired up for easier visualization of the probability distribution that exists in five-dimensional space. It is shown that the highest probability region of each panel shaded in black closely follows the cross-point of the 'true value' solid lines, demonstrating the accuracy of the estimation. In this model, some correlation among parameters can be inferred from the joint probability distribution contour plots. For instance, the high probability region of α_z and K_{XZ} form an inversely proportional relationship. And similar behavior is noted between the high probability regions of K_{XZ} and K_{YZ}.

In order to further investigate the accuracy of the estimated parameters, a separate simulated data set (D_7) was used for verification. The verification data was simulated without process or measurement noise in order to plot the 'true' measurement trajectories. Using the expected values in Table 5.5 and the known sequence of $X(t)$ from D_7, the expression profiles for $\hat{Y}(t)$ and $\hat{Z}(t)$ were predicted without process or measurement noise term. These trajectories were then compared with the simulated data. The plot is shown in Figure 5.11. It can be seen that by using the estimated parameter values, the expression profiles of $Y(t)$ and $Z(t)$ are predicted with very small error.

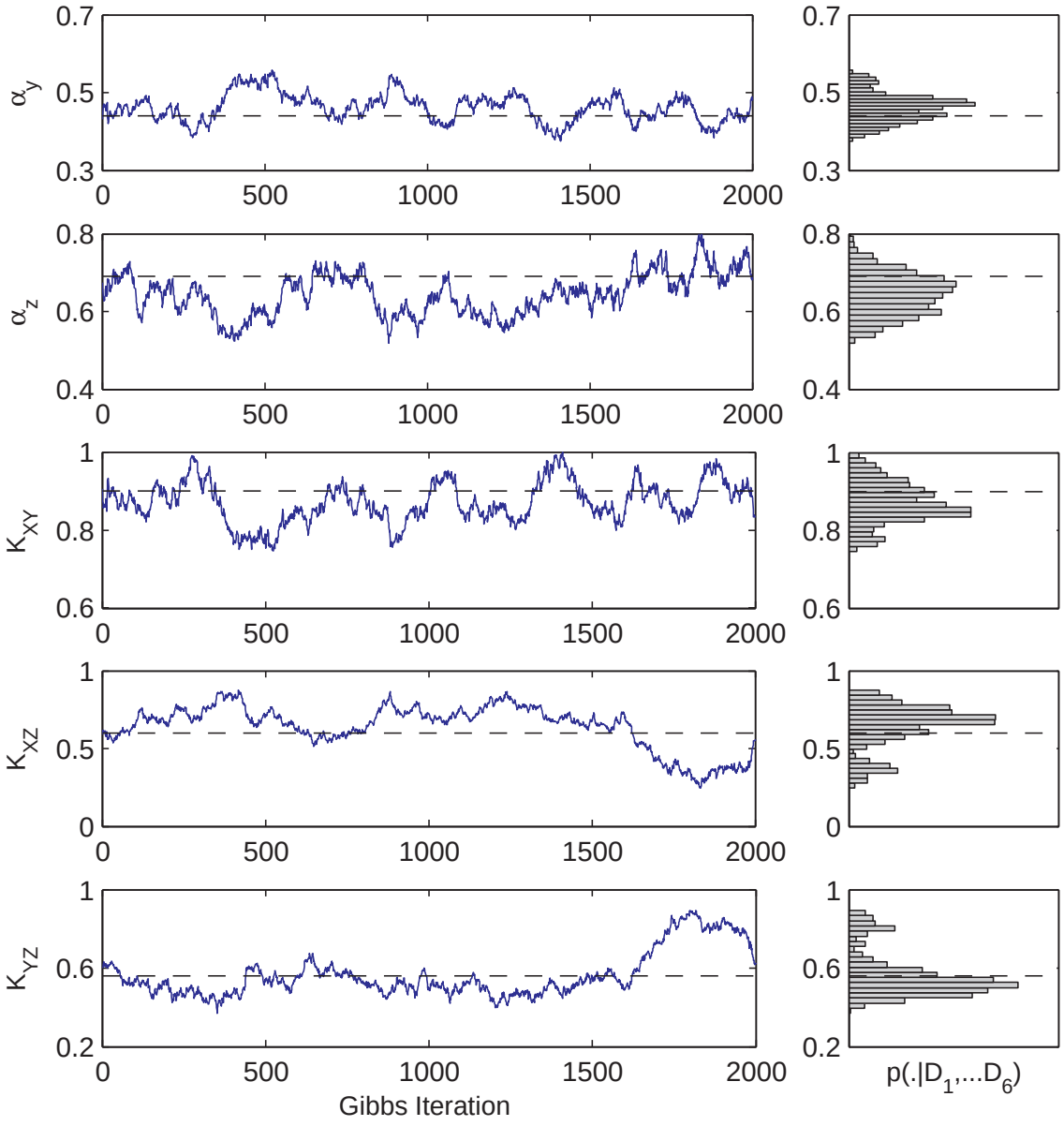

Figure 5.9: Plot of Gibbs sequences for 2000 iterations and the marginal distribution using of each process parameter approximated using the corresponding sequence. The dotted horizontal lines correspond to the true value of each process parameter used to simulate the experimental data of FFL model.

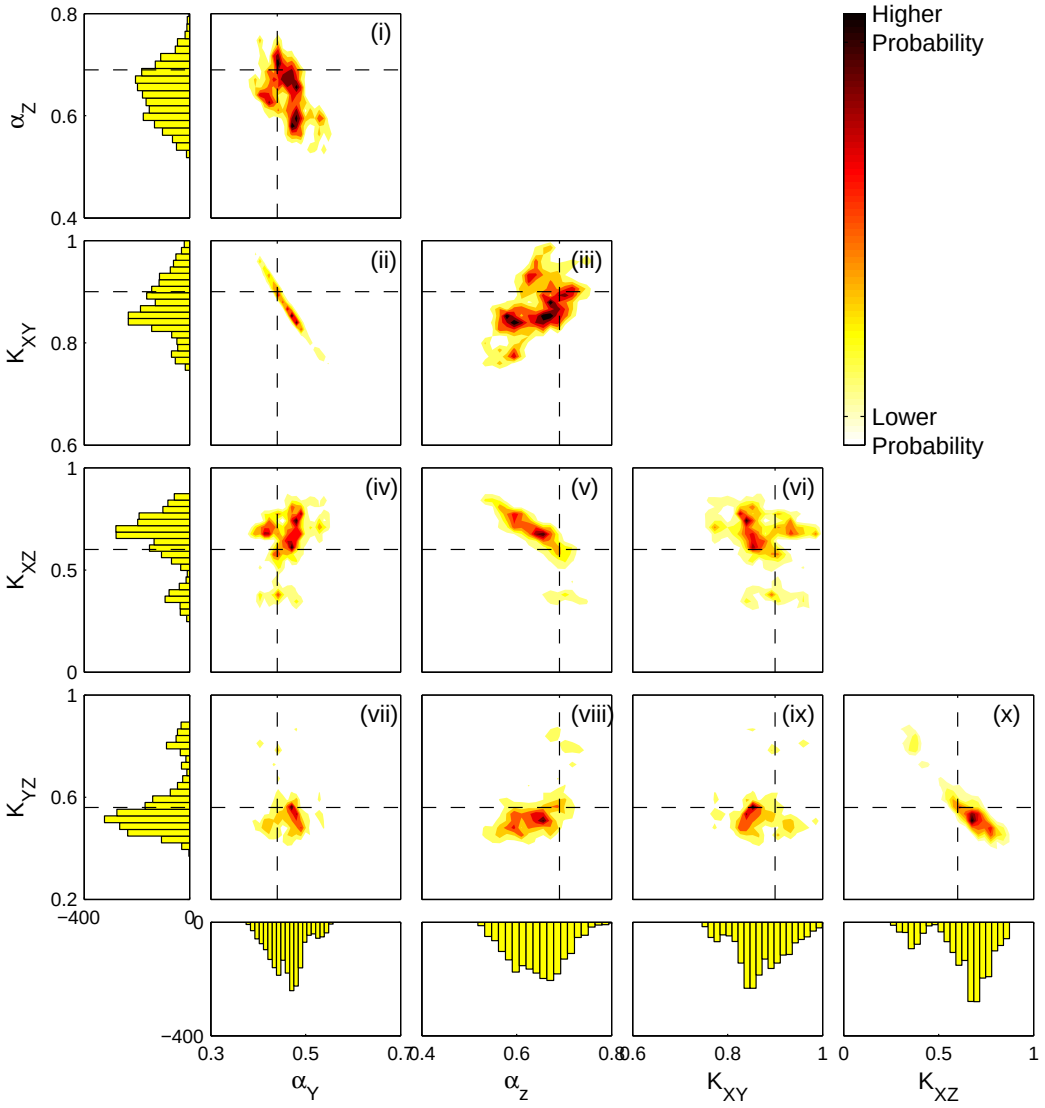

Figure 5.10: Joint distributions of FFL model parameters approximated using Metropolis-Hastings algorithm and Gibbs sampler. Each panel corresponds to the corresponding pair of parameters indicated on the x-axis and the y-axis. The region corresponding to higher probability is indicated with black (the higher end of the color bar on the right) and the lowest probability region is indicated with white (the lower end of the color bar on the right). The intersections of dotted lines represent the coordinates of true values of the parameter pairs.

Table 5.5: The maximum *a posteriori*, the expected mean and the 95% confidence interval calculated from each marginal distribution corresponding to the process parameter of FFL genetic regulatory network model.

Process Parameter	α_y	α_z	K_{XY}	K_{XZ}	K_{YZ}
Maximum *a posteriori*	0.469	0.659	0.861	0.701	0.513
↪Normalized Error	0.065	0.045	0.043	0.169	0.083
Expected Mean	0.461	0.644	0.871	0.641	0.565
↪Normalized Error	0.049	0.066	0.032	0.068	0.009
95% Confidence Interval - Lower Bound	0.340	0.543	0.765	0.307	0.424
95% Confidence Interval - Upper Bound	0.537	0.746	0.973	0.839	0.849

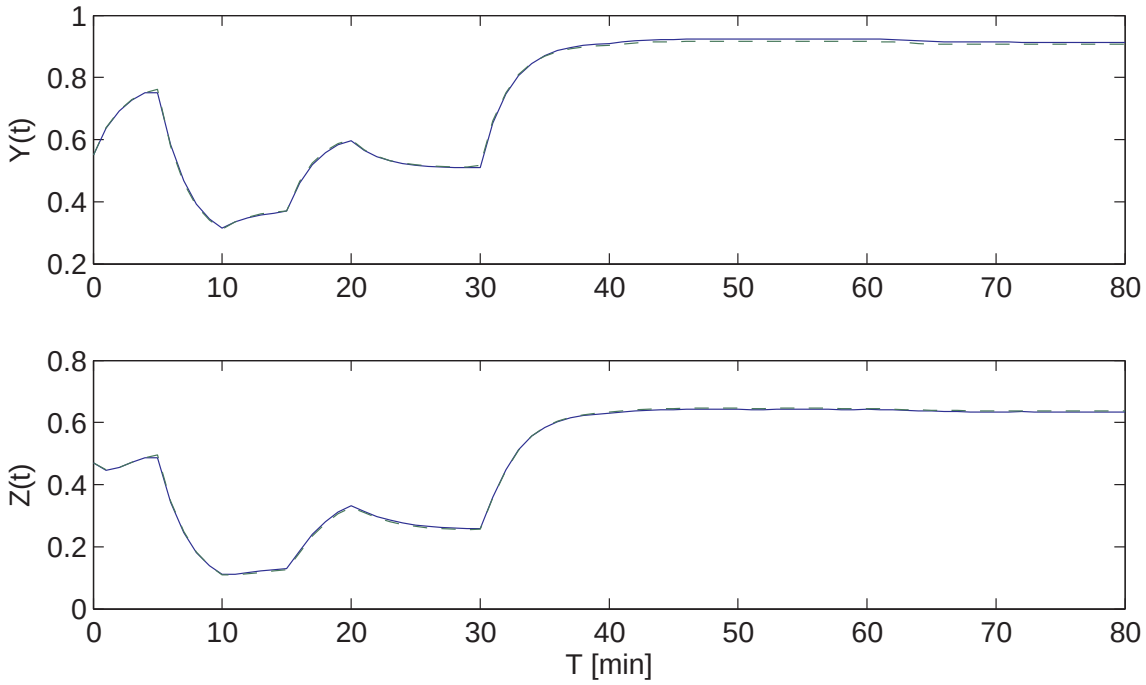

Figure 5.11: Using a known sequence of input variable $X(t)$ and the estimated parameters from Table 5.5, the gene expression profiles of $Y(t)$ and $Z(t)$ were predicted. They are compared with the simulated expression profile using the true parameter values. Solid line : true parameters. Dotted line: estimated parameters.

5.3 JAK-STAT Signal Transduction Pathway Model : Partially Observable States

There is a single experimental data set available from the literature which consists of the input time series, $u(t)$, representing the EpoR concentration profile and the two output variables, $y_1(t)$ and $y_2(t)$ measured at $t = 0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 25, 30, 40, 50, 60$ minutes ($N = 16$). The output variables are plotted in Figure 5.12, where 'x' denote y_1 and '+' denote y_2. The values of a_1, a_3 and a_4, were estimated in [42], using experimental data obtained from [51] and assuming $\tau = 4.001$ min, with Unscented Kalman Filter (UKF) estimation approach. They are reported to be $\hat{a}_1 = 0.0515 \pm 0.011$, $\hat{a}_3 = 3.39 \pm 0.882$ and $\hat{a}_4 = 0.35 \pm 0.092$. The estimation 95% confidence interval (1.96σ) was calculated assuming normal distribution of the process parameters. The trajectory of predicted output variables using this estimated value is shown in Figure 5.12 with solid and dashed curves.

Using these literature values, the prior distribution of each parameter was assigned as follows under the assumption that only the order of magnitude of estimated parameters are reliable.

$$p_0(a_1) = \mathcal{U}(0, 0.1), \tag{5.11}$$

$$p_0(a_3) = \mathcal{U}(0, 10), \tag{5.12}$$

$$p_0(a_4) = \mathcal{U}(0, 1). \tag{5.13}$$

The likelihood function was derived in an identical manner as previous case studies, assuming that the measurement noise is distributed normally with zero mean and standard deviation $\sigma = 0.1$. This value was assigned by examining the disagreement between the experimental data and the predicted output calculated by using the parameters reported in literature.

The Gibbs sampler was executed for 1000 iterations with 1000 M-H algorithm steps per parameter for evaluation of conditional distribution (The first 501 samples of the Markov chain was

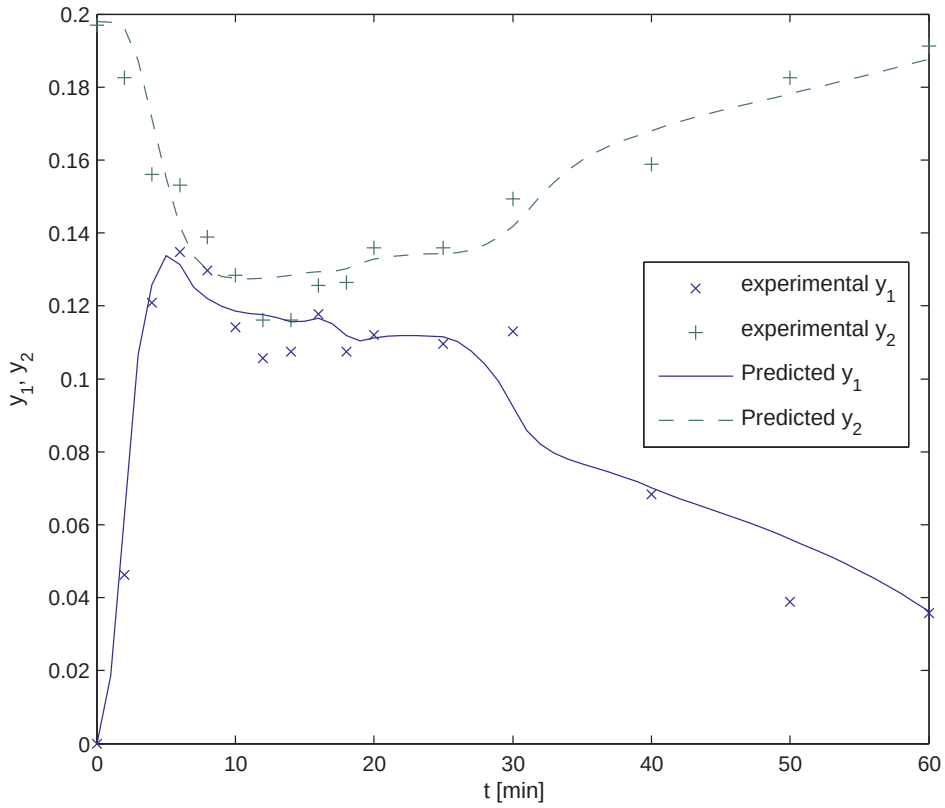

Figure 5.12: The experimental data, y_1 and y_2, obtained from [51] are denoted with 'x' and '$+$', respectively. The output trajectory of $\hat{y}_1$ (solid curve) and $\hat{y}_2$ (dashed curve) are predicted values calculated using the estimated values reported in [42].

discarded to remove the 'burn-in' effect). Figure 5.13 shows the joint probability distributions of pairs of the parameter visualized by the three contour plots. The maximum *a posteriori* values are indicated with dotted lines. The higher probability region is indicated by red and black (corresponds to the color bar), and the lowest probability region is indicated by white. The histograms on the outer margin of the figures are the marginal distribution. The magnified view of the marginal distribution of individual parameter are shown in Figure 5.14. The maximum *a posteriori*, expected mean and 95% HPD intervals are calculated and shown in Table 5.6. The approximated distributions of a_1 and a_4 converged to peaks which indicate that the uniform prior distribution was successfully assigned to contain the high probability region of the parameter space. Furthermore, the estimated results agree with the previous literature values, as the normalized 'differences'[9] are 0.109 and 0.014, respectively. However, for a_3, the marginal probability distribution did not converge to a peak where the highest probable region is easily identifiable.

An interesting aspect about this model is that the state, even though the JAK-STAT model has been generally known for its unobservable states, the first state x_1 is actually observable. From the measurements of y_1 and y_2, it becomes possible to compute the concentration profile of $x_1(t) = y_2(t) - y_1(t)$, which is shown in Figure 5.15. The experimental profile is compared with the predicted profile of x_1 using $\hat{\theta} = [0.0459, 9.351, 0.355]$. It is easy observe that the two values are in good agreement.

5.3.1 Comparison With Literature Parameter Values

For further investigation, the estimated parameter vectors from literature and those from the proposed Gibbs algorithm are examined in order to determine their ability to reliably represent the given experimental data. The three vectors are $\hat{\theta}_1 = [0.0515, 3.39, 0.35]$ (literature value),

[9]The author does not wish to use the term 'error' in this particular case, because the previous literature value is also an estimation, after all.

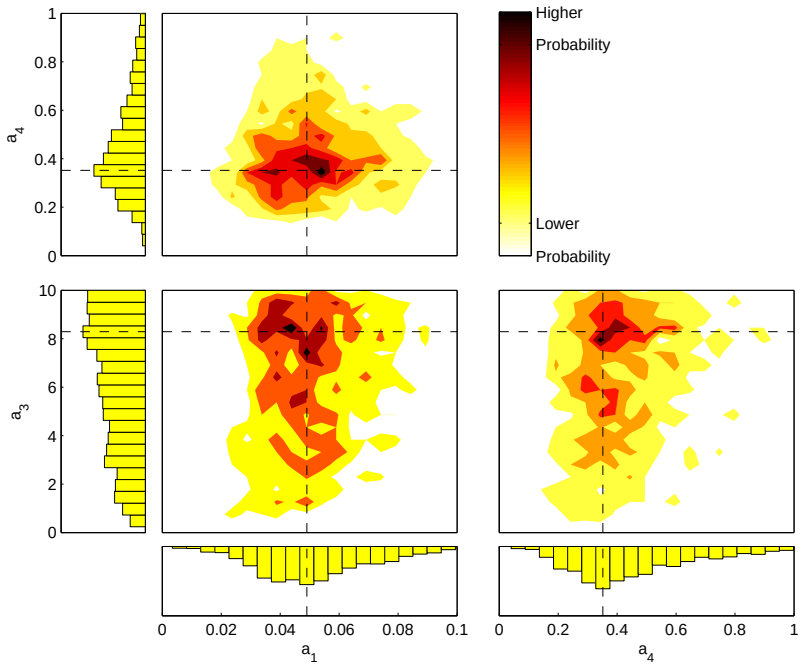

Figure 5.13: Joint distributions of JAK-STAT model parameters approximated using Metropolis-Hastings algorithm and Gibbs sampler. The data used for this analysis is the experimental data reported in previous literature. Each contour plot panel corresponds to the corresponding pair of parameters indicated on the x-axis and the y-axis. The region corresponding to higher probability is indicated with black (the higher end of the color bar) and the lowest probability region is indicated with white (the lower end of the color bar). The dotted lines represent the maximum *a posteriori* of the estimated probability distribution of true values of the parameter pairs. The histograms on the outer part of the figure represent the approximated marginal probability distributions identical to Figure 5.14.

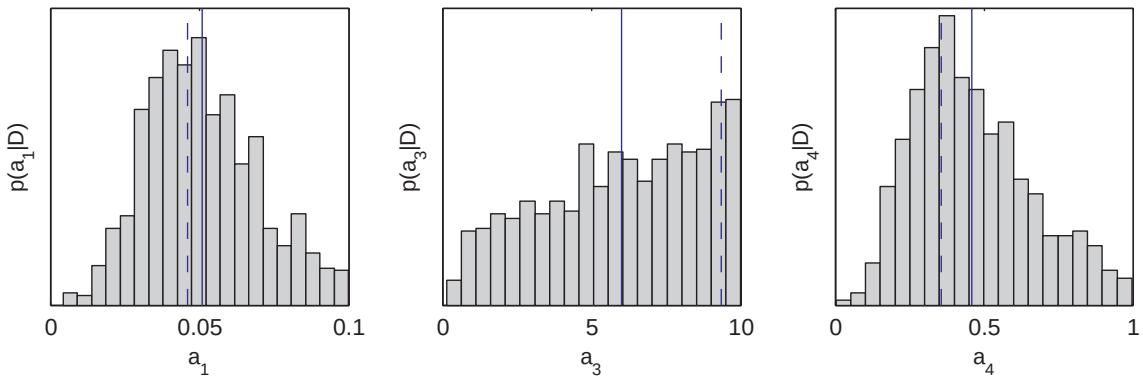

Figure 5.14: Approximated marginal distribution of JAK-STAT process parameters obtained from 1000 Gibbs sampler iterations. Each Gibbs iteration consisted of inner iteration of MH algorithm of 1000 samples per experiment per parameter. The expected mean is denoted with solid vertical line on each panel and the dashed vertical line denote the maximum *a posteriori* value.

Table 5.6: The maximum *a posteriori*, the expected mean and the 95% confidence interval calculated from each marginal distribution corresponding to the process parameter of JAK-STAT signal transduction pathway model.

Process Parameter	a_1	a_3	a_4
Maximum *a posteriori*	0.0459	9.351	0.355
Expected Mean	0.0509	6.004	0.460
95% Confidence Interval - Lower Bound	0.0144	0.682	0.137
95% Confidence Interval - Upper Bound	0.0928	9.942	0.962

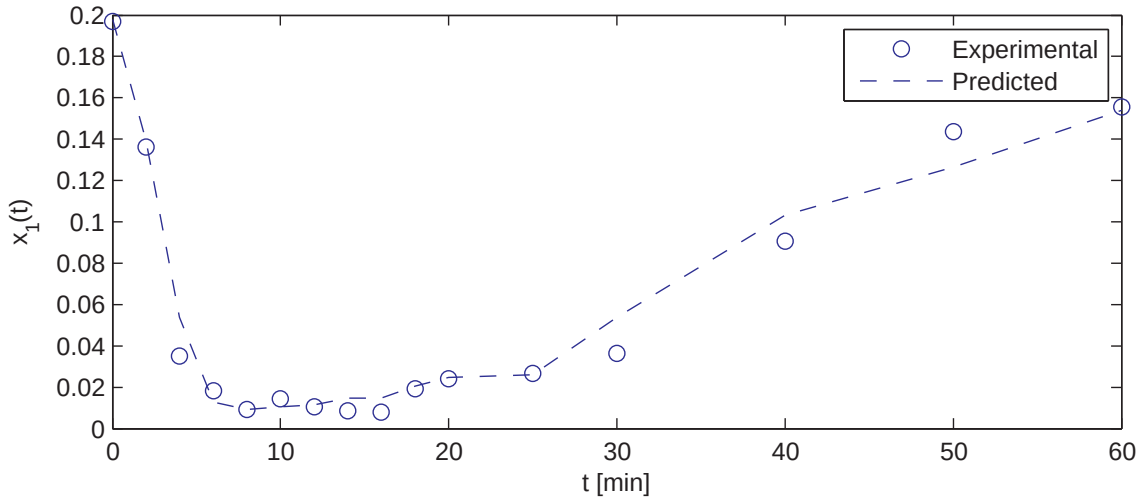

Figure 5.15: The concentration of $x_1(t)$ is computed from the experimental data of $y_1(t)$ and $y_2(t)$ is denoted with 'o' and the predicted $x_1(t)$ is shown with dotted line.

$\hat{\theta}_2 = [0.0459, 9.351, 0.355]$ (maximum *a posteriori*) and $\hat{\theta}_3 = [0.0509, 6.004, 0.460]$ (expected mean). Using these estimations, the output profiles are predicted by solving the state equations and output equations. The predicted trajectories are shown in Figure 5.16. Using the predictions, the sum of squared errors are calculated and presented in Table 5.7.

In Figure 5.16, the output profiles predicted using the different $\hat{\theta}$ vectors show similar trajectories even though $\hat{a}_3$ estimates are quite different. The scattering of the predicted trajectories is lesser in magnitude compared to the scattering of the experimental data points. Therefore, the likelihood values and the sum of squared errors, the quantitative representation of the disagreement between experimental data and the predicted data, are examined. It can be concluded that by using $\hat{\theta}_2$, in conjunction with the model equations, the experimental data are best represented. However, it is difficult to reliably estimate the confidence interval of nonlinear process parameters from a single data set.

The rather large disagreements between the three estimated values of $\hat{a}_3$ can be further analyzed by conducting sensitivity analysis of the model. If by varying this parameter value, no significant variation in predicted output can be detected, then one can conclude that the observations are insensitive to this parameter. Therefore, it can not be estimated. A given vector of parameters may be identifiable with some sets of experimental data but may become unidentifiable if the measurement noise is increased. It can also be inferred that the identifiability of process parameters is not an absolutely definable quality but depends on the quality of the data (e.g. noise variance, amount of sample points).

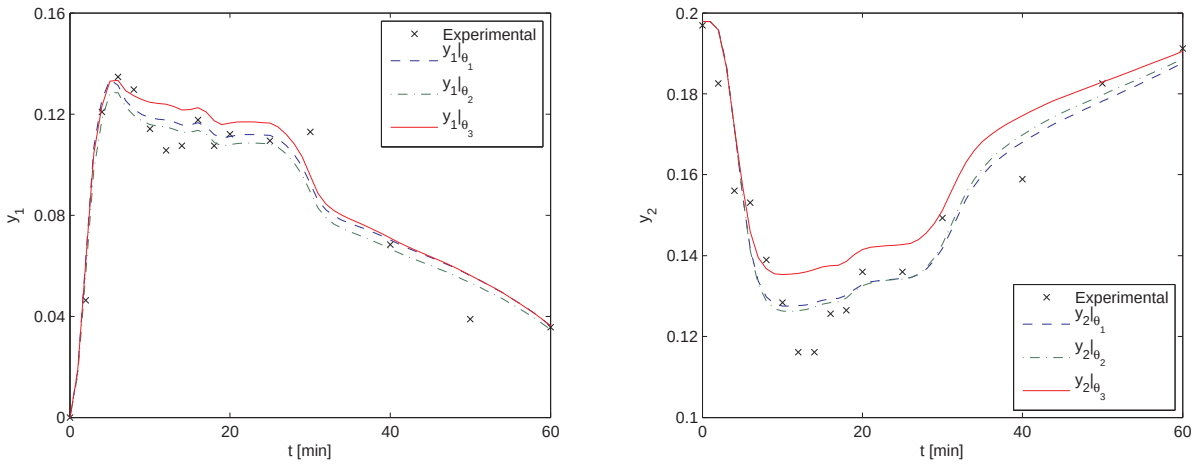

Figure 5.16: The output variables y_1 and y_2 are predicted using three different estimation of JAK-STAT process parameters; $\hat{\theta}_1 = [0.0515, 3.39, 0.35]$ (literature value, shown in blue dashed lines), $\hat{\theta}_2 = [0.0459, 9.351, 0.355]$ (maximum *a posteriori*, shown in green dotted lines) and $\hat{\theta}_3 = [0.0509, 6.004, 0.460]$ (expected mean, shown in red solid lines). The experimental data is denoted with 'x'.

Table 5.7: Likelihood values and sum of squared errors calculated for different estimates of JAK-STAT process parameters using the experimental data set. Two sets of likelihood values were computed using different noise variance, σ, 0.1 and 0.01. $\hat{\theta}_1 = [0.0515, 3.39, 0.35]$ (literature value), $\hat{\theta}_2 = [0.0459, 9.351, 0.355]$ (maximum *a posteriori*) and $\hat{\theta}_3 = [0.0509, 6.004, 0.460]$ (expected mean)

	$\hat{\theta}_1$	$\hat{\theta}_2$	$\hat{\theta}_3$
Likelihood $(\sigma = 0.1)$	1.807×10^{25}	1.923×10^{25}	1.227×10^{25}
Likelihood $(\sigma = 0.01)$	0	0	0
Sum of Squared Errors	0.0025	0.0023	0.0036

5.3.2 Quantitative Parameter Estimability and Sensitivity Analysis: Comparison with PDF

In order to study how sensitive the observed output variables are with respect to the change in parameter value, a preliminary result of sensitivity analysis is done by computing the sensitivity coefficient matrix as follows.

$$\begin{bmatrix} \frac{\partial y_1}{\partial a_1} & \frac{\partial y_1}{\partial a_3} & \frac{\partial y_1}{\partial a_4} \\ \\ \frac{\partial y_2}{\partial a_1} & \frac{\partial y_2}{\partial a_3} & \frac{\partial y_2}{\partial a_4} \end{bmatrix} \tag{5.14}$$

The partial derivatives are approximated by finite-difference method with respect to the previously reported parameter estimation, $\bar{\theta} = [\bar{a}_1, \bar{a}_3, \bar{a}_4]$.

$$\frac{\partial y_i}{\partial a_j} = \frac{y_i|_{0.90\bar{a}_j} - y_i|_{1.10\bar{a}_j}}{0.90\bar{a}_j - 1.10\bar{a}_j} \tag{5.15}$$

where $i = 1, 2$ and $j = 1, 3, 4$. The computed values are used to illustrate their relative magnitude to each other in Figure 5.17. In the top panel, the blue bars correspond to $\frac{\partial y_1}{\partial a_1}$, the green bars correspond to $\frac{\partial y_1}{\partial a_3}$ and the red bars correspond to $\frac{\partial y_1}{\partial a_4}$ at each sampling time. The bottom panel, analogously correspond to partial derivatives of y_2 with respect to the process parameters. It is shown that the partial derivatives of y_1 and y_2 with respect to a_3 are very small compared to the partial derivatives with respect to a_1 and a_4. Therefore, it is shown that varying a_3 does not affect the output as much as when a_1 or a_4 are varied. Such observation cannot be made readily when using traditional parameter estimation approaches where the distribution of parameters is assumed to be Gaussian and only the point-estimate is obtained.

Using the proposed algorithm, the entire probability distribution function is estimated, from which, the sensitivity of certain parameter can be diagnosed qualitatively through examining

the relative distribution of each parameter. This is demonstrated in Figure 5.18, which show the approximated marginal probability distribution obtained through simulated data sets of six independent experiments. The x-axis of each panel was fixed at their given order of magnitude to demonstrate the relative 'scattering' of each marginal distribution corresponding to their order of magnitude. For instance, a_1 is in the 10^{-1} order of magnitude and the corresponding panel (A) has a fixed axis at $(0, 0.1)$. To demonstrate this scattering of asymmetric probability distribution function in quantitative terms, a 'coverage-ratio' of each distribution was calculated where,

$$CR_1 = \frac{\left| \arg\max_{a_1} \hat{p}(a_1|D_1, \cdots, D_6) - \arg\min_{a_1} \hat{p}(a_1|D_1, \cdots, D_6) \right|}{10^{-1}}, \tag{5.16}$$

$$CR_3 = \frac{\left| \arg\max_{a_3} \hat{p}(a_3|D_1, \cdots, D_6) - \arg\min_{a_3} \hat{p}(a_3|D_1, \cdots, D_6) \right|}{10^{1}}, \tag{5.17}$$

$$CR_4 = \frac{\left| \arg\max_{a_4} \hat{p}(a_4|D_1, \cdots, D_6) - \arg\min_{a_4} \hat{p}(a_4|D_1, \cdots, D_6) \right|}{10^{0}}. \tag{5.18}$$

The resulting CR values are $0.263, 0.859$ and 0.125, for a_1, a_3, a_4 respectively. Thus, it is quantitatively shown that probability distribution of a_3 has the widest relative width, compared to other two parameters. This observation can be directly correlated to the smaller degree of sensitivity of a_3 shown through the sensitivity analysis. Subsequently, this conveys the fact that a_3 is not easily estimated with a reliable confidence interval with comparable degree of accuracy compared to the other two parameters.

The parameter estimability analysis is further examined by evaluating the scaled sensitivity co-

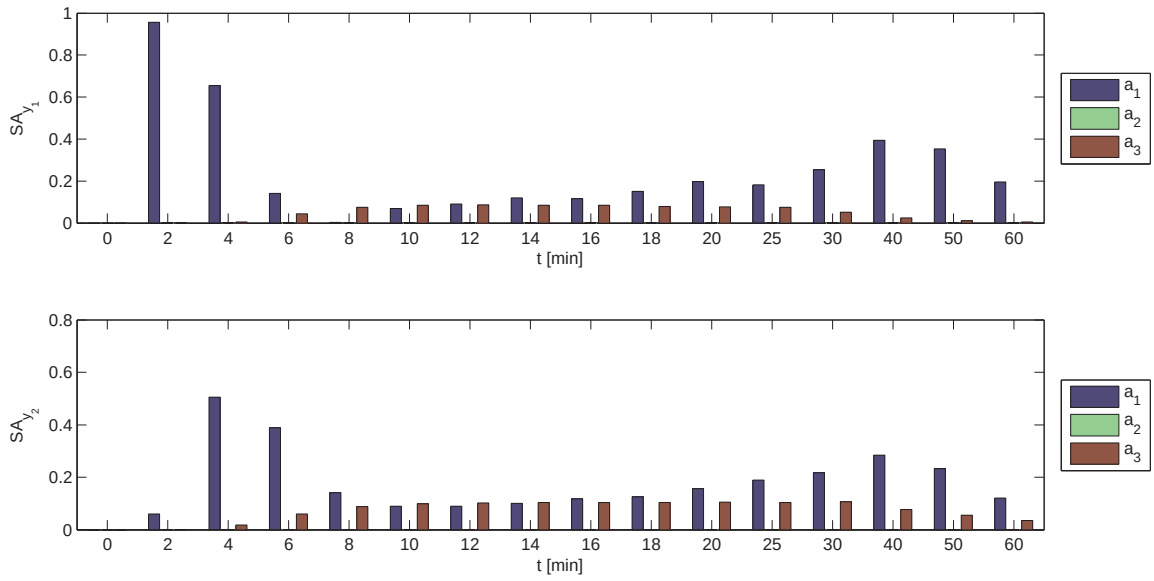

Figure 5.17: Sensitivity coefficients calculated using the finite difference method shown in (5.15). The top panel corresponds to the partial derivatives of y_1 and the bottom panel corresponds to the partial derivatives of y_2. The blue bars denote partial derivatives with respect to a_1; the green bars denote partial derivatives with respect to a_3; and the red bars denote partial derivatives with respect to a_4. The horizontal axis represents the sampling time and the vertical axis represents the value of the partial derivatives.

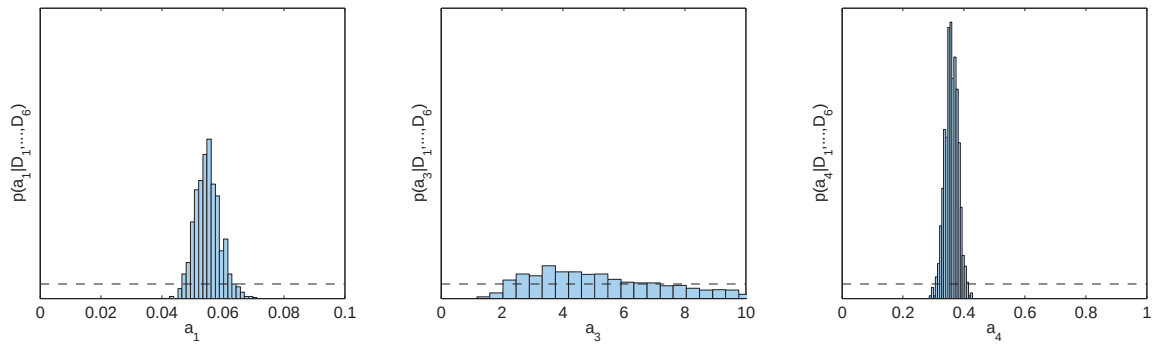

Figure 5.18: Approximated marginal distribution of JAK-STAT signal transduction pathway model parameters. The probability distributions are normalized and plotted in a window of $[0, \mathcal{O}(a_i)] \times [0, 20/\mathcal{O}(a_i)]$ where $i = 1, 3, 4$. The dotted lines represent the uniform prior distributions of a_1, a_3, a_4, $\mathcal{U}(0, 1), \mathcal{U}(0, 10), \mathcal{U}(0, 0.1)$, respectively.

efficient matrix, which is expressed as follows.

$$\bar{Z} \equiv \begin{bmatrix} \hat{Z}(0) \\ \hat{Z}(1) \\ \vdots \\ \hat{Z}(N-1) \end{bmatrix}, \tag{5.19}$$

$$\hat{Z}(c) = \begin{bmatrix} \frac{\hat{\theta}_1}{\eta_1}\frac{\partial \eta_1}{\partial \theta_1}|_{t=t_c} & \cdots & \frac{\hat{\theta}_k}{\eta_1}\frac{\partial \eta_1}{\partial \theta_k}|_{t=t_c} \\ \vdots & \ddots & \vdots \\ \frac{\hat{\theta}_1}{\eta_m}\frac{\partial \eta_m}{\partial \theta_1}|_{t=t_c} & \cdots & \frac{\hat{\theta}_k}{\eta_m}\frac{\partial \eta_m}{\partial \theta_k}|_{t=t_c} \end{bmatrix}, \tag{5.20}$$

where $\{\theta_1, \cdots, \theta_k\}$ are the model parameters; $\{\eta_1, \cdots, \eta_m\}$ are the output variables; $\{t_0, \cdots, t_{N-1}\}$ are the sampling times and $\hat{p}_i$ is either guess or literature parameter value. The rank of the orthogonalized scaled sensitivity coefficient matrix, $\bar{Z}$, is equal to the number of estimable parameters [55]. Also, the relative ease of estimability can be computed, such that the parameter corresponding to the column of $\bar{Z}$ has the largest magnitude (sum of squares of the element) is the most estimable. The rank of orthogonalized $\bar{Z}$ is equal to 3 for the JAK-STAT signal transduction pathway model, conveying that all 3 of the parameters are indeed estimable. The magnitude of $\bar{Z}$ is $M = [26.09,\ 25.02,\ 26.29]$. Such that the order is

$$M(3) > M(1) > M(2), \tag{5.21}$$

where $M(i)$ is the ith element in the vector. This rank of parameter estimability agrees with the rank of relative 'coverage-ratio' of the approximated marginal probability distribution. Such that the parameter that is the easiest to estimate have the smallest 'coverage-ratio' (have the most narrow relative distribution), which conveys the higher confidence in the parameter value.

5.3.3 Effect of Initial Conditions on the Algorithm's Performance

Within the scope of this thesis and its purpose of using the MCMC sampling methods, the Gibbs sampler has been shown to be affected by the initial value of the chain. This is similar to the gradient-search methods for parameter estimation tending to get stuck in local minima depending on the initial guess. However, MH algorithm and Gibbs sampler do not have the problem of getting 'stuck', instead the rate of convergence may be reduced so that more iterations are required in order to successfully approximate the desired target distribution. In order to study the effects of initial conditions, the JAK-STAT signal transduction pathway model parameters were estimated using four different initial conditions.

1. $\theta_0^{(a)} = [0,\ 0,\ 0]$
2. $\theta_0^{(b)} = [0,\ 0,\ 1]$
3. $\theta_0^{(c)} = [0,\ 10,\ 0]$
4. $\theta_0^{(d)} = [0,\ 10,\ 1]$

Since the order of the Gibbs iteration is '$a_1 \rightarrow a_3 \rightarrow a_4$' the starting value of the a_1 does not affect the behavior of the chain; it gets over-written during the first step of sampling from conditional distribution of a_1 given a_3, a_4 and the experimental data set. The experimental data set used in this case study is identical to the one used in previous section - published real experimental data. The initial conditions were chosen such that they all start from the far corners of the two dimensional parameter space of $[0,\ 10] \times [0,\ 1]$ where a_3 and a_4 belong to.

The algorithm was executed for 1000 iterations for all four of the chains and the moving standard deviation is calculated. The idea behind this analysis is quite simple. If the variance ([standard deviation]^2) of the samples from each individual chain and the variance of the samples from all of the chains converge to each other, this indicates that the chain has reached its steady state. In order to illustrate this point, Figure 5.19 shows the four individual standard deviation computed using each chain and also the standard deviation computed using all of the samples.

The horizontal axis indicates the iteration index at which the moving standard deviations were computed. For instance, at iteration 44, each individual moving standard deviation is computed using the chain samples from 1st iteration to 44th iteration. And for the overall moving standard deviation is computed using the chain samples from 1st to 44th iteration of all four chains (so the pool of which the standard deviation is calculated is four times larger). Each row in the figure corresponds to a_1, a_3 and a_4 respectively. The values corresponding to chain 1, 2, 3 and 4 are indicated with blue, green, red and aqua solid lines respectively, and the solid black lines indicate the overall moving standard deviation. The result of moving standard deviations is split up into two sections, where the first section corresponds to the moving standard deviations computed up to the 50th iteration, and the second section corresponds to the moving standard deviations computed from the 51st iteration to the 1000th iteration. This is because after about 30 iterations, there is hardly any differences among the moving standard deviations, and therefore the emphasis was made on the first 50 iterations. The figure indicates that regardless of the initial conditions, the chains promptly converge to their steady state value. It is predicted that similar result will be obtained, if the order of the Gibbs iteration is changed, so that a_1 is not the first random variable to be sampled, and the initial value of a_1 were to be varied. This insensitivity to the initial condition is a significant advantage to the proposed algorithm compared to the other gradient search based parameter estimation methods.

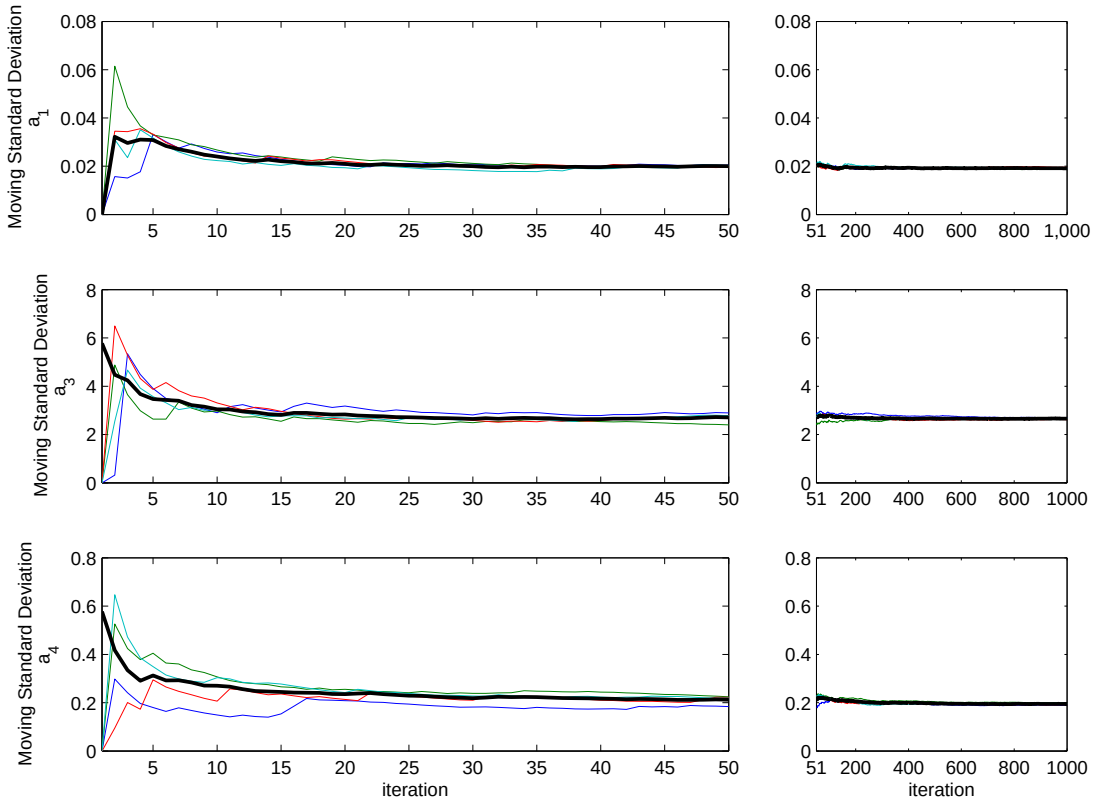

Figure 5.19: The moving standard deviations of four different Markov chains with starting initial conditions $[0,\ 0,\ 0], [0,\ 0,\ 1], [0,\ 1,\ 0]$ and $[0,\ 1,\ 1]$ are indicated with blue, green, red and aqua lines, respectively. The heavier black solid lines indicate the overall moving standard deviation of all four chains. Each row corresponds to parameter a_1, a_3 and a_4 from top row to bottom row. The first column corresponds to the moving standard deviations calculated up to 50th iterations, and the second column corresponds to the moving standard deviations calculated from 51st iterations up to 1000th iterations.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

- In order to estimate parameters of biological processes that have the tendency to yield data sets with small number of sample points and irregular sample intervals, Bayesian parameter estimation method was used.

- The Bayes Rule is adept at incorporating a priori information with the experimental data to yield posterior probability distribution of the parameter estimation. This characteristic was used to handle multiple experimental data sets and decrease the uncertainty in the estimation. Posterior distribution from each experiment was used as the prior distribution for the next estimation step. This sequential updating procedure allowed the final probability distribution of the parameters to be conditional on all of the data sets, providing a systematic method to merge information from multiple sources.

- The complex posterior distribution obtained through Bayes Rule is often analytically intractable. Thus, a random sampling method called Markov Chain Monte Carlo was used in order to numerically approximate various probability distributions. Two different instances of MCMC were implemented, Metropolis-Hastings (MH) algorithm and Gibbs Sampler.

- MH algorithm was used to approximate the univariate conditional probability distribu-

tions of process parameters. The approximated distributions were calculated under the assumption that except for a single candidate parameter, the other parameter values are conditionally known. MH algorithm was executed in conjunction with the Gibbs Sampler, and made up the inner level of iterative approximation steps. Gibbs Sampler made up the outer level of iterative approximation steps. These two MCMC sampling methods, implemented together, approximated the multi-dimensional probability distributions of nonlinear process parameters.

- The confidence interval of the parameter estimates is obtained straightforwardly from the full probability distribution. The overall shape of the distribution conveyed that the common Gaussian distribution assumption of nonlinear process parameters is incorrect.

- The shape of the full probability distribution agrees well with conventional sensitivity analysis, and estimability analysis, thus providing an alternative framework for analyzing nonlinear processes.

- The choice of initial guess of the parameters did not affect the performance of the proposed algorithm as heavily as it does on conventional parameter estimation method such as Maximum Likelihood Estimation or Nonlinear Regression.

- The high computational cost of MCMC was reduced by implementing multi-phase estimation of the Gibbs Sampler. As some Markov chains corresponding to some of the parameters in the parameter vector converge faster to their steady-state, these chains were removed from the inner level of iterations. Each parameter removed from the estimation procedure resulted in a reduction of computational cost by $1/m \times 100\%$.[10]

[10] m is the length of the parameter vector

6.2 Future Work

6.2.1 Further Investigation of the Algorithm

In this thesis, a heuristic approach of estimating the probability distribution of nonlinear process parameters using Gibbs Sampler in different phases was discussed. This approach is successful in reducing the large computational cost when there is a large discrepancy between the rates of convergence among the parameters. However, there are other directions that a further research endeavor may explore to address this discrepancy. For instance, the order of the parameters sampled using the Gibbs Sampler in the case study of Batch fermentation reaction model was fixed at $\mu_m \rightarrow k_s \rightarrow k_P \rightarrow Y_{XS} \rightarrow Y_{PX}$. It is interesting to note that the two of the parameters that are notoriously harder to estimate are placed in the earlier position, and one can investigate whether putting the parameters with higher estimability before the k_s and k_P will result in smaller uncertainty in the estimated parameters. A preliminary study shows that a k_P value that maximizes the likelihood function of the model is sensitive to the variation in the other parameters. This is in contrast to the parameter μ_m where the likelihood maximizing value of this parameters does not show wide variation when the parameters of other values are changed, including k_P. Therefore, if these parameters that are sensitive to the other parameters' variations are placed in the latter part of the sampling, it will allow the other parameters to establish relatively more accurate positions in the parameter space, thus resulting in more accurate estimation of the sensitive-to-others parameters.

6.2.2 Experiment Design

The uncertainty of the parameter estimates are conveyed by the probability distribution estimated through the approach discussed in this thesis. The quality of the data sets used in the case studies was poor due to their irregular sampling and small number of data points. From studying the variance of the estimated distribution and the quality of the data, one can derive a correlation between the two, by coming up with quantitative criteria for irregularity and sparsity of data

sets, and the normalized variance of the distributions of the parameters. If the total number of samples during an experiment is known *a priori*, then it is conceivable to think of developing an optimal sampling scheme to obtain a "desired" parameter distribution. This is part of our future work.

Bibliography

[1] U. Alon. *An Introductionto Systems Biology*. Chapman and Hall/CRC, 2007.

[2] C. Andrieu, N. De Freitas, A. Doucet, and M. I. Jordan. An Introduction to MCMC for Machine Learning. *Machine Learning*, 50:5–43, 2003.

[3] M. Baltes, R. Schneider, C. Sturm, and M. Reuss. Optimal experimental design for parameter estimation in unstructured growth models. *Biotechnology Progress*, 10(5):480–488, 1994.

[4] Y. Bard. *Nonlinear Parameter Estimation*. Academic Press, 1974.

[5] N. Barkai and S. Leibler. Robustness in Simple Biochemical Networks. *Nature*, 387:913–917, 1997.

[6] J. Beck and K. Arnold. *Parameter Estimation in Engineering and Science*. John Wiley, 1977.

[7] H. Bohr, J. Bohr, S. Brunak, R. M. J. Cotterill, H. Fredholm, B. Lautrup, and S. B. Petersen. A novel approach to prediction of the 3-dimensional structure of protein backbones by neural networks. *FEBS Letters*, 261(1):43–46, 1990.

[8] R. B. Burrows, G. R. Warnes, and R. C. Hanumara. Statistical Modelling of Biochemical Pathways. *IET Systems Biology*, 2007.

[9] Y. Cai, X. Liu, X. Xu, and K. Chou. Artificial neural network method for predicting protein secondary structure content. *Computers and Chemistry*, 26:347–350, 2002.

[10] J. Cao and H. Zhao. Estimating Dynamic Models for Gene Regulation Networks. *Bioinformatics*, 24:1619–1624, 2008.

[11] G. Casella and R. L. Berger. *Statistical Inference*. Brooks/Cole Publishing Company, Pacific Grove, California, 1990.

[12] C. Chen. *Linear System Theory and Design*. Holt, Rinehart and Winston, 1984.

[13] W. Chen, B. R. Bakshi., P. K. Goel, and S. Ungarala. Bayesian Estimation via Sequential Monte Carlo Sampling - Unconstrained Nonlinear Dynamical Systems. *Industrial and Engineering Chemistry Research*, 43:4012–4025, 2004.

[14] S. Chib and E. Greenberg. Understanding the Metropolis-Hastings Algorithm. *The American Statistician*, 49(4):327–335, November 1995.

[15] M. C. Coleman and D. E. Block. Bayesian Parameter Estimation with Informative Priors for Nonlinear Systems. *AIChE Journal*, 52(2):651–667, February 2006.

[16] P. Englezos and N. Kalogerakis. *Applied Parameter Estimation for Chemical Engineers*. Marcel Dekker, Inc., 2001.

[17] G. Fishman. *Monte Carlo: Concepts, Algorithms, and Applications*. Springer-Verlag, 1996.

[18] K. Fukushima. Cognitron: A self-organizing multilayered neural network . *Biological Cybernetics*, 20(3-4), 1975.

[19] A. R. Gallant. *Nonlinear Statistical Models*. John Wiley, 1987.

[20] A. P. Gasch, P. T. Spellman, C. M. Kao, O. Carmel-Harel, M. B. Eisen, G. Storz, D. Botstein, and P. O. Brown. Genomic Expression Programs in the Response of Yeast Cells to Environmental Changes. *Molecular Biology of the Cell*, 11:4241–4257, 2000.

[21] A. E. Gelfand, S. E. Hills, A. Racine-Poon, and A. F. M. Smith. Illustration of Bayesian Inference in Normal Data Models Using Gibbs Sampling. *Journal of American Statistical Associtation*, 85:972–985, 1990.

[22] A. E. Gelfand and A. F. M. Smith. Sampling-Based Approaches to Calculating Marginal Densities. *Journal of American Statistical Associtation*, 85:398–409, 1990.

[23] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1984.

[24] G. Grimmett and D. Stirzaker. *Probability and Random Processes*. Oxford University Press, 2001.

[25] R. R. Gupta and L. Achenie. A network model for gene regulation. *Computers and Chemical Engineering*, 31:950–961, 2006.

[26] W. K. Hastings. Monte Carlo Sampling Methods using Markov Chains and Their Applications. *Biometrika*, 57:97–109, 1970.

[27] A. Holmberg. On the practical idenetifiability of microbial growth models incorporating michaelis-menten type nonlinearities. *Mathematical Biosciences*, 62:23–43, 1982.

[28] B. Juang and L. Rabiner. An Introduction to hidden Markov models. *ASSP Magazine, IEEE*, 3(1):4–16, 1985.

[29] G. Kimmel and R. Shamir. A Block-Free Hidden Markov Model for Genotypes and Its Application to Disease Association. *Journal of Computational Biology*, 12(10):1243–1260, 2005.

[30] S. Kirkpatrick, C. Gelatt Jr., and M. Vecchi. Optimization by simulated annealing. *Science*, 220:671–679, May 1983.

[31] W. R. Kolk and R. A. Lerman. *Nonlinear System Dynamics*. Van Nostrand Reinhold, 1992.

[32] A. Krogh, M. Brown, I. Mian, K. Sjolander, and D. Haussler. Hidden Markov Models in Computational Biology. *J. Mol. Biol.*, 235:1501–1531, 1994.

[33] J. Lackie. *The Dicitonary of Cell and Molecular Biology*. Academic Press, 2007.

[34] S. Ö. Laursena, D. Webba, and W. F. Ramirez. Dynamic hybrid neural network model of an industrial fed-batch fermentation process to produce foreign protein. *Computers and Chemical Engineering*, 2006.

[35] L. Ljung. *System Identification: Theory for the User*. Prentice Hall, Upper Saddle River, NJ, 2nd edition, 1999.

[36] S. Mangan and U. Alon. Structure and function of the feed-forward loop network motif. *PNAS*, 100(21):11980–11985, 2003.

[37] I. A. Maraziotis, A. Dragomir, and A. Bezerianos. Gene networks reconstruction and time-series prediction from microarray data using recurrent neural fuzzy networks. *IET Systems Biology*, 1(1):41–50, 2007.

[38] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of State Calculations by Fast Computing Machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.

[39] I. J. Myung. Tutorial on maximum likelihood estimation. *Journal of Mathematical Psychology*, 47:90–100, 2003.

[40] R. M. Neal. Probabilistic Inference Using Markov Chain Monte Carlo Methods. Technical report, Department of Computer Science, University of Toronto, 1993.

[41] P. Van Laarhoven and E. Arts. *Simulated Annealing: Theory and applications*. Amsterdam: Reidel Publishers, 1987.

[42] M. Quach, N. Brunel, and F. d'Alche Buc. Estimating parameters and hidden variables in

non-linear state-space models based on ODEs for biological networks inference. *Bioinformatics*, 23(23):3209–3216, 2007.

[43] L. R. Rabiner. A tutorial on Hidden Markov Models and selected applications in speech recognition. In *Proceedings of the IEEE*, 1989.

[44] A. E. Raftery and S. Lewis. How Many iterations in the Gibbs Sampler? April 10, 1991; revised September 13, 1991, 1991.

[45] W. J. Rugh. *Nonlinear System Theory. The Volterra/Wiener Approach*. The Johns Hopkins University Press, 1981.

[46] G. Seber and C. Wild. *Nonlinear Regression*. John Wiley, 1989.

[47] S. S. Shen-Orr, R. Milo, S. Mangan, and U. Alon. Network motifs in the transcriptional regulation network of Escherichia coli. *Nature Genetics*, 31, May 2002.

[48] A. Siepel and D. Haussler. Combining Phylogenetic and Hidden Markov Models in Biosequence Analysis. *Journal of Computational Biology*, 11:413–428, 2004.

[49] D. Sivia. *Data Analysis : A Bayesian Tutorial*. Clarendon Press, Oxford, 1996.

[50] T.S. Söderström and P. Stoica. *System Identification*. Prentice Hall International, 1989.

[51] I. Swameye, T. G. Muller, J. TImmer, O. Sandra, and U. Klingmuller. Identification of Nucleocytoplasmic Cycling as a Remote Sensor in Cellular Signaling by Databased Modeling. *Proceedings of the National Academy of Sciences of the United States of America*, 100(3):1028–1033, 2003.

[52] J. Tomshine and Y. N. Kaznessis. Optimization of a Stochastically Simulated Gene Network Model via Simulated Annealing. *Biophysical Journal*, 91:3196–3205, 2006.

[53] J. V. Tu. Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. *Journal of Clinical Epidemiology*, 49(11):1225–1231, November 1996.

[54] J. Wu and J. Xie. Computation-Based Discovery of Cis-Regulatory Modules by Hidden Markov Model. *Journal of Computational Biology*, 15(3):279–290, 2008.

[55] K. Yao, B. Shaw, B. Kou, K. McAulley, and D. Bacon. Modeling Ethylene/Butene Copolymerization with Multi-site Catalysts: Parameter Estimability and Experiment Design. *Polymer Reaction Engineering*, 11(3):563–588, 2003.

[56] J. Zhang. Batch-to-batch optimal control of a batch polymerisation process based on stacked neural network models. *Chemical Engineering Science*, 63:1273–1281, 2008.

[57] Z. Zi and E. Klipp. SBML-PET: a Systems Biology Markup Language-based parameter estimation tool. *Bioinformatics*, 22(21):2704–2705, 2006.

Appendix A

Experimental Data Simulation

Simulating the experimental data of nonlinear process with irregular sampling time starts with the time vector T, where $0 \leq T \leq T_{\max}$. In order to simulate the irregular sampling time, $T \in \mathrm{R}^N$, MATLAB® 'RAND' function is used, where N is the number of samples.

Program A.1 MATLAB program that simulates nonlinear process experimental data with irregular sampling time. 'k' = number of experiments, 'theta' = parameter vector, 'N' = number of samples, 'T' = sampling time, 'init' = initial condition of the states, 'sigma' = white noise standard deviation, 'Tmax' = $T_{\max}$.

```
global N T theta sigma

for i = 1:k
    T    = sort([0;rand(N-1,1)]*Tmax);
    sol  = ode45(@(t,x) MODEL_ode(t,x,theta), [0 Tmax], init);
    states = deval(sol,T)';
    y = states + randn(size(states))*sigma;
    ExpData(:,:,i) = y;
end
```

'MODEL_ode' is a user-defined MATLAB function that handles the right side of the differential equation. Following are the functions for batch fermentation model, FFL model and JAK-STAT signal transduction pathway model, respectively.

Program A.2 MATLAB function that handles the ordinary differential equations of batch fermentation model.

```
function dxdt = MODEL_ode(t,x,theta)
    mu = theta(1)*x(2)/(theta(2)+x(2))*(1-x(3)/theta(3));
    dxdt = [mu*x(1);
     -mu*x(1)/theta(4);
    theta(5)*mu*x(1)];
end
```

Program A.3 MATLAB function that handles the ordinary differential equations of Feed-Forward Loop genetic regulation network model.

```
function dxdt = MODEL_ode(t,x,theta)
      B_y = 1;
      B_z = 1;
      H = 2;
      dxdt = [-theta(1)*y(1) + ...
                        B_y*((FFL_X_real(t)/theta(3)).^H)/...
                        (1+(FFL_X_real(t)/theta(3)).^H);
              -theta(2)*y(2) + ...
                        B_z*((FFL_X_real(t)/theta(4)).^H)/...
                        (1+(FFL_X_real(t)/theta(4).^H);
              ((y(1)/theta(5)).^H)/(1+y(1)/theta(5)).^H)];
end
```

Program A.4 MATLAB function that handles the ordinary differential equations of JAK-STAT signal pathway model.

```
function dxdt = JAK_STAT_ode(t,x,theta)

global tao;

if (t<tao)
dxdt = [ -theta(1)* x(1)* EpoR(t)*60 ;
         theta(1)* x(1)* EpoR(t)*60 - 2*x(2)^2;
         -theta(2)*x(3) + x(2)^2;
         theta(2)*x(3)
];
else
    dxdt = [ -theta(1)* x(1)* EpoR(t)*60...
                + 2*theta(3)*x(4);
         theta(1)* x(1)* EpoR(t)*60 - 2*x(2)^2;
         -theta(2)*x(3) + x(2)*x(2);
         theta(2)*x(3) - theta(3)*x(4)
];
end
```

The simulated data sets of each model are shown in Figure A.1, Figure A.2 and Figure A.3, respectively.

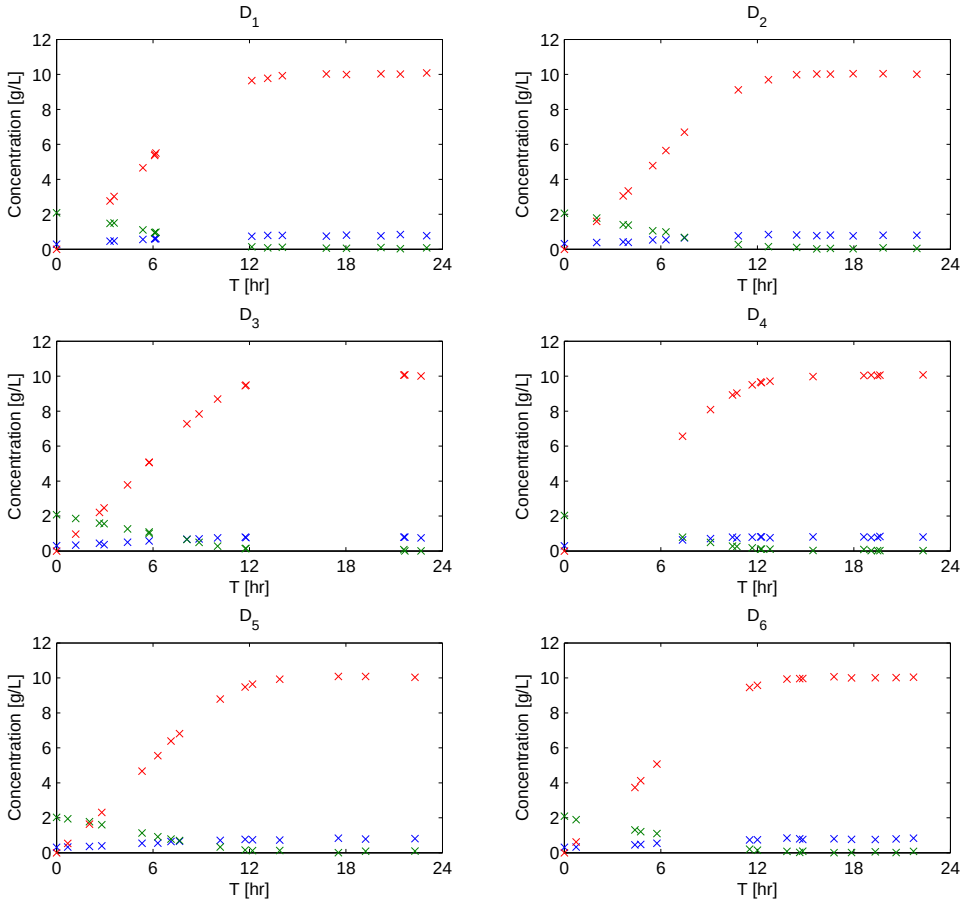

Figure A.1: Simulated data sets of Batch fermentation reaction model used in this thesis. The parameter vector value used for simulation is $\theta = [0.15,\ 0.50,\ 0.25,\ 0.25,\ 0.20]$. The blue 'x's correspond to $C_X(t)$, the green 'x's correspond to $C_S(t)$, and the red 'x's correspond to $C_P(t)$.

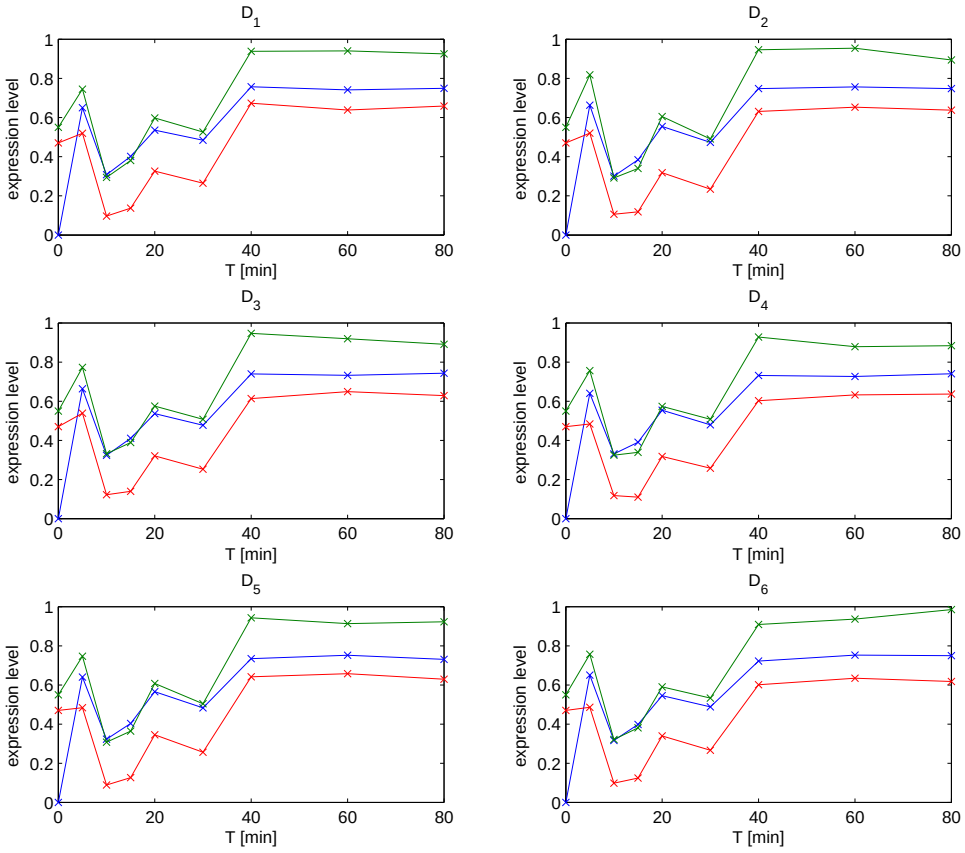

Figure A.2: Simulated data sets of Feed-Forward Loop genetic regulatory network model used in this thesis. The parameter vector value used for simulation is $\theta = [0.44,\ 0.69,\ 0.90,\ 0.60,\ 0.56]$. The blue lines correspond to $X(t)$, the green lines correspond to $Y(t)$, and the red lines correspond to $Z(t)$.

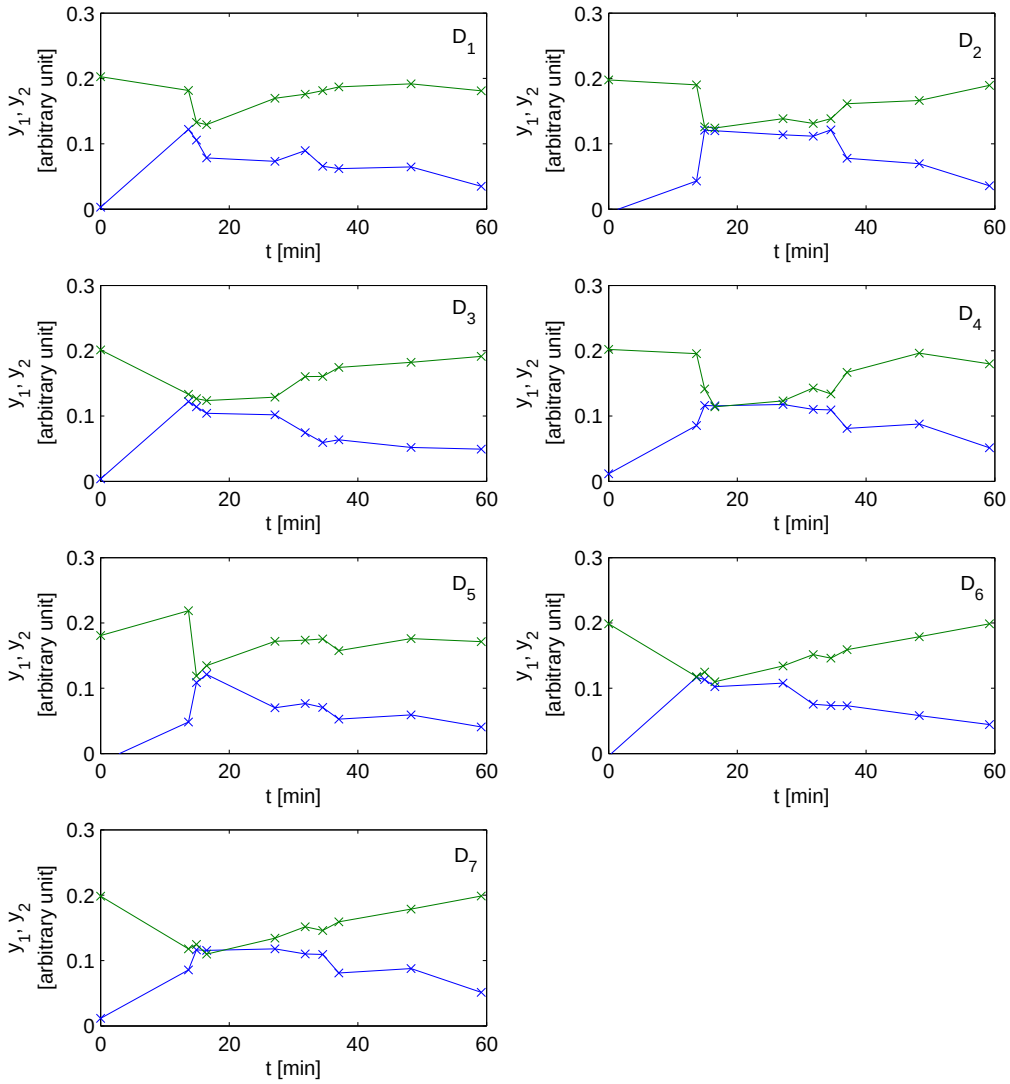

Figure A.3: Simulated data sets of JAK-STAT signal transduction pathway model used in this thesis. The parameter vector value used for simulation is $\theta = [0.0515,\ 3.39,\ 0.35]$. The blue lines correspond to $y_1(t)$, and the green lines correspond to $y_2(t)$. The first six sets, $D_1, D_2, D_3, D_4, D_5, D_6$, were used to illustrate the proposed algorithm and the last data set, D_7, was used to verify the estimated values.

Appendix B

Derivation of Likelihood Function for Nonlinear Dynamic Process

The likelihood function of a nonlinear dynamical model is obtained from assuming that the states are deterministic variables and that the measurement noise is Normally distributed [35]. For instance, the probability of observing the first experimental data set, D_1, given certain values for θ is desired, then the likelihood function is expressed as follows.

$$L(\theta \,|\, D_1) = p(y_1(t_0,\theta),\cdots,y_1(t_{N-1},\theta),\cdots,y_q(t_0),\cdots,y_q(t_{N-1},\theta)\,|\,\theta) \quad \text{(B.1)}$$

$$= \prod_{i=0}^{N-1}\prod_{j=1}^{q} p(y_j(t_i,\theta)\,|\,\theta) \quad \text{(B.2)}$$

The second equality follows from assuming that the measurements are independent from each other and thus the joint probability of all the measurements in D_1 is the product of individual probability of each measurement point. Then, using the output variable equations, the conditional probability of individual measurement point is expressed as follows

$$p(y_j(t_i,\theta)\,|\,\theta) = p(\eta_j(t_i)\,|\,\theta) \quad \text{(B.3)}$$

$$= \frac{1}{\sqrt{2\pi}\,\sigma_j}\exp\left(-\frac{\eta_j|_\theta(t_i)^2}{2\sigma_1^2}\right) \quad \text{(B.4)}$$

$$= \frac{1}{\sqrt{2\pi}\,\sigma_j}\exp\left(-\frac{(y_j|(t_i,\theta)-y_j(t_i))^2}{2\sigma_1^2}\right) \quad \text{(B.5)}$$

The second equality follows because measurement noise, η_1 is assumed to be Normally distributed with zero mean and variance σ_1^2. In the third equality, $y_j|_\theta(t_i)$ is the predicted output variable at time t_i given the θ value. The above expression defines the likelihood function of

a single measurement data of y_j at time t_i as a function of θ (obtained by solving the ordinary differential equations of the model). Substituting the output equations into (B.2), the likelihood, the following likelihood function of experimental data D was obtained.

$$L(\theta \mid D) \quad = \quad \frac{1}{(2\pi)^{m(N-1)/2} \prod_{j=1}^{q} \sigma_q^{N-1}} \times \exp\left(\sum_{i=0}^{N-1} \sum_{j=1}^{q} -\frac{(y_j(t_i,\theta) - y_j(t_i))^2}{2\sigma_j^2} \right)$$