UNIVERSITY OF SOUTHAMPTON

Probabilistic Inference in Models of Systems Biology

by

Xin Liu

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy

in the
Faculty of Engineering and Applied Science
Department of Electronics and Computer Science

December 2014

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING AND APPLIED SCIENCE
DEPARTMENT OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

by Xin Liu

In Systems Biology, it is usual to use a set of ordinary differential equations to characterize biological function at a system level. The parameters in these equations generally reflect the reaction or decay rates of a molecular species, while states characterize the concentration values of species of interest, e.g. mRNA, proteins and metabolites. Often parameter values are estimated from *in vitro* experiments which may not be true reflections of the *in vivo* environments. With internal states, some may not be accessible for experimental measurement. Hence there is interest in estimating parameter values and states from noisy or incomplete observations taken at inputs/outputs of a system. This thesis explores several probabilistic inference approaches to do this.

The study starts from a thorough investigation of the effectivenesses of the most commonly used one-pass inference methods, from which the non-parametric particle filtering approach is shown to be the most powerful method in the sequential category. After this study, the family of Approximate Bayesian Computation (ABC) methods, also known as likelihood-free batch approach, is reviewed chronologically and its advantages and deficiencies are summarized via a statistical toy example and two biological models. Additionally, a novel ABC method coupled with the sensitivity analysis technique has been developed and demonstrated on three periodic and one transient biological models. This approach has the potential to solve problem in high dimension by selectively allocating computational budget. In order to assess the capability of the proposed method in real-world problems, we have modeled the polymer pathway and conducted quantitative analysis via the proposed inference approach.

Contents

List of Figures

List of Tables

Acknowledgements

I would like to gratefully acknowledge my supervisor Prof. Mahesan Niranjan. My PhD training has benefited enormously from his expertise, enthusiasm and encouragement. His excitement and commitment to research set a strong example that I wish to achieve in academia.

I would like to thank to Dr. Ipsita Roy , Dr. Srinandan Dasmahapatra and Dr. Bing Chu for the regular advice and ideas. Their contributions have brought this thesis in a higher level.

I have been very fortunate to work with the group members in the CSPC group at the University of Southampton. I am benefited hugely from the discussions with Dr. Ke Yuan, Tayyaba Azim and Tim Matthews.

I would like to thank my colleagues in the groups, Yawwani Gunawardana, Chathurika Dharmagunawardhana Abdullah Alrajeh, Jianhao Xiong, Xiaoru Sun, Dr. Bassam Farran, Dr, Ali Hassan, Dr. Mustansa Ali Ghazanfar, Dr. Sung Uk Jung, Wangmu Liu, Dr. Yizhao Ni, Dr. Amirthalingam Ramanan and Dr. Daisy Tong who offered me their kindness helps during my PhD.

Special thanks to my girlfriend Bohui Zou for accompanying me through these tough days. I would like to thank my parents for their patience and support help me on every step that I made.

To my parents Wenlin Liu and Feng Lin and my girlfriend Bohui Zou

Chapter 1

Introduction

1.1 Problem statement

This thesis is about parameter estimation and inference from computational models of biological systems. In analyzing biological function at a system level, rather than at individual component level, we ofter use descriptions based on simultaneous ordinary differential equations. These equations contain parameters (e.g, decay rate or reaction rate of a molecular species) and states (concentrations of species of interest, e.g. mRNA, proteins and metabolites). Some of these parameters are estimated from knowledge of the reaction or *in vitro* experiments. Sometimes, the interest lies in measuring the concentrations of reaction intermediates. With experimental methods it is not always possible to either measure parameters *in vivo* settings or to observe all intermediates produced in complex biological phenomena. Hence, it is of interest to ask if we can derive computational methods by which parameters and states of a system can be accurately inferred from partial data, usually in some input-output forms. Additionally, measurements of biological variables are inherently noisy due to variabilities in experimental conditions as well as noise in instrumentation. Consequently, it is often desirable to quantify the uncertainty associated with any estimate made, either of the parameters or of the state variables.

In this thesis we study computational models of systems biology for state and parameter estimation using probabilistic inference methods. As many of these models/systems are dynamic in nature, the main focus of our work is within the state-space modeling formalism, consisting of continuous dynamical processes from which noisy observations are made at discrete points in time (Liu and Niranjan, 2012). The work due to Sitz et al. (2002) is a starting point for this formulation and subsequent algorithmic settings, in which we numerically integrate the underlying ODE system between time points where observations are made, assuming this part is deterministic. A particular example of using this formulation in systems biology is the work of Lillacci and Khammash (2010),

who used an extended Kalman filter (EKF) to analyze the heat shock response system of *E. coli*.

Advanced computational methodologies centered around the idea of Approximate Bayesian Computation (ABC) have attracted much interest since their first development in population genetics (Tavaré et al., 1997; Pritchard et al., 1999). They have been shown to be useful in systems biology for precisely the same problems we wish to address. The nature of the underlying biological system, and the way different molecular interactions appear as terms in the system of ODEs, cause a number of difficulties in applications of these type of models. The reproducibility of results claimed in the literature is sometimes problematic for this reason. How parameters cause such difficulties has been explored by Gutenkunst et al. (2007), who analyze a number of systems biology models and suggest that the set of parameters can be decomposed into having sloppy and stiff properties with respect to the behavior of the system. Meanwhile the ABC algorithms have advanced significantly in the probabilistic inference community including methods for adaptively setting convergence thresholds (Del Moral et al., 2012). In this dissertation we thoroughly explore the modern ABC algorithms, and their applicability to parameter estimation and inference in systems biology models, particularly in light of Gutenkunst et al. (2007)'s work on parameter behavior (stiff and sloppy). We address this issue by sensitivity analysis (Saltelli et al., 1999; Zi, 2011).

Finally, analysis of this type is only possible after a biological system has been described by ODEs. This requires carefully listing the molecular species involved in the system, and writing down the biochemical equations describing their interactions. In this thesis, we study a polymer production system by modeling the pathway with three differential equations and twenty parameters. Quantitative analysis including sensitivity analysis, parameter estimation and dependence of production on the external supply is carried out. The reliability of results is verified biologically by collaboratively work with Dr. Ipsita Roy from University of Westminster. However, due to the complexity of subpathway and insufficient experimental data, the process is still in progress. The model presented in this dissertation is part of early work towards modeling a dual polymer production pathway, in which two polymers will be simultaneously produced via a series of chemical reactions, e.g. energy releasing TCA cycle and glycolysis.

1.2 Contributions

Following are the four main contributions of this thesis:

- We have concluded the particle filter (PF) is more powerful than extended and unscented Kalman filters. This is done by initializing the PF with parameters far from their true values, while providing Kalman filters with a highly favorable

initialization which is the closest to the truth among all the particles. Moreover, comparisons are also carried out between the sequential and batch methods in terms of computational cost, accuracy and ability to handle noise. A paper based on this contribution is published as Liu and Niranjan (2012).

- A collection of Approximate Bayesian Computation (ABC) methods, also known as likelihood-free methods, has been studied theoretically and empirically. We track their chronological developments, and their mathematical derivations are also given. Moreover, the advantages or the deficiencies of methods are demonstrated by simple biological or statistical examples. Features of all considered approaches are summarized. A review paper has been submitted to *IEEE Transaction on Computational Biology and Bioinformatics* based on this work.

- We have developed a new method for ABC coupled with sensitivity analysis technique. Relying on the importance of parameters to the system, the computational budget is selectively partitioned so that the crucial parameters can be estimated with high precision. The effectiveness of the proposed method is demonstrated on three oscillatory models and one transient model taken from the systems biology literature. A submission to *PLoS Computational Biology* was made based on this work.

- We have developed a model for the polymer pathway and conducted the quantitative analysis including sensitivity analysis and parameter estimation based on the mathematical expressions. Moreover, we simulate the behavioral response to the different concentration levels of external fed species and substrate. The reliability of investigation is verified biologically by our collaborator Dr. Ipsita Roy, and a manuscript is preparing based on the achieved results.

1.3 Thesis organisation

The reminder of this thesis is organized as follows. Chapter 2 gives a literature review of biological systems modeling, state-space models, Bayesian inference and sensitivity analysis techniques. Chapter 3 reviews the popular ABC methods both theoretically and empirically. Chapter 4 presents a new ABC method to tackle issues with existing ABC methods. In Chapter 5, we illustrate the modeling of a biochemical system and conduct a quantitative investigation. Finally, Chapter 6 concludes the thesis, and points out avenues for future work.

Chapter 2

Literature Review

In this chapter, we introduce some biological systems to be worked with. Additionally, the state-of-the-art inference and learning algorithms under the state-space models framework are discussed.

2.1 Systems biology

Biological systems are interpreted via studying mechanisms of their components such as cells, molecules, proteins or mRNA. Quite often, the living organism of interest is so complicated that conducting experiments on it is impossible. Studies are therefore instead carried out on individual components of the organism that have been isolated from their biological environments, and an experimental setup of this kind is called *in vitro*. In this setting, measurements may be inconsistent with those that are from studies being conducted on the unified whole, namely *in vivo*.

This limitation motivated the development of a systematic strategy to interpret the biological problems. Systems Biology is an inter-disciplinary study including mathematical modeling, quantitative analysis and experimental validation motivated towards understanding biological problems at the system level.

The precursors of this field are Hodgkin and Huxley (1952) who developed a mathematical model for characterizing a dynamic movement of a neuronal cell along the axon. The work due to Noble (1960) is also seen as the foundation, in which Noble (1960) derived a computer solvable model of the heart pacemaker. This model has been verified by a huge amount of experimental data, they are therefore regarded as the pioneers of systems biology. Von Bertalanffy (1968) later proposed the systems theory which elucidates the definition of systems biology.

Insight into kinetic mechanism such as Michaelis-Menten kinetics (Michaelis and Menten, 1913) and enzyme-catalyzed kinetics (Roberts, 1977) is an important aspect of modeling

in Systems Biology. Particularly, the development of Systems Biology Markup Language (SBML) operating as a representation format provides a standard way to describe mathematical formulas in computational biology. This 'universal' language enables software to easily interpret models independently of their representations, and saves the cost for communicating between modeler and software engineer.

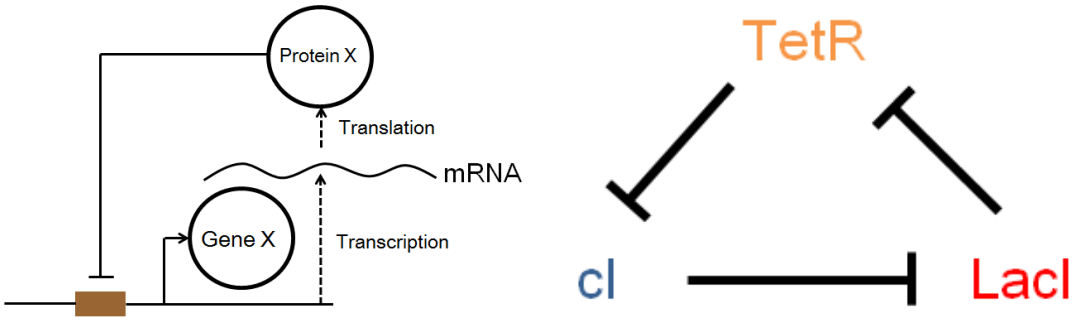
In addition, advanced technology platforms such as genomics, metabolomics and proteomics provide large quantities of high quality biological data. Consequently, the cross-fertilization of the quantitative expression, the experimental data and the statistical tools greatly aids interpretation of the biological problems of interest. For instance, the Physiome Project, a worldwide collaboration, is intended to deepen insight into human physiology. This project's efforts are targeted towards databasing the functional behavior of whole organism, and the development of integrated quantitative and descriptive modeling.

2.2 Biological systems

2.2.1 Gene regulatory network

In biological systems, genes and proteins are known to regulate each other, either along signalling pathways or in combinatorially acting to achieve selective actions (either in space or in time). Genes and transcription regulatory networks define how such regulation takes place. A particular example of such regulation is feedback known as auto-regulation (Alon, 2006), shown in Figure 2.1(a). In this simple auto-regulatory gene network, the DNA sequence is transcribed to produce the corresponding mRNA, which is then translated into protein. Quite generally, in auto-regulation, the concentration of the resulting protein influences the degree to which the gene is transcribed. As a result of this, the circuit is able to achieve faster transient response of how much protein is needed in the cell. Another example is the repressilator network shown in Figure 2.1(b) in which three genes mutually negatively regulate the activities of each other. LacI, the first repressor gene initially inhibits the second repressor gene, TetR. The product of the TetR protein decreases the expression of cI which is the third gene in the system. A closed loop is ultimately formed, as cI inhibits LacI.

In systems biology, these kinds of reaction are quantitatively captured by a set of ordinary differential equations (ODEs) and solutions of which help explain and interpret the cellular behaviors of systems at a system level. In order to strengthen the interpretation, several synthetic biological systems have been constructed, one of which is call the repressilator system and it is suggested to mimic the periodic oscillations by six differential equations associated with four parameters (Elowitz and Leibler, 2000). These equations are used to describe the change in species concentration with respect to



(a) (b)

FIGURE 2.1: (a): An illustration of a gene autoregulatory network (Alon, 2006). Protein X is the product of gene X, which binds a site in its promoter and acts as the repressor of the own transcription. (b): Example of gene regulatory network representing a cyclic loop for repressilator system.

time in which parameters reflect the transcription rates of proteins and the translation rates/decay rates of mRNA. We use this circuit as one of the examples in this dissertation and provide these mathematical descriptions in chapter 4 as equations (4.12) - (4.17).

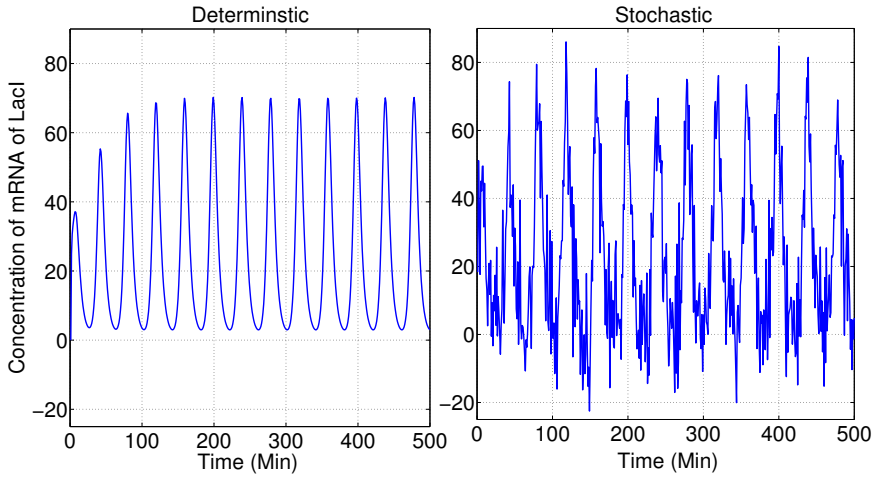


FIGURE 2.2: Concentration of mRNA in LacI gene, synthesized from the deterministic and stochastic models of the repressilator system, where the schematic graph of the repressilator system is given as Figure 2.1(b) and this system is also considered as an example for assessing the performance of inference approaches presented in chapter 4.

The behavior of the mRNA of LacI being captured by the deterministic repressilator model is shown in the left-hand side of Figure 2.2. In the graph, periodic repetition every 50 minutes is clearly observed and the behavioral uncertainty is completely ruled out in the simulation. McAdams and Arkin (1999); Elowitz and Leibler (2000) increased the realism of the model by proposing a stochastic repressilator model that takes into account the randomness of evolution which may be caused by noise in instrumentation or variability in experimental conditions. The simulation of this stochastic repressilator model is given in the right-hand side of Figure 2.2, in which the noise of system output

is evident.

If the system to be modeled has no randomness involved in its evolution, it is more appropriate to formulate the system deterministically. However, a few processes such as the stock market and medical data appear stochastic in time sequence, and the future states can evolve in various ways in comparison to the deterministic processes. The model can account for the behavioral uncertainty of these systems by treating them in a stochastic manner and defining the evolution dynamics probabilistically. Simulation of stochastic repressilator model is shown in the right-hand side of Figure 2.2. It can be seen from the graphs, the two models only differ in details of their behavior but show the same general trend of evolution.

2.2.2 Metabolic dynamics

The ability of living organisms to sustain life and digest is referred to as metabolism. Information about how substrates are utilized and catalyzed by enzymes to produce products is encoded in this metabolic pathway.

From a functional point of view, metabolism can be further categorized into Catabolism and Anabolism. The catabolic pathway breaks down substrates such as protein and nucleic acids to form smaller units, e.g. amino acids and nucleotides, finally releasing energy through further degradation. Anabolism refers to how end-products are synthesized, in which the small units and the released energy are utilized for constructing new large molecules.

One of the metabolic pathways rooted by glucose is shown in Figure 2.3. In this system, glucose, the starting material for reactions, is manipulated by seven enzymatic steps and converted to ethanol and glycerol.

Teusink et al. (1998) used this glycolysis pathway as an example to investigate the effectiveness of metabolic control analysis on quantifying the importance of intermediate metabolites. As an extension of this previous work, Teusink et al. (2000) refined the glycolysis pathway by adapting previous interpretation and examined the reliability of the proposed model. Similarly, mathematical expressions of the glycolysis pathway that satisfied the empirical investigations were developed. Hynne et al. (2001) studied the rate constants and maximum velocities at stationary state of glycolysis pathway by a simple fitting approach. Yang et al. (2007) inferred the intermediate metabolites and fluxes by using a Bayesian approach only given the input-output species.

2.2.3 Cell Cycle

The cell cycle is a sequence of biological events by which a cell duplicates (reproduction) and divides to produce two daughter cells. Specifically, in organisms whose cells have no

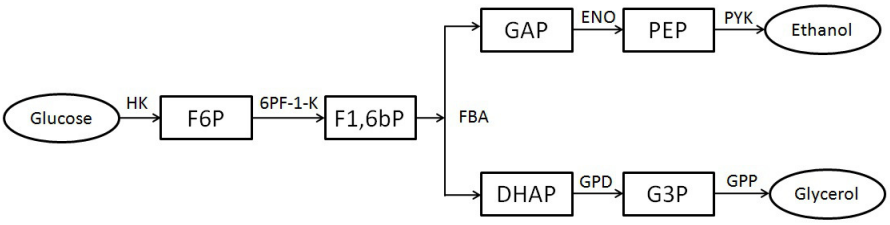


FIGURE 2.3: Metabolic pathway of glycolysis. Glucose is converted to glycerol and ethanol in the system. In the diagram, circles represent substrate and end-products, boxes are intermediated metabolites and chemicals above arrows are enzymes (Wu et al., 2005).

nucleus (prokaryotes), the reproduction of cells is accomplished by a particular process, namely binary fission. In this process, the DNA molecule replicates itself and attaches a copy to the cell membrane. Subsequently, two daughter cells with identical genetic materials are formed by pulling apart the mother cell. The division of prokaryotes takes place without nucleus, since the cell of prokaryotes does not contain nucleus.

For organisms whose cells have nucleii (eukaryotes), the cycle consists of three phases and is more complex than those of prokaryotes. Prior to reproduction and division, cells are in an initial quiescent phase, which is known as the resting phase. The preparations for reproduction and division are achieved via Gap1, Synthesis and Gap2 phases which constitute the interphase of cell cycle. For the sake of simplicity, they are denoted as G1, S and G2.

In G1, the cell grows in volume and starts to synthesize various enzymes which will be utilized in the S phase. Following the G1, during the S phase, all chromosomes are replicated via producing two (sister) chromatids for each, such that the DNA is completely doubled. At the pre-mitotic phase, in G2 phase, the cell rapidly grows and speeds up protein synthesis, and completes the preparation for Mitosis. As seen in Figure 2.4, checkpoints are used by the cell cycle which are dedicated to monitoring and manipulating the progress (Elledge, 1996). Specifically, checkpoints play a role in controlling the quality of DNA replication and are also responsible for repairing DNA damage. The process cannot continue if the checkpoint requirements are not fulfilled.

The final phase of the cell cycle is mitotic (M) consisting of mitosis and cytokinesis, during which the cell stops growing and consumes the cellular energy on division into two genetically alike daughter cells. More specifically, chromosomes in the mother cell are initially separated into two identical sets of chromosomes in mitosis, and each set is allocated in its own nucleus. Division of the mother cell into two daughter cells is carried out in cytokinesis, by which the cytoplasm, organelles and cell membrane are separated with each equally sharing the genetic materials.

Computational biologists have devoted their intensive efforts to modeling this cell cycle system in the past decade. Tyson (1991), being the pioneer in this field, mathematically

characterized the interaction of `cdc2` and `cyclin`. In the simplified cell cycle, this process is crucial, since it forms a heterodimer (maturation promoting factor) that acts as the regulator of the major events of the cell cycle. The intrinsic machinery of the cell cycle was further studied by Tyson and Novak (2001), in which the irreversibility of transitions occurring between two phases (G1 and S-G2-M) was assessed. Consequently, the simple model was enriched following this empirical investigation, becoming a reasonable model of the control systems in yeast cells, frog eggs, and cultured mammalian cells. Chen et al. (2000) modeled the regulation of the cyclin-dependent kinase that 'switches on' DNA synthesis and mitosis in the budding yeast, *Saccharomyces cerevisiae*, via nine nonlinear ordinary differential equations. Srividhya and Gopinathan (2006) proposed a time delay model with seven differential equations for characterizing the cell cycle in higher eukaryotes. One variable in these formulas describes the mass of the cell, which is usually regarded as the checkpoint of G2/M.

Additionally, several efforts towards integrating quantitative study and a 'wet-lab' environment have been undertaken by Cross (2003); Battogtokh and Tyson (2004). A comprehensive review of the empirical investigation of the cell cycle in computational biology can be found in Ingalls et al. (2007).

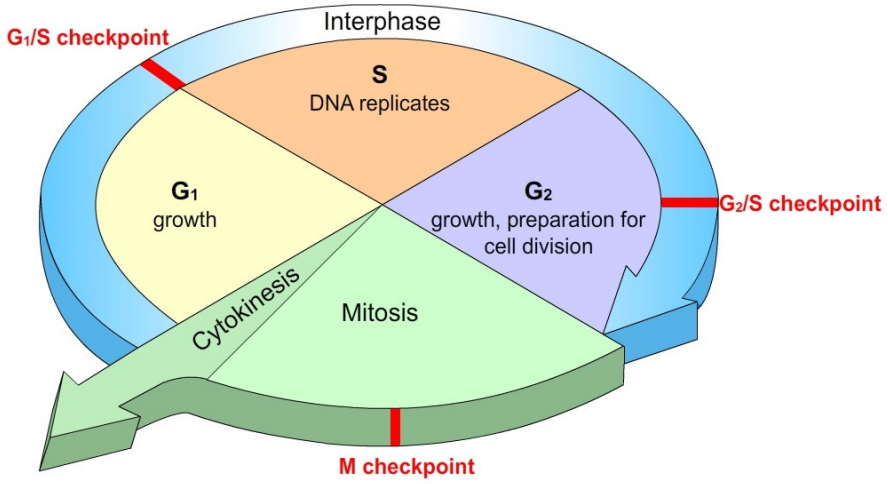


FIGURE 2.4: Schematic diagram of the cell cycle process, obtained from `Ricochet Science`'s website: `http://ricochetscience.com/category/diseases/cancer/`. Though the four phases are shown roughly equal in time, in fact the time spent in each phase is different. From the simulation, the relative times are: $\mathbf{G}_1$: 55%, $\mathbf{S}$: 15%, $\mathbf{G}_2$: 15% and $\mathbf{M}$: 15%.

Even though the rapid development of biotechnology makes the exploration of systems experimentally possible, however, the experimental cost would be sometimes extremely expensive. Consequently, the above limitation motivates the computational biology in which the biological system is mathematically described and the hypothesis can be verified via the simulation before carrying out the experiment. The state-space models, facilitating to produce the synthetic biological data, are introduced in the following

section.

2.3 State-space models

State-space models are general mathematic expression for describing data, and have been intensively used in many areas, such as system biology (Quach et al., 2007), computational finance (Shephard, 1996), environmental analysis (Wikle et al., 1998) and geophysical science (Ghil and Ide, 1997). In the setting of state-space models, the system of interest is consisted of two kinds of variables: latent (unobservable) variables, and visible (observable) variables. The series of latent variables, are referred to as states, while the sequence of visible variables, are known as observations. State-space models are dedicated to quantifying how observations arise from the states at a given time. Mathematical description of these problems are due to Bar-Shalom and Fortmann (1987); Kitagawa and Gersch (1996).

2.3.1 Linear dynamical systems

The simplest form of state-space models is the linear dynamical system (LDS) which formulates processes as

$$\begin{aligned} \mathbf{x}_t &= \mathbf{A}\mathbf{x}_{t-1} + \mathbf{w}_t, \\ \mathbf{y}_t &= \mathbf{C}\mathbf{x}_t + \mathbf{v}_t, \end{aligned} \tag{2.1}$$

where $\mathbf{x}_t \in \mathbb{R}^{n\times 1}$ and $\mathbf{y}_t \in \mathbb{R}^{m\times 1}$ are the states and observations of dynamics at time instance t. $\mathbf{A} \in \mathbb{R}^{n\times n}$ is the transition matrix and $\mathbf{C} \in \mathbb{R}^{m\times n}$ is the observation matrix. $\mathbf{w}_t$ and $\mathbf{v}_t$ are independent Gaussian random variables from $\mathcal{N}(0, \boldsymbol{Q})$ and $\mathcal{N}(0, \boldsymbol{R})$.

The set of linear dynamical systems is the typical approach for formulating data in the domains of control theory and machine learning. Among applications in those fields, control engineering is the area for most frequently using LDS. For example, Parker and Johnson (2009) proposed a scheme for decoupling LDSs in the context of control theory so as to cancel the effect of disturbance inputs and correct unwanted behavior. Liu et al. (2011) enriched LDS and developed a tool based in control theory for enhancing the controllability of complex networks. In addition, Shi and Lu (2010) claimed that LDS itself is capable of summarizing the electroencephalography (EEG) signals, in which EEG signals with disturbance are regarded as the observations. The use of LDS in EEG paved a way to classify signals given noisy observations.

In machine learning, Rauch et al. (1965) combined LDSs and Maximum Likelihood Estimator in tracking state behavior of system, and illustrated the effectiveness of this state-of-the-art solution by deploying to a numerical example. Zhang and Boukas (2009)

focused on finding the solution to estimate the unknown transition matrix in LDSs, success of the proposed approach was demonstrated by two toy numerical examples. In the review literature, Roweis and Ghahramani (1999) claimed that, due to their generality, typical problems in machine learning such as principle component analysis (PCA), independent component analysis models (ICA) and Kalman filtering can be formulated as LDSs.

In computational biology, LDSs also play a crucial role to characterize the dynamics studied; for instance, Yang et al. (2007) formulated a metabolic pathway into the LDSs where the model adopted the stoichiometry matrix as the coefficient $\mathbf{A}$. The concentrations of intermediate metabolites and end-products are mapped to states and observations in LDSs, respectively. Fluxes and enzymes consumed in reactions are indicated by the transition matrix $\mathbf{A}$, and observations related to states are quantified by the observation matrix $\mathbf{C}$. Similar treatments of LDSs in modeling gene expression data can be found in the literatures (Beal et al., 2005; Sanguinetti et al., 2006).

LDSs are also useful in describing biological signals such as the electroencephalography (EEG). For example, Sanei and Chamber (2007); Cheung et al. (2010) characterized the multi-channel of EEG signals as the observations of LDSs, yet the unmeasurable information in the cortex was treated as the latent variables and the parameters in model indicated the cortical connectivities.

2.3.2 General state-space models

Nonlinear state-space models with Gaussian noise

In its default setting, LDSs are unable to model numerous intricate problems in the real world which motivates the development of nonlinear state-space models, given as

$$\mathbf{x}_t = \mathbf{f}(\mathbf{x}_{t-1}) + \mathbf{w}_t, \tag{2.2}$$

$$\mathbf{y}_t = \mathbf{h}(\mathbf{x}_t) + \mathbf{v}_t, \tag{2.3}$$

where $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ and $\mathbf{h} : \mathbb{R}^n \to \mathbb{R}^m$. $\mathbf{w}_t$ and $\mathbf{v}_t$ are Gaussian noise with the same distributions presented in LDS. Transition of state in dynamics at time series is depicted by a nonlinear function $\mathbf{f}(\cdot)$, equation 2.2 is called the transition model. Observations respond to states which are quantified by function $\mathbf{h}(\cdot)$, also known as the observation model, given in equation 2.3. LDS can be seen as a particular case of nonlinear state-space models. We give two examples to illustrate how to describe dynamics with nonlinear state-space models.

Example 2.1. *The univariate non-stationary growth model is often adopted as a benchmark system to assess the performances of learning tools in the literature, such as Kitagawa and Gersch (1996); Arulampalam et al. (2002). The mathematical expression of*

this system is

$$x_t = \frac{1}{2}x_{t-1} + \frac{25x_{t-1}}{1+x_{t-1}^2} + 8\cos(1.2t) + w_t \tag{2.4}$$

$$y_t = \frac{x_t^2}{20} + v_t \tag{2.5}$$

where $w_t \sim \mathcal{N}(0, Q)$ *and* $v_t \sim \mathcal{N}(0, R)$. *Summarizing the univariate non-stationary growth system in nonlinear state-space models, functions* $f(\cdot)$ *and* $h(\cdot)$ *are given as:*

$$f(x_{t-1}) = \frac{1}{2}x_{t-1} + \frac{25x_{t-1}}{1+x_{t-1}^2} + 8\cos(1.2t), \tag{2.6}$$

$$h(x_t) = \frac{x_t^2}{20}, \tag{2.7}$$

Nonlinear state-space models with non-additive Gaussian noise

According to the relation between dynamics and noise, the nonlinear state-space models are developed to multiplicatively corrupt with Gaussian noise. The *stochastic volatility* model falls into this category.

Example 2.2. *The stochastic volatility model found itself as a useful tool to formulate problems in finance. For instance, Hull and White (1987) quantitatively analyzed option pricing by characterizing it in the stochastic volatility model. Further, numerous statistical learning tools employ the stochastic volatility model to demonstrate their capabilities for inference (Liu, 2001; Chib et al., 2002; Girolami and Calderhead, 2011). The stochastic volatility model is given as:*

$$x_t = \phi x_{t-1} + \eta_t \tag{2.8}$$

$$y_t = \epsilon_t \beta \exp\left(\frac{x_t}{2}\right) \tag{2.9}$$

where $\eta_t \sim \mathcal{N}(0, \sigma_\eta^2)$, $\epsilon_t \sim \mathcal{N}(0, 1)$ *and* β *is the rate constant. It can be seen from the formulations, that although noise is still additively imposed in the transition model, in the observation model it acts as a multiplier to observations. In addition, to express the stochastic volatility model with the nonlinear state-space model, functions are represented as:*

$$f(x_t) = \phi x_{t-1} \tag{2.10}$$

$$h(y_t) = \beta \exp\left(\frac{x_t}{2}\right) \tag{2.11}$$

Statistical expression of state-space models

From a statistical point of view, an alternative way of writing state-space formulations

of equation 2.2 - 2.3 is given below

$$\mathbf{x}_0 \sim p(\mathbf{x}_0|\boldsymbol{\theta}_0), \tag{2.12}$$

$$\mathbf{x}_t \sim p(\mathbf{x}_t|\mathbf{x}_{t-1}, \boldsymbol{\theta}_\mathrm{s}), \tag{2.13}$$

$$\mathbf{y}_t \sim p(\mathbf{y}_t|\mathbf{x}_t, \boldsymbol{\theta}_\mathrm{obs}), \tag{2.14}$$

where $\mathbf{x}_0$ is the initial condition of states, $t \in [0, \ldots, T]$ is a vector of time points, $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^m$. $\boldsymbol{\theta}_0$, $\boldsymbol{\theta}_\mathrm{s}$ and $\boldsymbol{\theta}_\mathrm{obs}$ are parameters involved in the initial states distribution, transition distribution and observation distribution, respectively. In this setting, the probability $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \boldsymbol{\theta}_\mathrm{s})$ is known as the transition distribution, similarly, $p(\mathbf{y}_t|\mathbf{x}_t, \boldsymbol{\theta}_\mathrm{obs})$ is the observation distribution. The graphical representation of the state-space models in a statistical context shown in Figure 2.5. It has found applications in control or heart

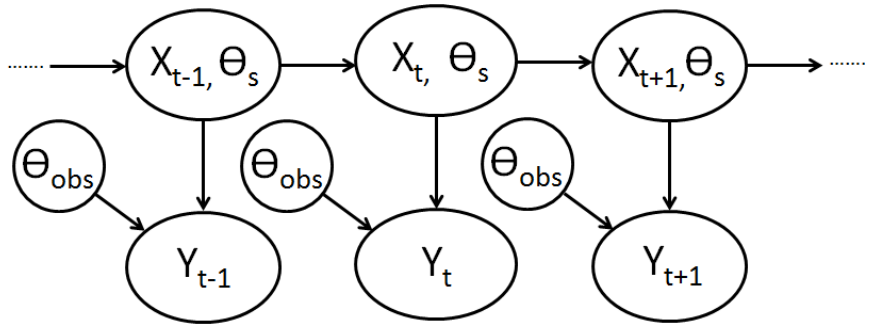


FIGURE 2.5: Graphical description of state-space models, including parameters for state model and observation model (shown as $\boldsymbol{\theta}_\mathrm{s}$ and $\boldsymbol{\theta}_\mathrm{o}$, respectively.).

beat modeling, taking external inputs such as disturbances from motor and abnormal heart beat pulse which are treated as factors of the transition distribution as well. Consequently, the transition distribution is further extended as $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \boldsymbol{\theta}_\mathrm{s}, \mathbf{u}_t)$, where the external inputs are termed as $\mathbf{u}_t$. In **Example 2.3**, we rewrite the aforementioned examples in probabilistic forms.

Example 2.3. *Linear state-space models in probabilistic form:*

$$p(\mathbf{x}_0|\boldsymbol{\theta}_\mathrm{s}) = \mathcal{N}(\mathbf{x}_0|\boldsymbol{\pi}_0, \boldsymbol{\Sigma}_0), \tag{2.15}$$

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}, \boldsymbol{\theta}_\mathrm{s}) = \mathcal{N}(\mathbf{x}_t|\mathbf{A}\mathbf{x}_{t-1}, \boldsymbol{Q}) \tag{2.16}$$

$$p(\mathbf{y}_t|\mathbf{x}_t, \boldsymbol{\theta}_\mathrm{obs}) = \mathcal{N}(\mathbf{y}_t|\mathbf{C}\mathbf{x}_t, \boldsymbol{R}) \tag{2.17}$$

where parameters for transition distribution denoted as $\boldsymbol{\theta}_\mathrm{s} = \{\boldsymbol{\pi}_0, \boldsymbol{\sigma}_0, \mathbf{A}, \boldsymbol{Q}\}$, and $\boldsymbol{\theta}_\mathrm{obs} = \{\mathbf{C}, \boldsymbol{R}\}$.

Example 2.4. *The univariate non-stationary growth model in the probabilistic form:*

$$p(x_0|\boldsymbol{\theta}_{\mathrm{s}}) = \mathcal{N}(x_0|\pi_0, \sigma_0), \tag{2.18}$$

$$p(x_t|x_{t-1}, \boldsymbol{\theta}_{\mathrm{s}}) = \mathcal{N}(x_t|ax_{t-1} + \frac{bx_{t-1}}{1+x_{t-1}^2} + c\cos(1.2t), Q), \tag{2.19}$$

$$p(y_t|x_t, \boldsymbol{\theta}_{\mathrm{obs}}) = \mathcal{N}(y_t|\frac{x_t^2}{20}, R), \tag{2.20}$$

where $\boldsymbol{\theta}_{\mathrm{s}} = \{\pi_0, \sigma_0, a, b, c, Q\}$, and $\theta_{\mathrm{obs}} = R$.

Example 2.5. *The stochastic volatility model in the probabilistic form:*

$$p(x_0|\boldsymbol{\theta}_{\mathrm{s}}) = \mathcal{N}(x_0|0, \frac{\sigma^2}{1-\phi^2}), \tag{2.21}$$

$$p(x_t|x_{t-1}, \boldsymbol{\theta}_{\mathrm{s}}) = \mathcal{N}(x_t|\phi x_{t-1}, Q), \tag{2.22}$$

$$p(y_t|x_t, \boldsymbol{\theta}_{\mathrm{obs}}) = \mathcal{N}(y_t|0, \beta^2\exp(x_t)), \tag{2.23}$$

where $\boldsymbol{\theta}_{\mathrm{s}} = \{\phi, Q\}$, and $\theta_{\mathrm{obs}} = \beta$.

2.3.3 Likelihood function

Likelihood, central to the operation of inference methods including Maximum Likelihood (ML), Expectation Maximization (EM), Particle Filter (PF), etc, represents the probability of event occurrence given a set of parameters. In the inference problems, this likelihood is expressed as a function denoted as $\mathcal{L}_{\boldsymbol{\theta}} = p(\mathbf{y}|\boldsymbol{\theta})$ that takes the parameters of a statistical model as variables. Let us assume that no Gaussian noise is imposed to dynamics in the transition model and the observation function $\mathbf{h}(\cdot)$ is an identity matrix $\mathbf{I}$, then the likelihood function can be derived as

$$\mathbf{x} = \mathbf{f}(\mathbf{x}_0, \boldsymbol{\theta}_{\mathrm{s}}), \tag{2.24}$$

$$\mathbf{y} = \mathbf{I}\mathbf{x} + \mathbf{v} \sim \mathcal{N}(0, \boldsymbol{\sigma}_{\mathbf{v}}^2), \tag{2.25}$$

$$p(\mathbf{y}|\boldsymbol{\theta}_{\mathrm{obs}}) = (2\pi)^{-\frac{\mathrm{n}}{2}}|\boldsymbol{\Sigma}|^{-\frac{1}{2}}\exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}. \tag{2.26}$$

Equation (2.26) is called *probability density function* (PDF) of likelihood function. Since the identity matrix solely describes how states are linearly related to observations, the system transition $\mathbf{f}(\cdot)$ directly replaces the term $\mathbf{I}\mathbf{x}_t$ in formulation.

A multivariate Gaussian case is represented as an illustration of likelihood distribution. Suppose the observed sequence of data is distributed in multivariate Gaussian having means and covariance matrix given as $\boldsymbol{\mu} = [-1.5, 2.5]$ and $\boldsymbol{\Sigma} = \begin{bmatrix} 2 & 1.15 \\ 1.15 & 0.8 \end{bmatrix}$; the surface of likelihood is shown in Figure 2.6.

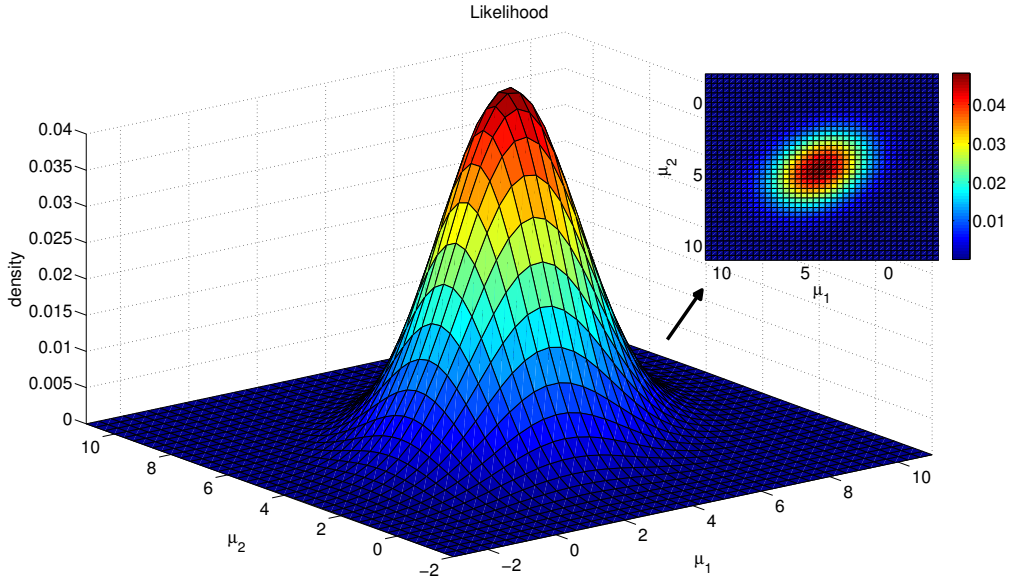


FIGURE 2.6: Surface of likelihood of the given multivariate Gaussian distribution, the region in red color depicts the higher probability for being the true values.

2.3.4 Inference problems in the state-space models

Suppose a biological system has the latent state variables, denoted as $\mathbf{x}$, which might consist of concentrations of chemical species of interest, e.g. mRNA, proteins and metabolites. Inference in systems biology is to identify those hidden information by given the observations $\mathbf{y}$, via a reverse engineering process.

Differential equations based state-space models

In computational biology, the dynamical systems are often characterized by a set of ordinary equations. The differential equations capture the changes in system with respect to time and the solution to these equations helps in explaining the behavior at the system level. More specifically, biological systems adapting state-space models with ODEs are described as

$$\dot{\mathbf{x}}_t = f(\mathbf{x}, \boldsymbol{\theta}_\mathrm{s}), \tag{2.27}$$

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \int_{t-1}^{t} f(\mathbf{x}, \boldsymbol{\theta}_\mathrm{s}) d\tau, \tag{2.28}$$

$$\mathbf{y}_t = h(\mathbf{x}_t) + \mathbf{v}_t, \tag{2.29}$$

where $\mathbf{v}_t$ and $\boldsymbol{\theta}_\mathrm{s}$ are the same Gaussian noise for the observation model and transition parameters as previously represented. As the underlying system is deterministic, no noise is added to the transition model.

For quantitative analyzing dynamics, even though all biological systems are the continuous-time processes, which are assumed to be observed at discrete instances in time. Following

Sitz et al. (2002) and other authors, the way to solve this problem is to numerically integrate the state dynamics between temporal points at which observations are made, as shown as equation 2.28.

State and parameter estimations

Statistically, the inference task is defined to compute the posterior distribution of the latent state. Following Bayes's rule, the posterior distribution of state is shown as

$$p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_{1:t}|\mathbf{x}_{1:t})p(\mathbf{x}_{0:t})}{\int p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t})p(\mathbf{x}_{1:t})}, \tag{2.30}$$

where unknown states and the sequence of observations are denoted as $\mathbf{x} = [\mathbf{x}_0, \ldots, \mathbf{x}_t]$ and $\mathbf{y} = [\mathbf{y}_1, \ldots, \mathbf{y}_t]$, respectively. Following Andriue et al. (2001), we adopt the notation $\mathbf{x}_{0:t}$ to refer to data in time sequence from 0 to t. The two terms in the numerator are the *likelihood* and the *prior distribution*, which can be expanded as

$$p(\mathbf{y}_{1:t}|\mathbf{x}_{1:t}) = \prod_{t=1}^{t} p(\mathbf{y}_t|\mathbf{x}_t), \tag{2.31}$$

$$p(\mathbf{x}_{0:t}) = p(\mathbf{x}_0) \prod_{t=1}^{t} p(\mathbf{x}_t|\mathbf{x}_{t-1}). \tag{2.32}$$

Parameters $\boldsymbol{\theta}_\mathrm{s}$, in their specifications, such as rate constants of biochemical reactions, synthesis and decay rates of macromolecules, delays incurred n transcription of genes and translation of proteins, and sharpness of nonlinear effects (Hill coefficient) are often unknown in dynamics. Hence, those need to be simultaneously estimated with the latent states. A common solution to this problem is to adopt the unknown parameters as additional states in the system and to impose an artificial dynamics on them (Kitagawa, 1998; Liu, 2001). This technique is known as the state extension (Sitz et al., 2002) and dynamics of parameters in the probabilistic form are described as

$$\boldsymbol{\theta}_{\mathrm{s},0} \sim p(\boldsymbol{\theta}_{\mathrm{s},0}|\boldsymbol{\theta}_{\mathrm{p},0}), \tag{2.33}$$

$$\boldsymbol{\theta}_{\mathrm{s,t}} \sim p(\boldsymbol{\theta}_{\mathrm{s,t}}|\boldsymbol{\theta}_{\mathrm{s,t\text{-}1}}, \boldsymbol{\theta}_\mathrm{p}), \tag{2.34}$$

where $\boldsymbol{\theta}_{\mathrm{s},0}$ is the initial guess of the parameters and is generated by $\boldsymbol{\theta}_{\mathrm{p},0}$. The parameters are further explored by the transition kernel which is determined by $\boldsymbol{\theta}_\mathrm{p}$.

Example 2.6. *Lotka (1925); Volterra (1926) initially proposed the predator-prey models, intended to formulate the continuously overlapping interactions of two species in an environment. Two first-order, yet nonlinear differential equations are used to describe*

this system, which is denoted as

$$\frac{dx}{dt} = \alpha x - \beta y, \tag{2.35}$$

$$\frac{dy}{dt} = \delta xy - \gamma y, \tag{2.36}$$

where x and y are the populations of prey and predator, respectively. This model has been extensively used in economics (Desai and Ormerod, 1998), computational finance (Lee et al., 2005) and ecology (Berryman, 1992).

A large number of parameter estimation methods in the field of computational biology make use of this predator-prey model to demonstrate their effectivenesses. An example of this is Toni et al. (2009) who employed the predator-prey model for testing performance of the proposed inference method, in which parameter vector $\boldsymbol{\theta}_s = \{\alpha, \beta, \delta, \gamma\}$ is to be estimated. ABC-SIS (Toni et al., 2009) (details are introduced in chapter 3) with an initial guess of parameter values drawn from a uniform distribution $\boldsymbol{\theta}_0 \sim \mathcal{U}(a, b)$, associated with transitions made randomly using a certain distance in each iteration, has yielded precise estimate of parameters. Mapping these algorithmic settings to equations 2.33-2.34, initial guess and transition distribution can be probabilistically described

$$p(\boldsymbol{\theta}_{\mathrm{s},0}|\boldsymbol{\theta}_{\mathrm{p},0}) = \mathcal{U}(\boldsymbol{\theta}_{\mathrm{s},0}|\boldsymbol{a}, \boldsymbol{b}), \tag{2.37}$$

$$p(\boldsymbol{\theta}_{\mathrm{s,t}}|\boldsymbol{\theta}_{\mathrm{s,t\text{-}1}}, \boldsymbol{\theta}_{\mathrm{p}}) = \mathcal{N}(\boldsymbol{\theta}_{\mathrm{s,t}}|\boldsymbol{\theta}_{\mathrm{s,t\text{-}1}}, \boldsymbol{\sigma}^2_{\mathrm{p}}), \tag{2.38}$$

where $\boldsymbol{\theta}_{p,0} = \{\boldsymbol{a}, \boldsymbol{b}\}$ and $\boldsymbol{\theta}_p = \{\boldsymbol{\sigma}^2_p\}$.

There are a few solutions for the problem of imposing artificial dynamics on unknown parameters, the one represented in equation 2.38 is known as the random walk kernel and is often chosen for the state inference (Sun et al., 2008) and parameter estimation problems (Andrieu et al., 2004). As an alternative, Gordon et al. (1993) suggested to specify the variance $\boldsymbol{\sigma}^2_{\mathrm{p}}$ of the Gaussian as $\boldsymbol{\sigma}^2_{\mathrm{p}} = KEN^{-2/d}$, where E is the discrepancy between the minimum and the maximum of the current particles (the concept of 'particle' will be introduced later in section A.1.4), K is a parametric constant, N is the number of particles and d is the dimension of parameter vector. In addition, Liu and West (2001) introduced the use of kernel smoothing with shrinkage for parameter evolution, details of this scheme is introduced in the section A.4 of Appendix A and a comparative study between this parameter evolution and the random walk can be found in chapter 4.

A variety of forms of state-space models characterizing the data of system are presented, and the rigorous mathematical description of the parameter estimation problem has been introduced in this section. In the following, several estimation methods being categorized as sequential and batch will be discussed.

2.3.5 State-space models with ODEs

The biological systems are often characterized by a set of ordinary differential equations (ODEs), which capture changes to a system with respect to time and solutions of which help to explain behavior at the system level. As mentioned in Chapter 2, we generally consider nonlinear state-space models of the biological systems in which the dynamics are deterministic and observations noisy. More specifically, the nonlinear state-space models adapted for systems biology are given as

$$\dot{\mathbf{x}}_t = f(\mathbf{x}_t, \boldsymbol{\theta}), \tag{2.39}$$

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \int_{t-1}^{t} f(\mathbf{x}, \boldsymbol{\theta}_t) d\tau, \tag{2.40}$$

$$\mathbf{y}_t = h(\mathbf{x}_t) + \boldsymbol{v}_t, \tag{2.41}$$

where state vector $\mathbf{x}_t$ may consist of concentrations of different molecular species at time t, and $\mathbf{y}_t$ quantifies the noisy observations relating to $\mathbf{x}_t$ via the output function $h(\cdot)$. $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_p\}$ is the parameter vector of dynamics. It is necessary to recall that $\boldsymbol{v}_t$ is the zero mean Gaussian noise corrupting observations, and its covariance matrix is denoted by $\boldsymbol{R}$. This $\boldsymbol{R}$ is a positive definite matrix, quantifying the quality of observations. Additionally, $\boldsymbol{R}$ is defined as a diagonal matrix due to the noise corrupting to each observation is assumed to be uncorrelated. Setting of $\boldsymbol{R}$ governs the performance of inference methods, investigation on this subject is given in section A.6.3.

2.4 Bayesian inference

Inference is the process of running certain algorithms or probability models on a set of observed data, such that the values of parameters and unobserved quantities such as concentration of species can be predicted. In this field, point estimators, for instance minimum mean squared error (MMSE) and maximum likelihood (ML), propose a single value as the 'best estimate' of an unknown. Rather than claim efficiency as the pragmatic advantage of point estimators, however, we prefer to emphasize their disadvantages such as relying on an explicit form of likelihood or loss function and conclusions are represented without characterizing uncertainty.

Bayesian inference is therefore motivated by making inferences from data using probability models for quantities about either observed or unobserved. The essence of Bayesian methods involves the use of probability for quantifying uncertainty in inferences, which allow us to interpret conclusion in a statistical way. In this thesis, probability distributions are notatedas $p(\boldsymbol{\theta}|\mathbf{y})$ or $p(\mathbf{x}|\mathbf{y})$, indicating the guess of parameters or states is conditional on the observations $\mathbf{y}$.

The probabilistic conclusions about $\boldsymbol{\theta}$ given $\mathbf{y}$ is referred to as the *posterior distribution*,

following Bayes' rule, which can be evaluated as

$$p(\boldsymbol{\theta}|\mathbf{y}) = \frac{p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathbf{y})}, \tag{2.42}$$

where $p(\mathbf{y}|\boldsymbol{\theta})$ is known as likelihood introduced in section 2.3.3 and $p(\boldsymbol{\theta})$ is called *prior distribution* which will be discussed in the following section. In addition, $p(\mathbf{y}) = \int p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}$ is the sum over all possible values of $\boldsymbol{\theta}$ (in the discrete cases, $p(\mathbf{y}) = \sum_{\theta} p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}$). In general, $p(\mathbf{y})$ is considered as a constant due to its independence from $\boldsymbol{\theta}$, with fixed $\mathbf{y}$, yielding the *unnormalized posterior distribution*, given as

$$p(\boldsymbol{\theta}|\mathbf{y}) \propto p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta}). \tag{2.43}$$

Equations 2.42 - 2.43 derive the primary mechanics of Bayesian inference, where the primary task is to form the prior distribution and likelihood in an appropriate way so that inference can perform smoothly.

2.4.1 Beta prior distribution

In Bayesian inference, the prior distribution quantifies the initial guess of unknown quantity which could be parameter or hidden state variable. $p(\boldsymbol{\theta})$ or $p(\mathbf{x})$ allow us to express any belief we have in the value of $\boldsymbol{\theta}$ or $\mathbf{x}$ prior to observing any data. More specifically, the prior distribution can be varied in accordance with different assumptions or knowledge. For instance, the prior distribution could be used to characterize the probability of tossing coins giving heads, without performing any trials. Here we illustrate the effect of prior on updating posterior via an example previously introduced in Rogers and Girolami (2011), where three possible assumptions include no information about the outcome of coin tosses, equal probability of either heads or tails and biased to flip heads more than tails can be made.

Each of these assumptions can be encoded by different prior distribution, moreover, the parameter θ implying the possibility of results can be any value between 0 and 1. Figure 2.7 shows the densities characterizing three different prior scenarios. As shown in the graph, the blue line is a uniform density and shows no preference for any particular value of θ. The red curve is shaped symmetrically centered on $\theta = 0.5$, which indicates a fair coin being tossed and value of θ is most likely allocating between 0.4 and 0.6. Rather than the first two cases, the green line clearly implies its preference that value of θ is most certainly greater that 0.5. This suggests that the coin is biased to give more heads than tails.

These continuous random variables that are varied between 0 and 1 can be formulated

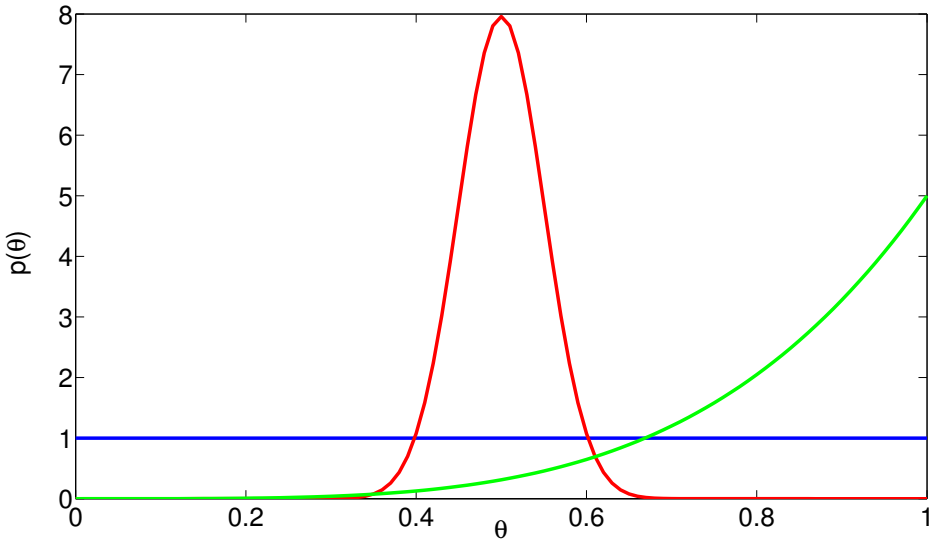


FIGURE 2.7: The prior densities $p(\theta)$ are evaluated using value of θ for three assumptions, where the no prior information, fair chance and biased assumption are shown as blue, red and green lines, respectively.

by the beta distribution, given as

$$p(\theta) = \text{Beta}(\theta|\alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1}, \tag{2.44}$$

where α and β are the parameters for determining the shape of density, and both must be positive. $\Gamma(\alpha)$ is the gamma function and defined as $\Gamma(\alpha) = \int_0^\infty x^{\alpha-1}e^{-x}dx$. This function is used to ensure the density $p(\theta)$ integrates to 1 and leads to the independence on data for prior. We will henceforth denote the beta distribution as $\mathcal{B}(\alpha, \beta)$, and the priors shown in Figure 2.7 are accordingly written as $p_1(\theta) = \mathcal{B}(\theta|1, 1)$, $p_2(\theta) = \mathcal{B}(\theta|50, 50)$ and $p_3(\theta) = \mathcal{B}(\theta|5, 1)$.

2.4.2 The exact posterior

In Bayesian inference, the process involves updating a prior distribution $p(\boldsymbol{\theta})$ to a posterior distribution $p(\boldsymbol{\theta}|\mathbf{y})$. From the mathematical convenience perspective, it is natural to expect that some general relations might hold between prior and posterior. That is, the posterior distribution should be expressed due to its same mathematical form as the prior distribution. This property is called *conjugacy*, which can be achieved by fixing likelihood to a particular form.

Consider the beta distribution example, it is referred as the conjugate prior to the binomial likelihood. In probability theory, the binomial distribution plays a role in characterizing density of the number of successes in a sequence of n independent 'yes' or 'no' experiments, with each trial yielding success with a parametric probability θ. Then

the binomial distribution is given as

$$p(k|\theta) = \text{Bin}(k|n,\theta) = \binom{n}{k}\theta^k(1-\theta)^{n-k}, \tag{2.45}$$

where n and k are the number of experiments performed and number of successes, respectively. $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ is called binomial coefficient. When n is set to one, the binomial distribution becomes a special case which is known as Bernoulli distribution and this success or failure experiment is called a Bernoulli trial.

Considering the expression given in equation (2.43), having beta distribution as prior and Bernoulli distribution as likelihood, posterior is written as

$$\begin{aligned} p(\theta|y=k) &\propto p(\theta)p(k|\theta) \\ &\propto \left[\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1}\right] \times \left[\binom{n}{k}\theta^k(1-\theta)^{n-k}\right] \\ &\propto \theta^{k+\alpha-1}(1-\theta)^{n-k+\beta-1}. \end{aligned} \tag{2.46}$$

Therefore, we have

$$p(\theta|y) = \frac{\Gamma(\alpha+\beta+n)}{\Gamma(\alpha+y)\Gamma(\beta+n-y)}\theta^{\alpha+y-1}(1-\theta)^{\beta+n-y-1}. \tag{2.47}$$

Notice the posterior is given the same form as prior distribution, i.e. $p(\theta|y) \sim \mathcal{B}(\theta|\delta,\gamma)$ where $\delta = \alpha + y$ and $\gamma = \beta + n - y$, and it shows how the posterior parameters are updated by incorporating the new number of heads y_t to the first prior parameter α and the new number of tails $n - y_t$ to the second prior parameter β.

We now illustrate how the posterior distribution $p(\theta|y)$ is accordingly updated for three different prior scenarios. In addition, the expected value and variance of θ are used to quantify the effect of priors. Supposing a random variable is drawn from a beta distribution with parameters α and β, i.e. $x \sim \mathcal{B}(\alpha,\beta)$, then its expected value and variance are computed as

$$\mathbb{E}[x] = \frac{\alpha}{\alpha+\beta}, \qquad \text{var}[x] = \frac{\alpha\beta}{(\alpha+\beta)^2+(\alpha+\beta+1)}. \tag{2.48}$$

Therefore, for all three scenarios, we can compute the expected value of θ (the notation of expected value $\mathbb{E}_{p_n(\theta)}[\theta]$ is simplified as $\mathbb{E}_n[\theta]$):

$$\mathbb{E}_1[\theta] = \frac{1}{2} = 0.5, \qquad \mathbb{E}_2[\theta] = \frac{1}{2} = 0.5, \qquad \mathbb{E}_3[\theta] = \frac{5}{6} = 0.833.$$

Similarly, the notation of variance $\text{var}_{p_n(\theta)}[\theta]$ is simplified as $\text{var}_n[\theta]$, and values are calculated as:

$$\text{var}_1[\theta] = 0.083, \qquad \text{var}_2[\theta] = 0.002, \qquad \text{var}_3[\theta] = 0.019.$$

Note that the expected value of θ under no prior knowledge and fair coin scenario is identical, of which the variance for the first scenario is much higher than another. This makes sense as both priors show no preference, leading to equal probability for head and tail. However, the lower variance of the fair coin implies less uncertainty in comparison to the no prior knowledge case. This is because most of the probabilities θ for second scenario are valued between 0.4 and 0.6, while θ for the first scenario can be any value between 0 and 1 causing more uncertainty.

From equation (2.47), we can update the posterior distribution toss by toss, via a beta distribution with parameter $\delta = \alpha + y_t$ and $\gamma = \beta + n - y_t$. Let the first coin toss ($n = 1$) be head ($y_t = 1$), then under the three different prior settings, we have

$$\mathbb{E}_{p_1(\theta|y_t)}[\theta] = 0.667, \quad \mathbb{E}_{p_2(\theta|y_t)}[\theta] = 0.505, \quad \mathbb{E}_{p_3(\theta|y_t)}[\theta] = 0.857,$$

with variances as:

$$\text{var}_{p_1(\theta|y_t)}[\theta] = 0.055, \quad \text{var}_{p_2(\theta|y_t)}[\theta] = 0.002, \quad \text{var}_{p_3(\theta|y_t)}[\theta] = 0.015.$$

The increase of expected value and decrease of variance under all three scenarios are naturally expected, since the current coin tossing is evidence to suggest that heads are more likely than tails and further decreases the uncertainty in belief.

Supposing the complete results of ten tossing trials are given as

$$\text{H, H, H, T, H, H, T, H, T, H.}$$

We show how the posterior distribution $p(\theta|y_t)$ is updated from the prior (previous posterior distribution $p(\theta|y_{t-1})$), incorporating the new observed coin tosses. In the graphs

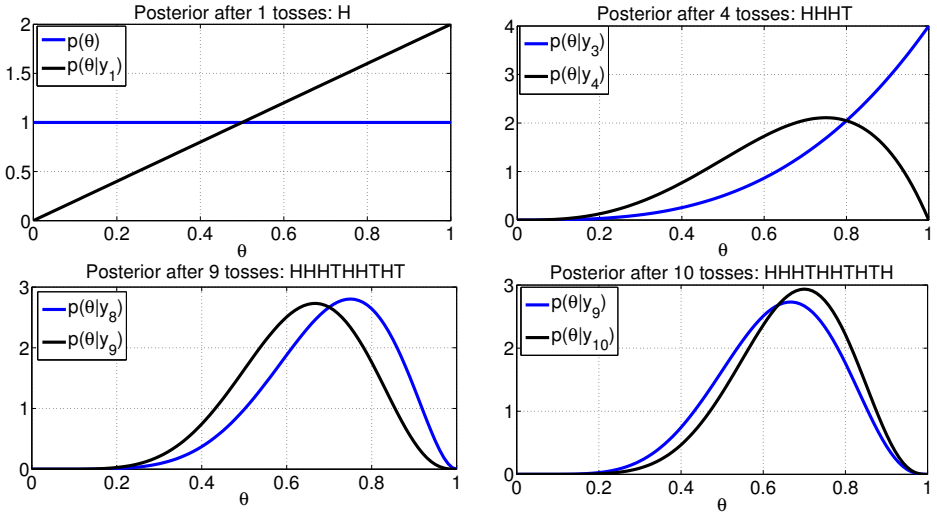


FIGURE 2.8: Posterior distribution $p(\theta|y_t)$ (shown as the red line) is evolved from the previous posterior distribution $p(\theta|y_{t-1})$) (shown as the blue line), incorporating the observation of coin tosses.

shown in Figure 2.8, the red line shows the posterior after one toss being seen, and the blue line is the prior or the previous posterior. It is consistent with the aforementioned calculation, the possibility θ dramatically increases after having a head tossing as evidence. The results up to toss 3 suggest that heads is much more likely than tails (3 out of 4), therefore, it seems unexpected for seeing tails on the fourth toss and the density has been 'pulled' back to the low values of θ. The similar decrease is also seen due to the arrival of tails at the ninth toss, while the expected value of θ jumps up a bit for tossing heads at the tenth trial.

For the fair coin prior, considering the same sequence of coin tossing, we show the evolution of posterior distribution in Figure 2.9. As shown in these graphs, we can see

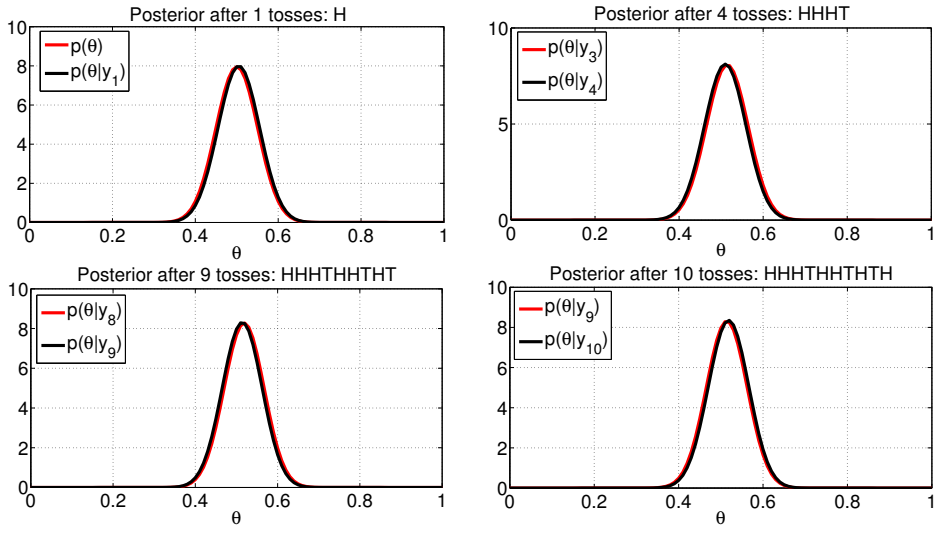


FIGURE 2.9: Posterior distribution $p(\theta|y_t)$ (shown as the red line) is evolved from the previous posterior distribution $p(\theta|y_{t-1})$) (shown as the blue line), under the fair coin prior setting.

very little change in the evolution of the posterior distribution. These slight changes are caused by the small amount of new observations. Recalling the parameters of beta prior distribution are set as $\alpha = 50$ and $\beta = 50$, this parametric setting implies that 50 successes out of 100 tosses, as a result of which 10 more tosses make little distinguishable difference. We will see how a large amount of tosses (e.g. 1000) changes posterior distribution under all three priors in the following discussion.

Following from these conclusions for the no prior knowledge and fair coin cases, for the biased prior of which the evolution of posteriors is shown in Figure 2.10, it is not surprising that an unexpected arrival of tails will decrease the expected value of θ and increase in uncertainty.

We observe that a new toss observation will be either too influential for changing posterior (scenario 1 and 3), or have no effect on updating posterior at all (scenario 2). We show the posterior distributions updated after 1000 tosses, under three priors. From the graphs, it notices that all posteriors are similar, especially the scenarios 1 and 3 which

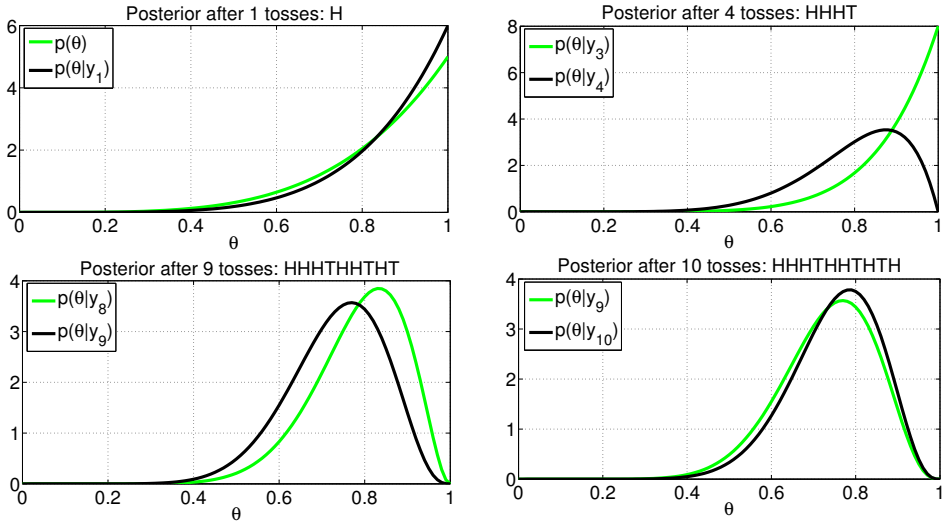


FIGURE 2.10: Posterior distribution $p(\theta|y_t)$ (shown as the red line) is evolved from the previous posterior distribution $p(\theta|y_{t-1})$) (shown as the blue line), under the biased prior setting.

are mostly identical to each other. The reason there is a difference between the posterior for scenario 2 and the other two is due to the fair coin prior having the most certainty (lowest variance), therefore, more data is needed to remove this strong belief and deliver the identical distributional shape.

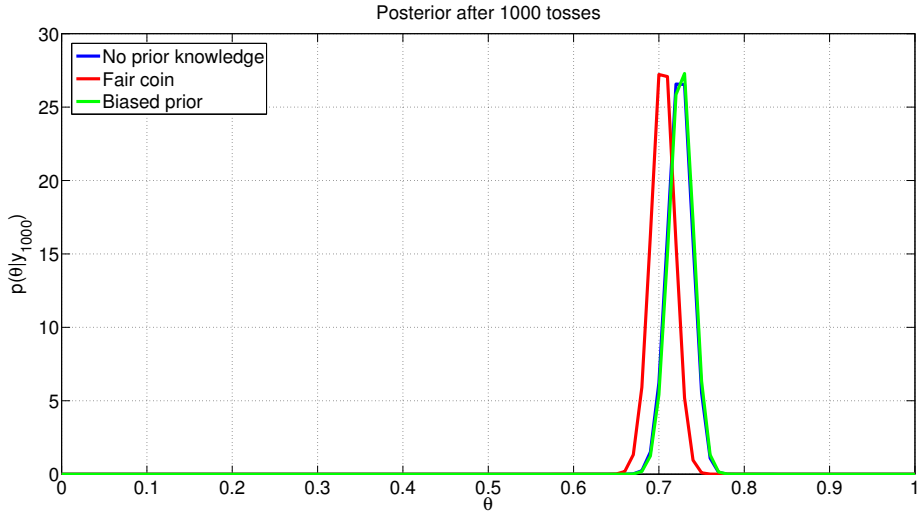


FIGURE 2.11: Posterior distributions updated after 1000 coins under all three prior settings.

In addition, as the number of observations increases, the initial guess (prior) contributes less to the posteriors and becomes unimportant. It makes perfect sense from the mathematical perspective, as the posterior is the result of multiplying the prior by the likelihood (ignoring the normalizing constant). As more and more data is involved in updating posterior, the likelihood becomes the product of individual likelihood for increasing ob-

servations, while the prior remains unchanged. Consequently, the likelihood dominates the posterior evolution.

2.4.3 Gaussian-Gaussian conjugated pair

The Gaussian-Gaussian is another widely used conjugated prior-likelihood pair. In the following, a discussion will be made on how posteriors are evolved with the arrival of observations, associated with the mathematical convenience of conjugated Gaussian-Gaussian pair.

Let us assume that a set of Gaussian random variables $\mathbf{y} = \{y_1, \ldots, y_N\}$ (these variables are assumed independent to each other) follows the distribution $\mathbf{y} \sim \mathcal{N}(\mu, \sigma^2)$, for which the mean μ is unknown and the variance σ^2 is fixed. The likelihood, that serves as a function to evaluate the probability of the observations given μ, is written as

$$p(\mathbf{y}|\mu) = \prod_{n=1}^{N} p(y_n|\mu) = \frac{1}{(2\pi\sigma^2)^{N/2}} \exp\left\{-\frac{1}{2\sigma^2}\sum_{n=1}^{N}(y_n - \mu)^2\right\}. \tag{2.49}$$

It is noticeable that the likelihood takes the form of the exponential of a quadratic form in μ. In order to pursue the mathematical convenience, we use the Gaussian distribution for the prior $p(\mu)$, given as

$$p(\mu) = \mathcal{N}(\mu|\mu_0, \sigma_0^2) = \frac{1}{\sqrt{2\pi}\sigma_0^2} \exp\left\{-\frac{(\mu - \mu_0)^2}{2\sigma_0^2}\right\}, \tag{2.50}$$

where μ_0 is the mean of $p(\mu)$ and corresponds to an initial arbitrary guess for μ. σ_0^2 is the variance of the prior and implies the reliability of this initial guess. This Gaussian prior is the conjugate distribution for the Gaussian likelihood function, which is due to the posterior being a product of two exponentials of quadratic functions of μ and so also is Gaussian. The relevant derivation is given in the following.

Following Bayes' rule shown in equation (2.43), we consider the log of the posterior

$$\ln p(\mu|\mathbf{y}) = \ln p(\mathbf{y}|\mu) + \ln p(\mu), \tag{2.51}$$

where the first term in right hand side can be further extended as

$$\ln p(\mathbf{y}|\mu) = -\frac{1}{2}\ln 2\pi\sigma^2 - \frac{1}{2\sigma^2}\sum_{n=1}^{N}(y_n - \mu)^2, \tag{2.52}$$

and the second term can also be written as

$$\ln p(\mu) = -\frac{1}{2}\ln 2\pi\sigma_0^2 - \frac{1}{2\sigma_0^2}(\mu - \mu_0)^2. \tag{2.53}$$

Substituting equations (2.52) - (2.53) into equation (2.51), we have

$$\begin{aligned}\ln p(\mu|\mathbf{y}) &= -\frac{1}{2\sigma^2}\sum_{n=1}^{N}(y_n-\mu)^2 - \frac{1}{2\sigma_0^2}(\mu-\mu_0)^2 + \text{const} \\ &= -\frac{1}{2\sigma^2}\sum_{n=1}^{N}\left(y_n^2 - 2y_n\mu + \mu^2\right) - \frac{1}{2\sigma_0^2}\left(\mu^2 - 2\mu\mu_0 + \mu_0^2\right) + \text{const} \\ &= -\frac{1}{2\sigma^2}\sum_{n=1}^{N}y_n^2 + \frac{\mu}{\sigma^2}\sum_{n=1}^{N}y_n - \frac{N\mu^2}{2\sigma^2} - \frac{\mu^2}{2\sigma_0^2} + \frac{\mu\mu_0}{\sigma_0^2} - \frac{\mu_0^2}{2\sigma_0^2} + \text{const}, \end{aligned} \tag{2.54}$$

after re-arrangement and grouping, we have

$$\ln p(\mu|\mathbf{y}) = \underbrace{\frac{N\mu}{\sigma^2}\sum_{n=1}^{N}y_n + \frac{\mu\mu_0}{\sigma_0^2}}_{\text{first-order terms}} \underbrace{-\frac{N\mu^2}{2\sigma^2} - \frac{\mu^2}{2\sigma_0^2}}_{\text{second-order terms}} + \text{const}. \tag{2.55}$$

The terms $-\frac{1}{2\sigma^2}\sum_{n=1}^{N}y_n^2$ and $-\frac{\mu_0^2}{\sigma_0^2}$ in equation (2.54) are irrelevant to μ, therefore these two terms are grouped into constant. In addition, as a function of μ, equation (2.55) is again a quadratic form, where the expression is completely characterized by the mean and variance. This rearrangement is so called 'completing the square', in which a quadratic form defining the exponent terms in Gaussian distribution is given, and the goal becomes to determine the corresponding mean and variance. This task can be achieved by mapping the concrete expression to the exponent in a general Gaussian distribution $\mathcal{N}(x|\mu,\sigma^2)$ given as

$$-\frac{(x-\mu)^2}{2\sigma^2} = -\frac{x^2}{2\sigma^2} + \frac{x\mu}{\sigma^2} + \text{const}. \tag{2.56}$$

Mapping the equation (2.55) to equation (2.56), denoting $p(\mu|\mathbf{y}) \sim \mathcal{N}(\mu_N, \sigma_N^2)$, we can have

$$\sigma_N^2 = \frac{1}{\frac{N}{\sigma^2}+\frac{1}{\sigma_0^2}} = \frac{\sigma^2\sigma_N^2}{N\sigma_0^2+\sigma^2} \tag{2.57}$$

$$\mu_N = \sigma_N^2\left(\frac{\mu_0}{\sigma_0^2} + \frac{N\sum_{n=1}^{N}y_n}{\sigma^2}\right) = \frac{\sigma^2}{N\sigma_0^2+\sigma^2}\mu_0 + \frac{N\sigma_0^2}{N\sigma_0^2+\sigma^2}\bar{\mathbf{y}}. \tag{2.58}$$

We next illustrate how the posterior distribution $p(\mu|\mathbf{y})$ changes under different priors or evolves by incorporating increasing numbers of observations. Supposing the observations follow a Gaussian distribution $\mathbf{y} \sim \mathcal{N}(5, 0.03)$, and similarly we also consider one strongly informative prior $\mu \sim \mathcal{N}(3,1)$ and one non-informative prior $\mu \sim \mathcal{N}(0,5)$.

Figure 2.13 shows the evolution of posteriors after observing one, four, seven and ten data. From the graphs, it is noticeable that the posterior distribution is 'shrunk' towards the real mean value $\mu = 5$ along with less uncertainty, as we observe more and

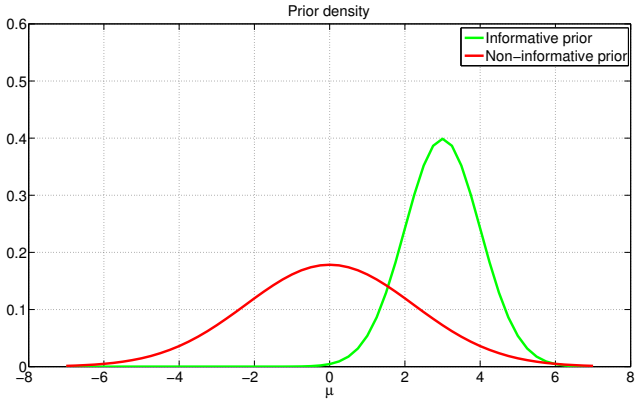


FIGURE 2.12: Prior densities for two different settings, where the informative and non-informative priors are shown as green and red lines, respectively.

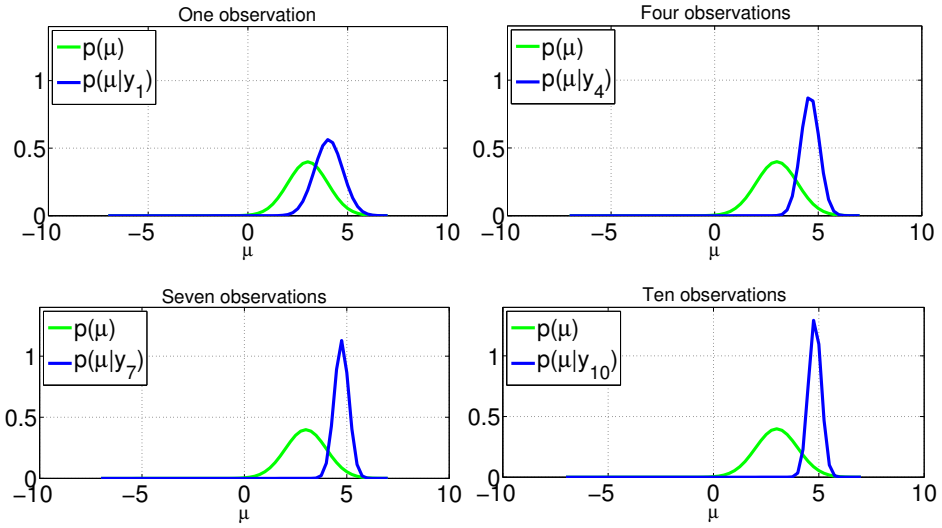


FIGURE 2.13: Posterior distributions are updated by incorporating different number of observations, under the informative prior.

more data. Moreover, the posteriors after seeing seven and ten observations are barely distinguishable, this is because the term $\bar{\mathbf{y}}$ in the mean of posterior shown in equation (2.58) becomes stable and changes slightly, when a certain amount of observations is used to update the posterior.

Comparing to the informative prior, we carry out the same investigation on the non-informative prior case, for which the results are shown in Figure 2.14. As we expected, the evolved posterior after observing several data points becomes stable and the final density implies that five is most likely to be the value of mean of observations $\mu = 5$. Even though two different priors are involved to examine how posteriors change with respect to the increasing observations, the effect of informative/non-informative prior on posterior evolution is diminished as more data is seen.

In the following discussion, we illustrate how the posterior changes in response to the different types of observations. Let us assume the observation in this example is vector-

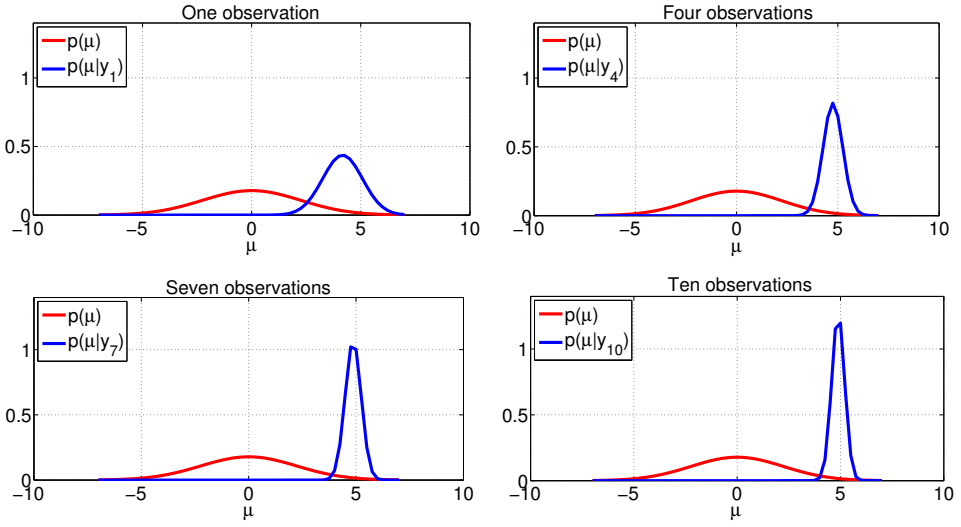


FIGURE 2.14: Posterior distributions are updated by incorporating different number of observations, under the non-informative prior.

valued data, more specifically, two observations, $\mathbf{y}_1 \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $\mathbf{y}_2 \sim \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$. Given a non-informative prior distribution as $\boldsymbol{\mu} \sim \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0) = \mathcal{N}(0, 10\mathbf{I})$ where $\mathbf{I}$ denotes the identity matrix, and equally reliable observations through covariance as $\boldsymbol{\Sigma}_{y,1} = \boldsymbol{\Sigma}_{y,2} = 0.01\mathbf{I}$, we can have posterior distribution $p(\boldsymbol{\mu}|\mathbf{y}_1, \mathbf{y}_2)$ as shown in Figure 2.15.

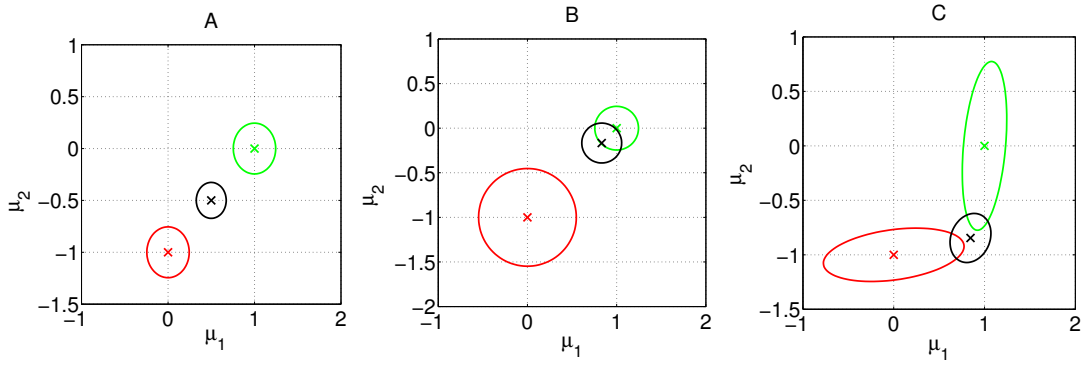


FIGURE 2.15: Posterior distribution $p(\boldsymbol{\mu}|\mathbf{y}_1, \mathbf{y}_2)$ is inferred by incorporating observations which are generated by fixed means $\boldsymbol{\mu}_1 = (0, -1)$ and $\boldsymbol{\mu}_2 = (1, 0)$. The densities of first, second observations and posterior are shown as the red, green and black circles, respectively. A: Equally reliable observations. B: The second observation $\mathbf{y}_2$ is more reliable than the first observation $\mathbf{y}_1$. C: $\mathbf{y}_1$ is more reliable in the vertical direction, while $\mathbf{y}_2$ is more reliable in the horizontal direction.

From the Figure 2.15.A, when the components in observation have the same level of

certainty or uncertainty, the posterior lies between the densities of the two observations. Conversely, when $\mathbf{y}_1$ is less reliable than $\mathbf{y}_2$ via setting $\mathbf{\Sigma}_{y,1} = 0.1\mathbf{I}$ and $\mathbf{\Sigma}_{y,2} = 0.01\mathbf{I}$, we can see from Figure 2.15.B that the posterior moves towards the density of $\mathbf{y}_2$. Moreover, different components can have a specific 'preferred' direction, for instance we define $\mathbf{y}_1$ has less uncertainty on the vertical direction in comparison to $\mathbf{y}_2$ and $\mathbf{y}_2$ is more reliable on the horizontal direction. This can be achieved by setting the covariance matrix as

$$\mathbf{\Sigma}_{y,1} = 0.01 \begin{pmatrix} 10 & 1 \\ 1 & 1 \end{pmatrix}, \quad \mathbf{\Sigma}_{y,2} = 0.01 \begin{pmatrix} 1 & 1 \\ 1 & 10 \end{pmatrix}. \tag{2.59}$$

By favoring different components in observations, the posterior shown as Figure 2.15.C implies that components will contribute more to posterior in the directions with less uncertainty.

2.4.4 Other distributions in the exponential family

In general, the posterior distribution $p(\theta|y)$ has no closed-form expression, and it is problematic to compute such a distribution due to the intractable integral for the normalizing constant $p(y)$. Thus much research uses conjugate priors to yield an analytical solution to the posterior, such that it can be computed without evaluating the denominator in Bayes' rule. Apart from the aforementioned beta-binomial and Gaussian-Gaussian conjugate prior-likelihood pairs, there are some more common conjugate pairs and we briefly introduce some of them in the following.

In section 2.4.3, we have introduced the use of Gaussian-Gaussian conjugate pair to infer the mean of Gaussian given a set of random Gaussian variables with known variance. We now wish to infer the variance with a fixed mean. Recalling that the likelihood, when used as a function of variance σ^2, can be rewritten as

$$\begin{aligned} p(\mathbf{y}|\sigma^2) &= \prod_{n=1}^{N} \mathcal{N}(y_n|\mu, \sigma^2) \\ &\propto \frac{1}{\sigma^N} \exp\left\{ -\frac{1}{2\sigma^2} \sum_{n=1}^{N} (y_n - \mu)^2 \right\}. \end{aligned} \tag{2.60}$$

For the sake of simplicity, we use the precision that is defined as $\lambda \equiv \frac{1}{\sigma^2}$ to describe Gaussian distribution, rather than the variance σ^2. Bishop (2006) suggests that the *gamma* distribution should be adopted as the conjugate prior for this likelihood, which is proportional to the power of λ and the exponential of a linear function of λ. The gamma distribution is written as

$$\mathrm{Gam}(\lambda|a, b) = \frac{1}{\Gamma(a)} b^a \lambda^{a-1} \exp(-b\lambda), \tag{2.61}$$

where $\Gamma(a)$ is the gamma function that is introduced in section 2.4.1 ensuring that the gamma distribution is correctly normalized. Now considering a $\text{Gam}(\lambda|a_0, b_0)$ as prior distribution, multiplying by the likelihood shown in equation (2.60), we can write the expression of posterior distribution as

$$p(\lambda|\mathbf{y}) = \lambda^{a_0-1}\lambda^{N/2}\exp\left\{-b_0\lambda - \frac{\lambda}{2}\sum_{n=1}^{N}(y_n-\mu)^2\right\}. \tag{2.62}$$

This expression can be formed into the distribution $\text{Gam}(\lambda|a_N, b_N)$. By noting the terms with respect to equation (2.61), we have

$$p(\lambda|\mathbf{y}) \sim \text{Gam}(\lambda|a_N, b_N) \tag{2.63}$$

$$a_N = a_0 + \frac{N}{2} \tag{2.64}$$

$$b_N = b_0 + \frac{1}{2}\sum_{n=1}^{N}(y_n-\mu)^2. \tag{2.65}$$

From equations (2.63) - (2.65), one can interpret that the curvature of posterior is increased by adding values $N/2$ to prior parameter a_0. Additionally, b implies how wide the gamma distribution spreads out, and b_N therefore quantifies the uncertainty of posterior after having N observations. Rather than working with the precision λ, if the variance σ^2 is used in the likelihood function, its corresponding conjugate prior is called the *inverse gamma distribution*. Here we work with the precision, due to its convenience for denotation.

We now progressively make the task harder, that is how to choose a conjugate prior if both the mean and the precision are unknown. Following the hints for finding the appropriate prior in aforementioned examples, we first decompose the likelihood into the functions related to μ and λ

$$\begin{aligned} p(\mathbf{y}|\lambda) &= \prod_{n=1}^{N}\left(\frac{\lambda}{2\pi}\right)\exp\left\{-\frac{\lambda}{2}(y_n-\mu)^2\right\} \\ &\propto \left[\lambda^{1/2}\exp\left(-\frac{\lambda\mu^2}{2}\right)\right]^N \exp\left\{\lambda\mu\sum_{n=1}^{N}y_n - \frac{\lambda}{2}\sum_{n=1}^{N}y_n^2\right\}. \end{aligned} \tag{2.66}$$

We now encode the joint distribution $p(\mu, \lambda)$ as the same form as likelihood shown in equation (2.66), to identify the parametric settings for conjugate prior.

$$\begin{aligned} p(\mu,\lambda) &= p(\mu|\lambda)p(\lambda) \\ &\propto \underbrace{\exp\left\{-\frac{\beta\lambda}{2}(\mu - c/\beta)^2\right\}}_{\text{first component}} \underbrace{\lambda^{\beta/2}\exp\left\{-\left(d - \frac{c^2}{2\beta}\right)\lambda\right\}}_{\text{second component}}. \end{aligned} \tag{2.67}$$

From equation (2.67), we can therefore define $p(\mu|\lambda)$ and $p(\lambda)$ as Gaussian and gamma

distributions, respectively. More specifically, the first component in equation (2.67) encodes the distribution $p(\mu|\lambda)$ and the second component corresponds to the distribution $p(\lambda)$. Consequently, we have

$$p(\mu|\lambda) = \mathcal{N}(\mu|\mu_0, (\beta\lambda)^{-1}) \tag{2.68}$$

$$p(\lambda) = \text{Gam}(\lambda|a, b), \tag{2.69}$$

where $\mu_0 = c/\beta$, $a = 1 + \beta/2$, $b = d - c^2/2\beta$ and c, d and β are user-specific constants. The distribution shown in equation (2.68) is known as *Gaussian-gamma* distribution. Figure 2.16 shows some plots for Gaussian-Gamma distribution obtained with different prior settings, from which we can observe that a in Gamma distribution controls the shape of density, while the parameter b influences the level of distribution spread. Notice

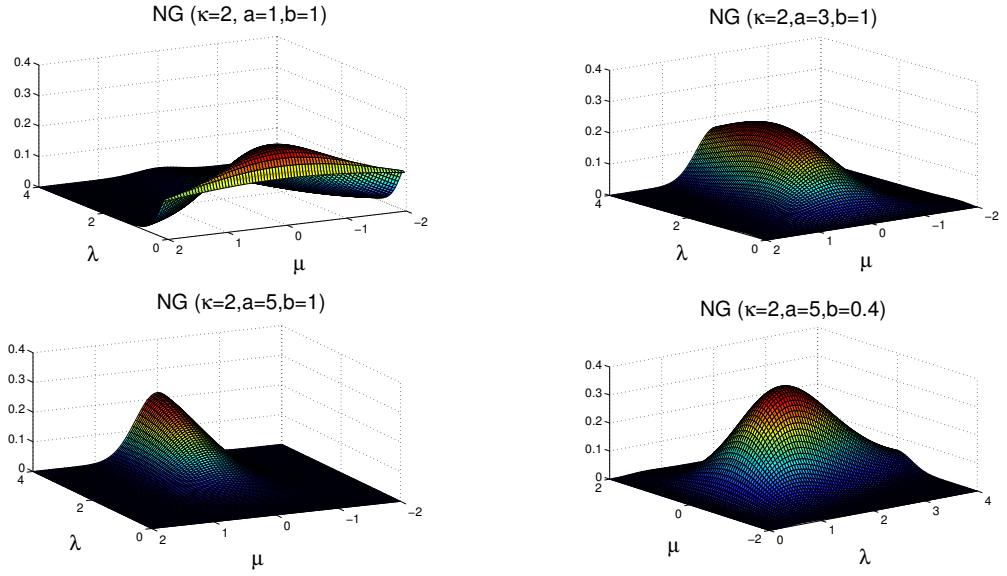


FIGURE 2.16: Gaussian-Gamma distributions. Graphs are plotted with different prior settings.

that this distribution holds a coupling property between the precision of μ and the value of λ, as the precision of mean μ for Gaussian-gamma distribution is a function of λ.

When the random Gaussian variables follow the multivariate Gaussian distribution $\mathbf{X} \sim \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1})$, with an unknown mean and known precision, the conjugate prior is another Gaussian distribution. If the mean is known, while the precision is unknown, then the conjugate prior is *Wishart* distribution. When both the mean and the precision are unknown, following the same principle of reasoning to the univariate case, the conjugate prior is known as *Gaussian-Wishart* distribution. More details about these conjugated pairs can be found in Bishop (2006).

The gamma distribution is also used as the conjugate prior for Poisson distribution, of which the major use is to model the incidence rate of disease in epidemiology. In general, Poisson distribution quantifies the probability of numbers of events occurring

in a fixed duration or space. Assuming all events occurring with a known average rate and independently in the time being, the Poisson distribution likelihood function is written as

$$\begin{aligned} p(\mathbf{y}|\theta) &= \prod_{n=1}^{N} \frac{1}{y_n!} \theta^{y_n} e^{-\theta} \\ &\propto \theta^{\sum_{n=1}^{N} y_n} e^{N\theta} \\ &\propto \theta^{S} e^{N\theta}, \end{aligned} \tag{2.70}$$

where $S = \sum_{n=1}^{N} y_n$ and $\mathbf{y}$ a positive integer vector, e.g. $y_n = 0, 1, 2, \ldots$, and parameter θ defines the rate of occurrence, In order to have the posterior distribution $p(\theta|\mathbf{y})$ in the same form as prior, following the derivation given in Gelman et al. (2013), we therefore define the conjugated prior for Poisson likelihood as

$$p(\theta) \propto \theta^{a-1} e^{-b\theta}, \tag{2.71}$$

noticing that equation (2.71) has the identical form of gamma distribution with parameters a and b. Similarly to the beta-binomial case, here we can express posterior as $p(\theta|\mathbf{y}) \sim \text{Gam}(\mathbf{y}|a + NS, b + N)$.

Apart from introduced distributions, there are some more widely used conjugate prior-likelihood pairs such as Dirichlet-Multinomial. Even though the conjugacy provides several attractive advantages such as computational convenience and intuitive interpretation, however, both prior and likelihood have to be characterized according to the problems of interest. More specifically, the prior should be set in accordance with how well we know about problems before seeing any data, any constraints applied on prior and whether the underlying system is discretized or continuous. Likewise, under the state-space models, the likelihood function is written in response to the type of noise corrupting observations. It can also be expected that increasing observed data can gradually remove the effect of prior on determining posterior. Consequently, the inference should be performed on priors which satisfy reality, rather than for computational convenience.

2.5 Inference methods

Computational modeling of biological systems is about quantitatively describing biochemical systems using differential equations (Kitano, 2002). Knowledge of biological processes is captured in these equations, and when their solutions match measurements of the system of interest it helps confirm our understanding of mechanism at the systems level. Examples of such models include cell cycle progression (Chen et al., 2000), integrate and fire generation of heart pacemaker pulses (Zhang et al., 2000) and cel-

lular behavior in synchrony with the circadian cycle (Leloup and Goldbeter, 2003). A particular appeal of modeling is that models can be interrogated with *what if* type questions to improve our understanding of the system, or can be used to make quantitative predictions in domains in which measurements are unavailable.

A central issue in developing computational models of biological systems is setting states such as concentration of proteins, metabolite and mRNA, and parameters such as rate constants of biochemical reaction, synthesis and decay rates of macromolecules, delays incurred in transcription of genes and translation of proteins, and sharpness of nonlinear effects (Hill coefficient). Parameter values are usually determined by conducting *in vitro* experiments (Niedenthal et al., 1996; Wadsworth et al., 2001; Tseng et al., 2002; Wiedenmann et al., 2004). When parameter values are not available from experimental measurements, modelers often resort to hand-tuning during the model development process and publish the range of values of a parameter required to achieve a match between model output and observed data. In this setting, however, we encounter two difficulties. First, parameters measured by *in vitro* experiments may be far away from those obtained *in vivo*. Second, some parameters of the system may not be amendable in the experimental setting. These limitations motivate the use of statistical tools to infer the unknown parameters in the systems.

As previously discussed, one way of setting parameters systematically is based on techniques for search and optimization. For example, Mendes and Kell (1998) compared several optimization based algorithms for estimating parameters along biochemical pathways and concluded that no single approach significantly outperforms other available approaches. Follow up work on a reasonably large system is described in Moles et al. (2003) where 36 parameters of a nonlinear biochemical dynamical model are optimized using stochastic global optimization using evolutionary strategies. Similar work on a developmental gene regulation circuit is described by Fomekong-Nanfack et al. (2009); Ashyraliyev et al. (2008), and a spline approximation based method for learning the parameters of enzyme kinetic model in a cell cycle system is described in Zhan and Yeung (2011).

An alternate approach is the use of probabilistic Bayesian formulations to quantify uncertainties in the process of estimating parameters. Work described by Golightly and Wilkinson (2005); Dewar et al. (2010); Barenco et al. (2006); Vyshemirsky and Girolami (2008); Jayawardhana et al. (2008); Lawrence et al. (2006) falls into this category. While maximum likelihood estimation has been the popular tool (Muller et al., 2004; Muller and Timmer, 2004; Baker et al., 2005; Bortz and Nelson, 2006) with probabilistic methods, approximate Bayesian treatment via variational Bayes (VB) and Markov Chain Monte Carlo methods have also been explored. With time varying or dynamical systems, some authors have pointed out advantages of sequential estimation models, formulating the problem as state and parameter estimation in a state-space modelling framework (Quach et al., 2007; Sun et al., 2008; Lillacci and Khammash, 2010). Kalman filtering and its

variants, and nonparametric particle filtering have been applied in this setting (Liu and Niranjan, 2012; Nakamura et al., 2009; Yang et al., 2007).

2.5.1 Approximate Bayesian computation methods

Apart from the deterministic approximation methods, in the batch category, a class of data-driven algorithms, namely approximate Bayesian computation algorithms (ABC), has been widely used to address the parameter estimation problem. ABC methods are ideal for systems for which it is possible to synthesize data but the computation of likelihood is computational expensive or intractable, therefore, such class of methods is also known as likelihood-free.

Pritchard et al. (1999) initially invented the ABC method associated with a simple rejection criterion, in order to tackle the parameter estimation problem in genetics. Since the coalescent tree that is used for conveying the inheritance relationships between alleles of gene in genetics is of high dimensionality and the human chromosome dataset is large, all state-of-the-art methods were infeasible. As an alternate, Pritchard et al. (1999) evaluated the coalescent model and yielded sets of pseudo-data by using a collection of samples for parameters. If the current pseudo-data is identical to the real data, then the corresponding sample is accepted for representing the posterior of parameter, but otherwise rejected. The key conceptual idea of ABC depends on this acceptance/rejection scheme, which operates without the likelihood evaluation or other deterministic approximations. This simple rejection based ABC method (called ABC-rejection) is only feasible for systems with low parameter dimensionality. To alleviate this issue, Wall (2000) considered a distance metric to quantify the discrepancy the real dataset and pseudo-data, associated with a threshold ϵ. According to Wall (2000), if the distance between the pseudo-observation and the real data is lower than the threshold ϵ, then the corresponding sample can be treated as an estimate of the parameter.

Even though the concepts of distance metric and threshold were introduced to enhance the capability of ABC-rejection, the tuning of threshold is tricky, as one needs to compromise between accuracy and computational expense by tweaking this threshold. Beaumont (2003) proposed their ABC method, which uses a local weighted regression adjustment to ABC-rejection, which is called ABC-regression. This method projects all estimations from ABC-rejection into a regression model and operates a local adjustment to improve the posterior estimates.

However, both ABC methods have a serious deficiency in that their computational efficiency dramatically decreases if the prior distribution for generating the samples is non-informative. A possible solution, Marjoram et al. (2003) developed their treatment of ABC in context of MCMC, namely ABC-MCMC. Although the transition chain moves samples around the space, as long as a significant number of iterations are run, samples

are guaranteed to converge to the target distribution. However, this ABC method inherits the disadvantages of MCMC including slow chain mixing and dependence on the covariance of transition.

A more powerfully ABC based method was introduced by Sisson et al. (2007), which merges the previous innovative SMC sampler (Del Moral et al., 2006) in the ABC setting, known as ABC-PRC. Despite ABC-PRC overcoming the problems evident in ABC-MCMC, its original version violated the condition of SMC sampler and thus leaded to a biased estimation. In this case, ABC algorithms such as ABC-SIS (Toni et al., 2009), ABC-PMC (Beaumont et al., 2009) and ABC-SMC (Del Moral et al., 2012) which use a similar framework as ABC-PRC have been proposed. Also, Sisson et al. (2009) corrected their ABC-PRC method.

Particularly, the capability of ABC methods on parameter estimation in the context of biological systems was highlighted by Toni et al. (2009). In Chapter 3, we thoroughly review this rich collection of algorithms and provide the empirical suggestion of using these approaches in systems biology. Meanwhile, as the discussion seen in chapter 3, ABC type methods, being data-driven solutions, require fine tuning of the acceptance criterion to strike a balance between accuracy and computational complexity. This dilemma, however, motivates the algorithm proposed in Chapter 4.

2.6 Sensitivity analysis

From the perspective of model, parameters, or combinations of parameters, can be classified as sloppy or stiff with respect to how sensitive the outputs respond to the variations of parameter values. Sensitivity analysis is a class of techniques for quantifying the properties of parameters in terms of stiff or sloppy, and which has been applied to various areas such as economic modeling for decision making (Triantaphyllou and Sánchez, 1997), social sciences (Kennedy, 2003; Saisana et al., 2005), chemistry (Saltelli et al., 2005; Komorowski et al., 2011) and engineering (Becker et al., 2011). We focus on its use on systems biology, and briefly introduce it below.

Local sensitivity analysis

Sensitivity analysis can be carried out locally or globally. From a practical perspective, the local methods carry out the sensitivity analysis by calculating the partial derivatives of system dynamics with respect to parameters, associated with operating points in the parameter space. Sensitivity index is often used as the quantitative measurement of the sensitivity, and its definition is given as

$$S_i = \frac{\partial \mathbf{x}_i}{\partial \theta} = \lim_{\Delta\theta \to 0} \frac{\mathbf{x}_i(\theta + \Delta\theta) - \mathbf{x}_i(\theta)}{\Delta\theta} \cong \frac{\mathbf{x}_i(\theta + \Delta\theta) - \mathbf{x}_i(\theta)}{\Delta\theta}, \tag{2.72}$$

where $\mathbf{x}_i$ is the ith system output, θ is the parameter of interest and Δ is the variation considered in parameter. It can be seen from this definitional derivative that an accurate approximation can be achieved by using a significantly small $\Delta\theta$, however, the choice of $\Delta\theta$ value is heavily empirical.

Dickinson and Gelinas (1976); Rabitz et al. (1983) proposed an alternative way to compute the sensitivity index, in which a differential equation of sensitivity index is constituted, given as

$$\frac{\partial S_i}{\partial t} = \frac{\partial}{\partial t}\left(\frac{\partial \mathbf{x}_i}{\partial \theta}\right) = \frac{\partial}{\partial \theta}\left(\frac{\partial \mathbf{x}_i}{\partial t}\right) = \frac{\partial \dot{\mathbf{x}}}{\partial \theta}, \tag{2.73}$$

where $\dot{\mathbf{x}}$ is the ODEs of the dynamics of interest. This differential equation is often numerically solvable. Dickinson and Gelinas (1976); Rabitz et al. (1983) empirically suggested that the initial conditions of equation 2.73 are required to assign the reasonable values based on the relationship between θ and $\mathbf{x}_0$.

A closely related method of sensitivity analysis used in some biological problems is metabolic control analysis which aims to measure the dependence of state variables in metabolic networks, e.g fluxes and species concentrations, on kinetic parameters. This dependence is quantitatively described by the control coefficient, which is defined as

$$C_\theta^{\mathbf{x}} = \frac{\theta}{\mathbf{x}} \times \frac{\partial \mathbf{x}}{\partial \theta} \tag{2.74}$$

This method was derived to quantify the metabolic networks (Kacser and Burns, 1973; Heinrich and Rapoport, 1974; Fell and Sauro, 1985; Reder, 1988), and it was subsequently applied to study other biological processes, e.g. cell signaling (Ihekwaba et al., 2004) and genetic networks (Swameye et al., 2003).

In addition, an innovative sensitivity analysis technique based on principal component analysis (PCA) was proposed for analyzing sensitivity by given the inferred particles for unknown parameters (Toni et al., 2009). This method has applied to the repressilator gene regulatory network and the MAP kinase signaling pathway (Secrier et al., 2009), and successfully discriminated the sloppy and stiff parameters of these two systems. However, the reliability of this approach depends on the precision of inference. If the parameters of the system are inaccurately estimated, the sensitivity analysis result is not robust. Due to this limitation, we cannot use PCA-based sensitivity analysis technique in our proposed inference method described in chapter 4, since our approach requires to provide the reliable sensitivity analysis result given the relatively low precision inference. In addition, we carry out a comparative study on this PCA-based method and details can be found in section 4.4.

Global sensitivity analysis

It is evident that the local sensitivity analysis techniques are heavily dependent on the

operating points considered. However, due to the sophisticated mechanisms in most biological systems, values of parameters can be greatly varied and precise determination is impossible. which may lead to unreliable results. In order to overcome this difficulty, sensitivity analysis techniques are then carried out with tolerance for the substantial variation of parameters, results obtained within this strategy are classified as the global sensitivity analysis.

A straightforward global sensitivity analysis technique is to assign the parameter values by picking the samples from a particular range so as to mimic variations in reality. This simple scheme is inefficient for generating samples that fully cover the parameter space. Alternatively, Latin hypercube sampling was proposed by McKay et al. (1979); Iman et al. (1981), in which the parameter space is split into equal intervals and samples from each individual are drawn with equal probability. This grid sampling method outperforms the random scheme in terms of diversity, since the randomly drawn samples may neglect some intervals in space.

By assuming a monotonic relationship between parameters and system outputs, the correlation coefficient (CC) also plays its role in sensitivity analysis and is calculated as

$$r_{\theta,\mathbf{y}} = \frac{\mathrm{Cov}(\boldsymbol{\theta}, \mathbf{y})}{\sqrt{\mathrm{Var}(\boldsymbol{\theta})\mathrm{Var}(\mathbf{y})}} = \frac{\sum_{i=1}^{N}(\theta_i - \bar{\boldsymbol{\theta}})(\mathrm{y}_i - \bar{\mathbf{y}})}{\sqrt{\sum_{i=1}^{N}(\theta_i - \bar{\boldsymbol{\theta}})^2 \times \sum_{i=1}^{N}(\mathrm{y}_i - \bar{\mathbf{y}})^2}}, \tag{2.75}$$

where N is the number of samples picked, and $\bar{\boldsymbol{\theta}}$ and $\bar{\mathbf{y}}$ are mean of samples and system outputs, respectively. Value of correlation coefficient is between -1 and 1, where a positive coefficient implies that an increase in the parameter values leads to a corresponding growth of the system outputs. In contrast, negative value indicates that an increase in parameter value decreases system outputs. Based on this concept, rank correlation coefficient (RCC) that is obtained by using the rank-transformed data, and partial rank correlation coefficient (PRCC) whose coefficient is calculated after eliminating the linear effects on $\mathbf{y}$ caused by parameters except the underlying one, are further involved in sensitivity analysis. Details can be found at Saltelli et al. (2008); Marino et al. (2008) The family of correlation coefficient methods were successfully applied to study the TCR signaling pathway (Zheng and Rundell, 2006), an HIV model (Blower and Dowlatabadi, 1994; Blower et al., 2000) and pharmacokinetic model (Kiparissides et al., 2009).

Sobol' method, was invented by Sobol' (1990) to handle the sensitivity problem for nonlinear and non-monotonic systems. The idea of Sobol' method is to decompose system dynamics $f(\boldsymbol{\theta})$ into summands of various simplified functions which take different combinations of parameters in increasing dimensionality. This can be mathematically described as

$$f(\boldsymbol{\theta}) = f_0 + \sum_{i=1}^{n_p} f_i(\theta_i) + \sum_{i=1}^{n_p}\sum_{j=i+1}^{n_p} f_{ij}(\theta_i, \theta_j) + \ldots + f_{1,\ldots,n_p}(\theta_1, \theta_2, \ldots, \theta_{n_p}), \tag{2.76}$$

where f_0 is a constant. The total variance D is defined as

$$D = \int_{\boldsymbol{\Theta}^{n_p}} f^2(\boldsymbol{\theta}) d\boldsymbol{\theta} - \left(\int_{\boldsymbol{\Theta}^{n_p}} f(\boldsymbol{\theta}) d\boldsymbol{\theta} \right)^2, \tag{2.77}$$

where the integrals of each summand over its own variables is zero, that is the second term in equation 2.77 is cancelled (Saltelli and Bolado, 1998). Additionally, by partitioning the system dynamics, then partial variances can be individually calculated by following

$$D_{i_1 i_2 \ldots i_s} = \int \ldots \int f^2(\theta_{i_1}, \theta_{i_2}, \ldots, \theta_{i_s}) d\theta_{i_1} d\theta_{i_2} \ldots d\theta_{i_s} \tag{2.78}$$

The approximation of this integral can be achieved with Monte Carlo integration, and then the sensitivity index can be computed as

$$S_{i_1 i_2 \ldots i_s} = \frac{D_{i_1 i_2 \ldots i_s}}{D} \tag{2.79}$$

This fraction explains about how total variance is apportioned to the individual kinetic parameter or their combinations. Higher value indicates that the corresponding parameter or combination is more crucial to characterize system behavior. Although Sobol' method can converge to the analytical solution (Kim et al., 2010), the dependence on Monte Carlo approximation may be highly computational-demanding (Zi, 2011).

Another variance-based approach, Fourier amplitude sensitivity test (FAST), provides a way to partially overcome the computational complexity of the Sobol' method. Specifically, FAST transforms system outputs to the frequency domain and characterizes kinetic behaviors by using Fourier series. By doing so, approximation of sensitivity index can be achieved by the summation of Fourier series. In similarly to Sobol' method, FAST analyzes the sensitivity index by using the Monte Carlo method, as a result, the computational complexity is much higher than other approaches where it needs to tradeoff between accuracy and efficiency in most cases. Comparing to Sobol' method, FAST can achieve the reliable sensitivity analysis result with cheaper computational expense, if only the first-order sensitivity index is required to compute (Saltelli and Bolado, 1998). Details of the first-order sensitivity index and FAST are given in chapter 4. Consequently, this method is adopted in our work to perform the sensitivity analysis.

2.7 Discussion

In this chapter, we outlined the main tasks which are consisted in systems biology. As the first step in the roadmap of systems biology, we set out to show the different forms of state-space models and their mathematical descriptions. Combined with the discretization technique, state-space models possess the ability to mimic the noisy observations

of biological systems.

From the perspective of the thesis, we roughly categorized the inference methods in terms of how they process data. We gave an overview of sequential methods, beginning with the most famous family of Kalman filters, the method ends up to a powerful sampling-based filtering technique, *i.e.* particle filter. Additionally, we also reviewed the approaches for estimating parameter in a batch fashion, which span a wide range from the most straightforward ML algorithm to the latest RMHMC algorithm, as well as modern methods, featuring the likelihood-free, ABC methods were introduced.

Chapter 3

Approximate Bayesian Computation Methods

This chapter is a review of Approximate Bayesian Computational (ABC) methods, a class of powerful algorithms for Bayesian inference that do not require explicit computation of likelihood. We provide a tutorial introduction to a variety of algorithms centered around the ABC idea and illustrate their relative performances on a Gaussian mixture model and two models of system biology: the widely used Lokta-Volterra model and the Heat Shock Response model considered in the previous chapter.

3.1 Basic ABC methods

The basic idea in ABC algorithms is to sample the unknown from a prior distribution, $\theta \sim \pi(\theta)$, synthesize data from the model under study, $\mathbf{X}^* \sim f(\mathbf{x}_0, \theta^*)$, where $\mathbf{x}_0$ is the initial condition and $f(\cdot, \cdot)$ is the model, and accept θ^* as a sample for the posterior if the synthesized data $\mathbf{X}^*$ is close enough in some sense to the observations $\mathbf{X}$. In our discussion with Systems Biology models, we will focus on $f(\cdot, \cdot)$ being a set of ordinary differential equations which can be numerically integrated. We will also use Euclidean distance between the synthesized and the observations as measure of discrepancy. Variety of the ABC frameworks are derived by considering different sampling strategies and adapting the threshold at which acceptance decisions are made.

In order to retain the consistency of notation throughout the descriptions in chapter 3, we assume that the dimension of parameter in system is one, i.e. $D_p = 1$, therefore, the scalar θ_t denotes a sample of the parameter at the time instant t and the collection of samples is described as $\boldsymbol{\theta}$. However, in most real world situations, the dimension of parameters in system is always greater than one, as a result, the scalar θ_t is naturally extended to a vector, denoted as $\boldsymbol{\theta}_t$, and the collection of samples becomes a matrix,

given as $\boldsymbol{\Theta}$.

3.1.1 ABC-Rejection algorithm

In its earliest form (Tavaré et al., 1997), the generated particle θ^* was accepted only if $\mathbf{X}^*$ was identical to the observations $\mathbf{X}$. It became immediately evident that this is an inefficient procedure because thousands of trails needed to be performed before accepting one of the generated particles. A modification to the scheme, introduced by Pitt and Shephard (1999) was to define a threshold ϵ and accept particles when the discrepancy between $\mathbf{X}^*$ and $\mathbf{X}$ was within this. This variant of the method is normally referred to as the ABC-rejection algorithm shown in Algorithm 1.

Algorithm 1 ABC-rejection

`Input:` $\pi(\theta)$, ϵ, $\mathbf{x}_0$, N_{rej}, $\mathbf{X}$, $\rho(\cdot,\cdot)$ and $f(\cdot,\cdot)$.
`Output:` $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_{N_{reg}}\}$

n=1
Repeat
1. Draw $\theta^* \sim \pi(\theta)$
2. Synthesize $\mathbf{X}^* \sim f(\mathbf{x}_0, \theta^*)$
3. Evaluate discrepancy $d = \rho(\mathbf{X}^*, \mathbf{X})$
4. **if** $d \leq \epsilon$ **then**
5. $\quad \theta_n = \theta^*$
6. $\quad n = n + 1$
7. **end if**

until $n = N_{rej}$

The procedure of ABC-rejection is illustrated in Figure 3.1, which shows the steps taken by ABC-rejection to approach the posterior distribution.

The function $\rho(\cdot,\cdot)$ in Algorithm 1 quantifies a distance between synthesized the true observations. The Euclidean distance is often chosen for this metric function in the context of systems biology, which is given by

$$\rho(\mathbf{X}^*, \mathbf{X}) = \sum_{i=1}^{N_{OT}} \|\mathbf{x}_i^* - \mathbf{x}_i\|, \tag{3.1}$$

where $\|\mathbf{x}_i^* - \mathbf{x}_i\|$ denotes the *norm* of error between $\mathbf{x}_i^*$ and $\mathbf{x}_i$, and N_{OT} is the number of data points of observations. Intuitively, tolerance ϵ plays a crucial role in performing the ABC-rejection. If ϵ approaches zero, the approximated posterior distribution becomes infinitesimally close to the target distribution. This is achieved at the expense of more computation, therefore, efficiency becomes an issue. Likewise high precision cannot be expected with tolerance ϵ increasing considerably.

Example 4.1 *Take the deterministic Lotka-Volterra model (Lotka, 1925; Volterra, 1926) as an example and which is also considered in* **Example 2.6** *of chapter 2. The ODEs*

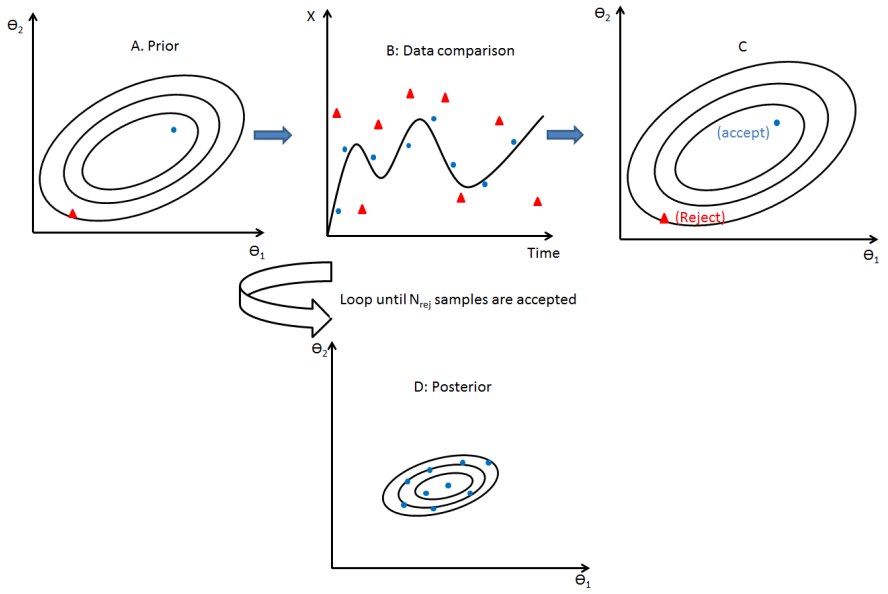


FIGURE 3.1: The procedure of ABC-Rejection algorithm: The graph.A shows the prior distribution of parameters (there are two parameters in this illustration i.e. $\boldsymbol{\theta} = [\theta_1, \theta_2]$) where the dot and triangle are samples drawn from the prior distribution. It is easy to see from the graph.B that is the simulation yielded from the dot parameter set is sufficiently close to the true data, thus the dot sample for parameter is accepted. In contrast the simulation from the triangle set mimic poorly the true data, and are therefore rejected. By running through the process, the posterior distribution is obtained and shown in the graph.D, and turns out to be narrower than the prior distribution.

of this Lotka-Volterra model are described as

$$\dot{\mathbf{x}}_1 = f(x_1) = \alpha x_1 - x_1 x_2, \quad \dot{\mathbf{x}}_2 = f(x_2) = x_1 x_2 - \beta x_2, \tag{3.2}$$

where x_1 and x_2 are two species in a system. In particular, observations of the system are corrupted by Gaussian noise which is generated from $\mathcal{N}(0, 0.05)$. The time length for synthesizing the pseudo-observations is 100 min, sampling at regular intervals of 0.2 minutes, that is the number of data for representing the pseudo-observations is 500. We generate the observations with the true model: $\alpha = \beta = 0.5$. Starting from the same prior distribution for picking particles, i.e $\pi(\boldsymbol{\theta}) \sim \mathcal{U}(0.2, 0.9)$, we examine the performance of ABC-rejection under different epsilons: $\epsilon = 300, 100, 20, 10$. The inference under each epsilon is carried out 10 times. The results of inferences are shown in Figure 3.2 where the ability of ABC-rejection to estimate parameters is varying considerably with respect to the epsilon value. The performance of ABC-rejection in terms of efficiency under

ϵ	Computational time (second)	Acceptance rate
300	4.434±0.11	100%±0
100	4.493±0.14	95.6%±1.6
20	154.4±18.1	2.43%±0.2
10	727.6±35.3	0.05%±0.02

different tolerances is summarized and listed in Table 3.1.1. We can conclude that if the

coarse acceptance criterion is employed, the algorithm appears as an efficient method with high acceptance rate, however, a few undesired particles are also accepted to represent the posterior. In contrast, the particles are able to narrowly circle around the true point when the harsh tolerance is taken; yet, this high precision requires more computational expense and the decline of acceptance rate is evident.

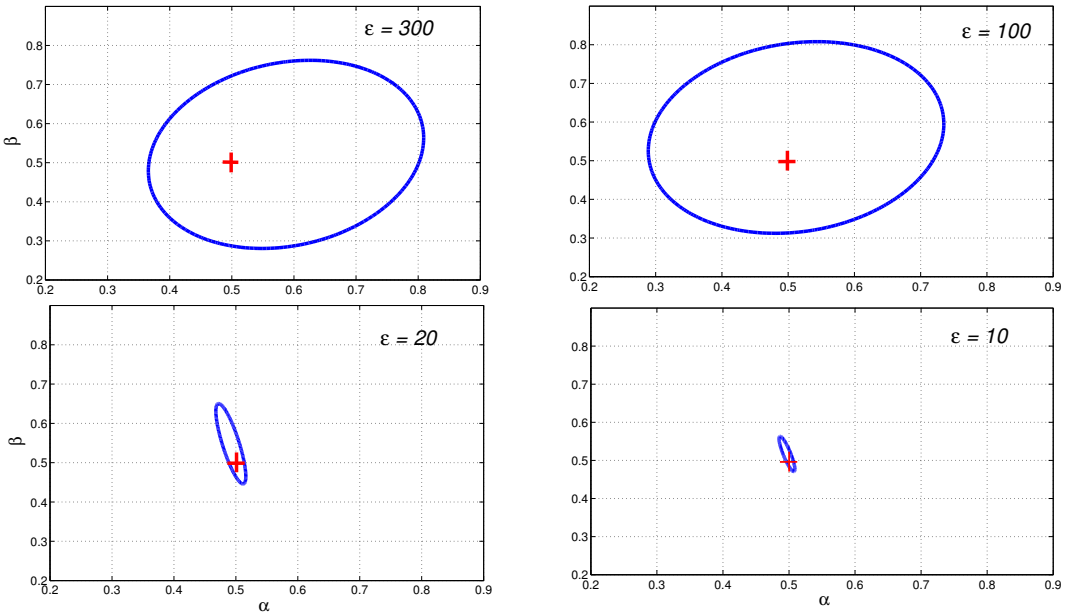


FIGURE 3.2: Illustration of the posterior distribution $p(\boldsymbol{\theta}|\mathbf{X})$ *(blue ellipses)* obtained under different tolerances ϵ, where the small *red cross* implies the point of true parameters.

This example illustrates the capability of ABC-rejection on parameter estimation without likelihood evaluation, however, it can be challenging to strike a balance between computational efficiency and accuracy.

The tradeoff made for acceptance rate and precision limits the widespread use of ABC-rejection. Beaumont (2003) introduced the modification of standard rejection ABC method associated with a local regression adjustment. This so-called ABC-regression method and ABC-rejection appear fundamentally in collecting samples for approximating the posterior distribution. With making use of the local correction, ABC-regression is allowed to afford a relatively large tolerance ϵ. The innovation of this algorithm is in characterizing the relationship between the collection of accepted samples and their corresponding discrepancies via a linear regression model. We escape the detail of ABC-regression method in this section, due to the high similarity. The derivation, interpretation, pseudo-code and illustrative example for ABC-regression can be found in the section A.9 of Appendix 6.

3.1.2 ABC-MCMC algorithm

Although the local regression process has the effectiveness for correcting estimate with only a negligible computational cost, an informative prior must be set for ABC-rejection and ABC-regression. When samples are generated from a non-informative prior distribution, thousands of attempts will be rejected leading to inefficiency. As a solution of this difficulty, Marjoram et al. (2003) performed MCMC algorithm in the ABC setting, namely the ABC-MCMC.

In ABC-MCMC, a Markov chain transition kernel $q(\theta^*|\theta_t)$ of invariant distribution $p(\theta_t|\mathbf{X})$ is allowed to apply on each sample so that $\int q(\theta^*|\theta_t)p(\theta_t|\mathbf{X}) = p(\theta^*|\mathbf{X})$, leading to the samples still being distributed according to the posterior of interest (Andriue et al., 2001). When the current distance between pseudo-observations and true dataset is less than the tolerance ϵ, the proposal is taken as the sample for parameter with an acceptance probability, defined as

$$h(\theta^*, \theta) = 1 \wedge \frac{p(\theta^*)q(\theta|\theta^*)}{p(\theta)q(\theta^*|\theta)} = \min\left(1, \frac{p(\theta^*)q(\theta|\theta^*)}{p(\theta)q(\theta^*|\theta)}\right). \tag{3.3}$$

The steps for executing ABC-MCMC algorithm are given in Algorithm 2.

Algorithm 2 ABC-MCMC

`Input`: $\theta_1 \sim \pi(\theta)$, ϵ, $\mathbf{x}_0$, N_{mcmc}, $\mathbf{X}$, $k(\cdot)$, $\rho(\cdot,\cdot)$ and $f(\cdot,\cdot)$.
`Output`: $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_{N_{mcmc}}\}$

t=1
Repeat
1. Move $\theta_t \rightarrow \theta^*$: $\theta^* \sim k(\theta_t)$ where $k(\cdot)$ is given as equation A.37
2. Synthesize $\mathbf{X}^* \sim f(\mathbf{x}_0, \theta^*)$
3. Calculate distance $d = \rho(\mathbf{X}^*, \mathbf{X})$
4. **if** $d \leq \epsilon$ **then**
5. generate indicator $u \sim \mathcal{U}(0, 1)$
6. **if** $u < \min\left(1, \frac{\pi(\theta^*)k(\theta_t|\theta^*)}{\pi(\theta_t)k(\theta^*|\theta_t)}\right)$ **then**
7. $\theta_{t+1} = \theta^*$
8. $t = t + 1$
9. **else**
10. $\theta_{t+1} = \theta_t$
11. $t = t + 1$
12. **end if**
13. **end if**

until $t = N_{mcmc}$

It can be seen that, apart from the acceptance criterion conducted by ABC, the proposed samples must additionally be accepted according to the MH acceptance probability. The use of this acceptance probability makes the Markov chain satisfy the detailed balance condition, and guarantees its convergence to the stationary distribution $p(\theta|\mathbf{X})$. We prove this in what follows, and these standard derivations are described in (Marjoram

et al., 2003).

Taking the notations previously mentioned, the detailed balance can be described

$$p(\theta|\mathbf{X})r(\theta \to \theta^*) = p(\theta^*|\mathbf{X})r(\theta^* \to \theta), \tag{3.4}$$

where $r(\theta \to \theta^*)$ is the MH transition mechanism, given as

$$r(\theta \to \theta^*) = k(\theta^*|\theta)p(\mathbf{X}|\theta^*)h(\theta, \theta^*). \tag{3.5}$$

Substituting equation 3.5 into the left-hand side of the detailed balance condition, we have

$$\begin{aligned}
p(\theta|\mathbf{X})r(\theta \to \theta^*) &= p(\theta|\mathbf{X})k(\theta^*|\theta)p(\mathbf{X}|\theta^*)h(\theta, \theta^*) \\
&= \frac{p(\mathbf{X}|\theta)\pi(\theta)}{p(\mathbf{X})}k(\theta^*|\theta)p(\mathbf{X}|\theta^*)h(\theta, \theta^*) \\
&= \frac{p(\mathbf{X}|\theta)\pi(\theta)}{p(\mathbf{X})}k(\theta^*|\theta)p(\mathbf{X}|\theta^*)\frac{\pi(\theta^*)k(\theta|\theta^*)}{\pi(\theta)k(\theta^*|\theta)} \\
&= \frac{p(\mathbf{X}|\theta^*)\pi(\theta^*)}{p(\mathbf{X})}p(\mathbf{X}|\theta)k(\theta|\theta^*) \\
&= p(\theta^*|\mathbf{X})p(\mathbf{X}|\theta)k(\theta|\theta^*) \\
&= p(\theta^*|\mathbf{X})p(\mathbf{X}|\theta)k(\theta|\theta^*)h(\theta^*, \theta) \\
&= p(\theta^*|\mathbf{X})r(\theta^* \to \theta),
\end{aligned} \tag{3.6}$$

which holds the equality.

Example 4.3 *Deploying all aforementioned methods to the Lokta-Volterra model, starting from a relatively non-informative prior, we attempt to illustrate the advantage of the ABC-MCMC. In this case, the initial* $\boldsymbol{\theta}_1$ *is generated from a Gaussian distribution* $\boldsymbol{\theta}_1 \sim \mathcal{N}(0.7, 0.01^2)$ *and the tolerance* ϵ *is set to 20. For illustrating the outperformance of ABC-MCMC, ABC-rejection draws the samples from the identical prior. As shown in Figure 3.3, the posterior estimate from ABC-MCMC converges to the true value from a relatively far place after a few iterations. Moreover, it can be seen from the contour of the posterior shown in the right panel of Figure 3.3 that ABC-rejection performs similarly in terms of accuracy. The gain of Markov transition kernel* $k(\cdot)$ *is evident in the computational efficiency, where ABC-MCMC takes approximately 1300 iterations with an acceptance rate of* 7.69% *to collect 100 samples, whereas 24000 iterations are taken by ABC-rejection to collect 100 samples, yielding an acceptance rate of* 0.42%*. We also note that even though the tolerance is set to a value that is identical to* **Example 4.1** ($\epsilon = 20$)*, however, ABC-rejection performed worse in this study in comparison to the results shown in Figure 3.2. This is because the prior distribution is non-informative in this simulation, while it is set to a relatively favorable prior (easier to generate samples that is close to the true values) in the previous study.*

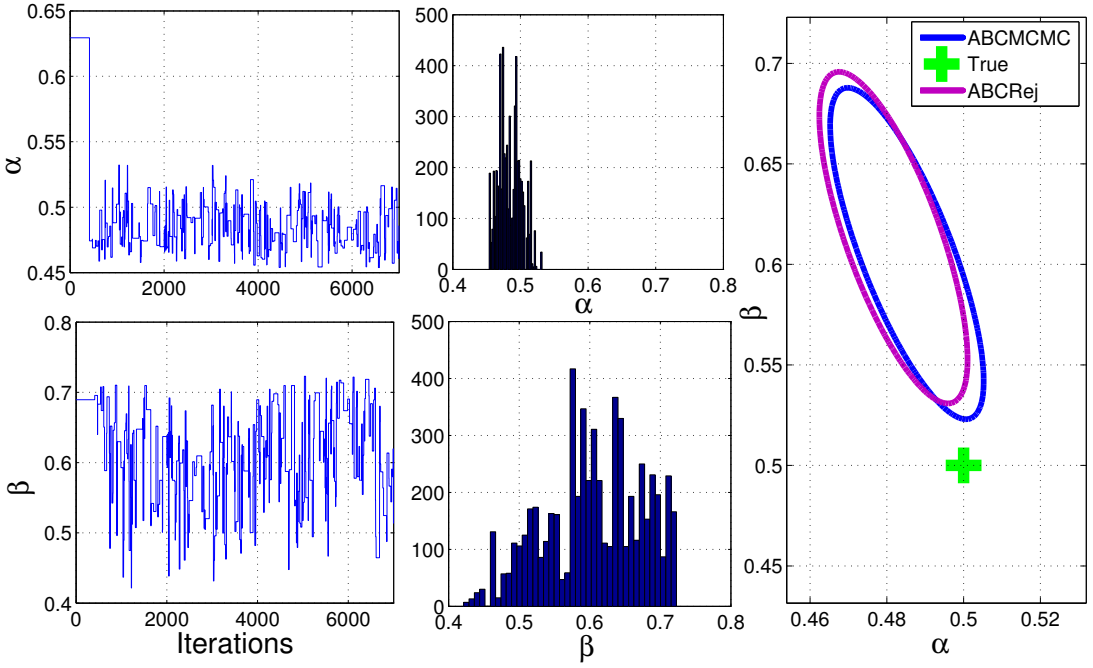


FIGURE 3.3: *First column:* the trajectories of estimations for α and β from the ABC-MCMC algorithm. *Second column:* the histograms of the inferred samples, notice that the estimates obtained in the *burn-in* phase are not included in the histograms. *Last column:* The comparative study of performances between ABC-MCMC and ABC-rejection in term of the contour of particles.

Interestingly, as shown in graphs, even though two parameters are simultaneously estimated, it becomes immediately apparent that the variance of samples for α is considerably lower than the one for β. This phenomenon is known as *sloppiness/stiffness*, and such property gives a rise to the concept of *sensitivity analysis*, which motivates our innovative ABC-based inference algorithm. This algorithm will be introduced in Chapter 4.

3.2 Advanced ABC methods

A variety of deficiencies, such as the simple sampling method, the curse of dimensionality and the slow Markov chain mixing, limit the widespread use of basic ABC methods on parameter estimation of biological systems (Beaumont, 2010). The advanced ABC methods were developed by focusing on determining the acceptance criterion, either deterministically or automatically. In addition, the adaptive transition kernel is also merged in the SMC based ABC methods.

3.2.1 ABC-partial rejection control (ABC-PRC) algorithm

ABC-MCMC is able to partially alleviate the constraint on informativeness of prior, however, it suffers from the slow Markov chain mixing. A toy example was considered by (Sisson et al., 2007; Beaumont et al., 2009; Del Moral et al., 2012), and we took this problem to illustrate this particular issue with ABC-MCMC, which is represented below.

Example 4.4 *Suppose it is of interest to estimate the mean μ of a mixture Gaussian model with a prior distribution $\pi(\cdot)$, given as*

$$f(x^*|\mu) = \frac{1}{2}\mathcal{N}(x^*;\mu,1) + \frac{1}{2}\mathcal{N}(x^*;\mu,0.01), \quad \pi(\mu) \sim \mathcal{U}(-10,10), \tag{3.7}$$

where $\mathcal{N}(x^*;\mu,\sigma^2)$ is the one-dimensional normal probability density function of mean μ and variance σ^2, evaluated at x^*, and $\mathcal{U}(a,b)$ is the uniform distribution on the interval $[a,b]$. The true observation x is assumed to be zero (i.e. $x = 0$), therefore, the true posterior distribution is given as

$$p(\mu|x) \propto p(x|\mu)p(\mu) = \{\mathcal{N}(0;\mu,1) + \mathcal{N}(0;\mu,0.01)\}\,\mathcal{I}_{[-10,10]}(\mu). \tag{3.8}$$

where $\mathcal{I}_{[-10,10]}(\mu)$ is the indicator function, returning one if μ is in the interval $[-10,10]$, otherwise, zero. In this toy example, the $\mathbb{L}_1$ distance is used as the function to measure the discrepancy between x^* and x, since $x = 0$, $\rho(x^*,x) = |x^* - x| = |x^*|$. In the ABC setting, the posterior distribution is approximated by the tolerance, given as

$$\begin{aligned} p_\epsilon(\mu|x) &\propto p(\rho(x,x^*) \le \epsilon|\mu)p(\mu) \\ &= p(|x^*| \le \epsilon|\mu)\mathcal{I}_{[-10,10]}(\mu) \\ &= p(-\epsilon \le x^* \le \epsilon|\mu)\mathcal{I}_{[-10,10]}(\mu), \end{aligned} \tag{3.9}$$

where $p(-\epsilon \le x^* \le \epsilon|\mu)$ indicates the region $p(x|\mu)$ that is between $p(-\epsilon|\mu)$ and $p(\epsilon|\mu)$. Such region is blacked in Figure 3.4, and it can be counted via the cumulative distribution. For the mathematical convenience, we shift the distribution of interest whose mean is μ to the zero mean Gaussian distribution. Such shifting can be done by subtracting the original variables by μ, i.e. $\mu - \mu = 0$. Additionally, it is due to the function of interest is a mixture Gaussian model, as a result of which we need to shift all distributions to zero mean Gaussian. After the shifting, this region can be calculated as

$$p(-\epsilon \le x^* \le \epsilon|\mu) = \Phi(\epsilon-\mu) - \Phi(-(\epsilon+\mu)) + \Phi(10(\epsilon-\mu)) - \Phi(-10(\epsilon+\mu)) \tag{3.10}$$

where $\Phi(\cdot)$ is the cumulative distribution function of the normal Gaussian distribution. For moving the samples, the MH transition kernel following the random walk is given as

$$k(\mu^*|\mu_n) \sim \mathcal{N}(\mu_n, 0.15^2). \tag{3.11}$$

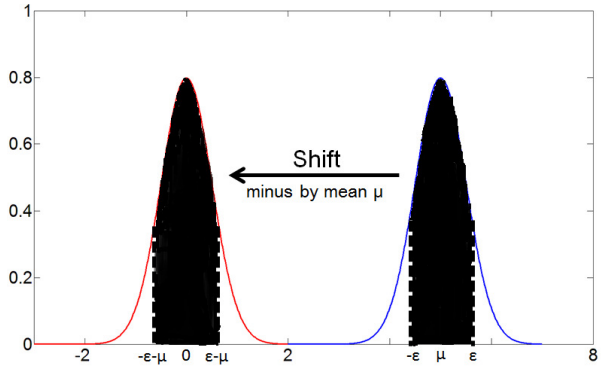


FIGURE 3.4: Illustrative graph shows how to calculate the region of interest after shifting by misusing the mean μ.

Results shown in Figure 3.5 were obtained by setting tolerance as $\epsilon = 0.025$ *and the number of MCMC iterations as* $N_{mcmc} = 20000$. *It can be easily observed from the*

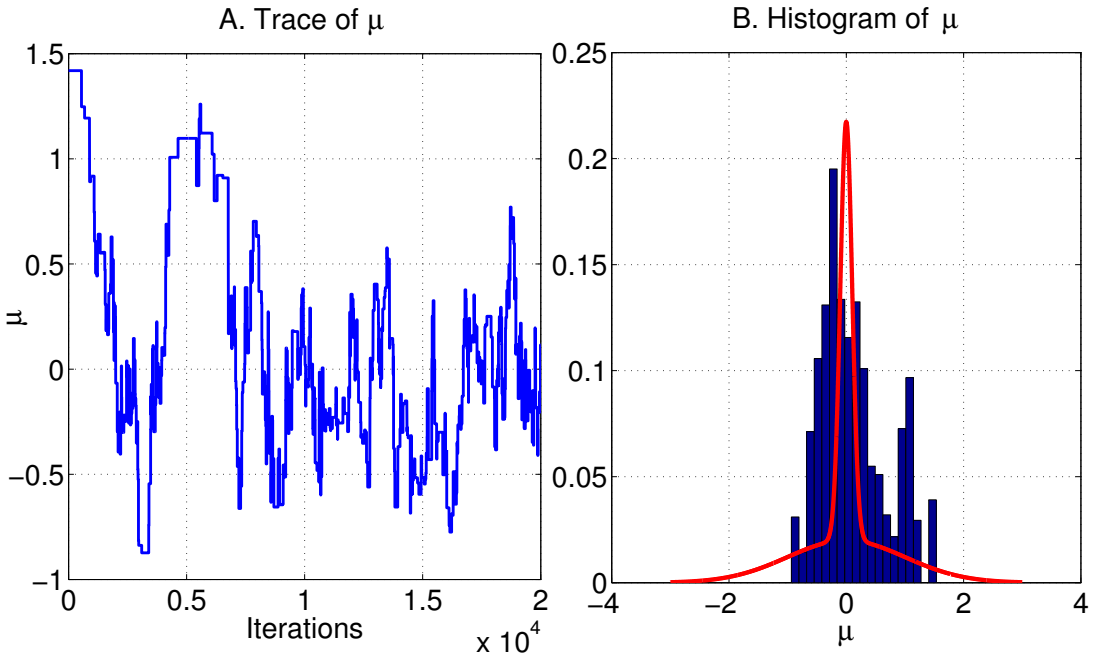


FIGURE 3.5: A: Trajectory of samples for μ obtained from 20000 ABC-MCMC iterations. B: Histogram of the samples, and the true target distribution $p_\epsilon(\mu|x)$ is shown as the red solid line.

graphs that, after converging to the true value, the samples again move to the place in space away from the true value leading to the multi-modal distribution. Moreover, no samples visit the tails of distributions within 20000 *ABC-MCMC iterations. As this toy example is deliberately designed for illustrative purpose, the true distribution* $p(\mu|x)$ *can be approximated by* $p_\epsilon(\mu|x)$ *whose analytical solution is given as equation 3.10. The target distribution is shown as the red curve in Figure 3.5, which provides evidence to illustrate that ABC-MCMC suffers from the slow-mixing chain problem in this simple example.*

To overcome this problem, SMC method has been recently merged into the ABC framework and details are introduced below. In brief, the philosophy of SMC is to gradually approach the posterior of interest via a series of intermediary distributions, which are constituted by a collection of particles $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_{N_{smc}}\}$. In the intermediary phase, each particle will be perturbed around the space through the transition kernel and its importance is weighted based on how well it can represent the posterior. Those fittest realizations are 'encouraged' to characterize the target distribution by frequent selection, whilst those with negligible weights are discarded. Moreover, within the ABC framework, through bypassing the evaluation of the likelihood, the target distribution $p(\theta|\mathbf{X})$ is approximated as $p_\epsilon(\theta|\rho(\mathbf{X}^*, \mathbf{X}) \leq \epsilon)$. Intuitively, when the prior distribution is non-informative and the tolerance ϵ is small, the computational complexity immediately becomes an issue.

In the earliest form of SMC sampling ABC methods, Sisson et al. (2007) derived an innovative ABC approach by combine the previously proposed SMC sampler (Del Moral et al., 2006) with a partial rejection control scheme (ABC-PRC) at which the acceptance criterion is specified as a sequence of tolerances $\boldsymbol{\epsilon} = \{\epsilon_1, \ldots, \epsilon_T\}$. A smooth approach to the target posterior can be expected with this tolerance path, rather than a jump caused with a specific value of ϵ. Basically, ABC-PRC draws the particles from the previous population by considering their weights, and perturbs those particles around the space using the transition kernel, $\theta^{**} \sim k(\theta^*)$. The pseudo-observations are synthesized from the underlying model, $\mathbf{X}^* \sim f(\mathbf{x}_0, \theta^{**})$, where $\mathbf{x}_0$ is the initial condition and $f(\cdot, \cdot)$ is the dynamics. Particle θ^{**} is accepted and weighted if the discrepancy between synthetic data $\mathbf{X}^*$ and true dataset $\mathbf{X}$ is lower than the current tolerance ϵ_t. When the diversity issue appears, i.e. all particles collapse to a few values and this problem can be formulated as $\{\sum_{i=1}^{N_{smc}} (w_t^i)^2\}^{-1} \leq \frac{N_{smc}}{2}$, particles are resampled associated with their corresponding weights. The procedure of ABC-PRC is diagrammatically illustrated in Figure 3.6.

A variety of SMC-based ABC methods were developed with different weighting processes for pursuing unbiasedness and adaptivity. The earliest sequential sampling ABC method, ABC-PRC, however, can be shown to have inappropriate weight evaluation leading to biased estimation. The proof is given below. Considering the SMC sampler (Del Moral et al., 2006), the weight evaluation is given as

$$w_t^i \propto \frac{\pi(\theta_t^i) L_{t-1}(\theta^*|\theta_t^i)}{\pi(\theta^*) k_t(\theta_t^i|\theta^*)}, \tag{3.12}$$

where $\pi(\cdot)$ is the prior distribution for generating the initial particles and $L_{t-1}(\cdot)$ is known as the backward Markov transition kernel defining the probability for moving particles from θ_t^i to θ^*. Following the suggestion from Sisson et al. (2007), the Gaussian random walk kernel is adopted for $L_{t-1}(\theta^*|\theta_t^i)$ and $k_t(\theta_t^i|\theta^*)$.

Beaumont et al. (2009) pointed out that the threshold ϵ_t for accepting particles violates

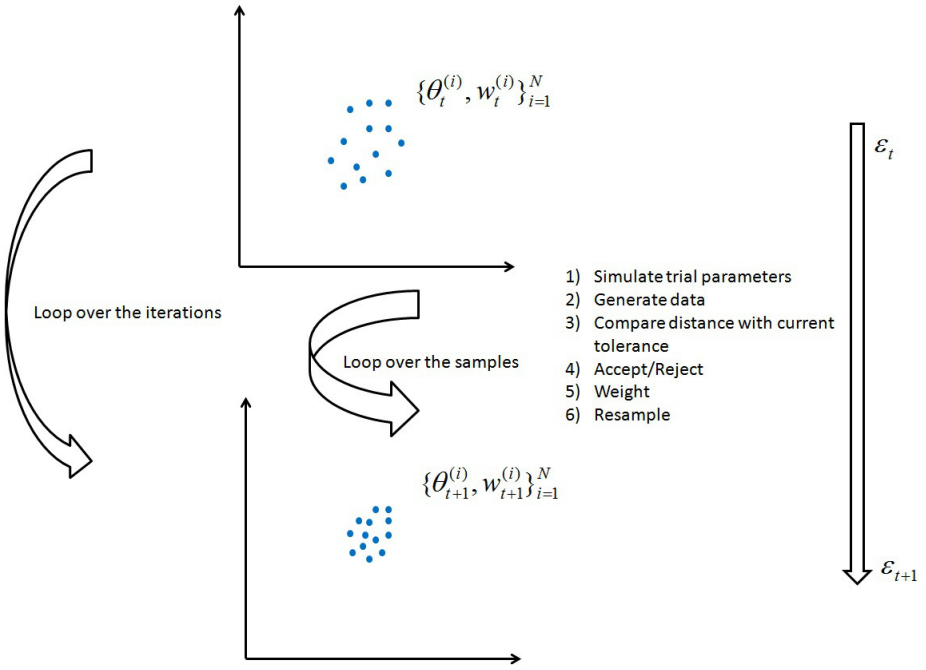


FIGURE 3.6: In the beginning of each iteration, particles are picked from the previous population associated with weights, subsequently, those particles are perturbed by the transition kernel. The pseudo-observations are synthesized by solving the ODEs of system under study, if the discrepancy between synthetic dataset and true observation is lower than the current acceptance criterion, then the particles are retained. When the effective sample size N_{eff} is smaller than a threshold, i.e. $\{\sum_{i=1}^{N_{smc}} (w_t^i)^2\}^{-1} \leq \frac{N_{smc}}{2}$, a resampling step will be carried out on the particles and reset the weights to $1/N_{smc}$.

the condition of using this weighting scheme from the previous SMC sampler (Del Moral et al., 2006)[1].

Let us assume an extreme case in which ϵ is set to zero, then only if $\mathbf{X}^* = \mathbf{X}$ the particle θ_{t-1} would be taken as θ_t (for elaborating, here, θ^{**} is instead denoted as θ_{t-1}). To show the biasedness, we first define the joint density $p(\theta_t, \theta_{t-1})$ as

$$\begin{aligned} p(\theta_t, \theta_{t-1}) &= p(\theta_{t-1})p(\theta_t|\theta_{t-1}) \\ &= \pi(\theta_{t-1}|\mathbf{X})p(\theta_t|\theta_{t-1}, \mathbf{X}) \\ &= \pi(\theta_{t-1}|\mathbf{X}) \frac{k_t(\theta_t|\theta_{t-1}) f(\mathbf{X}^*|\theta_t)\mathbb{I}(\mathbf{X}^* = \mathbf{X})}{\int k_t(\theta_t|\theta_{t-1}) f(\mathbf{X}^*|\theta_t)\mathbb{I}(\mathbf{X}^* = \mathbf{X}) d\theta_t} \\ &\propto \pi(\theta_{t-1}|\mathbf{X}) k_t(\theta_t|\theta_{t-1}) f(\mathbf{X}|\theta_t). \end{aligned} \tag{3.13}$$

We further consider an arbitrary integrable function $r(\cdot)$, having variable θ_t with weights

[1] As the weight is defined as $w_t = \frac{p_t(\boldsymbol{\theta}_t)}{q_t(\boldsymbol{\theta}_t)}$, for approximating the target distribution at the time t, in the original SMC sampler, Del Moral et al. (2006) achieves the approximation by marginalizing the particles coupled with an artificial constructed backward transition kernel L_{t-1}. That is $p_t(\theta_{1:t}) = p_t(\theta_t)\prod_{j=1}^{t-1} L_j(\theta_j|\theta_{j+1})$. However, when using ϵ_t, the distribution of current particles θ_t is approximated as $p_{\epsilon_t}(\theta_t) = \pi_{\epsilon_t}(\theta_t, \mathbf{X}^*|\mathbf{X}) \propto \pi(\theta_t) f(\mathbf{X}^*|\theta^*)\mathbb{I}(\rho\mathbf{X}^*, \mathbf{X})$, so it can not be described as the marginal distribution in θ_t.

w_t, and expectation of $r(\cdot)$ is written as

$$\begin{aligned}
\mathbb{E}[r(\theta_t)w_t] &\propto \iint r(\theta_t)p(\theta_t,\theta_{t-1})w_t d\theta_t d\theta_{t-1} \\
&\propto \iint r(\theta_t)\pi(\theta_{t-1}|\mathbf{X})k_t(\theta_t|\theta_{t-1})f(\mathbf{X}|\theta_t)\frac{\pi(\theta_t)L_{t-1}(\theta_{t-1}|\theta_t)}{\pi(\theta_{t-1})k_t(\theta_t|\theta_{t-1})}d\theta_t d\theta_{t-1} \\
&\propto \iint r(\theta_t)f(\mathbf{X}|\theta_{t-1})\pi(\theta_{t-1})k_t(\theta_t|\theta_{t-1})f(\mathbf{X}|\theta_t)\frac{\pi(\theta_t)L_{t-1}(\theta_{t-1}|\theta_t)}{\pi(\theta_{t-1})k_t(\theta_t|\theta_{t-1})}d\theta_t d\theta_{t-1} \\
&\propto \iint r(\theta_t)f(\mathbf{X}|\theta_{t-1})f(\mathbf{X}|\theta_t)\pi(\theta_t)L_{t-1}(\theta_{t-1}|\theta_t)d\theta_t d\theta_{t-1} \\
&\propto \iint r(\theta_t)\pi(\theta_t|\mathbf{X})f(\mathbf{X}|\theta_{t-1})L_{t-1}(\theta_{t-1}|\theta_t)d\theta_t d\theta_{t-1} \\
&\propto \int r(\theta_t)\pi(\theta_t|\mathbf{X}) \times \left\{\int f(\mathbf{X}|\theta_{t-1})L_{t-1}(\theta_{t-1}|\theta_t)d\theta_{t-1}\right\}d\theta_t. \qquad (3.14)
\end{aligned}$$

Principally , the unbiased result should be denoted as

$$\mathbb{E}[r(\theta_t)w_t] \propto \int r(\theta_t)\pi(\theta_t|\mathbf{X})d\theta_t, \qquad (3.15)$$

where only the particles at t time instant θ_t are involved. If the backward transition kernel $L_{t-1}(\theta_{t-1}|\theta_t)$ is irrelevant to the current particles θ_t, i.e. the integral $\int f(\mathbf{X}|\theta_{t-1})L_{t-1}(\theta_{t-1}|\theta_t)d\theta_{t-1}$ is always a constant, then the expectation of function $r(\theta_t)$ could be unbiased as shown in equation 3.14. However, the backward transition kernel adopts the random walk scheme, that is $L_{t-1}(\theta_{t-1}|\theta_t) = k_t(\theta_{t-1}|\theta_t) \sim \mathcal{N}(\theta_t, \sigma_k^2)$, which violates the condition. Consequently, the weights from equation 3.12 certainly produce biased estimates.

Example 4.5 *In order to illustrate the biased estimate made by ABC-PRC, we consider the previously studied Gaussian mixture model. By using the identical algorithmic settings such as* $x = 0$, $\sigma_k^2 = 0.15^2$ *and* $\rho(x^*, y) = |x^*|$, *ABC-PRC is carried out with 10 consecutive iterations, associated with the sequence of tolerances starting from* $\epsilon_1 = 2$ *down to* $\epsilon_{10} = 0.01$. *The number of particles* N_{smc} *is set to 1000. Results are given in Figure 3.7. The posterior distribution approximated by tolerance* $p_{\epsilon_t}(\theta|\boldsymbol{X})$ *can be solved analytically as given in equation 3.10, and simulations from different tolerance values* ϵ_t *are shown as the purple curves. We also draw the exact posterior distribution by considering* $\epsilon_t = 0$ *as the red dashed line. As seen in the graph, only in the first iteration particles can roughly represent the posterior distribution. However, in successive iterations a failure to cover the distributional tail is clearly observed. Consequently, we discern that ABC-PRC yields biased estimations. One positive outcome of this study is that, unlike ABC-MCMC, ABC-PRC has the appealing property of being capable of avoiding the multi-modal estimation.*

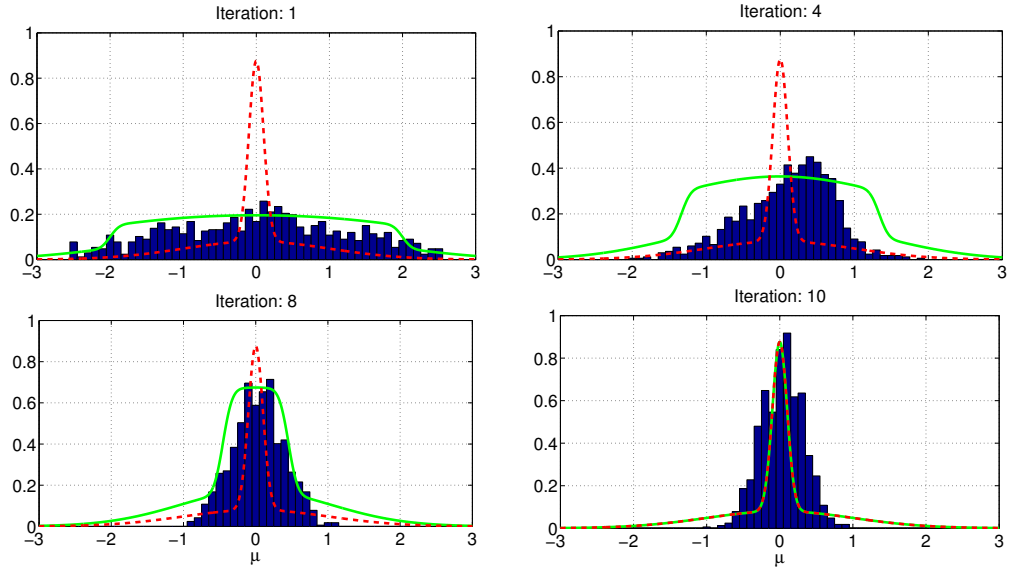


FIGURE 3.7: Histograms of particles obtained from the ABC-PRC in 1^{st}, 4^{th}, 8^{th} and 10^{th} iterations for estimating mean μ of a Gaussian mixture model. The red dash line represents the true posterior distribution $p(\mu|x)$ whose explicit expression is given as equation 3.8, while the green solid line shows the posterior distribution approximated by the current tolerance $p_{\epsilon_t}(\mu|x)$ and of which the expression is given as equation 3.9.

3.2.2 ABC-sequential importance sampling (ABC-SIS) algorithm

Since the weighting scheme adopted in the original ABC-PRC leads to biased estimates, two similar methods, *population Monte Carlo* (Cappé et al., 2004) based method, namely ABC-PMC, and *sequential Monte Carlo* based algorithm known as ABC-SIS were proposed by Beaumont et al. (2009) and Toni et al. (2009). Sisson et al. (2009) issued a correction to the original ABC-PRC algorithm. Due to the high similarity among these algorithms, we only mention ABC-SIS as a representative paradigm of ABC-PMC and corrected ABC-PRC.

Aiming to yield unbiased estimation of the target distribution, ABC-SIS abandons the constructed backward transition kernel for approximating the intermediary distributions. Alternatively, it carries out this approximation by straightforwardly using the prior distribution and the current tolerance ϵ_t, denoted as

$$p_t(\theta_t) = \frac{\pi(\theta_t)}{B} \sum_{j=1}^{B} \mathbb{I}(\rho(\mathbf{X}_j^*, \mathbf{X}) \leq \epsilon_t), \tag{3.16}$$

where $\pi(\theta_t)$ is the prior distribution for generating the initial particles for parameter; B is the number of sets of pseudo-observations synthesized; $\mathbb{I}(\cdot)$ is the indicator function; ϵ_t is the threshold for accepting particles to represent the intermediary distribution $p_t(\theta_t)$.

In the *importance sampling* based methods, the weight is defined as

$$w_t(\theta_t) = \frac{p_t(\theta_t)}{q_t(\theta_t)}, \tag{3.17}$$

where $q_t(\theta_t)$ is called the proposal distribution and the numerator is approximated by equation 3.16. The fundamental concept in importance sampling is to encourage the frequent selection of crucial samples, which is accomplished by choosing an appropriate proposal distribution. Following the derivation introduced by Andriue et al. (2001), the proposal distribution is formulated as

$$q_t(\theta_t) = \int q_{t-1}(\theta_{t-1})k(\theta_t|\theta_{t-1})d\theta_{t-1}, \tag{3.18}$$

here $q_t(\theta_t)$ is adopted as the marginal distribution of the previous proposal distribution. Since the previous proposal distribution $q_{t-1}(\theta_{t-1})$ is the approximation of the intermediary distribution at time $t-1$, i.e. $p_{t-1}(\theta_{t-1})$, $q_t(\theta_t)$ can be regarded as the distribution obtained from perturbing $p_{t-1}(\theta_{t-1})$. Replacing the $q_{t-1}(\theta_{t-1})$ as $p_{t-1}(\theta_{t-1})$ in equation 3.18, the proposal distribution is rewritten as

$$q_t(\theta_t) = \int p_{t-1}(\theta_{t-1})k(\theta_t|\theta_{t-1})d\theta_{t-1}. \tag{3.19}$$

Here the integral $\int p_{t-1}(\theta_{t-1})d\theta_{t-1}$ is always analytically intractable. To address this, by applying the Monte Carlo approximation (Doucet et al., 2001), we have

$$\begin{aligned}\int p_{t-1}(\theta^i_{t-1})d\theta^i_{t-1} &\approx \frac{1}{N}\sum_{j=1}^{N}\delta_{\widehat{\theta^j}\sim p_{t-1}(\theta_{t-1})}(\theta^j_{t-1})w(\theta^i)\\ &\approx \frac{1}{N}\sum_{j=1}^{N}w(\theta^j_{t-1}),\end{aligned} \tag{3.20}$$

N is the number of particles and $\{\theta^i_{t-1},\ i=1,\ldots,N\}$ are realizations from the distribution $p_{t-1}(\theta_{t-1})$. Since $\{\widehat{\theta^j},\ j=1,\ldots,N\}$ are taken from $p_{t-1}(\theta_{t-1})$, therefore, each $\widehat{\boldsymbol{\theta}^j}$ belongs to θ^i_{t-1}. Consequently, $\delta_{\widehat{\theta^j}\sim p_{t-1}(\theta_{t-1})}(\theta^i_{t-1})$ always equals one. By making use of this approximation, the proposal distribution at time t can therefore be given as

$$q_t(\theta^i_t) \approx \frac{1}{N}\sum_{j=1}^{N}w(\theta^j_{t-1})k(\theta^i_t|\theta^j_{t-1}). \tag{3.21}$$

Defining $b_t(\theta_t) = \sum_{j=1}^{B}\mathbb{I}(\rho(\mathbf{X}^*_j,\mathbf{X}) \le \epsilon_t)$, and substituting equation 3.16 and 3.21 into

the weight calculation, we have

$$\begin{aligned} w_t(\theta_t^i) &= \frac{p_t(\theta_t^i)}{q_t(\theta_t^i)} \\ &= \frac{\pi(\theta_t^i) b_t(\theta_t^i)}{\frac{B}{N}\sum_{j=1}^{N} w(\theta_{t-1}^j) k(\theta_t^i|\theta_{t-1}^j)} \\ &\propto \frac{\pi(\theta_t^i)}{\sum_{j=1}^{N} w(\theta_{t-1}^j) k(\theta_t^i|\theta_{t-1}^j)}. \end{aligned} \tag{3.22}$$

If the system of interest is formulated deterministically, the particle θ_t^i is used to generate the pseudo-observations only once and therefore B is set to 1. When the target system is stochastic, θ_t^i is utilized for synthesizing the pseudo-observations more than once, which means that $B > 1$.

Notice that these three algorithms only differ in the resampling process triggered by the effective sample size, but not in kind. In the beginning of each iteration $t \neq 0$, all algorithms pick particles from the previous population with their corresponding weights before moving these particles by the transition kernel. In ABC-PRC, beyond this picking strategy, if a severe degeneracy is observed (i.e. the effective sample size is lower than a threshold), particles will be resampled according to the current weights. However, in the other two methods, ABC-PMC and ABC-SIS, this additional resampling is canceled, since Beaumont et al. (2009); Toni et al. (2009) claimed that the resampling in response to degeneracy is unnecessary as it is already performed at the beginning of each iteration. The block for describing ABC-SIS steps is given in Algorithm 3.

Algorithm 3 ABC-SIS

`Input`: $\boldsymbol{\theta}_0 = \{\theta_0^1, \ldots, \theta_0^{N_{smc}}\} \sim \pi(\boldsymbol{\theta})$, $\mathbf{x}_0$, $\mathbf{X}$, $k(\cdot)$, $\rho(\cdot,\cdot)$, $f(\cdot,\cdot)$, T and $\boldsymbol{\epsilon} = \{\epsilon_1, \ldots, \epsilon_T\}$
`Output`: $\boldsymbol{\theta}_T = \{\theta_T^1, \ldots, \theta_T^{N_{smc}}\}$
for $t = 1, \ldots, T$ **do**
  for $i = 1, \ldots, N_{smc}$ **do**
    1. Draw $\boldsymbol{\theta}^*$ from $\boldsymbol{\theta}_{t-1}$ according to weights $\boldsymbol{w}_{t-1}$
    2. Move $\boldsymbol{\theta}^* \rightarrow \boldsymbol{\theta}^{**}$: $\boldsymbol{\theta}^{**} \sim k(\boldsymbol{\theta}^*)$ where $k(\cdot)$ is given as equation A.30
    3. Synthesize $\mathbf{X}^* \sim f(\mathbf{x}_0, \theta^{**,i})$
    4. Calculate distance $d = \rho(\mathbf{X}^*, \mathbf{X})$
    5. **if** $d \leq \epsilon_t$ **then**
    6. $\quad \theta_t^i = \theta^{**,i}$
    7. $\quad$ compute weight $w_t^i = \frac{\pi(\theta_t^i)}{\sum_{j=1}^{N} w_{t-1}^j k(\theta_t^i|\theta_{t-1}^j)}$
    8. **end if**
  end for
end for

ABC-PRC and ABC-SIS could be infeasible in some complex problems, as these two algorithms perturb particles by the random walk and finding the covariance matrix $\boldsymbol{\sigma}_k^2$ of this kernel requires a fine hand-tuning process. The posterior may diverge by using a large $\boldsymbol{\sigma}_k^2$, whereas, if $\boldsymbol{\sigma}_k^2$ is set to small, particles are highly correlated which can not cover

the space of the target distribution well. On the other hand, ABC-PMC specifies the adaptive movement of particles by taking twice the variances of the previous population as the diagonal elements of the covariance matrix for the random walk.

Example 4.6 *To illustrate how ABC-SIS overcomes biased estimation which is encountered by the ABC-PRC, we examine the performance of the ABC-SIS on the same problem, as well as the algorithmic settings. Outputs from ABC-SIS are plotted in Figure 3.8. As shown in the graphs, ABC-SIS fully covers the target distributions in all iterations, even the distributional tails where ABC-PRC fails to explore. Consequently, the modification of weight calculation successfully removes the bias of the ABC sampling approach.*

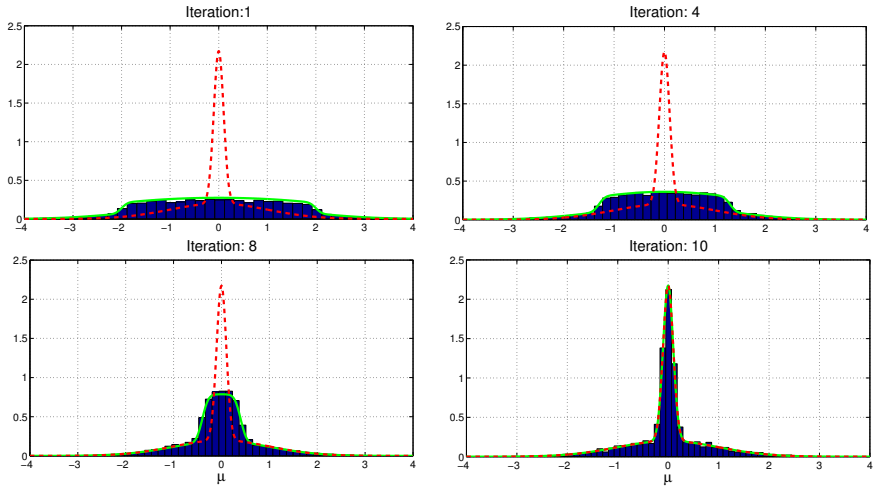


FIGURE 3.8: Histograms of particles obtained from the ABC-SIS in 1^{st}, 4^{th}, 8^{th} and 10^{th} iterations for estimating mean μ of a Gaussian mixture model. The red dash line represents the true posterior distribution, that is $p(\mu|x)$, while the green solid line shows the approximated posterior distribution, i.e. $p_{\epsilon_t}(\mu|\rho(x^*, x) \leq \epsilon_t)$.

3.2.3 ABC-sequential Monte Carlo sampler (ABC-SMC) algorithm

Apparently, the transition kernel $k(\cdot)$ and the tolerance ϵ play significantly crucial roles in determining the performance of all ABC methods. For the SMC-based ABC methods, the schedule of tolerances $\boldsymbol{\epsilon} = \{\epsilon_1, \ldots, \epsilon_T\}$ needs to be carefully designed, as large decreases lead to low acceptance rate, and conversely, small step sizes in the schedule require more iterations. Moreover, the computational complexity of weight calculation is quadratic in the number of particles, which may become an issue with large number of particles.

To address these problems, Del Moral et al. (2012) proposed the innovative SMC-based ABC method which reduces the computational complexity to linear in the number of particles. The idea of the 'linear' weight calculation also provides an automatic way of finding the path of tolerance.

This adaptive ABC-SMC algorithm is theoretically underpinned by the previous SMC sampler (Del Moral et al., 2006), in which weights are evaluated as

$$w_t^i \propto w_{t-1}^i \frac{p_t(\theta_t^i) L_{t-1}(\theta_{t-1}^i|\theta_t^i)}{p_{t-1}(\theta_{t-1}^i) k_t(\theta_t^i|\theta_{t-1}^i)}, \tag{3.23}$$

similarly, $L_{t-1}(\cdot)$ is the backward Markov kernel, and following the suggestion from Del Moral et al. (2006), an optimal choice for this backward kernel is chosen as

$$L_t^{\text{opt}}(\theta_t|\theta_{t+1}) = \frac{p_t(\theta_t) k_{t+1}(\theta_{t+1}|\theta_t)}{\int p_t(u) k_{t+1}(u|\theta_t) du}. \tag{3.24}$$

Unfortunately, the analytical solution of the integral in equation 3.24 is intractable. As one of the possible solutions, an MCMC kernel of invariant distribution $p_{t+1}(\cdot)$ for the transition kernel $k_{t+1}(\cdot)$ is considered for approximating the optimal backward kernel $L_t^{\text{opt}}(\cdot)$. Consequently, the weight can be approximated as following

$$\begin{aligned}
w_t^i &\propto w_{t-1}^i \frac{p_t(\theta_t^i) L_t(\theta_{t-1}^i|\theta_t^i)}{p_{t-1}(\theta_{t-1}^i) k_t(\theta_t^i|\theta_{t-1}^i)} \\
&\propto w_{t-1}^i \frac{p_t(\theta_t^i)}{p_{t-1}(\theta_{t-1}^i)} \times \frac{p_t(\theta_{t-1}^i) k_t(\theta_t^i|\theta_{t-1}^{i)})}{p_t(\theta_t^i)} \\
&\times \frac{1}{k_t(\theta_t^i|\theta_{t-1}^i)} \\
&\propto w_{t-1}^i \frac{p_t(\theta_{t-1}^i)}{p_{t-1}(\theta_{t-1}^i)} \\
&\propto w_{t-1}^i \frac{\sum_{m=1}^{M_{smc}} \mathcal{I}_{\epsilon_t}(\mathbf{X}_{m,t-1}^{*,i}, \mathbf{X})}{\sum_{m=1}^{M_{smc}} \mathcal{I}_{\epsilon_{t-1}}(\mathbf{X}_{m,t-1}^{*,i}, \mathbf{X})}
\end{aligned} \tag{3.25}$$

where $\mathbf{X}^* \in \mathbb{R}^{M_{smc} \times N_{smc} \times D_s \times N_{OT}}$ are the pseudo-observations, and $\mathbf{X}_{m,t-1}^{*,i}$ can be interpreted as the m^{th} synthetic outputs generated by the i^{th} parameter particle at t time instant θ_{t-1}^i. $\mathcal{I}_{\epsilon_t}(\mathbf{X}_{m,t-1}^{*,i}, \mathbf{X})$ is an indicator function that returns one if the discrepancy between pseudo-observation $\boldsymbol{X}_{m,t-1}^{*i}$ and data $\mathbf{X}$ is less than the tolerance ϵ_n, zero otherwise. Symbol M_{smc} here is the integer factor, functioning to operate M_{smc} SMC filters in parallel (Andrieu and Johanes, 2008).

Benefiting from the weight calculation, ABC-SMC is able to adaptively select the current tolerance level ϵ_t. The idea behind this automatic scheme is to determine an appropriate reduction of the tolerance level based on the proportion of particles surviving under the current tolerance. If a large amount of particles remain 'alive', it implies the acceptance criterion is relatively loose and it is safe to make a jump for the next tolerance level. In contrast, if the ratio of 'alive' particles is low, this means particles are less likely to describe the posterior, therefore, a tiny movement should be considered. Such process is mathematically described as $\text{PA}(\mathbf{X}_t, \epsilon_{t+1}) \leq \alpha \text{PA}(\mathbf{X}_t, \epsilon_t)$.

For intentionally moving particles toward the target distribution, ABC-SMC determines

the diagonal elements of covariance matrix for the random walk as the variances of the previous population, which is denoted as

$$\text{var}(\boldsymbol{\theta}) = \mathbb{E}[\boldsymbol{\theta}_t^2] - (\mathbb{E}[\boldsymbol{\theta}_t])^2, \tag{3.26}$$

where the expectation expands as

$$\mathbb{E}[\boldsymbol{\theta}_t] = \sum_{i=1}^{N_{smc}} w_t^i \theta_t^i. \tag{3.27}$$

Substituting equation 3.27 into equation 3.26, the variance is formulated as

$$\begin{aligned} \text{var}(\boldsymbol{\theta}) &= \mathbb{E}[\boldsymbol{\theta}_t^2] - (\mathbb{E}[\boldsymbol{\theta}_t])^2 \\ &= \sum_{i=1}^{N_{smc}} (w_t^i \theta_t^i)^2 - (\sum_{i=1}^{N_{smc}} w_t^i \theta_t^i)^2 \end{aligned} \tag{3.28}$$

Consequently, the algorithm for ABC-SMC is shown in Algorithm 4 where the notations are clarified in Table 3.2. A comparative study of ABC-SMC against other methods is introduced in section 3.3.1, particularly, a competition between ABC-SMC and ABC-SIS is represented in section 3.3.2.

Algorithm 4 ABC-SMC

`Input`: Details are listed in Table 3.1 and set $t = 1$

`Output`: $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_{N_{smc}}\}$

Set ϵ_1 to an arbitrary enough large value

Repeat

1. If $\{\sum_{i=1}^{N_{smc}} (w_t^i)^2\}^{-1} \leq N_T$, then resample $\boldsymbol{\theta}_t$ according to their weights $\boldsymbol{w}_t$; details can be found in Kitagawa (1998);
2. Move $\boldsymbol{\theta}_t \rightarrow \boldsymbol{\theta}_{t+1}$: $\boldsymbol{\theta}_{t+1} \sim k(\boldsymbol{\theta}_t)$, where $\boldsymbol{\sigma}_k^2$ is determined by equation 3.28.
3. Synthesize $\mathbf{X}_{t+1}^* \sim f(\mathbf{x}_0, \boldsymbol{\theta}_{t+1})$
4. Compute the ratio of $\boldsymbol{\theta}_{t+1}$ remaining 'alive' under current tolerance ϵ_t, such function is defined as $\text{PA}(\mathbf{X}_{t+1}^*, \epsilon_t)$, details are given in A-D:

A. compute $M_{smc} \times N_{smc}$ distance matrix $\mathbf{D} = \rho(\mathbf{X}_{t+1}^*, \mathbf{X})$ using equation 3.1

B. compute $M_{smc} \times N_{smc}$ indicator matrix $\boldsymbol{I} : I_{m,n} = \mathcal{I}(d_{m,n}, \epsilon_t)$ where $\mathcal{I}(\cdot) = 1$ if $d_{m,n} \leq \epsilon_t$, otherwise, $\mathcal{I}(\cdot) = 0$.

C. constitute $1 \times N_{smc}$ summation vector $\boldsymbol{v}$: $\boldsymbol{v}_n = \sum_{m=1}^{M_{smc}} I_{m,n}$

D. $\text{PA}(\mathbf{X}_{t+1}^*, \epsilon_t) = \frac{\sum_{n=1}^{N_{smc}} (\boldsymbol{v}^n \neq 0)}{N_{smc}}$.

5. Determine ϵ_{t+1} by solving $\text{PA}(\mathbf{X}_{t+1}^*, \epsilon_{t+1}) \leq \alpha_{\text{SMC}} \text{PA}(\mathbf{X}_{t+1}^*, \epsilon_t)$.
6. **if** $\epsilon_{t+1} \geq \epsilon_{\text{Ta}}$ **then**
   Calculate weights $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t \frac{\text{PA}(\mathbf{X}_t^*, \epsilon_{t+1})}{\text{PA}(\mathbf{X}_t^*, \epsilon_t)}$ and $t = t + 1$
7. **else**
   $\boldsymbol{\theta} = \boldsymbol{\theta}_t$ and set $\epsilon_t = \epsilon_{\text{Ta}}$.

until $\epsilon_t \leq \epsilon_{\text{Ta}}$.

TABLE 3.1: Definitions of Inputs

Method	Input
ABC-SMC	1. Number of particles N_{smc}; Initial tolerance ϵ_0, target tolerance ϵ_{Ta}; integer factor M_{smc}; tolerance reduction factor α_{smc}. 2. $\theta_0^i \sim \pi(\theta),\ i = 1, \ldots, N_{smc}$, initial particles for parameters. 3. $\mathbf{X}_0 : D_s \times N_{smc} \times M_{smc}$, initial condition for system states. 4. True system observations $\mathbf{X}$: $D_s \times \text{N}_{\text{OT}}$. 5. Resampling threshold $N_T = N_{smc}/2$

TABLE 3.2: List of Notations

ABC-SMC			
Description	Symbol	Function	Dimension
number of particles used for SMC	N_{smc}	represent prior/posterior distribution of parameter	scalar
integer factor	M_{smc}	number of SMC filters are used in parallel	scalar
initial particles for parameters	$\boldsymbol{\theta}_0$	initially use for generating solution of dynamics	$D_p \times N_{smc}$
initial condition of states	$\mathbf{x}_0$	states of dynamics at 1^{st} time instant	$D_s \times 1$
weights vector	$\boldsymbol{w}$	represent the importances of particles for parameters	$1 \times N_{smc}$
tolerance reduction factor	α_{smc}	calculate the next tolerance level	scalar
synthetic system output	$\mathbf{X}^*$	synthetic system output obtained by solving ODEs of dynamics associated with the current particles of parameters	$M_{smc} \times N_{smc} \times D_s \times N_{OT}$
distance matrix	$\mathbf{D}$	represent discrepancy between each synthetic and real datasets	$M_{smc} \times N_{smc}$
indicator matrix	$\boldsymbol{I}$	show if the underlying discrepancy is less than epsilon	$M_{smc} \times N_{smc}$
summation vector	$\boldsymbol{v}$	summation of elements in indicator matrix as column-wise	$1 \times N_{smc}$

3.3 Quantitative performance comparison

In this section, we examine the performances of ABC approaches on parameter estimation to the heat shock response model in increasing dimensionality of the unknown parameters. This biological system is used as an example in the section A.4 of Appendix A for quantifying how an initialization influences the performance of the three sequential

inference methods. In this study, this system is used for exploring how the tolerance schedule dominates the performance of the ABC algorithms.

3.3.1 Two unknown parameters case

We first consider the relatively simple case in which two parameters are assumed unknown and the other four parameters are assigned to values from the literature (El-Samad et al., 2006). The algorithmic settings for generating the synthetic dataset are the same in all simulations. As seen from the previous study described in the section A.4 of Appendix A, the difficulty of estimating the parameters rapidly grows with respect to the prior distribution and the dimensionality of unknown parameters. To ensure the identifiability of inference, only stiff parameters are assumed unknown (analysis of sensitivity of the heat shock model is given in chapter 4). Hence the inference task is focused on two unknown parameters space (k_d and α_d), while the remaining four parameters are fixed to their true values, i.e. $K_d = 3$ and $\alpha_d = 0.015$. In addition, each unknown parameter is assigned the flat non-informative prior. All simulations were carried out with `MATLAB`®on an Intel®Xeon$^{\text{TM}}$W3520 @ 2.67GHz with 12 GB RAM computer.

Implementation details are following given

- *ABC-rejection* generates samples from the uniform distribution $\theta_{k_d} \sim \mathcal{U}(0, 10)$ and $\theta_{\alpha_d} \sim \mathcal{U}(0, 1)$. The samples with a population of 1,000 are accepted with tolerance $\epsilon = 0.7$.

- *ABC-regression* uses the same prior for generating the samples as ABC-rejection. Mean of dataset is taken as the summary statistics and tolerance $\epsilon = 0.7$. The population of samples for representing posterior is 1,000.

- *ABC-MCMC* initializes the inference from the place where $\boldsymbol{\theta}_0 = [3.5, 0.02]$. The random walk proposals are utilized for parameters, in particular, $\mathcal{N}(\theta_{t-1}, 0.0001^2)$ for k_d and $\mathcal{N}(\theta_{t-1}, 0.000007^2)$ for α_d. We employ 2,000 MCMC iterations associated with the tolerance ϵ as 10.

- *ABC-PRC* generates particles from the uniform distribution $\theta_{k_d} \sim \mathcal{U}(-10, 10)$ and $\theta_{\alpha_d} \sim \mathcal{U}(-1, 1)$ with a population of 1,000. The path of tolerances starts from 4 and goes down to 0.7 within 10 iterations, reducing consistently in each iteration. The transition kernels for moving particles use the simplest random walk kernel, specifically, $\mathcal{N}(\theta_{t-1}, 2^2)$ for k_d and $\mathcal{N}(\theta_{t-1}, 0.01^2)$ for α_d. The threshold for triggering resampling is set to 500.

- *ABC-PMC* and *ABC-SIS* use the algorithmic settings which are identical to ABC-PRC.

- *ABC-SMC* uses the same prior distributions for synthesizing particles, for which population is 1,000 and M_{smc} is 10. The target tolerance ϵ_T is 10 and the discount factor α_{smc} for determining the next tolerance is 0.99 and the threshold for performing resampling is 500.

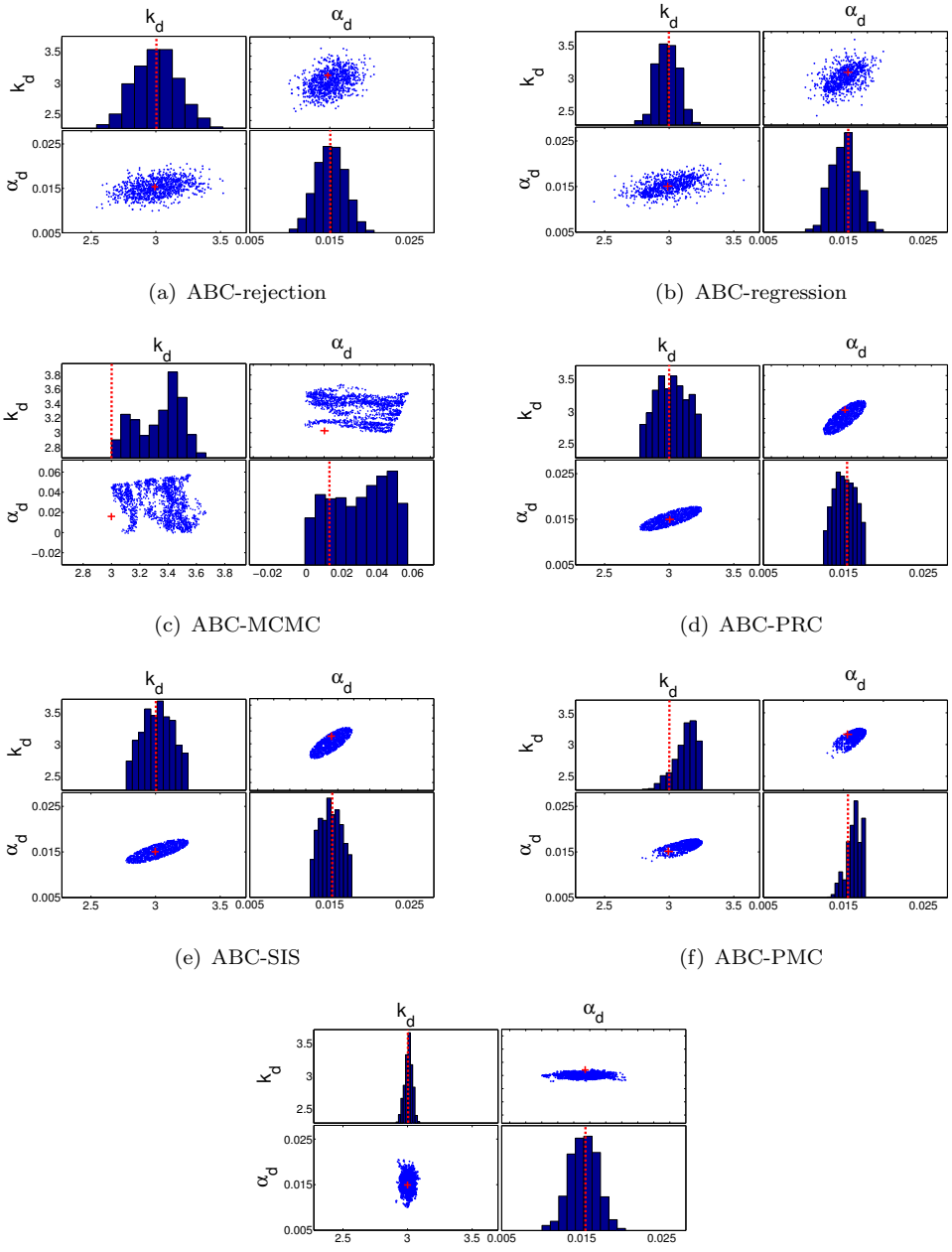
The priors for generating the initial samples in ABC-rejection, ABC-regression and ABC-MCMC are tuned to strike a balance between accuracy and efficiency. In other words, to deliver comparable results with an affordable computational cost, the optimal initial conditions are chosen for these non-sophisticated approaches. In addition, tolerance for ABC-MCMC is adapted to a relatively larger value, because otherwise the proposed samples can barely survived under the tolerance considered in other non-sophisticated methods.

Figure 4.9 shows the estimates from seven ABC methods. Apparently, the non-sophisticated methods (ABC-rejection and ABC-regression) produce the estimations which successfully recover the unknowns and are in good agreement with the results obtained from the advanced approaches (ABC-PRC, ABC-SIS, ABC-PMC and ABC-SMC), while ABC-MCMC fails to make accurate inferences. As expected, all SMC based algorithms outperform the non-sophisticated methods in terms of accuracy, and the unknown parameters are inferred by the advanced algorithms with a high similarity.

As previously claimed, the computational complexity heavily depends on the acceptance criterion, and so we further assess the number of model evaluations required to fulfill the target tolerance. Since the evaluation of the complex system is usually expensive, therefore, the comparison is carried out after accounting for the cost of synthesizing pseudo-observations. In order to make a fair comparison, each approach is run 10 times and the results are shown in Figure 3.10.

As shown in the graph, more trials are attempted by the basic methods to find the fittest samples, while the majority of SMC based approaches consume less cost for simulating pseudo-observations. We note that ABC-PMC, however, is an exception. The significant increase of computational cost is caused by the transition kernel $k(\cdot)$ which utilizes twice the variance of the empirical population as the covariance matrix $\boldsymbol{\sigma}_k^2$. Such adaptivity negatively influences the rate of convergence when the prior (or posterior distributions in the early iterations) is widely distributed leading to the large $\boldsymbol{\sigma}_k^2$, subsequently, many proposals are rejected. ABC-SIS and ABC-PRC, benefiting from an appropriate $\boldsymbol{\sigma}_k^2$ of transition kernel, are clearly observed the outperformance in terms of computational efficiency. However, this cannot be expected in the case with high dimension, since finding an appropriate $\boldsymbol{\sigma}_k^2$ might be impossible. This claim is supported by the investigation stated in section 4.2.2.

ABC-SMC seemingly accomplishes an attractive balance between adaptivity and efficiency in this example. Due to the 'no rejection' strategy of ABC-SMC, uninfluenced



(a) ABC-rejection

(b) ABC-regression

(c) ABC-MCMC

(d) ABC-PRC

(e) ABC-SIS

(f) ABC-PMC

(g) ABC-SMC

FIGURE 3.9: Full posterior distributions of parameters k_d and α_d of the heat shock model obtained by seven methods, where the red dash lines and the red '+' both denote the true values of parameters. The scatter plots in each of the parameters are mirror images about the diagonal histograms.

by the large transitions in the first few iterations, all particles are kept. In addition, most realizations are assigned non-zero weights by the importance evaluation coupled with the loose tolerances. As a result, the large amount of 'surviving' particles leads to a rapid reduction of tolerance in the early stage. Subsequently, due to the wide distribution of particles and decreased tolerance, only few particles are weighted as non-zero, which results in a serious drop of the effective sample size. Hence, a resampling step

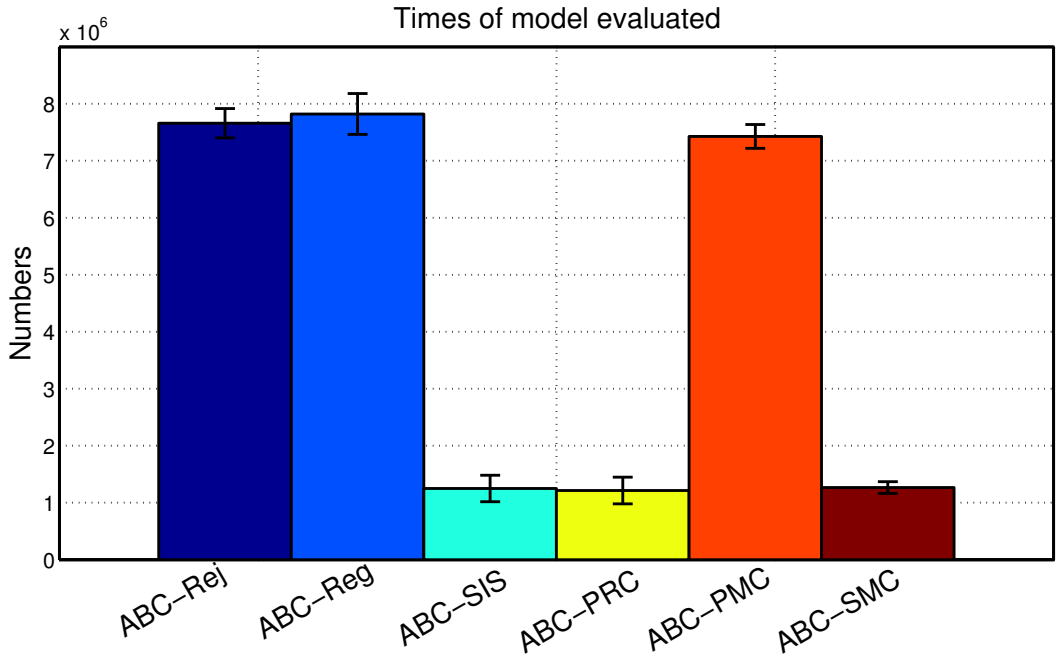


FIGURE 3.10: Number of model evaluations required to achieve the target tolerance by six ABC methods. We have not included ABC-MCMC results because the estimations were poor quality.

is triggered to enrich the diversity and it prevents the particles from collapsing. When a slow tolerance reduction is observed meaning that the particles almost achieved the convergence, the algorithm should be stopped by the fulfillment of tolerance. In other words, an appropriate target tolerance can greatly boost the computational efficiency, since less attempts will be made to satisfy the criterion with negligible movement. Although this determination is highly case dependent, a heuristic solution was suggested by Del Moral et al. (2012) to stop the algorithm when the drop of tolerance is less than 1.5%.

3.3.2 Three unknown parameters case

In the previous example we found ABC-SIS and ABC-SMC perform similarly in both accuracy and computational efficiency, therefore, a comparison is further carried out to discriminate their abilities on parameter estimation by increasing the dimension of unknown parameters. The study introduced below is to identify parameters k_d, α_d and α_0 with assuming the remaining three are known. The method of generating the synthetic dataset is identical to the two unknown parameters case, and the prior and the transition kernel for α_0 are identical to those for α_d that are described in last section. Comparison is carried out by setting two tolerance paths for ABC-SIS, in which the initial tolerance ϵ_0 and the final tolerance ϵ_T are identical to ABC-SMC ($\epsilon_0 = 480$ and $\epsilon_T = 10$). Since ABC-SMC on average requires 240 iterations to reach ϵ_T in an automatic manner, for a fair comparison, ABC-SIS declines the tolerance from ϵ_0 down to ϵ_T in

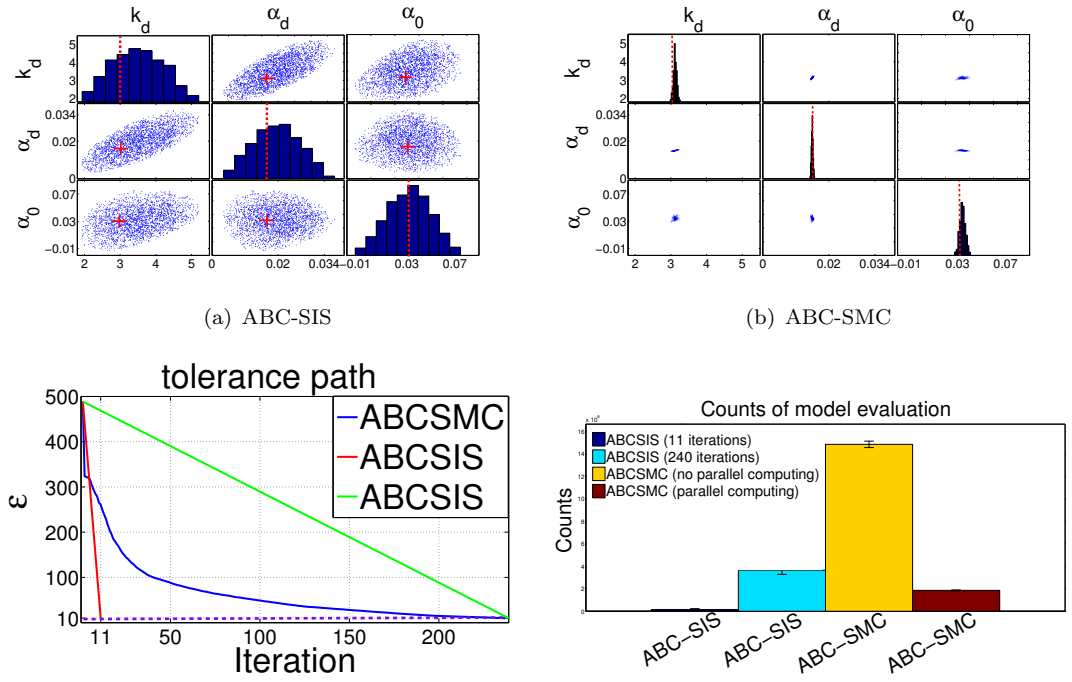
regular intervals (total number of intervals is 240). In addition, we further assess how ABC-SIS benefits from a manually-chosen tolerance sequence, via a tolerance schedule defined as $\boldsymbol{\epsilon} = [488, 440, 392, 344, 296, 249, 201, 153, 105, 57, 10]$ which is shown as the red line in Figure 3.11(c).

Posterior distributions of parameters are shown in Figure 3.11(a) and 3.11(b). As seen from the graphs, since ABC-SMC adaptively utilizes the variance of the empirical distribution for the transition kernel, it greatly outstrips ABC-SIS in terms of the uncertainty of estimation. From the perspective of efficiency, the computational advantage of ABC-SMC cannot be expected in this higher dimensional example. The side effect from the adaptivity of ABC-SMC is clearly shown in Figure 3.11(d), in which substantially more model evaluations were taken by ABC-SMC. Nevertheless, benefiting from the 'no rejection' strategy, ABC-SMC can be ran in parallel which partially alleviates this computational complexity. By using 8 cores for the parallel computing, the time duration for simulating pseudo-observations 14,000,000 times was cut to one eighth of its original value. Even though the computational expense is still ten times greater than ABC-SIS with well designed tolerance schedule, efficiency can be further boosted by using more cores.

We note that the tolerance sequence used governs the computational performance of ABC-SIS, better performance is evident with a suitable tolerance path, whereas a negative influence is observed if the tolerances are inappropriately chosen.

In summary, from the comparisons carried out on the heat shock response system, it is clear that ABC-SMC has the superior performance in terms of accuracy, and the adaptivity further increases its appeal. The computational efficiency of ABC-SMC, however, substantially decreases in the problems with high dimensionality. Majority of SMC type ABC methods (except ABC-PMC) are more attractive in tackling simple problems, however, deterministically assigning tolerance sequences and the tuning of transition kernel limit their appeals for complex problems. The non-sophisticated methods are somewhat straightforward solutions in the inference problem, rather than the advanced methods.

From the study conducted in the section A.4 of Appendix A, we note that the particle filtering algorithm is capable of precisely estimating the parameters of heat shock model, regardless of the combination of the two unknowns. Additionally, the PF, benefiting from the one-pass data visiting scheme, can expect an improvement on the computational efficiency. The family of ABC algorithms, except ABC-MCMC, are reliable for accurately inferring two or three unknown parameters of heat shock system. The superior performance of ABC methods, however, is highly dependent on the algorithmic settings, e.g. the tolerance and the transition kernel. Moreover, similarly to the comparative study represented in section A.7, ABC approaches might struggle with efficiency due to their batch nature, and the increase of computational complexity with respect to the growth of dataset volume is considerably greater than the PF. Since ABC methods



(a) ABC-SIS

(b) ABC-SMC

(c) path of tolerance

(d) counts of model evaluation

FIGURE 3.11: (a) and (b): Posterior distributions of the parameters k_d, α_d and α_0 obtained by ABC-SIS and ABC-SMC, where the red dash lines are the true values and the red '+' is the location that particles should center. (c): Paths of tolerance, in which two schedules with different reductions are considered in ABC-SIS. (d): Counts of model evaluation that are carried out by ABC-SIS and ABC-SMC to reach the target tolerance.

bypass the evaluation of likelihood, therefore, these approaches alleviate the statistical assumption of additive noise and become suitable for the problems without sufficient prior knowledge.

3.4 Discussion

In this chapter, we thoroughly investigated the approximate Bayesian computation methods both theoretically and empirically. Starting from the simplest ABC-rejection approach, we chronologically converge to the most powerful adaptive ABC-SMC algorithm. The features of all mentioned ABC methods are summarized in Table 3.3.

Interestingly, in the study of ABC-regression method, the use of summary statistics heavily effects the performance of algorithm. When the mean of data is the only adopted statistics term, the method performs the best. Insights on the influence of *summary statistics* represent an avenue for further study of ABC type methods and which may make it an appropriate treatment in the context of systems biology.

Table 3.3: ABC methods summary

Method	Time	Advantage	Disadvantage
ABC-Rejection	1997	Straightforward, simply compare the simulation with real dataset.	1. Sensitive to the choice of prior. 2. Performance completely depends on choice of tolerance ϵ.
ABC-Regression	2002	More accurate than ABC-Rejection.	1. Sensitive to the choice of prior. 2. Performance depends on choice of summary statistics $\boldsymbol{s}$.
ABC-MCMC	2003	Partially alleviates the dependence of prior distribution by using the transition kernel $k(\cdot)$ for perturbing samples.	The transition kernel needs to be tuned. Possibly trap in a region with rare opportunity to jump out if small σ_k^2 is used, while the acceptance rate will be low if the σ_k^2 is large.
ABC-PRC	2009	1. Candidate samples are drawn from the previous iteration with importance weights, efficiently eliminate the candidates which negligibly contribute to posterior. 2. The use of tolerance schedule makes algorithm gradually approach the target.	1. Tweak the covariance of transition kernel function $k(\cdot)$. 2. Superior performance can be only achieved by considering an appropriate tolerance path.
ABC-SIS	2009		
ABC-PMC	2009	The covariance matrix $\boldsymbol{\sigma}_k$ of the transition kernel $k(\cdot)$ is adaptively determined.	Computational complexity caused by simultaneously using the adaptive transition kernel and the fixed tolerance schedule.
ABC-SMC	2012	1. Adaptively select the tolerance level and transition kernel. 2. Weight computation depends on the ratio of survived particles.	Highest computation complexity among algorithms for the real biological systems.

Chapter 4

Approximate Bayesian Computation coupled with Sensitivity Analysis

In this chapter, we propose a three stage strategy inference framework by considering the approximate Bayesian computation methods coupled with sensitivity analysis technique. A systematic re-allocation of computational effort is suggested to achieve a decent compromise between accuracy and computational efficiency. The effectiveness of the proposed method is demonstrated on three oscillatory models and one transient response model taken from the Systems Biology literature.

4.1 Parameter Sensitivity

As suggested by Gutenkunst et al. (2007), parameter or combination of parameters, can be decomposed into having sloppy and stiff properties through sensitivity analysis (Saltelli, 2002; Gutenkunst et al., 2007; Marino et al., 2008). We demonstrate these parameter properties on the simple Lokta-Volterra system, given as

$$\begin{aligned} \frac{dx}{dt} &= \alpha\, x \;-\; x\, y \\ \frac{dy}{dt} &= x\, y \;-\beta\, y, \end{aligned}$$

where, x and y are the populations of predator and prey respectively. We simulated a set of synthetic data associated with different parameter values drawn from the interval [0 1], and evaluated the error between datasets synthesized from the arbitrary values and the true values (i.e. [0.5 0.5]). Equal output contours on the error surface are shown in Figure 4.1. Along the major axis of the ellipses, which is largely parallel to the variable β,

we see that the output can tolerate large changes in parameter values. Along the minor axis, approximately parallel to the parameter α, the model is sensitive to the changes in parameter values. Thus in this illustration, β is termed as a sloppy parameter and

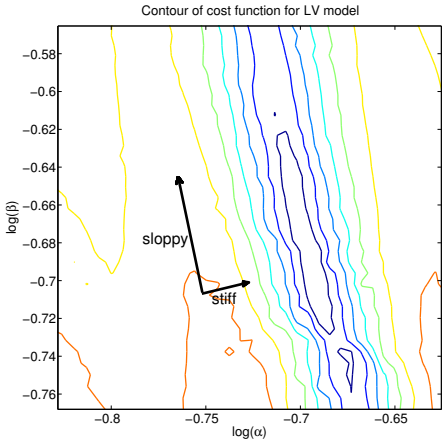


FIGURE 4.1: Contours on the error surface between true and synthesized data as the parameters move away from their true values to illustrate stiff and sloppy parameters in the Lokta-Volterra model. Error is minimum when the parameters are set to their true values at $\log(\alpha) = -0.7$ and $\log(\beta) = -0.7$. Contours are approximately ellipsoidal. Along the sloppy axis, which is dominated by β, the error varies slowly as a function of parameters, whereas along the stiff axis (along which the model has greater sensitivity to parameters), the variation in error is steep. By observing this, we would regard β as a sloppy parameter.

α, a stiff parameter. This conclusion also means that when ABC methods are applied to the Lokta-Volterra model, the uncertainty in the estimation for β is greater than the estimation for α (simulation can be found in chapter 3).

Sensitivity analysis can be carried out in various ways. A straightforward approach to quantify sensitivity is based on the gradients with respect to kinetic parameters that can be computed numerically. Such *local* sensitivity analyses, determined by the gradients at operating points of interest, are often unsuitable for dynamical biological systems which can undergo large changes in operating regimes during the processes of interest (Zi, 2011). Toni et al. (2009) used a PCA-based sensitivity analysis technique to differentiate stiff/sloppy parameters in the context of Systems Biology. This method has similar limitations (as discussed in section 4.4). Thus in this work we use variance partitioning method, known as the extended Fourier amplitude sensitivity test (eFAST) introduced by Saltelli et al. (1999); Saltelli (2002), which is a refined version of the Fourier Amplitude Sensitivity Test (FAST) (Cukier et al., 1973). The idea in this method is to quantify the statistical variance of the model output when the parameters are allowed to traverse a wide range in their input space. Briefly, this is achieved by tagging different frequencies to different parameters, allowing changes in the parameters by changing them at the respective frequencies, and quantifying the changes in output in the frequency domain. The variation due to each parameter shows up in the output as amplitudes of the Fourier coefficients when moving along the search trajectories of the parameter

space.

4.1.1 Extended Fourier Amplitude Sensitivity Test (eFAST)

The *extended Fourier amplitude sensitivity test* (eFAST) (Saltelli et al., 1999), is one of the popular sensitivity analysis techniques based on variance decomposition, being applicable for the nonlinear and non-monotonic systems. The algorithm initially partitions the total variance of the dataset, evaluating what fraction of the variance can be determined by variations in the parameter of interest. This quantity, known as the sensitivity index, is calculated as

$$\mathbf{S} = \frac{\mathrm{Var}_{\boldsymbol{\theta}}[\mathrm{E}(\mathbf{Y}|\boldsymbol{\theta})]}{\mathrm{Var}(\mathbf{Y})} = \frac{D_i}{D} \tag{4.1}$$

'Translating' this definition into eFAST, the sensitivity is assessed by picking the samples for the parameter of interest with the highest frequency $\omega_{\max}$, while the samples for the rest of the parameters are selected with the complementary frequencies $\boldsymbol{\omega}_{-i}$. This process is repeated until the samples of each parameter is drawn with highest frequency once. An illustrative example of this cycling process is shown in Figure 4.2.

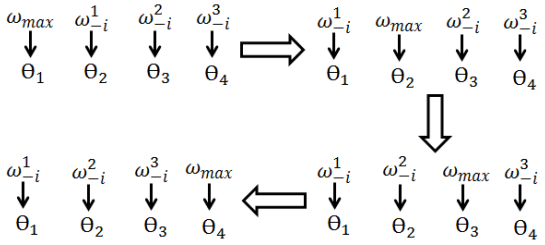


FIGURE 4.2: When we wise to evaluate the sensitivity of parameter θ_1, its samples are drawn with the highest frequency $\omega_{\max}$, while the samples for other parameters $\boldsymbol{\theta}_{-i} = \{\theta_2,\ \theta_3,\ \theta_4\}$ in the system are picked using the complementary frequencies $\boldsymbol{\omega}_{-i} = \{\omega^1_{-i},\ \omega^2_{-i},\ \omega^3_{-i}\}$. Through this process, all parameters in system should be assigned to the highest frequency once.

Taking this sampling strategy, eFAST claims to be capable of apportioning the total variance (term D in equation 4.1) into the partial variance (term D_i in equation 4.1) caused by individual variation of the parameter. Algorithmically, the parameter sensitivity with respect to a specific state is evaluated by a fraction, given as equation 4.1, where the numerator is the variance of outputs of the specific state. More specifically, the outputs adopted for numerator are synthesized by the parameter samples which are drawn from the frequency vector by setting the underlying parameter to the highest frequency. The denominator of this fraction is the summation of output variances of the same state, and these outputs are generated by different parameter samples, which are drawn from all possible combinations of frequency settings. For example, if we need to assess the sensitivity of parameter θ_1 with respect to the second state $\mathbf{x}_2$ of a system

that has four parameters, it can be calculated as

$$\mathbf{S}_2^1 = \frac{\hat{\sigma}_2^1}{\hat{\sigma}_2^1 + \hat{\sigma}_2^2 + \hat{\sigma}_2^3 + \hat{\sigma}_2^4}, \tag{4.2}$$

where the subscript of σ shows which state in system is under study, and the superscript implies which parameter is sampled using the high frequency (i.e. the parameter of interest).

The sensitivity in some cases consists of the the first-order sensitivity index that calculates the standalone effect of the underlying parameter, and the higher-order sensitivity index that captures the interaction among the underlying parameter and other parameters of any order. For example, as seen in Figure 4.3, if the sensitivity of parameter θ_1 is under study, the system sensitivity index S_T of an arbitrary model with three parameters can be decomposed into the total sensitivity index of parameter θ_1 and the index of complementary parameters (all parameters in system except the underlying θ_1), denoted as S_{T_1} and S_{-1}. In addition, the total sensitivity index S_{T_1} can be further specified as the first-order index S_1, the second-order indices $\{\mathrm{S}_{12},\ \mathrm{S}_{13}\}$ and the third-order index S_{123} (Saltelli, 2002).

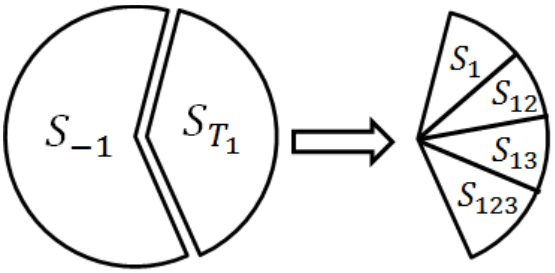


FIGURE 4.3: Sensitivity of parameter in an arbitrary system that has three parameters. Of interest is the parameter θ_1, therefore, the total effect of θ_1 consists of the first-order index S_1, second-order indices $\{\mathrm{S}_{12},\ \mathrm{S}_{13}\}$ and the third-order index S_{123}. The S_{-1} indicates the effect of complementary parameters on system outputs when the underlying parameter is θ_1.

Unfortunately, the nature of FAST prohibits the individual calculation of indices except the first-order. Consequently, Saltelli (2002) derived the extended FAST algorithm to solve this incapability, in which even though the individual higher-order indices are still impossible to calculate, the total sensitivity index S_{T_1} can be evaluated by subtracting the total indices of complementary parameters from the 'unit circle'. Algorithmically, the sensitivity index of complementary parameters $\boldsymbol{\theta}_{-1}$ can be approximated by picking the samples for $\boldsymbol{\theta}_{-1}$ with the complementary frequencies and evaluating the variance caused by the variations of $\boldsymbol{\theta}_{-1}$. Consequently, denoting the variance of complementary parameters as D_{-1}, the total sensitivity index of θ_1 can be calculated as

$$S_{Ti} = \frac{D - D_{-i}}{D} \tag{4.3}$$

More specifically, a heuristic strategy for selecting samples for parameter was also derived by Saltelli (2002) which considerably boosted the performance in comparison to the original version. Saltelli et al. (1999) initially proposed a sinusoidal function for generating samples, given as

$$\begin{aligned}\boldsymbol{\theta} &= G(\sin(\boldsymbol{\omega}\boldsymbol{s})) \\ &= \frac{1}{2} + \frac{1}{\pi}\arcsin(\sin(\boldsymbol{\omega}\boldsymbol{s})),\end{aligned} \tag{4.4}$$

where $\boldsymbol{\omega}$ is a $\mathrm{D_p} \times 1$ vector of frequencies assigning to parameter vector $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_{\mathrm{D_p}}]$. From the implementation, the high frequency for specifying the underlying parameter is called the maximum frequency $\omega_{\max}$, computed as $(\mathrm{N_{se}} - 1)/2\mathrm{M_e}$, where $\mathrm{N_{se}}$ specifies the number of samples drawn from the function $G(\cdot)$. $\mathrm{M_e}$ is the interference factor (from the empirical investigation (Saltelli, 2002; Marino et al., 2008), is usually used as 4 or 6) and acts as the remover for numerical amplitude from superposing of waves. The low frequencies for $\boldsymbol{\theta}_{-i}$ are set in the range $[1\ \omega_{-i,\max}]$ with a regular increment $\frac{\omega_{-i,\max}}{\mathrm{D_p}}$, where $\omega_{-i,\max} = \frac{\omega_{\max}}{2\mathrm{M_e}}$. Terms $\boldsymbol{s}$ in equation 4.4 defines a $1 \times \mathrm{N_{se}}$ scalar vector from $-\pi$ to π.

This sampling approach can be seen as an exploration of space in a grid scheme. For visualizing the distribution of samples drawn from this strategy, we consider a 2-dimensional toy example and set frequencies of state variables as $\boldsymbol{\omega}_1 = [1\ :\ 1\ :\ 20]$ and $\boldsymbol{\omega}_2 = [20\ :\ 1\ :\ 40]$. To aid intuition, the number of samples drawn from function $G(\cdot)$ is defined in a low dimension $\mathrm{N_{se}} = 25$. As seen from the graph, an even distribution of

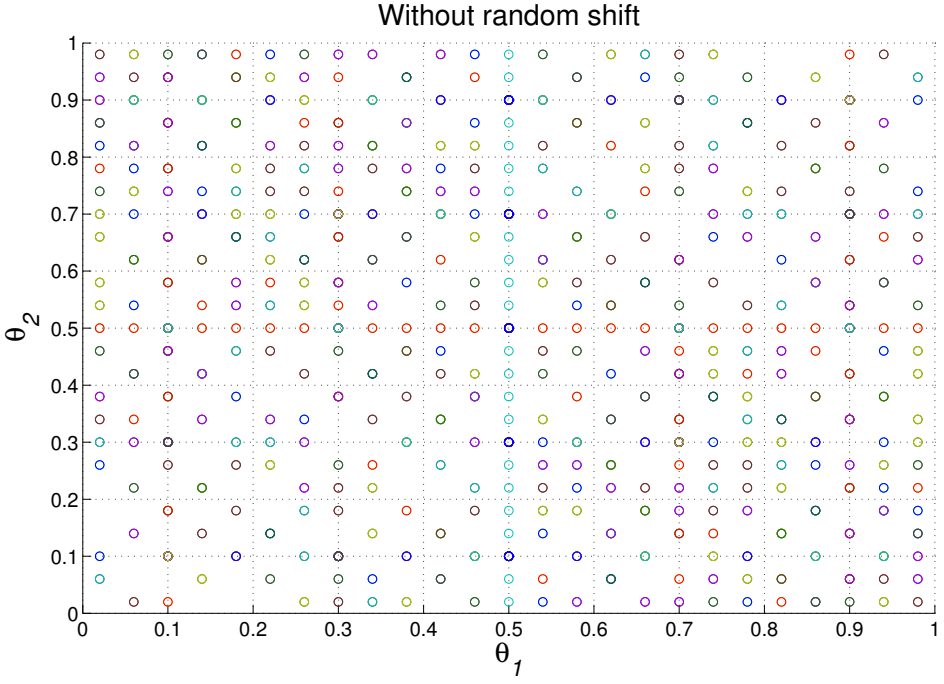


FIGURE 4.4: Distribution of samples for 2-dimensional variables toy example, points are drawn from the function without random shift.

samples in the space is evident.

The diversity of samples from the original sinusoidal function becomes a serious issue, and therefore, various combinations of frequencies are taken as compensation so that the samples are distributed as widely as possible in the unit hypercube parameter space. However, the frequency for sampling function is positively correlated to the sample size, as a result of which, one has to make a trade-off between the diversity of samples and computational cost. This problem was addressed by (Saltelli et al., 1999), in which a random shifting factor is adopted in the sinusoidal function, given as

$$\begin{aligned} \boldsymbol{\theta} &= G(\sin(\boldsymbol{\omega s} + \boldsymbol{\varphi})) \\ &= \frac{1}{2} + \frac{1}{\pi} \arcsin(\sin(\boldsymbol{\omega s} + \boldsymbol{\varphi})), \end{aligned} \tag{4.5}$$

where $\boldsymbol{\varphi}$ is the $\mathrm{D_p} \times \mathrm{N_{se}}$ matrix for random phase-shifting uniformly distributed in $[0, 2\pi]$. With identical frequency vectors, samples drawn from the function coupled with random phase-shifting can more thoroughly cover the parameter space. In order to illustrate the clear advantage of the sampling approach with randomness, we consider the previously studied 2-dimensional toy example with the fixed frequencies $\omega_1 = 10$ and $\omega_2 = 20$. The number of samples chosen from the function increases to 2049 (Note that $\mathrm{N_{se}}$ has to be an odd number, since the calculation of Fourier amplitude is carried out in pairs and the median element is isolated), yet other algorithmic settings remain the same. Figure 4.5 shows the distribution of samples with/without random phase-shifting: it is easily seen that the sinusoidal function coupled with randomness holds a better space exploring capability.

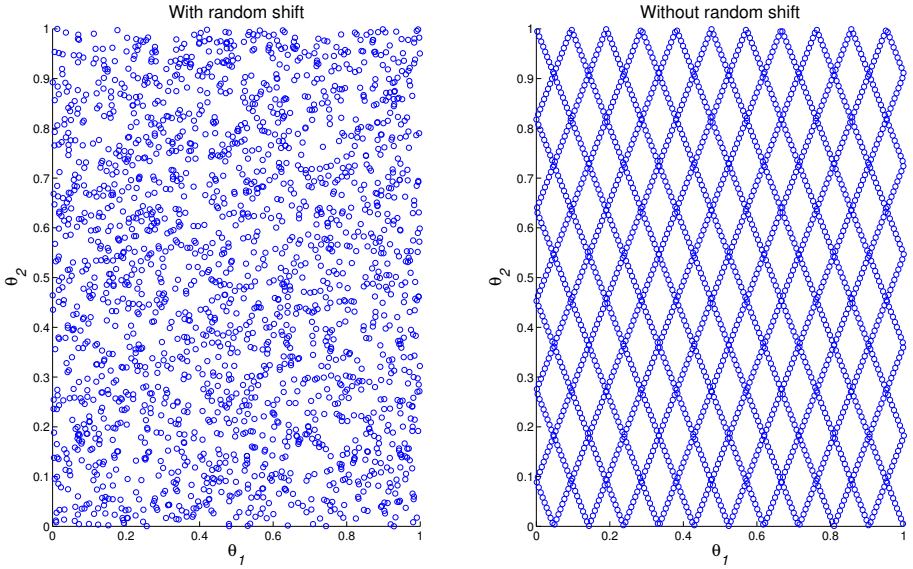


FIGURE 4.5: Distributions of samples for 2-dimensional variables toy example, points are obtained with/without random shift.

Since the entire samples are drawn from the $\mathrm{D_p}$ dimensional unit hypercube, i.e. $\boldsymbol{K}^i = (\boldsymbol{\theta}|0 \leq \theta_i \leq 1;\ i = 1, \ldots, \mathrm{D_p})$, for casting the realizations to the real-value space, samples must be scaled. In the cases studied here, we assume that the samples of parameters should always satisfy the uniform distribution, i.e. $\theta \in \mathcal{U}(a, b)$. Taking the appealing property of the uniform distribution, the sample from the unit hypercube can be easily converted to the real-value space by the mapping process shown below

$$\theta_i = (b - a) * \theta_i - a. \tag{4.6}$$

As the complexity of the dataset analyzed by eFAST is high, for the purposes of elaboration, the information about the structure of the dataset is clarified. Specifically, the dataset synthesized has five dimensions and is defined as $\mathbf{Y}(\mathrm{N_{se}}, \mathrm{N_{OT}}, \mathrm{D_s}, \mathrm{D_p}, \mathrm{N_r})$. For example, a point in the dataset $y(n, t_1, s_2, p_3, r_4)$, describes the value of state s_2 at time instant t_2, and pseudo-observations are synthesized by using n^{th} parameter vector $(n = 1, \ldots, \mathrm{N_{se}})$ from r_4^{th} $G(\cdot)$ function, which produces samples for the parameters by specifying the frequency of θ_{p_3} as $\omega_{\max}$.

The practical details for running eFAST are split into two parts: the model-evaluation and the sensitivity index calculation. The first half is shown in Algorithm 5 and the remaining half is given in Algorithm 6. Due to the complexity of eFAST, we list the variables needed to be initialized in Table 4.1 and all used variable notations are clarified in Table 4.2.

TABLE 4.1: Definitions of initializations for eFAST

Method	Input
eFAST	1. T is the number of interested time instances. 2. Number of samples $\mathrm{N_{se}}$; Number of search curves $\mathrm{N_r}$; Interference factor $\mathrm{M_e}$; Maximum frequency can be assigned for the underlying parameter $\omega_{\max} = \frac{(\mathrm{N_{se}}-1)}{2\mathrm{M_e}}$; 3. Maximum frequency can be assigned for the parameters in system except the underlying parameter $\boldsymbol{\theta}_{-i}$: $\omega_{\max,\text{-i}} = \frac{\omega_{\max}}{2\mathrm{M_e}}$

4.1.2 PCA based technique

Toni et al. (2009) introduced a principal component analysis based local sensitivity analysis technique that was dedicated to sequential Monte Carlo methods. In this approach, stiffness/sloppiness is quantified via the eigenvectors (principal components, PCs) of the covariance of the accepted particles from the last iteration, this covariance-variance matrix is denoted as $\boldsymbol{\Sigma} \in \mathbb{R}^{\mathrm{D}_p \times \mathrm{D}_p}$. PCA orthogonalizes parameters in the Euclidean space, thus, the first eigenvector indicates the direction of the highest variance. From the point of view of SMC, the first eigenvector shows the direction in which the particles distribution shows maximum variance. In contrast, the last eigenvector conveys the direction

Algorithm 5 model-evaluation phase in eFAST

- Initialize the inputs, details are given in Table 4.1.

1. `Model evaluation`

for $i = 1, \ldots, \mathrm{D_p}$ **do**

A. **for** $r = 1, \ldots, \mathrm{N_r}$

B. Define the frequency a $\mathrm{D_P} \times 1$ vector $\boldsymbol{\omega}$: set frequency for the underlying parameter θ_i: $\omega_i = \omega_{\max}$

C. Define the frequency vector $\boldsymbol{\omega}$: Set frequencies $\boldsymbol{\omega}_{-i}$ for the complementary elements in parameter space $\boldsymbol{\theta}_{-i}$; Details are described in the text of eFAST method.

D. Generate a $\mathrm{D}_p \times \mathrm{N_{se}}$ Matrix of random phase shift $\boldsymbol{\varphi} \sim \mathcal{U}(0, 2\pi)$. [equation (4.5)]

E. Generate a $1 \times \mathrm{N_{se}}$ vector of variables $\boldsymbol{s}$ according to $\boldsymbol{s} = \pi(2 \times (1 : \mathrm{N_{se}}) - \mathrm{N_{se}} - 1)/\mathrm{N_{se}}$.

F. Generate the parameter samples $\boldsymbol{\theta}$ $\boldsymbol{\theta} = 0.5 + \frac{1}{\pi} \arcsin(\sin(\boldsymbol{\omega}\boldsymbol{s} + \boldsymbol{\varphi}))$ [equation (4.5)]

G. Map back values of $\boldsymbol{\theta}$ from STEP **1.F** to their real values, following inverse cumulative density function and concern the pre-defined distribution of $\boldsymbol{\theta}$ [equation (4.6)]

H. **for** $n = 1, \ldots, \mathrm{N_{se}}$ Synthesize the current system output $\mathrm{N_{OT}} \times \mathrm{D_s}$ matrix by integrating system dynamics $\mathbf{Y}_{\mathrm{temp}} = \int_0^{\mathrm{T}} f(\boldsymbol{x}_0, \boldsymbol{\theta}_n) dt$

I. $\mathbf{Y}_{n,i,r} = \mathbf{Y}_{\mathrm{temp}}$

end for

end for

end for

After `Model Evaluation` step, a 5-D synthetic dataset $\mathbf{Y}(\mathrm{N_{se}}, \mathrm{N_{OT}}, \mathrm{D_s}, \mathrm{D_p}, \mathrm{N_r})$ is obtained. Example for interpreting this data construct can be found in text.

of the the smallest variance. Consequently, specifying the eigenvalue corresponding to i^{th} column of eigenvectors as λ_i, the variation in the collection of particles at the last iteration can be proportionally explained by i^{th} eigenvector as

$$\frac{\lambda_i}{\mathrm{trace}(\boldsymbol{\Sigma})}. \tag{4.7}$$

A larger proportion is due to a greater contribution to variance in the population, and correspondingly, plays a more crucial role in determining uncertainty. Of interest is to carry out the sensitivity analysis relying on the last column of eigenvector, since it extends the least in the space of the posterior distribution (i.e. has the least uncertainty). As a result, the parameter that contributes the most to this eigenvector is seen as stiff to the system. Similarly, the parameter which dominates the first eigenvector is regarded as the sloppy parameter.

In order to quantitatively study how parameters constitute the eigenvector, one needs to project the eigenvectors onto the raw parameters. Specifically, denoting the normalized

Algorithm 6 sensitivity index calculation in eFAST

2. `Compute sensitivity index`

A. Define two indices for utilizing symmetry property: $\mathrm{N_q} = \frac{\mathrm{N_{se}}-1}{2}$ and $\mathrm{N_0} = \frac{\mathrm{N_{se}}+1}{2}$.

for $k = 1, \ldots, \mathrm{D_s}$ **do**
  for $t = 1, \ldots, \mathrm{N_{OT}}$
    for $i = 1, \ldots, \mathrm{D_p}$
      for $r = 1, \ldots, \mathrm{N_r}$
        for $j = \omega_{\max} : \omega_{\max} : \mathrm{M_e} * \omega_{\max}$

D. Compute A_j^r by using equation

C. $\mathrm{A}_j^r = \frac{1}{\mathrm{N_{se}}}\{y(\mathrm{N_0}, t, s, i, r) + \{\sum_{q=1}^{\mathrm{N_q}} [y(\mathrm{N_0}+\mathrm{q}, t, s, i, r) + y(\mathrm{N_0}-\mathrm{q}, t, s, i, r)] \times \cos j\frac{\pi}{\mathrm{N_{Se}}}q\}\}$

D. $\mathrm{B}_j^r = \frac{1}{\mathrm{N_{se}}}\{\sum_{q=1}^{\mathrm{N_q}} [y(\mathrm{N_0}+\mathrm{q}, t, s, i, r) - y(\mathrm{N_0}-\mathrm{q}, t, s, i, r)] \times \sin j\frac{\pi}{\mathrm{N_{Se}}}q\}$

        end for
        for $j = 1 : (\mathrm{N_{se}} - 1)/2$

E. $\mathrm{A}_\mathrm{o}^r = \frac{1}{\mathrm{N_{se}}}\{y(\mathrm{N_0}, t, s, i, r) + \{\sum_{q=1}^{\mathrm{N_q}} [y(\mathrm{N_0}+\mathrm{q}, t, s, i, r) + y(\mathrm{N_0}-\mathrm{q}, t, s, i, r)] \times \cos j\frac{\pi}{\mathrm{N_{Se}}}q\}\}$

F. $\mathrm{B}_\mathrm{o}^r = \frac{1}{\mathrm{N_{se}}}\{\sum_{q=1}^{\mathrm{N_q}} [y(\mathrm{N_0}+\mathrm{q}, t, s, i, r) - y(\mathrm{N_0}-\mathrm{q}, t, s, i, r)] \times \sin j\frac{\pi}{\mathrm{N_{Se}}}q\}$

        end for

G. compute partial variance $\hat{\boldsymbol{D}}_{i,r}^{t,k} = 2\sum \mathrm{A}_j^r + \mathrm{B}_j^r$

H. compute overall variance $\hat{\boldsymbol{D}}_{\mathrm{overall},r}^{t,k} = 2\sum \mathrm{A}_\mathrm{o}^r + \mathrm{B}_\mathrm{o}^r$

      end for

I. compute the partial variance over $\mathrm{N_r}$ search curve: $\hat{D}_i^{t,k} = \mathrm{mean}(\hat{\boldsymbol{D}}_{i,r}^{t,k})$

J. compute the overall variance over $\mathrm{N_r}$ search curve: $\hat{D}_{\mathrm{overall}}^{t,k} = \mathrm{mean}(\hat{\boldsymbol{D}}_{\mathrm{overall},r}^{t,k})$

K. compute the importance of parameter θ_i for appraising k^{th} state in time instance t: $\mathrm{s}_i^{t,k} = \hat{D}_i^{t,k}/\hat{D}_{\mathrm{overall}}^{t,k}$

    end for
  end for
end for

- Output $\mathrm{D_p} \times \mathrm{N_{OT}} \times \mathrm{D_s}$ sensitivity index matrix S

eigenvectors and the corresponding eigenvalues as

$$\text{eigenvector} = \begin{bmatrix} v_{1,1} & v_{1,2} & \ldots & v_{1,p} \\ v_{2,1} & v_{2,2} & \ldots & v_{2,p} \\ \vdots & \vdots & \ldots & \vdots \\ v_{p,1} & v_{p,2} & \ldots & v_{p,p} \end{bmatrix} \quad \text{and} \quad \text{eigenvalue} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_p \end{bmatrix}, \tag{4.8}$$

the contribution from parameter θ_j to the last eigenvector is then calculated by

$$\chi_j = \frac{v_{j,p}^2}{\sum_{n=1}^{p} v_{n,p}^2} \tag{4.9}$$

where χ_j is called eigenparameter and represents the square of the eigenvector being reflected in the direction of the real parameter θ_j.

Since no extra computational cost is paid for the sampling, this PCA-based sensitivity analysis performs well computationally. Moreover, stiffness/sloppiness can be appro-

TABLE 4.2: List of variable notations for eFAST

eFAST			
number of resampling curves	N_r	how many resampling curves are utilized in parallel	scalar
number of samples picked from each curve	N_{se}	use to produce variation of parameter	scalar
frequency vector	$\boldsymbol{\omega}$	assign to parameters for computing amplitudes	$D_p \times 1$
highest frequency	ω_{max}	possible highest frequency can be assigned to the underlying parameter	scalar
highest complementary frequency	$\omega_{\text{-i,max}}$	possible highest frequency can be assigned to parameters except the underlying one	scalar
curve function	$G(\cdot)$	produce the values for parameters	none
incremental factor vectors	$\boldsymbol{s}$	construct the angles for $G(\cdot)$	$1 \times N_{se}$
random shift matrix	$\boldsymbol{\varphi}$	construct the angles for $G(\cdot)$	$D_p \times N_{se}$
interference factor	M_e	remove the amplitude to be computed from superposing	scalar
initial particles for parameters	$\boldsymbol{\theta}_0$	initially use for generating solution of dynamics	$D_p \times N_{se}$
synthetic dataset	$\mathbf{Y}$	dataset is generated by using all samples from curves	$N_{se} \times N_{OT} \times D_s \times D_p \times N_r$
sensitivity index matrix	$\mathbf{S}$	sensitivity of each parameter for different state at particular time instance	$D_p \times N_{OT} \times D_s$

priately differentiated by this algorithm if the accepted particles precisely estimate the parameters. However, when the particles fail to recover the unknown parameters, an incorrect sensitivity analysis from this PCA-based method is assigned. Investigation on this deficiency can be found in section 4.4.

4.1.3 ABC methods enhanced by sensitivity analysis

A new ABC based approach is developed by exploiting the fact that the values of the sloppy parameters can vary in a reasonable range, while the stiff parameters are determinants for evaluating the behavioral response and are therefore required to be precisely

assigned. This method can be seen as a selective allocation of the computing budget for sloppy and stiff parameters. It has three strategies, in which all parameters of a mode are simultaneously estimated alongside a coarse acceptance criterion. In the second phase of the eFAST technique, in order to differentiate the stiffness/sloppiness, the insensitive parameters are fixed to the values that are the mean of the posterior from the coarse analysis. In the final step, the stiff parameters are re-estimated by considering tighter error tolerances. Consequently, for ABC methods, this favorable allocation of computation budget alleviates the manual tuning for balancing accuracy and efficiency. Our approach is shown in Figure 4.6, and the pseudo-code is given in Algorithm 7.

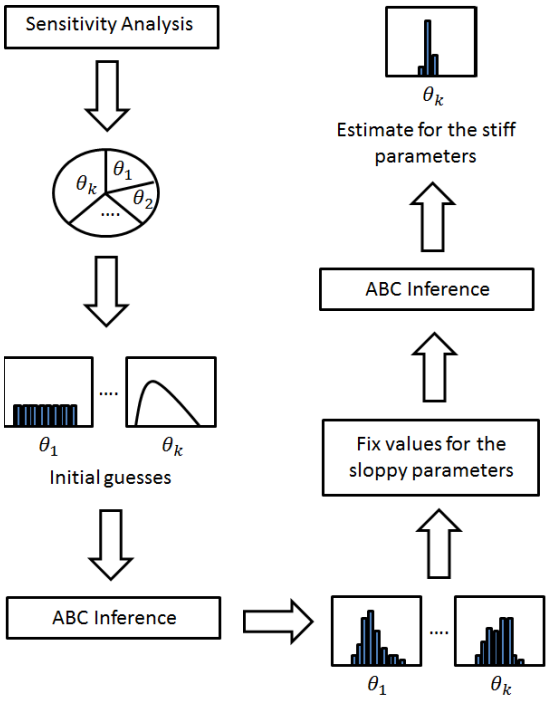


FIGURE 4.6: Computational steps in the proposed approach: Starting from an initial distribution of parameter values, we carry out a coarse approximate Bayesian Computation (ABC) estimation of parameters. Following this, using sensitivity analysis we identify sloppy and stiff parameters of the system. The sloppy parameters are fixed to values determined by the coarse analysis. In the final stage, we estimate the stiff parameters of the system by running the ABC method to tighter error tolerance. This achieves a selective partitioning of the computational budget, and reliable estimates can be achieved within reasonable times.

To illustrate the proposed method, we study four models taken from systems biology literatures. Three of those models are oscillatory processes: oscillations caused by transcriptional delay of an autoregulator (Monk, 2003), the repressilator circuit studied in the synthetic biology literature (Elowitz and Leibler, 2000) and the progression of cell cycle regulation (Leloup and Goldbeter, 2003). The fourth is the cellular response to heat shock model, a system we have considered in our past work using extended Kalman and particle filter approaches (El-Samad et al., 2006; Lillacci and Khammash, 2010; Liu and Niranjan, 2012). These cover a range of parameter values and complexities in the

Algorithm 7 ABC-SMC coupled with SA

`Input`: Initial parameters $\boldsymbol{\Theta}_0 \in \mathbb{R}^{\mathrm{D_p} \times \mathrm{N_{smc}}}$;
Small values for α_{smc} and N_{smc}
Other algorithmic settings defined in Table.3.1 and 4.1
`Output`: $\boldsymbol{\Theta}^T = \{\boldsymbol{\theta}_1^T, \ldots, \boldsymbol{\theta}_{N_{\mathrm{smc}}}^T\}$

Run eFAST

1. Carry out the eFAST on the system of interest. [Alg.5-6]
2. Group $\boldsymbol{\theta} \in \mathbb{R}^{D_{\mathrm{p}} \times 1} \rightarrow$ sloppy $\boldsymbol{\theta}_{\mathrm{sp}} \in \mathbb{R}^{D_{\mathrm{sp}} \times 1}$ and stiff $\boldsymbol{\theta}_{\mathrm{st}} \in \mathbb{R}^{D_{\mathrm{st}} \times 1}$.

Run 1st ABC-SMC

3. Apply ABC-SMC on $\boldsymbol{\theta}$ to propose the inference $\boldsymbol{\Theta}^*$. [Alg.4]
4. Fix sloppy parameters $\boldsymbol{\theta}_{\mathrm{sp}}$ in $\boldsymbol{\Theta}^T$ to corresponding values in $\boldsymbol{\Theta}^*$.

Run 2nd ABC-SMC

5. Increase values of α_{smc} and N_{smc}.
6. Apply ABC-SMC on $\boldsymbol{\theta}_{\mathrm{st}}$ to propose the inference $\boldsymbol{\Theta}^{**}$. [Alg.4]
7. Fix stiff parameters $\boldsymbol{\theta}_{\mathrm{st}}$ in $\boldsymbol{\Theta}^T$ to corresponding values in $\boldsymbol{\Theta}^{**}$.

End

structure of the nonlinear dynamical equations characterizing the systems. The following sections highlight the distinct feature of our method through several comparative studies.

4.2 Case study

In this section, we demonstrate the effectiveness of proposed method via three periodic and one transient biological systems, and the details of implementation are given in section D.3 of Appendix D. The Matlab code for producing results shown following discussion can be downloaded from Code link.

4.2.1 Delay-driven oscillatory system

The first model we consider is an oscillatory system in which periodic oscillations are caused by transcriptional delays. Such a system was formulated by Monk (2003) for quantitatively explaining oscillations of the tumor suppressor and related transcription factors `p53`, `Hes1` and `NF-`κ`B`. This oscillatory system is characterized by the delay differential equations, given as below

$$\begin{aligned} \frac{dm}{dt} &= \frac{1}{1 + (p(t-\tau)/p_0)^n} - \mu_m m(t) \\ \frac{dp}{dt} &= m(t) - \mu_p p(t), \end{aligned} \tag{4.10}$$

where, $m(t)$ and $p(t)$ describe the concentrations of mRNA and protein level in the system, their corresponding decay rates are shown as μ_m and μ_p. τ represents a transcription (and translation) delay and this lag is caused by the process of protein activation

so as to manipulate the transcription. The Hill coefficient and its threshold are denoted as n and p_0, respectively. The implementation details for performing this simulation are given in section D.3.

We assess the performance of ABC+SA in terms of accuracy of estimation and ability to reconstruct system behavior. Particularly, the precision of an estimate is quantitatively measured by the relative root mean square error (RRMSE), given as

$$\text{RRMSE} = \frac{\sqrt{\frac{\sum_{i=1}^{N}(\hat{\theta}_i - \theta_{\text{true}})^2}{N}}}{\theta_{\text{true}}}. \tag{4.11}$$

In addition, for the sake of comparison, the original ABC-MCMC is also carried out on parameter estimation associated with various tolerances, either a coarse acceptance criterion or a small tolerance. The analysis of this delay-driven autoregulation system is shown in Figure 4.7, as seen in the pie chart, μ_p and n are the stiff parameters while μ_m and P_0 are sloppy.

Marino et al. (2008) asserted that eFAST sometimes artificially yields a few negligible sensitivity indices for a parameter and these artifacts may derive from aliasing and interference effects. Consequently, Marino et al. (2008) introduced a 'dummy' parameter to eliminate this artifact. This factor does not really appear in models and has no effect on system behavior in any way, whilst the evolution of 'dummy' parameter is a standard practice in the screening methods in the family of global sensitivity analysis techniques (chapter 4 in Saltelli et al. (2000)).

In simulation, for a fair comparison, the small tolerance of both methods is identically set as $\epsilon = 1000$ and the coarse acceptance criterion for ABC+SA method is defined as $\epsilon = 3000$. Histograms of estimations obtained are shown in the A, B and C columns of Figure 4.7, from which it can be easily observed from the graphs that the ABC-MCMC associated with large ϵ performs poorly. For the original algorithm, however, the precision of estimations is comparable to ABC-MCMC+SA method whose posterior distributions center around the true values for the stiff parameters. In terms of accuracy, our proposed method is not clearly advantageous but still greatly outperforms ABC-MCMC in terms of computational efficiency, as shown in Figure 4.7.F. We further examine the performance of the original algorithm under a comparable computational cost, such that it takes a similar amount of time, and to this end the tolerance ϵ is set to 1400. Unsurprisingly, by utilizing this loose tolerance, ABC-MCMC can only deliver one precise estimation, yet fails to do so for the other three parameters.

Inferring the value of the parameter is essential to simulate the hidden states in the system, therefore, the ability of the parameter estimation algorithms to characterize the system behavior is further examined in this work. Curves in Figure 4.8 show the set of system outputs synthesized by utilizing the true values proposed in literature, and the inferred parameter values from a previous study. The distinct outperformance of

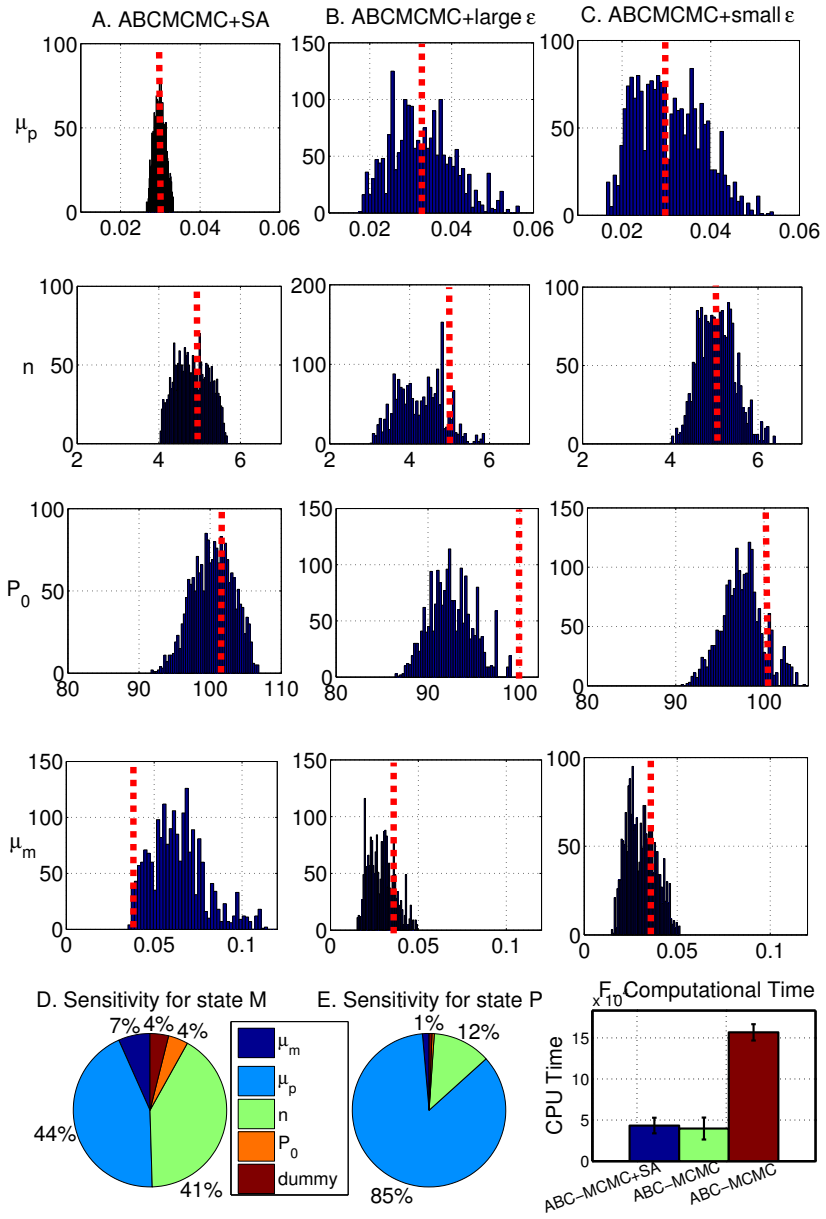


FIGURE 4.7: Sensitivity analysis and parameter estimation on the delay-driven p53 oscillatory model. Column A: Estimation of parameters μ_p, n, P_0 and μ_m from ABC-MCMC combining with SA. Column B: Estimation of the same parameters from ABC-MCMC with large tolerance ϵ. Column C: Estimation of the same parameters from ABC-MCMC with small tolerance ϵ. C and D: Average sensitivities of states (mRNA and protein) with respect to each of the parameters shown as pie charts (see text for technical details of the dummy variable). F: Computational times for the proposed method and ABC-MCMC associated with different tolerances (The green and red bars show the results of large ϵ and small ϵ, respectively).

the proposed method and ABC-MCMC with small tolerance is evident. However, due to the relatively inaccurate estimation of sloppy parameters, an offset in the synthetic dataset with respect to the 'real' one occurs in the last few time instants.

In addition, in order to quantitatively analyze these results, we calculate the RRMSE of inferences and list them in Table 4.3. The similarly low RRMSE values of stiff parameters

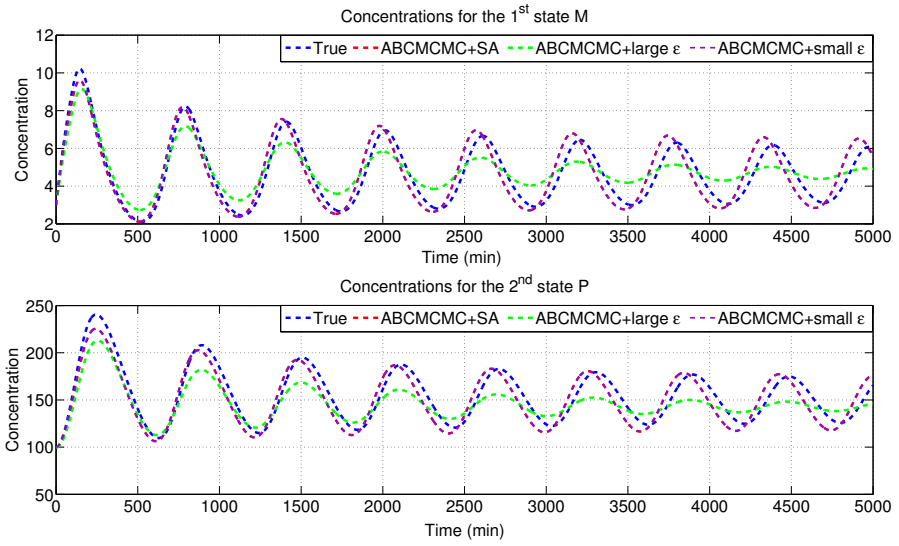


FIGURE 4.8: The curves for the concentrations of mRNA and protein in the delay-driven `p53` oscillatory model (noise-free). Simulations are produced by using true values (blue line) and inferred parameter values from ABC-MCMC+SA (red line), ABC-MCMC with loose tolerance (green line) and tight tolerance (purple line). Curves illustrate that the proposed method and the original algorithm with small ϵ perform highly similar (overlapped) in characterizing system behavior, while the imprecise parameter estimation from ABC-MCMC coupled with large ϵ delivers a bad reflection of process.

from the proposed method and ABC-MCMC with small tolerance explains why their corresponding synthetic datasets overlap with one another. Moreover, even though ABC-MCMC with loose acceptance criterion wins over the other two types of methods for some parameters (e.g accuracy of P_0 is highest of the three and the inference of μ_m is precisely in comparing to ABC-MCMC+SA), however, the sloppy parameters contribute less in characterizing system behavior, resulting in the corresponding synthetic dataset having the lowest precision.

TABLE 4.3: RRMSE of inferences from the proposed and original methods

parameter	ABC+SA	ABC+small ϵ	ABC+large ϵ
μ_p	$0.04 \pm 1.5 \times 10^{-5}$	$0.03 \pm 0.8 \times 10^{-5}$	$0.27 \pm 2.6 \times 10^{-8}$
n	$0.03 \pm 2.6 \times 10^{-5}$	$0.09 \pm 4.3 \times 10^{-4}$	$0.15 \pm 2.2 \times 10^{-3}$
P_0	$0.08 \pm 3.7 \times 10^{-5}$	$0.28 \pm 2.9 \times 10^{-3}$	$0.06 \pm 5.8 \times 10^{-4}$
μ_m	$0.58 \pm 7.3 \times 10^{-4}$	$0.27 \pm 6.4 \times 10^{-4}$	$0.26 \pm 1.7 \times 10^{-5}$

Consequently, for these simple scenarios, the proposed method is not remarkably accurate but a better computational efficiency is achieved.

4.2.2 Repressilator system

The deterministic repressilator system is constructed as a synthetic gene regulatory circuit which can sustain oscillations by the mutual repression of gene transcription (Elowitz and Leibler, 2000). This system consists of six differential equations, from three pairs of mRNA and protein, and has four parameters. This system was analysed by Toni et al. (2009) to demonstrate their ABC-SIS approach. The equations of the system are

$$\frac{dm_1}{dt} = -m_1 + \frac{\alpha}{1+p_3^n} + \alpha_0 \tag{4.12}$$

$$\frac{dp_1}{dt} = -\beta(p_1 - m_1) \tag{4.13}$$

$$\frac{dm_2}{dt} = -m_2 + \frac{\alpha}{1+p_1^n} + \alpha_0 \tag{4.14}$$

$$\frac{dp_2}{dt} = -\beta(p_2 - m_2) \tag{4.15}$$

$$\frac{dm_3}{dt} = -m_3 + \frac{\alpha}{1+p_2^n} + \alpha_0 \tag{4.16}$$

$$\frac{dp_3}{dt} = -\beta(p_3 - m_3), \tag{4.17}$$

where, α_0, α, n and β are the four parameters to be estimated from noisy observations of the six state variables, $m_1, \ldots, p_3$.

Figure 4.9.A describes the average sensitivity of parameters with respect to state m_1 in the system. Likewise, as shown in Figure 4.9.B, the sensitivity can be specifically evaluated at each time instant. Only the average sensitivity result takes part in this example but the decomposition of sensitivity is sometimes useful, e.g. to catalyze the specific reaction for achieving a rapid growth of species at a particular phase.

Considering these sensitivity results for ABC-SMC (Del Moral et al., 2012), we intend to initially fix the values of the sloppy parameters α and β taken from the first coarse search, given as 1035 and 5.698. A greater computational effort by increasing of N_{smc} and M_{smc} leads to precise estimates for the stiff parameters α_0 and n, where the results are shown in Figure 4.9.C-D. It is clear that the means of the posterior distributions of the stiff parameters converge to their true values.

Moreover, for analyzing the effect of the sloppy/stiff properties on parameter estimation, we further conduct simulations for all possible combinations of sensitivity assignments. Simulations are run 10 times for each combination, and results are summarized in Table 4.4. We note that success can only be achieve if the algorithmic setting has the appropriate sensitivity assignment and at least one stiff parameter is precisely inferred.

In addition, a fair comparative study between ABC-SMC+SA and ABC-SIS for pa-

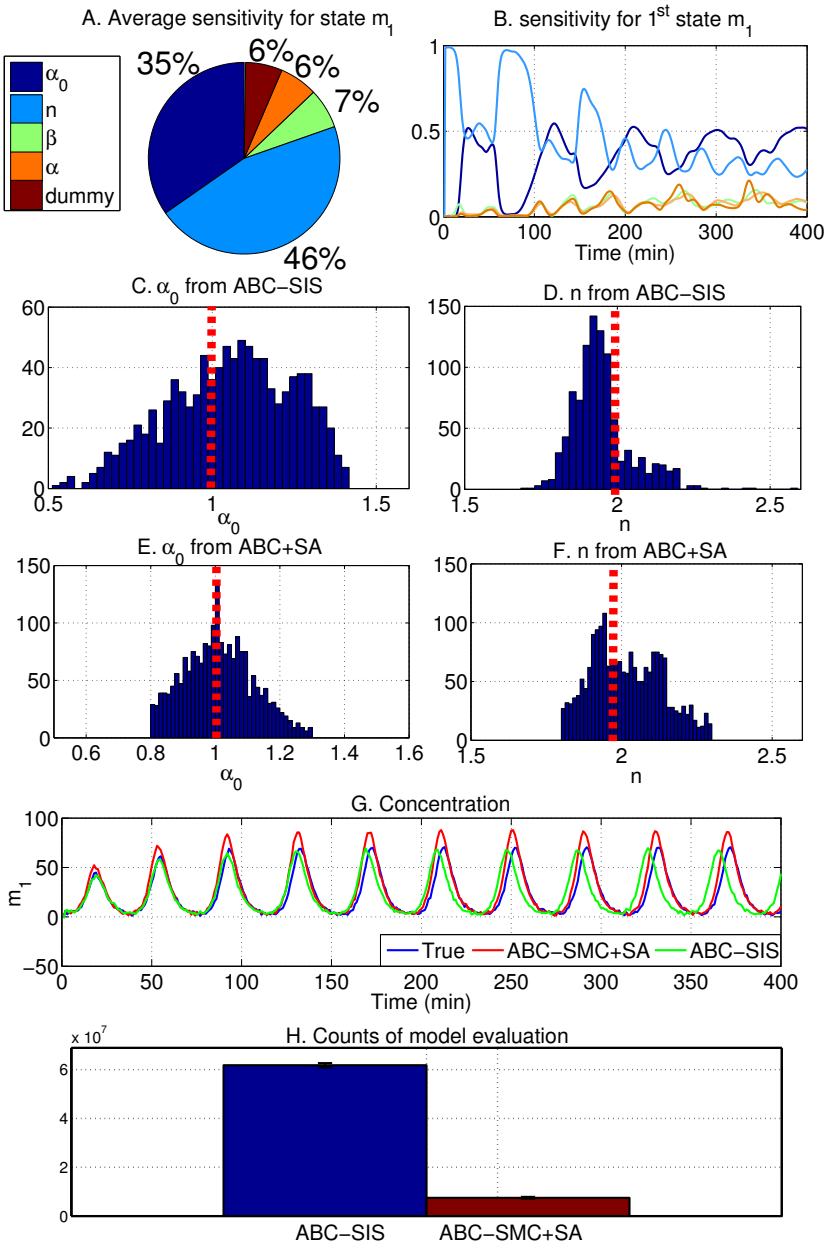


FIGURE 4.9: Results of sensitivity, parameter estimation and reproduction of the repressilator system. A: Pie graph shows the average sensitivity of parameters with respect to the state m_1. B: Curves represent the sensitivity of parameters for state m_1 at each time instant. C and D: Histograms show the estimations of the stiff parameters α_0 and n from ABC-SIS. E and F: Histograms for the same stiff parameters from ABC-SMC+SA. G: Simulations of state m_1 using true values, inferred values from ABC-SMC+SA and ABC-SIS. H: Counts of model evaluation taken by ABC-SIS and ABC-SMC+SA to achieve the final tolerance ϵ_T.

rameter estimation on repressilator system is conducted by setting identical algorithmic conditions, including regime of dataset, prior distribution and target tolerance. The results of stiff parameters from ABC-SIS are given in Figure 4.9.E-F. It is clear that the inferences from ABC-SIS successfully recover the true values of parameters, and are also similar to the results published in the literature (section 3.2 in Toni et al. (2009)).

TABLE 4.4: Parameter estimation to repressilator system within different sensitivity appointments

Combinations[a]							
Sloppy		Stiff		Mean of estimates[b]			
α_0	α	n	β	0.43±4.5 (×)	1900±2000 (×)	3.0±0.06 (×)	8.2±22.1 (×)
α_0	n	α	β	0.3±10(×)	2.3±0.6 (×)	1500±4700 (×)	3.7±8.8(×)
α_0	β	n	α	-0.1±4.7(×)	7.8±7.9(×)	3.2±0.2(×)	2100 ± 6100(×)
α	β	n	α_0	1600±2100 (×)	5.9±7.1(×)	2.3±0.4 (✓)	1.4±1.9(✓)
α	n	β	α_0	1700±1800(×)	2.1±0.2(×)	4.7±16.1(✓)	0.5±2.8(×)
n	β	α	α_0	2.8±0.3(×)	8.4±35(×)	1600±2600(×)	-1.6±2.6(×)

[a] True values: $\alpha_0 = 1$; $\alpha = 1000$; $\beta = 5$; $n = 2$
[b] Indicator: success (✓); failure (×).

Similarly to the previous study, the abilities of methods to recreate system dynamics is considered. Simulated system dynamics by using the true and inferred values are shown in Figure 4.9.G. Even though the proposed method is problematic in producing the identical amplitude while an offset in periodicity is observed in the simulation of ABC-SIS, these two methods can still be seen as capable of mimicking the system outputs.

As suggested by Toni et al. (2009), the prior distribution and transition kernel need to be chosen with care, otherwise the inference would be unacceptable. In the original literature, this claim was investigated using the stochastic Lotka-Volterra model with different prior distributions, where promising results can only be obtained with particular priors. In this repressilator example, we struggled with tuning the algorithm to deliver satisfactory accuracy, because the details of the transition kernel and tolerance schedule are not provided in the original work. This algorithmic setting is appropriate from the viewpoint of accuracy, but requires an unaffordable computational cost. In order to quantify how ABC-SIS is influenced by this algorithmic setting, we compare the computational expense between ABC-SIS and our proposed method using identical initial and final tolerances. Since ABC-SMC on average requires 140 iterations to reach ϵ_T we reduce the tolerance from ϵ_0 down to ϵ_T using 140 steps at regular intervals. The results are shown in Figure 4.9.H, from which it can be easily seen that the number of evaluations made by ABC-SIS is approximately seven times greater than ABC-SMC. We note that the conclusion drawn from this comparison is somehow opposite to the investigation presented in section 3.3.2, because the dimension of unknown parameters of the repressilator model (4 unknown parameters) is higher than the previously considered system (3 unknown parameters). Additionally, the repressilator example uses a more uninformed prior than the heat shock system, and therefore ABC-SIS expends considerable computation on searching for acceptable particles due to the 'non-fittest' chosen transition kernel and tolerance schedule.

Consequently, although the use of sensitivity analysis has no significant effect for ABC-SMC on improving the accuracy of estimation in this repressilator example, it does

provide inspiration for how the problem in high dimensions may be tackled. In comparison to ABC-SMC, ABC-SIS requires a great deal of manual tweaking to achieve satisfactory precision, while the computational complexity of ABC-SIS is much higher than ABC-SMC when using an inappropriate algorithmic setting.

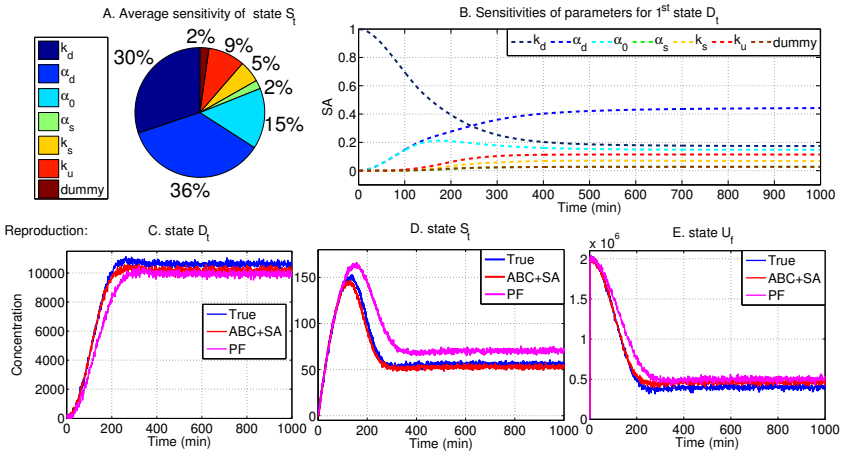
4.2.3 Heat shock response system

In the section A.4 of Appendix A, we considered the heat shock protein response system as a biological example for demonstrating the effectivenesses of Kalman algorithms and particle filter. When all parameters were unknown, being the hardest case considered, the non-parametric PF is able to recover four unknowns of six parameters. In order to discriminate the abilities of the proposed method and PF, a comparative study is carried out on the heat shock response system under the assumption that all parameters are unknown.

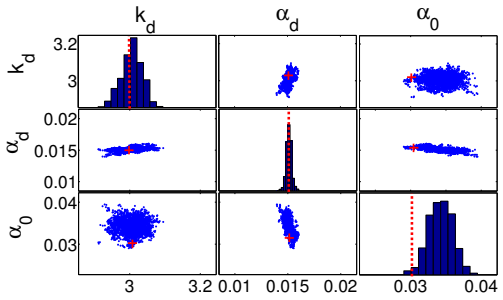
Figure 4.10(a).A describes the average sensitivity of parameters with respect to state S_t in the system, as shown in graph, the parameters α_d, k_d and α_0 are sensitive for producing system outputs, which are thus required to be precisely inferred. Making use of identical algorithmic settings (Liu and Niranjan, 2012), the estimation of stiff parameters from the proposed method are shown in Figure 4.10(b) and the results of sloppy parameters are given in Figure 4.10(c). It can be easily seen that the particles of the stiff parameters converge the true values (α_0 is inferred with relatively low precision and large variance, this is due to its less significance in behaving system dynamics, in comparison to the other two stiff parameters), whereas it fails to recover the true values of the sloppy parameters. Additionally, in the previously studied example, when the state S_t is hidden in observations, PF was found to be incapable of precisely inferring the parameters k_d and α_d. Given the sensitivity of parameters with respect to the state S_t, it is naturally expected this failure, since the behavior of S_t is governed by these stiff unknown parameters. Consequently, if S_t is hidden in the observations, its corresponding stiff parameters are impossible to estimate.

In particular, assuming all parameters unknown, PF (four of six) seemingly wins the battle over ABC-SMC+SA (three of six) in terms of successful inferences. Figure 4.10(a).C-E suggests that the proposed method slightly outperforms PF in terms of re-creating system dynamics, especially state S_t.

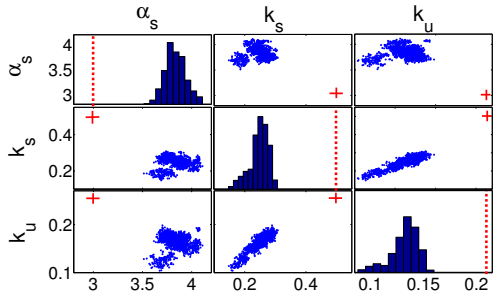
Consequently, the typical one-pass inference methods, e.g. particle filter and extended Kalman filter appear to be capable of efficiently producing the most promising estimations of partial parameters. While ABC-SMC+SA loses the competition in the count of successes, it does benefit from re-allocation of computational budget facilitated by the sensitivity analysis, and the method is capable of precisely estimating stiff parameters.



(a) sensitivity analysis and state reproduction for heat shock system



(b) stiff parameters of heat shock system from ABC+SA



(c) sloppy parameters of heat shock system from ABC+SA

FIGURE 4.10: Sensitivity analysis, inference of parameters and system re-characterization of heat shock model. (a) A: Average sensitivities of parameters with respect to state S_t. B: Sensitivities of parameters for S_t at each time instant. C-D: Reproduction of state S_t by using true values, estimates from ABC+SA and particle filter respectively. (b) and (c): Scatterplots and Histograms for the stiff parameters (k_d, α_d and α_0) and sloppy parameters (α_s, k_s and k_u). The red lines indicate the true values of parameters and the red '+' implies the location of the true parameters.

4.2.4 Deterministic cell cycle system

A system with high dimension of parameters is the ideal scenario for using the proposed method to address the inference problem. In this work, the effectiveness of ABC-SMC+SA approach on parameter estimation in a complex scenario is illustrated by the

cell cycle system.

The philosophy of cell cycle system can be found in the section 2.2.3 of chapter 2, and the model considered in this work mathematically describes the cytokinesis appearing in *Saccharomyces cerevisiae*. The formula of this process consists of six ordinary differential equations with twenty parameters (Jacquet et al., 2003), of which the details are described in Appendix D. Eight parameters are constructed without physical interpretation and assigned identical values and therefore we seek only to estimate the other twelve parameters.

Partial results of average sensitivities of parameters with respect to three state variables are shown in Figure 4.11.A-C. It is clear that parameters V_{kx} and V_{px} are most significant for characterizing state M, while V_p and V_{ks} are crucial in the behavior of states M^* and MN.

ABC-SMC, associated with the flat prior, either non-informative or informative, is unable to terminate the ODE solver. This is due to, in high dimensional systems, the parameter values chosen for the ODE solver, i.e. Runge-Kutta method, being less probable for the algorithm to achieve convergence and this difficulty causes infinite computational consumption towards finding a solution. When ABC methods tackle inference problems involving complex systems then prior to performance they should be tuned to find a particular parameter value interval that can deliver the convergence of ODE solver.

As an alternative to the original ABC-SMC, by combining it with sensitivity analysis we may partially alleviate. It is unnecessary to tune the prior until the ODE solver is terminated. Instead, the estimator determines the sloppy parameters by adopting the values where the inference algorithm gets trapped.

The estimates of the two stiff parameters are shown in Figure 4.11.D-E (the complete results are given in Figure D.2 - D.3 of Appendix D), as seen from the graphs, inferences achieve a good convergence to the true values. Surprisingly, as shown in Figure 4.12.B and E, the method produces an inaccurate estimation of parameter V_{ks} having the stiff properties, as well as in parameter V_p.

Particularly, the effect of imprecise estimations on system characterization is shown in Figure 4.12.C, in which the divergence is evident. Such divergence may be due to the greater values of V_p and V_{ks} resulting in the forward reaction contributing much more than the backward reaction.

To increase precision of its inferences, ABC-SMC+SA method further reduces the dimensionality of the unknown by fixing the two stiff parameters with accurate estimates, and re-identifying the parameters causes a failure of system characterization. Particles of these two unknown parameters, as shown in Figure 4.12.A and D, ultimately center around their true values. Consequently, the synthetic dataset is capable of capturing

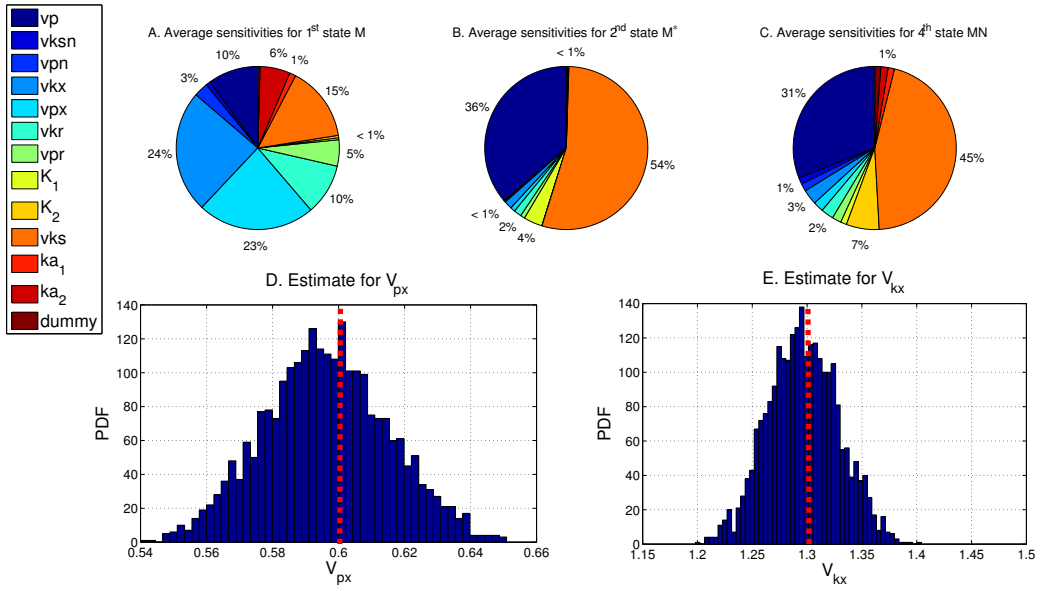


FIGURE 4.11: Results for sensitivity analysis and parameter estimation of cell cycle model. A, B and C: Average sensitivity status of parameters for states M, M^* and MN. (other three graphs will be presented in Figure D.1 of Appendix D); D and E: Histogram graphs for estimates of the stiff parameters V_{px} and V_{kx} from ABC-SMC+SA, of which the true values are highlighted by the red lines in the figures.

the dynamics better when all stiff parameters are precisely inferred, where the results are shown in Figure 4.12.F.

In the algorithmic sense, the proposed method is sensitive to the noise corrupting observations. When the noise variance multiplier is set to 0.01, the algorithm performed decently. However, increases of this multiplier negatively influences the method and values greater than 0.1 causes a failure in inference.

In addition, the proposed method using kernel smoothing with shrinkage of parameter evolution, i.e. equation A.41, outperforms the random walk kernel in terms of convergence and adaptivity. This kernel proposes particles according to the mean and variance of the previous posterior distribution, and resulting transition influences the specific direction of particle perturbation automatically 'shrinking' the step size with decreasing variance. The random walk kernel, however, perturbs particle with a non-specific direction and a fixed distance in each iteration, as a result of which it may struggle with the multi-modal issue. A comparative study of two evolutions is carried out on the cell cycle system, in which the posterior distributions are plotted in the waterfall effect. The evidence given in Figure 4.13 supports the conclusion, and the advantage of the proposed method in terms of convergence is apparent.

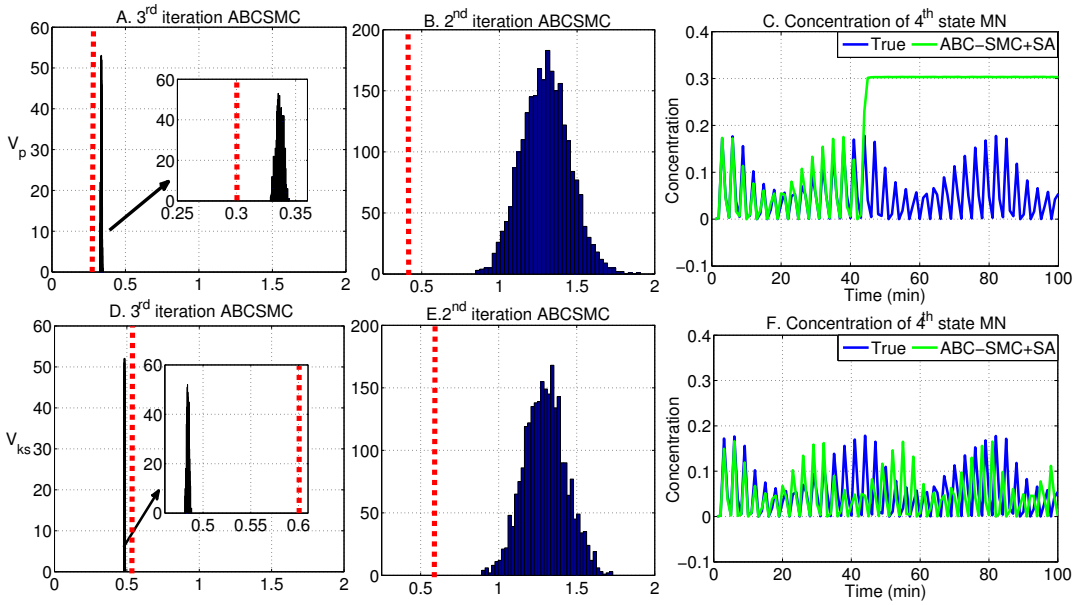


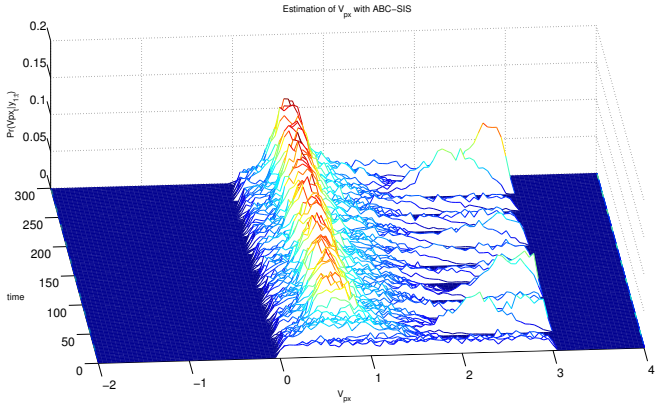
FIGURE 4.12: System reproduction and parameter estimation for cell cycle system from the second and third parses. A and D: Histogram graphs of the promising inferences for parameters V_p and V_{ks} from 3^{rd} estimate iteration. In figure, the original windows use the same x-axis as the B and E for comparison purpose. Histograms of realizations are zoomed-in and shown in the small windows. B and E: Histogram graphs of the imprecise inferences from 2^{nd} estimate iteration. C: Curves of the concentration for state MN synthesized by using the true values and estimations from the 2^{nd} parse. Clearly, simulations diverge after a few iterations. F: Synthetic outputs of state MN, while which is simulated by utilizing the values from the 3^{rd} parse.

4.3 How the choice of M_{smc}/N_{smc} affects performance

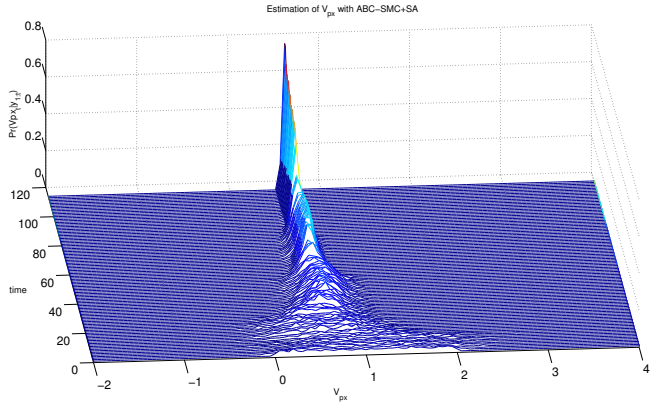
In ABC-SMC algorithm, the M_{smc} and N_{smc} would prefer to be set as large as possible, but are unfortunately limited by the computational budget, which strikes a balance between efficiency and accuracy. The use of M_{smc} and N_{smc} is highly case dependent, and an one-for-all answer of selecting this algorithmic setting is sometimes impossible. The solution of this is difficult, but can be investigated by empirically testing the performance of different parameterizations.

A comparison of various settings is carried out on the repressilator system, in which ABC-SMC is performed without SA and the amount of particles, i.e. the multiplication of M_{smc} and N_{smc}, is fixed as 40,000. More specifically, the particles are distributed in three combinations: $M_{smc} = 20$ and $N_{smc} = 2,000$, $M_{smc} = 200$ and $N_{smc} = 200$, or $M_{smc} = 2,000$ and $N_{smc} = 20$. Performances of these three combinations in terms of tolerance reduction and count of iterations are given in Figure 4.14, and the accuracy of inferences is summarized in Table 4.5.

As seen in the graphs, from the perspective of computational efficiency, a greater budget is required with decreasing values of M_{smc}. This is due to lower value of M_{smc} causing a higher probability of particles being 'killed' and having a zero weight. This in turn results



(a) random walk kernel



(b) kernel smoothing with shrinkage

FIGURE 4.13: Estimations of V_{px} in the cell cycle system are shown in waterfall effect. (a) Results from ABC-SIS associated with the random walk kernel, in which the multi-modality is evident. (b) Results from ABC-SMC with kernel smoothing. This transition of parameter offers a better convergence property.

in tolerance taking longer to converge due to smaller decrements being applied. For instance, considering the large M_{smc} example, *i.e.* $M_{smc} = 2000$, evaluating importance using equation 3.25, the zero weight barely appears. Intuitively, the greater non-zero proportion of the weight vector implies that most particles fulfill the current acceptance criterion, therefore, a large decrement should be taken for the next tolerance level.

TABLE 4.5: Comparison of RRMSE for different values of M_{smc}/N_{smc}

		RRMSE				
M_{smc}	N_{smc}		α_0	n	β	α
2000	20	Paras	20.8±72.2	4.8±0.15	7.8±0.3	4.1±0.02
200	200		16.1±46.1	3.5±0.005	6.2±7.7	3.1±0.15
20	2000		12.4±56.9	3.6±0.16	5.8±0.4	2.2±0.67

From the perspective of accuracy, when gaining a larger number of particles, the diversity of realizations increases and better performance in terms of precision is naturally expected. This conclusion is illustrated by the comparison of RRMSE of inferences

shown in Table 4.5, in which settings where $N_{smc} = 2000$ perform best, with higher values generally outperforming lower values.

Moreover, aiming only to assess the effect of M_{smc} and N_{smc} on ABC-SMC, the algorithm is run without SA and adopts a relatively loose tolerance, as a result of which, inferences are relatively inaccurate.

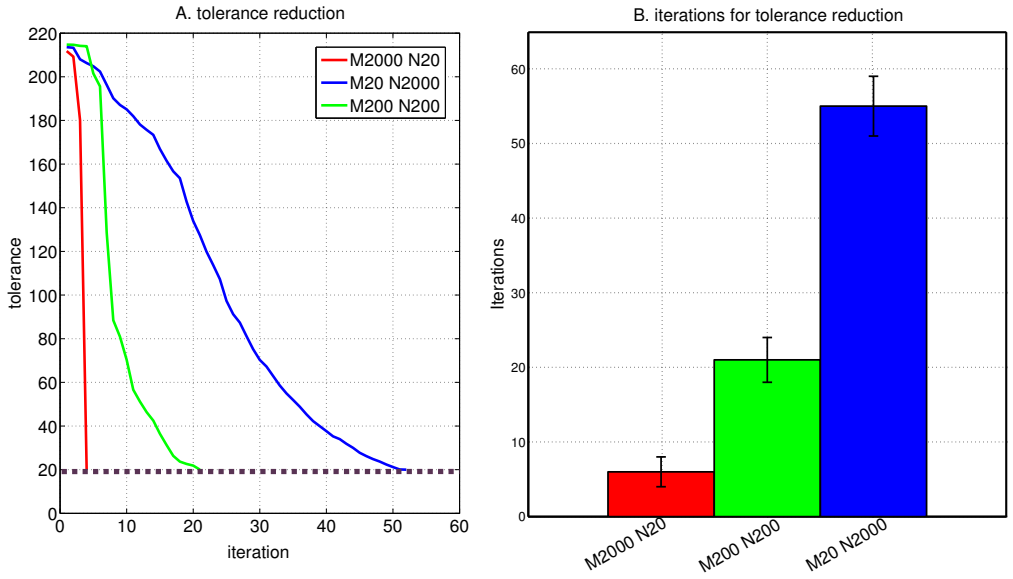


FIGURE 4.14: A: Tolerance paths received from different combinations of M_{smc} and N_{smc}, where the target epsilon ϵ_{Ta} is set to 20. B: Counts of iterations are taken by different combinations of M_{smc} and N_{smc} to achieve the ϵ_{Ta}.

4.4 Why use eFAST for sensitivity analysis?

Sensitivity analysis can be carried out globally, or locally, to appraise the significance of parameters in dynamics. eFAST, which operates globally, was performed to investigate this property so as to guide the re-allocation of the computational budget. From the local perspective, apart from the gradient based solutions, Toni et al. (2009); Secrier et al. (2009) undertook a kinetic study associated with the principle component analysis and demonstrated its success on some models from systems biology literatures.

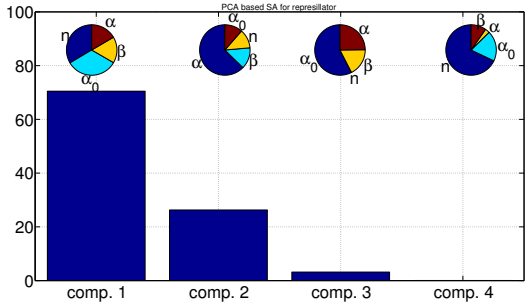
In brief, this method quantifies the sensitivity according to how parameters contribute most to the eigenvector of the covariance matrix of the particles. More specifically, for the smallest eigenvalue, the corresponding eigenvector indicates the direction of certainty. Consequently, a parameter is regarded as stiff when it contributes to this eigenvector the most. Details on the operation of this PCA-based sensitivity analysis technique can be found in section 2.6 of chapter 2.

Its simple implementation and negligible computational requirements are the major virtues of the PCA-based method, but it is strongly dependent on the precision of

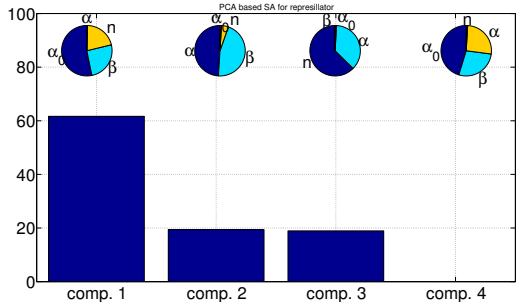
inference. When inaccurate estimates are fed to such methods, the reliability of analysis will be negatively affected. An illustration is carried out on the repressilator system, in which we deploy the PCA-based approach on two sets of inferences, one of which achieves a good convergence to the true values, while another does not.

The effectiveness of PCA-based method, in an ideal algorithmic environment, is clearly observed in Figure 4.15(a) where parameters n and α_0 are seen as stiff, since these two parameters contribute to the eigenvector which has the smallest eigenvalue (i.e. comp.4 in graph) the most. This result is identical to the one from eFAST and from the literature (Toni et al., 2009). When this method is used with imprecise inferences, as shown in Figure 4.15(b), the significances of β and α are greater than n and are unreliable. In contrast, eFAST delivers the global solution of sensitivity analysis, and barely struggles with the local minimum.

To facilitate the proposed inference method, given inaccurate estimates, the superior performance of sensitivity analysis approach is required to work with a selective computational allocation. To this end, eFAST takes priority.



(a) Correct PCA based sensitivity analysis for repressilator



(b) Incorrect PCA based sensitivity analysis for repressilator

FIGURE 4.15: Sensitivity analysis for repressilator by using PCA-based technique Toni et al. (2009). (a) Correct sensitivity analysis, one is the same as one presented in Toni et al. (2009). (b) Incorrect sensitivity analysis, since the failure is caused by concerning unreliable posterior population for inferred parameters.

4.5 Discussion

In this chapter, we proposed an approximate Bayesian computation coupled with sensitivity analysis inference method for parameter estimation problems in the context of systems biology. When all the model parameters are simultaneously estimated, the estimator is often required to strike a balance between accuracy and computational efficiency. The method works on the fact that each model parameter has different significances for characterizing dynamics, those which are dominant are categorized as stiff, others are sloppy. This suggests that the values of parameters that are more critical (stiff parameters) need to be determined with care, while the sloppy parameters need not be estimated to high precision. To facilitate such inference, we propose a three stage strategy in which sloppy parameters of a model are estimated in a coarse search followed by sensitivity analysis to guide the selective computational budget allocation and re-estimation of the stiff parameters to tighter error tolerances.

The effectiveness of the proposed method is demonstrated on three oscillatory models and one transient response model taken from the systems biology literature. In the simple problem, e.g. the delay-driven oscillatory system, the outperformance of this ABC+SA method in terms of accuracy was not observed. Nevertheless, this inference algorithm does indeed lead to an improvement in the computational efficiency. The major contribution of this work is the introduction of a re-allocation scheme for computational budget, which allows to reduce the dimensionality of unknown parameters, potentially paving a way for parameter estimation on more complex systems.

However, it is envisioned that the method proposed may lose its appeal without a validation assessment. As exemplified in our case study of cell cycle system, the method found itself only capable of partially reaching the true values, and an extra inference iteration was carried out to compensate for this imprecision. In reality, inaccurate inferences are usually hard to realize, therefore, a validation assessment becomes necessary to work with the proposed method to check the reliability of inferences. The temporary yet non-systematic solution of this problem is to perform the algorithm with successive iterations and then verify the success by integrating information from dynamics re-characterization.

In addition, when parameters in the system have no noticeable difference in sensitivity, the advantage of proposed method cannot be expected. In the cell cycle system, for instance, parameters equally contribute to the output of state RA (sensitivity analysis with respect to RA is given in Figure D.1 of Appendix D), as a result of which, all parameters need to be determined with care.

Chapter 5

Modeling a polymer pathway

In this chapter, we conduct a mathematical modeling of intracellular polymer synthesis in Alcaligenes eutrophus bacterium. The development of the model described in section 5.1 closely follows the work of Leaf and Srienc (1997), and includes the basic biochemical reaction descriptions for completeness. The novel contributions of our work is carried out the quantitative analysis including sensitivity analysis and parameter identification to this polymer pathway, via the proposed approach presented in chapter 4.

5.1 Biochemical pathway modeling

An enzyme-catalyzed pathway for producing polymers is considered in this work. Since disposable plastics has become a serious issue in the past decades, as the solution of this problem, Polyhydroxybutyrate (PHB), benefiting from its biodegradable property, is widely used as the material for producing plastics. Several bacteria such as *Escherichia coli* and *Ralstonia eutrophus* are used as the bacterial fermentation for producing PHB. Figure 5.1 shows a *Alcaligenes eutrophus* bacterium based pathway for synthesizing PHB, in which glucose is initially fed as the substrate. Glucose is next converted to form pyruvate through glycolysis reaction, with the release of free energy. As the product of glycolysis, pyruvate is a crucial intersection in metabolic reactions, and determines the end-product. Of interest is to analyze the sub-routine in pathway for producing PHB from acetyl-CoA, in which three enzymatic reactions, i.e. thiolase, reductase and synthase, are taking place (Poirier et al., 1995).

Several quantitative tools and culturing methods have been deployed to understand the mechanism of PHB pathway. A representative work due to van Wegen et al. (2001), experimentally measured the concentrations of intermediates in this polymerization process, performing in *Escherichia coli*. In this work, the concentration ratio of acetyl-CoA/CoA is claimed as the most sensitive for the PHB production. Kessler and Witholt

(2001) reported the diversity of regulatory mechanisms of PHB metabolism, and examined the strategies of the transcription levels and enzymatic levels to exert regulation under various microorganisms. Shang et al. (2007) discussed the role of dissolved oxygen in fermentation, and suggested that this factor is as crucial as the functions of carbon and nitrogen for batch-fermentation. Moreover, variety of models for this PHB pathway are proposed by using different kinetic expressions. Gombert and Nielsen (2001) considered the Michaelis-Mention kinetic for characterizing PHB synthesis, and various values of parameters in expressions need to be adapted under different physiological conditions. Wlaschin et al. (2006) quantitatively elucidated the connections between enzymes and metabolites of PHB network by viewing the metabolism as a weighted sum of elementary modes.

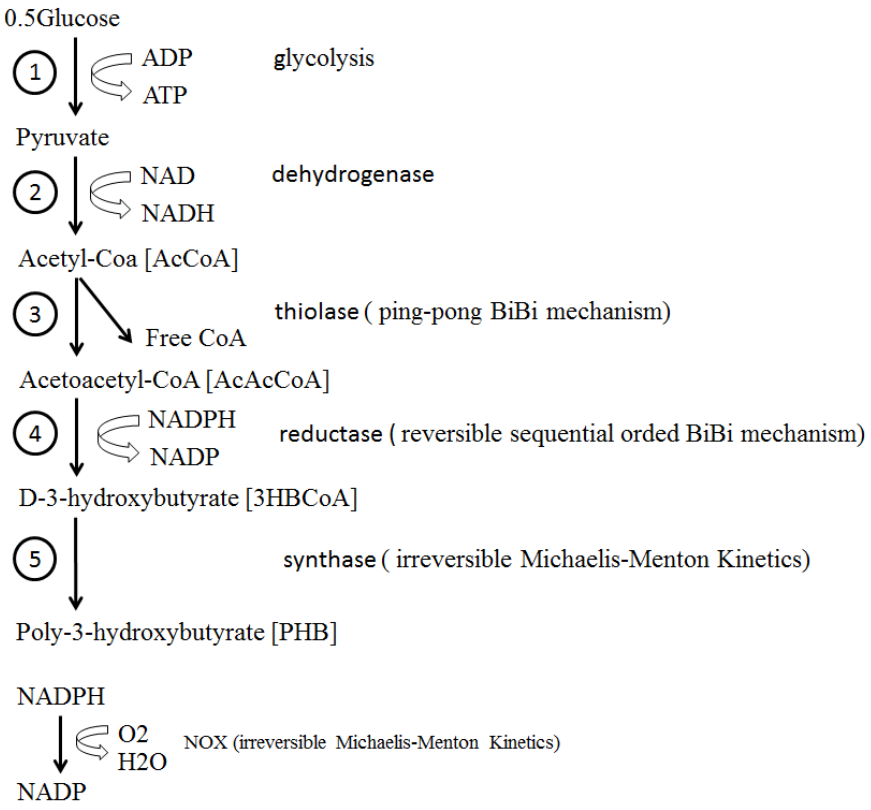


FIGURE 5.1: Metabolic pathway transforming from glucose to poly-3-hydroxybutyrate. In this system, ADP and NADPH are used to power the process. For modeling purposes, Acetoacetyl-Coa, D-3-hydroxybuyrate and Poly-3-hydroxybutyrate are simplified as AcAcCoA, 3HBCoA and PHB, respectively.

Leaf and Srienc (1997) modeled the PHB pathway through a principle driven method, in which the system is formulated by three differential equations and twenty parameters. In this work, we also follow the identical kinetics modelling method to formulate this polymerization process, as a result, the difference between our expression and the previous model is minor. The procedure of modelling is described following. In order to describe the change of flux concentration with respect to time, we first introduce a set

of balance equations of the pathway intermediates, denoted as

$$\frac{d[\text{AcAcCoA}]}{dt} = v_{\text{thiolase}} - v_{\text{reductase}} - \mu[\text{AcAcCoA}] \tag{5.1}$$

$$\frac{d[\text{3HBCoA}]}{dt} = v_{\text{reductase}} - v_{\text{synthase}} - \mu[\text{3HBCoA}] \tag{5.2}$$

$$\frac{d[\text{PHB}]}{dt} = v_{\text{synthase}} - \mu[\text{PHB}], \tag{5.3}$$

where v denotes the rate of reaction and μ represents the rate of self-dilution caused by expansion of the biomass during growth (Fredrickson, 1976). Each balance equation generally indicates increase or decrease of its particular intermediate during reaction. For solving such balance equations, the rates of reactions are necessarily formulated explicitly. The modeling of reactions has been previously addressed by the intensive use of Michaelis-Menten expressions, which appears unrealistic, therefore, we choose to develop more complex expressions to describe these mechanisms

Quite often, in modeling of enzyme-catalyzed reaction, solution is exploited by the number of substrates and products and mechanisms between these components (Cleland, 1963). The symbols are often used to describe the kinetic information, such as

$$\mathbf{A} \rightleftharpoons \mathbf{P} \qquad \text{Uni Uni} \tag{5.4}$$

$$\mathbf{A} \rightleftharpoons \mathbf{P} + \mathbf{Q} \qquad \text{Uni Bi} \tag{5.5}$$

$$\mathbf{A} + \mathbf{B} \rightleftharpoons \mathbf{P} \qquad \text{Bi Uni} \tag{5.6}$$

$$\mathbf{A} + \mathbf{B} \rightleftharpoons \mathbf{P} + \mathbf{Q} \qquad \text{Bi Bi} \tag{5.7}$$

Apart from classifying the number of substrates and products, in enzyme kinetics, four types of mechanisms are also defined (Roberts, 1977). If all substrates are participated before any products are released, the reaction is then called *sequential*. The mechanism is known as *ordered* when the addition of substrates and leaving of products follows an obligatory order. Correspondingly, if the substrates are participated and the products are released without an obligatory order, such mechanism is called *random*. The most complex one is known as *Ping Pong*, in which the release of one or more products happens before all substrates participated. In the next section we illustrate how three rate reactions that appear in the PHB pathway are modeled.

5.1.1 Modeling of enzyme-catalyzed reactions

In this section, we discuss the methodology for characterizing enzyme catalyzed reaction which closely follows the presentation in Roberts (1977). For the sake of completeness, this modeling process is described.

In this polymerization system, thiolase is the enzyme for catalyzing the reaction for converting acetyl-CoA (AcCoA) to form acetoacetyl-CoA (AcAcCoA) along with releasing

part of co-enzymes (CoA). This reaction is shown as the step 3 in Figure 5.1 which is mathematically described as

$$\mathrm{AcCoA} + \mathrm{AcCoA} \rightleftharpoons \mathrm{AcAcCoA} + \mathrm{CoA} \tag{5.8}$$

Since this thiolase reaction occurs satisfying the Ping-Pong Bi-Bi mechanism (Davis et al., 1987), we can simply draw a diagrammatical description given in Figure 5.2, in which E denotes the thiolase enzyme. A and B are acetyl-CoA, which are the sub-

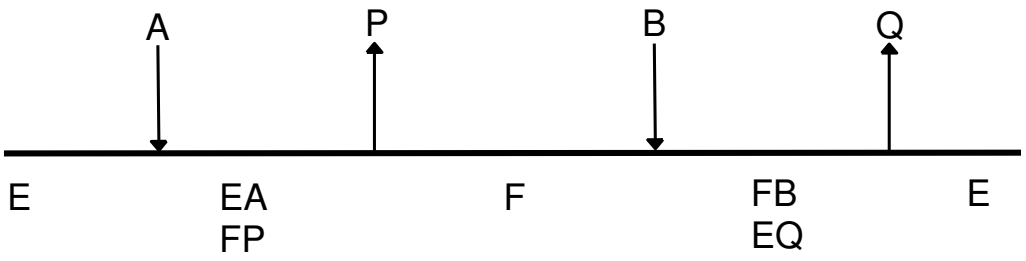


FIGURE 5.2: The diagrammatical description of Ping-Pong Bi-Bi mechanism for thiolase reaction.

strates of this reaction. P represents acetoacetyl-CoA, the first product in the reaction, while Q denotes CoA, the second product in this sub-pathway. The EA, FP, FB and EQ are intermediate complexes (Michaelis-Menten complex) (Roberts, 1977) after the participation of substrates or the departure of products.

To consider the structure of a reaction, it is convenient to use a diagrammatic method, namely the King-Altmen method (King and Altman, 1956) Specifically, enzymes in the reaction are formed in different geometrical patterns with arrows to represent the possible interconversions of species. Each arrow in patterns is labelled with a rate constant and any substrates/products may be consumed in the corresponding step. If the underlying step is reversible, a backward arrow is attached following the same philosophy. Consequently, in the King-Altmen procedure, the thiolase reaction can be formed as Figure 5.3 The next step in the King-Altmen method is to propose all possible ways

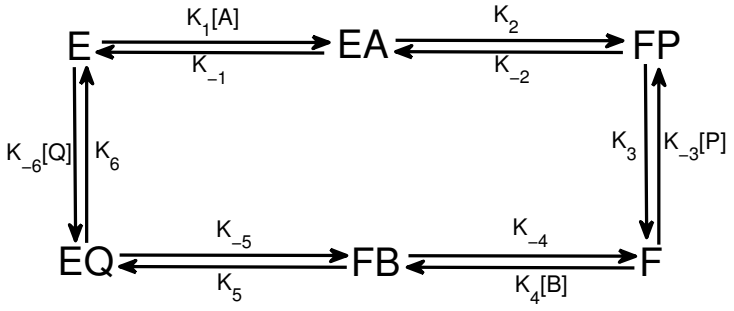


FIGURE 5.3: Thiolase reaction formed in the King-Altmen procedure.

that each enzyme can be connected. Two conditions are applied: First, the number of lines in pattern for connecting enzymes should be one line less than the number of species. Second, the underlying enzyme can not be the starting point and the finishing point at one time, therefore, close loops are not allowed in the King-Altmen description.

The diagram of thiolase reaction can then be decomposed into six patterns. The possible patterns for species E and EA in reaction, are given in Figure 5.4 below The patterns

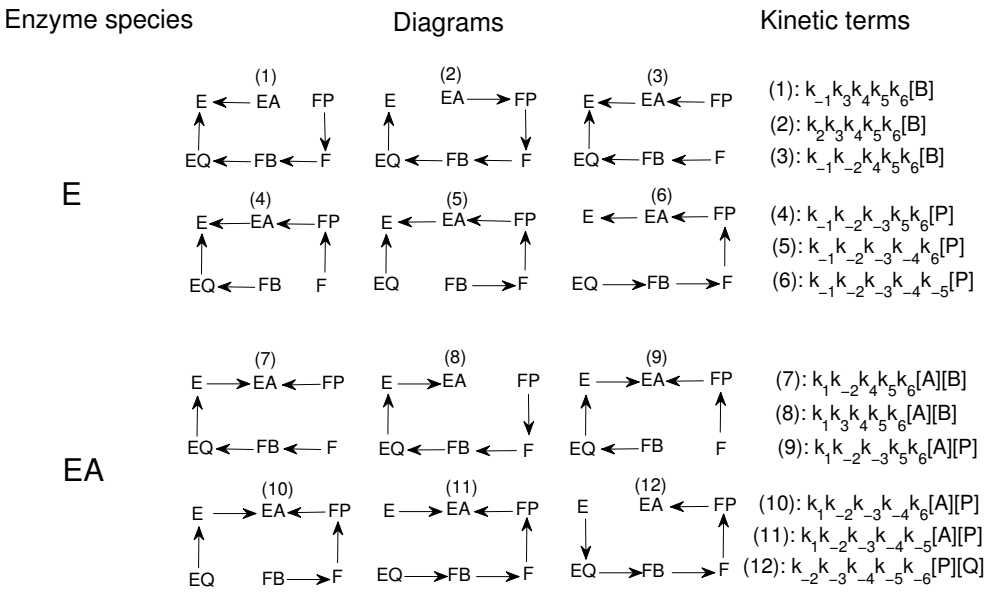


FIGURE 5.4: Possible patterns of species E and EA in the thiolase reaction and their corresponding kinetic expressions.

of intermediates FP and F are given in Figure 5.5.

All possible patterns of enzyme species FB and EQ are shown in Figure 5.6. As more substrates and products are consumed or released the number of possible patterns drawn from the King-Altmen method increases considerably. Therefore, finding the correct geometric patterns is the central challenge in modeling enzymatic reaction.

In this thiolase-catalyzed reaction, of interest is to derive the rate equation of the first product, i.e. AcAcCoA (P). From the scheme shown in Figure 5.3, the differential equation describing the rate of production of AcAcCoA (P) can be written as

$$\frac{d[\mathrm{P}]}{dt} = k_3[\mathrm{FP}] - k_{-3}[\mathrm{F}][\mathrm{P}]. \tag{5.9}$$

Concentration of a species of interest, say [FP], is expressed as a fraction of the total concentration of all species in the medium/volume. The fraction is given by the ratio of the sum of all kinetic terms relating to the species of interest to the sum of all kinetic

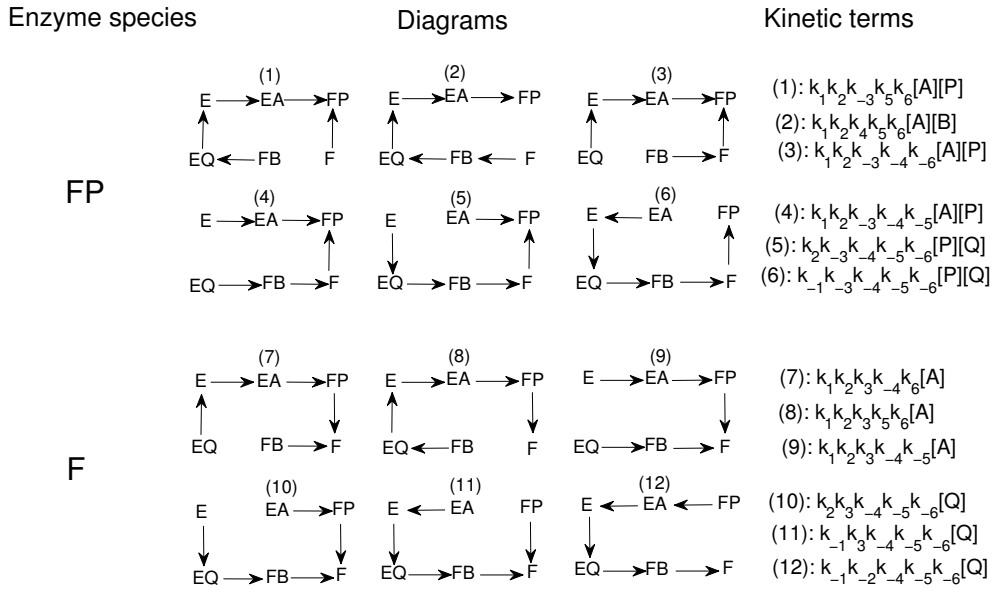


FIGURE 5.5: Possible patterns of species FP and F in the thiolase reaction and their corresponding kinetic expressions.

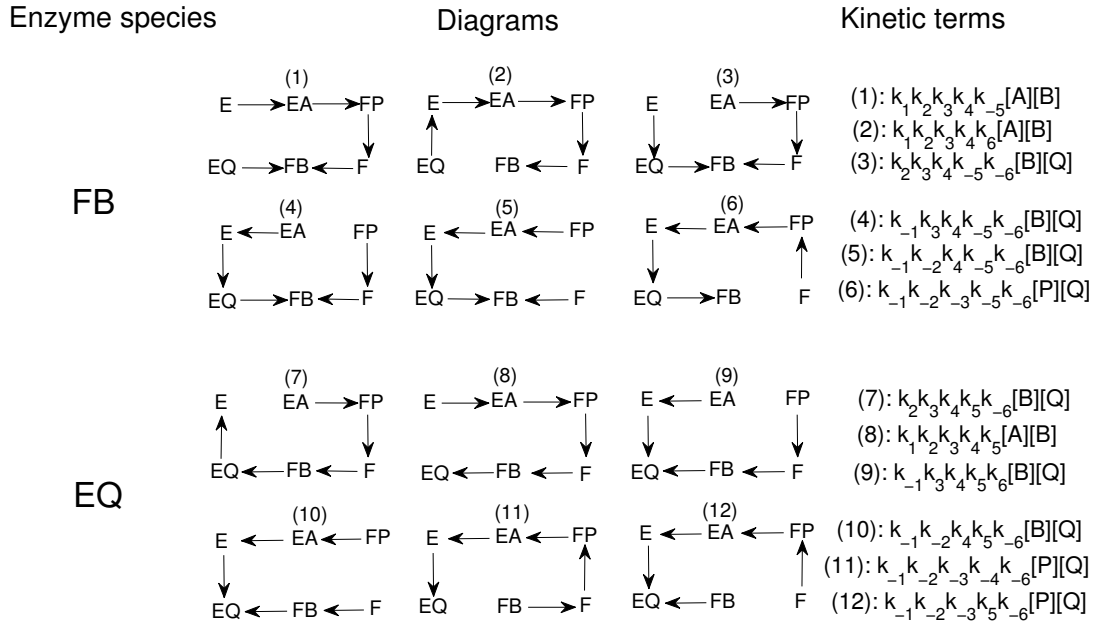


FIGURE 5.6: Possible patterns of species FP and F in the thiolase reaction and their corresponding kinetic expressions.

terms in the system.

$$[\mathrm{FP}] = \frac{\text{Kinetic terms of FP}}{\Sigma} \times [\mathrm{E}_0], \quad [\mathrm{F}] = \frac{\text{Kinetic terms of F}}{\Sigma} \times [\mathrm{E}_0]. \tag{5.10}$$

where $[\mathrm{E}_0]$ denotes the total concentration of species and Σ represents all kinetic terms in system. Further expanding the equation 5.10 by using kinetic terms of species FP

and F shown in Figure 5.5, we can rewrite equation 5.10 as

$$\begin{aligned}\frac{d[\mathrm{P}]}{dt} = \frac{[\mathrm{E_0}]}{\Sigma} \cdot (&k_1k_2k_{-3}k_3k_5k_6[\mathrm{A}][\mathrm{P}] + k_1k_2k_3k_4k_5k_6[\mathrm{A}][\mathrm{B}] \\ &+ k_1k_2k_{-3}k_3k_{-4}k_{-6}[\mathrm{A}][\mathrm{P}] + k_1k_2k_{-3}k_3k_{-4}k_{-5}[\mathrm{A}][\mathrm{P}] \\ &+ k_2k_{-3}k_3k_{-4}k_{-5}k_{-6}[\mathrm{P}][\mathrm{Q}] + k_{-1}k_{-3}k_3k_{-4}k_{-5}k_{-6}[\mathrm{P}][\mathrm{Q}]) \\ - \frac{[\mathrm{E_0}]}{\Sigma} \cdot (&k_1k_2k_{-3}k_3k_{-4}k_6[\mathrm{A}][\mathrm{P}] + k_1k_2k_{-3}k_3k_5k_6[\mathrm{A}][\mathrm{P}] \\ &+ k_1k_2k_{-3}k_3k_{-4}k_{-5}[\mathrm{A}][\mathrm{P}] + k_2k_{-3}k_3k_{-4}k_{-5}k_{-6}[\mathrm{P}][\mathrm{Q}] \\ &+ k_{-1}k_{-3}k_3k_{-4}k_{-5}k_{-6}[\mathrm{P}][\mathrm{Q}] + k_{-1}k_{-2}k_{-3}k_{-4}k_{-5}k_{-6}[\mathrm{P}][\mathrm{Q}]) \\ = \frac{[\mathrm{E_0}]}{\Sigma} \cdot (&k_1k_2k_3k_4k_5k_6[\mathrm{A}][\mathrm{B}] - k_{-1}k_{-2}k_{-3}k_{-4}k_{-5}k_{-6}[\mathrm{P}][\mathrm{Q}]). \end{aligned} \tag{5.11}$$

Notice that in the equation 5.11, only one positive and one negative terms are shown in the numerator. Intuitively, the positive term indicates reactions occurring in the forward direction, which is quantified by the concentration of products and their corresponding rate constants. The negative term indicates the reaction being taken place in reverse, calculated by multiplying product concentrations and all reverse rate constants. These two terms expression is always true in for all substrate-product mechanisms, except for *random* kinetics.

For deriving the final rate equation, we further define $num_1 = k_1k_2k_3k_4k_5k_6[\mathrm{E_0}]$ and $num_2 = k_{-1}k_{-2}k_{-3}k_{-4}k_{-5}k_{-6}[\mathrm{E_0}]$. Using this variable transformation, the rate of production can be rewritten as

$$\frac{d[\mathrm{P}]}{dt} = \frac{num_1 \cdot [\mathrm{A}][\mathrm{B}] - num_2 \cdot [\mathrm{P}][\mathrm{Q}]}{\Sigma}, \tag{5.12}$$

where the complete expression of Σ is given as

$$\begin{aligned}\Sigma = &k_{-1}k_3k_4k_5k_6[\mathrm{B}] + k_2k_3k_4k_5k_6[\mathrm{B}] + k_{-1}k_{-2}k_4k_5k_6[\mathrm{B}] \\ &+ k_{-1}k_{-2}k_{-3}k_5k_6[\mathrm{P}] + k_{-1}k_{-2}k_{-3}k_{-4}k_6[\mathrm{P}] + k_{-1}k_{-2}k_{-3}k_{-4}k_{-5}[\mathrm{P}] \\ &+ k_1k_{-2}k_4k_5k_6[\mathrm{A}][\mathrm{B}] + k_1k_3k_4k_5k_6[\mathrm{A}][\mathrm{B}] + k_1k_{-2}k_{-3}k_5k_6[\mathrm{A}][\mathrm{P}] \\ &+ k_1k_{-2}k_{-3}k_{-4}k_6[\mathrm{A}][\mathrm{P}] + k_1k_{-2}k_{-3}k_{-4}k_{-5}[\mathrm{A}][\mathrm{P}] + k_{-2}k_{-3}k_{-4}k_{-5}k_{-6}[\mathrm{P}][\mathrm{Q}] \\ &+ k_1k_2k_{-3}k_5k_6[\mathrm{A}][\mathrm{P}] + k_1k_2k_4k_5k_6[\mathrm{A}][\mathrm{B}] + k_1k_2k_{-3}k_{-4}k_6[\mathrm{A}][\mathrm{P}] \\ &+ k_1k_2k_{-3}k_{-4}k_{-5}[\mathrm{A}][\mathrm{P}] + k_2k_{-3}k_{-4}k_{-5}k_{-6}[\mathrm{P}][\mathrm{Q}] + k_{-1}k_{-3}k_{-4}k_{-5}k_{-6}[\mathrm{P}][\mathrm{Q}] \\ &+ k_1k_2k_3k_{-4}k_6[\mathrm{A}] + k_1k_2k_3k_5k_6[\mathrm{A}] + k_1k_2k_3k_{-4}k_{-5}[\mathrm{A}] \\ &+ k_2k_3k_{-4}k_{-5}k_{-6}[\mathrm{Q}] + k_{-1}k_3k_{-4}k_{-5}k_{-6}[\mathrm{Q}] + k_{-1}k_{-2}k_{-4}k_{-5}k_{-6}[\mathrm{Q}] \\ &+ k_1k_2k_3k_4k_{-5}[\mathrm{A}][\mathrm{B}] + k_1k_2k_3k_4k_6[\mathrm{A}][\mathrm{B}] + k_2k_3k_4k_{-5}k_{-6}[\mathrm{B}][\mathrm{Q}] \\ &+ k_{-1}k_3k_4k_{-5}k_{-6}[\mathrm{B}][\mathrm{Q}] + k_{-1}k_{-2}k_4k_{-5}k_{-6}[\mathrm{B}[\mathrm{Q}] + k_{-1}k_{-2}k_{-3}k_{-5}k_{-6}[\mathrm{P}][\mathrm{Q}] \\ &+ k_2k_3k_4k_5k_{-6}[\mathrm{B}][\mathrm{Q}] + k_1k_2k_3k_4k_5[\mathrm{A}][\mathrm{B}] + k_{-1}k_3k_4k_5k_6[\mathrm{B}][\mathrm{Q}] \\ &+ k_{-1}k_{-2}k_4k_5k_{-6}[\mathrm{B}][\mathrm{Q}] + k_{-1}k_{-2}k_{-3}k_{-4}k_{-6}[\mathrm{P}][\mathrm{Q}] + k_{-1}k_{-2}k_{-3}k_5k_{-6}[\mathrm{P}][\mathrm{Q}]. \end{aligned} \tag{5.13}$$

Rewriting the terms in Σ by grouping kinetic terms relating to similar reactants give in the expression, we have Σ as

$$\begin{aligned}\Sigma = &(coef\mathrm{A})[\mathrm{A}] + (coef\mathrm{B})[\mathrm{B}] + (coef\mathrm{P})[\mathrm{P}] + (coef\mathrm{Q})[\mathrm{Q}] \\ &+ (coef\mathrm{AB})[\mathrm{AB}] + (coef\mathrm{PQ})[\mathrm{PQ}] + (coef\mathrm{AP})[\mathrm{AP}] + (coef\mathrm{BQ})[\mathrm{BQ}],\end{aligned} \tag{5.14}$$

in terms of new coefficients

$$\begin{aligned}coef\mathrm{A} &= k_1k_2k_3(k_{-4}k_6 + k_5k_6 + k_{-4}k_{-5}) \\ coef\mathrm{B} &= k_4k_5k_6(k_{-1}k_3 + k_2k_3 + k_{-1}k_{-2}) \\ coef\mathrm{P} &= k_{-1}k_{-2}k_{-3}(k_5k_6 + k_{-4}k_6 + k_{-4}k_{-5}) \\ coef\mathrm{Q} &= k_{-4}k_{-5}k_{-6}(k_2k_3 + k_{-1}k_3 + k_{-1}k_{-2}) \\ coef\mathrm{AB} &= k_1k_4(k_{-2}k_5k_6 + k_3k_5k_6 + k_2k_5k_6 \\ &\qquad + k_2k_3k_{-5} + k_2k_3k_6 + k_2k_3k_5) \\ coef\mathrm{PQ} &= k_{-3}k_{-6}(k_{-2}k_{-4}k_{-5} + k_2k_{-4}k_{-5} + k_{-1}k_{-4}k_{-5} \\ &\qquad + k_{-1}k_{-2}k_{-5} + k_{-1}k_{-2}k_{-4} + k_{-1}k_{-2}k_5) \\ coef\mathrm{AP} &= k_1k_{-3}(k_{-2}k_5k_6 + k_{-2}k_{-4}k_6 + k_{-2}k_{-4}k_{-5} \\ &\qquad + k_2k_5k_6 + k_2k_{-4}k_6 + k_2k_{-4}k_{-5}) \\ coef\mathrm{BQ} &= k_4k_{-6}(k_2k_3k_{-5} + k_{-1}k_3k_{-5} + k_{-1}k_{-2}k_{-5} \\ &\qquad + k_2k_3k_{-5} + k_{-1}k_3k_5 + k_{-1}k_{-2}k_5).\end{aligned} \tag{5.15}$$

Substituting equation 5.14 into equation 5.12, the rate reaction of product is specified as

$$\frac{d[\mathrm{P}]}{dt} = \frac{num_1 \cdot [\mathrm{A}][\mathrm{B}] - num_2 \cdot [\mathrm{P}][\mathrm{Q}]}{\begin{aligned}\{&(coef\mathrm{A})[\mathrm{A}] + (coef\mathrm{B})[\mathrm{B}] + (coef\mathrm{P})[\mathrm{P}] + (coef\mathrm{Q})[\mathrm{Q}] \\ &+ (coef\mathrm{AB})[\mathrm{AB}] + (coef\mathrm{PQ})[\mathrm{PQ}] + (coef\mathrm{AP})[\mathrm{AP}] + (coef\mathrm{BQ})[\mathrm{BQ}]\}\end{aligned}}. \tag{5.16}$$

Specifically, following the suggestion from Roberts (1977), the maximum velocity for a particular direction can be defined as a fraction, in which the numerator is the specified direction of the underlying reaction and the denominator is the coefficient for species involved in the director. In this case, the maximum velocities of the forward V_1 and backward direction V_2, when the substrates A and B or products P and Q are saturating, are given as

$$V_1 = \frac{num_1}{coef\mathrm{AB}}, \quad V_2 = \frac{num_2}{coef\mathrm{PQ}}, \quad k_{eq} = \frac{num_1}{num_2}, \tag{5.17}$$

where k_{eq} indicates the rate constant when the system is in the equilibrium.

By multiplying the rate equation by $\frac{num_2}{coef\mathrm{AB} \cdot coef\mathrm{PQ}}$, the numerator of equation 5.16 is

transformed as

$$\begin{aligned}\text{numerator} &= \frac{num_1 \cdot [\text{A}][\text{B}] \cdot num_2}{coef\text{AB} \cdot coef\text{PQ}} - \frac{num_2 \cdot num_2 \cdot [\text{P}][\text{Q}]}{coef\text{AB} \cdot coef\text{PQ}} \\ &= \frac{num_1 \cdot [\text{A}][\text{B}] \cdot num_2}{coef\text{AB} \cdot coef\text{PQ}} - \frac{num_1 \cdot num_2 \cdot [\text{P}][\text{Q}]}{k_{eq} \cdot coef\text{AB} \cdot coef\text{PQ}} \\ &= V_1V_2 \cdot [\text{A}][\text{B}] - (V_1V_2 \cdot [\text{P}][\text{Q}])/k_{eq}. \end{aligned} \tag{5.18}$$

And the denominator is given as

$$\begin{aligned}\text{denominator} = &\frac{coef\text{A} \cdot [\text{A}] \cdot num_2}{coef\text{AB} \cdot coef\text{PQ}} + \frac{coef\text{B} \cdot [\text{B}] \cdot num_2}{coef\text{AB} \cdot coef\text{PQ}} \\ &+ \frac{coef\text{P} \cdot [\text{P}] \cdot num_2}{coef\text{AB} \cdot coef\text{PQ}} + \frac{coef\text{Q} \cdot [\text{Q}] \cdot num_2}{coef\text{AB} \cdot coef\text{PQ}} \\ &+ \frac{coef\text{AB} \cdot [\text{AB}] \cdot num_2}{coef\text{AB} \cdot coef\text{PQ}} + \frac{coef\text{PQ} \cdot [\text{PQ}] \cdot num_2}{coef\text{AB} \cdot coef\text{PQ}} \\ &+ \frac{coef\text{AP} \cdot [\text{AP}] \cdot num_2}{coef\text{AB} \cdot coef\text{PQ}} + \frac{coef\text{BQ} \cdot [\text{BQ}] \cdot num_2}{coef\text{AB} \cdot coef\text{PQ}} \end{aligned} \tag{5.19}$$

At this point, we further consider the Michaelis constant for representing the rate constant. In particular, the Michaelis constant, being the most widely adopted notation in computational modeling, is more like a complex expression rather than an actual kinetic description. In this denotation scheme, the rate constant of a reactant is defined as the ratio of the relevant coefficients. Moreover, the Michaelis constant is always named after the remaining letter after other terms are canceled out from the numerator and denominator. For the denominator shown as equation 5.19, we have

$$K_b = \frac{coef\text{A}}{coef\text{AB}}, \quad K_a = \frac{coef\text{B}}{coef\text{AB}}, \quad K_q = \frac{coef\text{P}}{coef\text{PQ}}, \quad K_p = \frac{coef\text{Q}}{coef\text{PQ}} \tag{5.20}$$

Particularly, in enzyme-catalyzed reaction, a few enzymes may act as inhibitors so that the equilibrium of the system can be achieved. In this case, inhibitors influence the substrate A and product P, causing the reaction of these two reactants to occur in reverse direction (Davis et al., 1987). Consequently, the following transformations can be written down

$$\begin{aligned}\frac{coef\text{AP}}{coef\text{PQ}} &= \frac{coef\text{AP}}{coef\text{P}} \cdot \frac{coef\text{P}}{coef\text{PQ}} = \frac{1}{\frac{coef\text{P}}{coef\text{AP}}} \cdot K_q = K_{ia} \cdot K_q \\ \frac{coef\text{BQ}}{coef\text{AB}} &= \frac{coef\text{BQ}}{coef\text{B}} \cdot \frac{coef\text{B}}{coef\text{AB}} = \frac{1}{\frac{coef\text{B}}{coef\text{BQ}}} \cdot K_a = K_{iq} \cdot K_a \end{aligned} \tag{5.21}$$

Understanding of these inhibition constants is quite straightforward. It is due to the letters of denominator are in different directions, and constant can be seen as the inhibitory rate for reactant which is named by the remaining letter in denominator after canceling out from numerator. For example, the letters A and P in $coef\text{AP}$ are in different directions, and P in denominator $coef\text{AP}$ is canceled by the numerator $coef\text{P}$.

Consequently, the remaining is A and the fraction $\frac{coefP}{coefAP}$ is therefore defined as the inhibition rate of substrate A.

Substituting equation 5.20 and 5.21 into the denominator of the rate equation, we have

$$\begin{aligned}
\text{denominator} = & \frac{coefA \cdot [A] \cdot num_2}{coefAB \cdot coefPQ} + \frac{coefB \cdot [B] \cdot num_2}{coefAB \cdot coefPQ} \\
& + \frac{coefP \cdot [P] \cdot num_2}{coefAB \cdot coefPQ} + \frac{coefQ \cdot [Q] \cdot num_2}{coefAB \cdot coefPQ} \\
& + \frac{coefAB \cdot [AB] \cdot num_2}{coefAB \cdot coefPQ} + \frac{coefPQ \cdot [PQ] \cdot num_2}{coefAB \cdot coefPQ} \\
& + \frac{coefAP \cdot [AP] \cdot num_2}{coefAB \cdot coefPQ} + \frac{coefBQ \cdot [BQ] \cdot num_2}{coefAB \cdot coefPQ} \\
= & V_2 \cdot [A] \cdot K_b + V_2 \cdot K_a \cdot [B] \\
& + \frac{V_1 \cdot [P] \cdot K_q}{K_{eq}} + \frac{V_1 \cdot [Q] \cdot K_p}{K_{eq}} \\
& + V_2 \cdot [A] \cdot [B] + \frac{V_1 \cdot [P] \cdot [Q]}{K_{eq}} \\
& + \frac{V_1 \cdot [A] \cdot [P] \cdot K_q}{K_{ia} K_{eq}} + \frac{V_2 \cdot [B] \cdot [Q] \cdot K_a}{K_{iq}}.
\end{aligned} \tag{5.22}$$

Consequently, we can formulate the rate reaction for producing AcAcCoA as

$$\begin{aligned}
v_{\text{thiolase}} = & \frac{V_1 V_2 \cdot [A][B] - (V_1 V_2 \cdot [P][Q])/k_{eq}}{\{V_2 \cdot [A] \cdot K_b + V_2 \cdot K_a \cdot [B] + \frac{V_1 \cdot [P] \cdot K_q}{K_{eq}} + \frac{V_1 \cdot [Q] \cdot K_p}{K_{eq}} + V_2 \cdot [A] \cdot [B]} \\
& + \frac{V_1 \cdot [P] \cdot [Q]}{K_{eq}} + \frac{V_1 \cdot [A] \cdot [P] \cdot K_q}{K_{ia} K_{eq}} + \frac{V_2 \cdot [B] \cdot [Q] \cdot K_a}{K_{iq}}\}
\end{aligned} \tag{5.23}$$

where A and B denote the concentration of acetyl-CoA, P is the first product of this sub-pathway and represents the concentration of acetoacetyl-CoA, and Q is the second product and indicates the concentration of released CoA.

Following the suggestion of Haywood et al. (1988) that the reaction of synthesizing 3HBCoA is a *sequential ordered* Bi-Bi mechanism, we can derive the equation of rate reaction of 3HBCoA by using a similar procedure, which is given as

$$\begin{aligned}
v_{\text{reductase}} = & \frac{V_1 V_2 \cdot [A][B] - (V_1 V_2 \cdot [P][Q])/k_{eq}}{\{V_2 \cdot K_{ia} \cdot K_b + V_2 \cdot K_b \cdot [A] + V_2 \cdot [B] \cdot K_a + V_2 \cdot [A] \cdot [B] + \frac{V_1 \cdot [P] \cdot K_q}{K_{eq}}} \\
& + \frac{V_1 \cdot K_p \cdot [Q]}{K_{eq}} + \frac{V_1 \cdot [P] \cdot [Q]}{K_{eq}}\}
\end{aligned} \tag{5.24}$$

where A is the first substrate of this reaction and denotes the concentration of acetoacetyl-CoA, B is the second substrate which is the concentration of externally supplied NADPH. P and Q represent the concentrations of 3HBCoA and $NADP^+$, respectively.

Gerngross et al. (1994); Wodzinska et al. (1996) claimed that the synthase enzyme remains covalently linked to the polymer chain when it grows and that therefore the

product macromolecule is more likely to be insoluble in an aquatic environment. As a result, the PHB is produced solely by diffusing from 3HBCoA. Exploiting this truth, Leaf and Srienc (1997) claimed that the rate reaction for producing PHB can be seen as a simple irreversible Michaelis-Menten kinetics, which is given as

$$v_{\text{synthase}} = \frac{V_1 \cdot [\text{3HBCoA}]}{K_m + [\text{3HBCoA}]}, \tag{5.25}$$

where V_1 is the maximum velocity towards the PHB production and K_m is the concentration of PHB when the reaction rate is half of the V_1.

5.2 Quantitative analysis

Wang and Lee (1997); Wong et al. (1999) claimed that PHB production in *Alcaligenes eutrophus* starts from a rapid cellular growth with few PHB being synthesized, and finally achieves an accumulative phase in which the cellular growth becomes low and velocity of PHB production hits maximum. Unfortunately, the substantial uncertainties of parameters in this transition limit the experimental exploration of the process with respect to these parameters. Quantitative descriptions offer an opportunity to thoroughly explore the dynamics.

5.2.1 Model output

In the previous chapters, we have studied several biological systems such as the repressilator system acting the periodic behaviors along its self-regulation and the heat shock response system achieving the transient status after the transition caused by heat shock. When the values of parameters in these models are assumed to be known, subsequently, the outputs of systems are solely governed by the initial conditions. This PHB pathway, however, is completely different from all aforementioned systems, since its outputs are additionally influenced by the externally fed species, *i.e.* NADPH. In order to illustrate how the PHB production is affected in response to the concentration of NADPH, we carried out the simulations by considering the infinitely and limited external species supply, *i.e.* setting the initial concentration of NADPH to 200 and 2, respectively. The time interval for synthesizing the outputs is set to 400 hours, where the typical time length in real life is approximately 50 hours. Following the literature (Leaf and Srienc, 1997), the initial concentration of AcCoA is set to 200, while other species are zero. Results of this simulation is shown in Figure 5.7, as given in the graph, the concentration of PHB rapidly grows in the initial phases, then the curve starts to drop after achieving the maximum. This decline is caused by the dilution. Moreover, the peak of PHB production with limited NADPH is reached earlier than the one with abundant supply. This is expected because the intermediate species hydroxybutyrate, being the substrate

of producing PHB, is synthesized by consuming NADPH. If NADPH is insufficiently provided, the hydroxybutyrate is stopped synthesizing due to the shortage of NADPH, subsequently, its maximum volume is early reached.

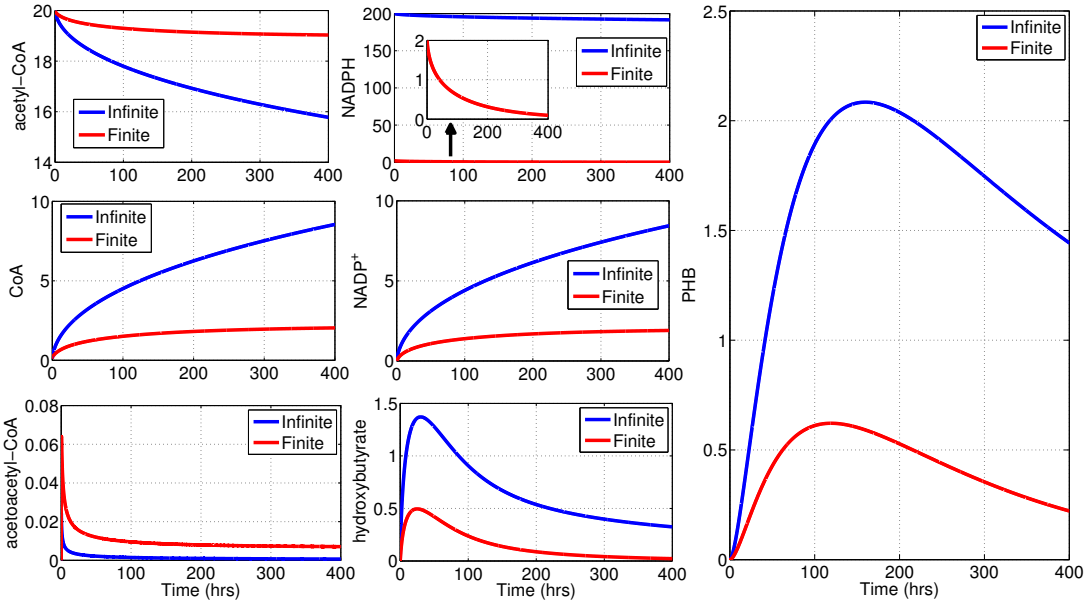


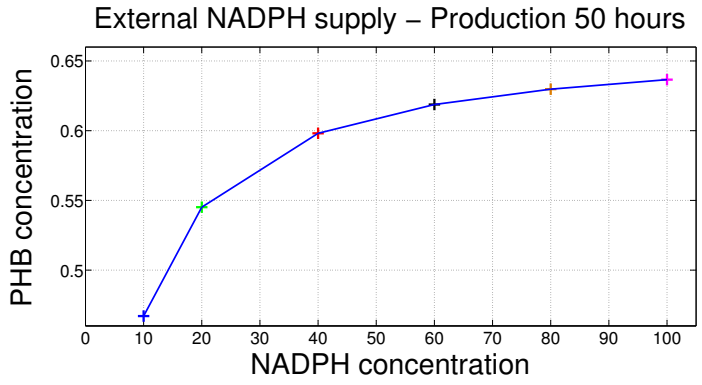
FIGURE 5.7: Concentrations of species in PHB pathway, which are synthesized by solving ODEs with consideration to the infinite or limited external supply (NADPH).

In addition, we further assess how the PHB production changes in response to different levels of the externally supplied NADPH. The study is performed by comparing the maximum volume of PHB production that is produced by supplying NADPH with low, moderate and abundant levels. The results are shown in Figure 5.8(a), which are similar to the results shown in Figure 5.7 where the abundant NADPH supply is generally advantageous for stimulating the production of PHB. However, the rate of increase of maximum volume slows down with the extra feeding of NADP, meaning that a steady-state is achieved. We also consider the combinatory influence of the initial substrate AcCoA and NADPH on PHB production, as shown in Figure 5.8(b), the amount of production only appears evidently different when the supply of external input is small. Through abundant feeding, more substrate can be converted to the end-product.

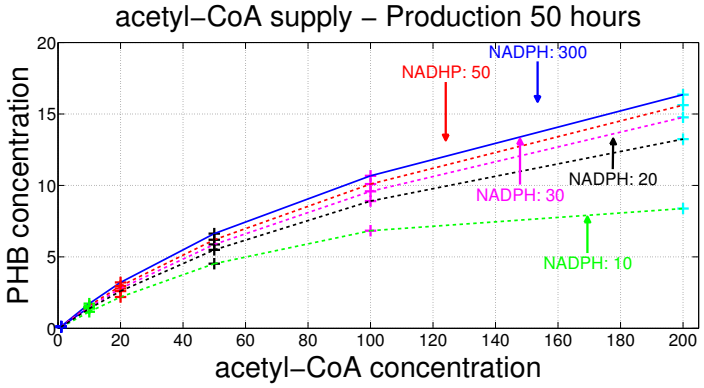
5.2.2 Sensitivity analysis

In chapter 4, we have claimed that our proposed method is ideal for applying to the parameter estimation problem with high dimensions, therefore, this PHB system is employed to test the effectiveness of our inference method by assuming all parameters are unknown.

In order to run the inference method, the sensitivity of parameters needs to be quantified, similarly, the eFAST plays the role in analyzing the sensitivity. For applying the eFAST



(a) PHB production with levels of NADPH



(b) PHB production with levels of acetyl-CoA and NADPH

FIGURE 5.8: (a): Maximum volumes of PHB concentrations generated by using various levels of NADPH supply. (b): Maximum volumes of PHB concentrations generated by feeding various initial substrate AcCoA and NADPH.

approach, we set the numbers of search curves (N_r) and samples picked from each curve (N_{se}) to 5 and 2049, respectively. Results are shown in Figure 5.9. It appears that production is predominantly governed by three maximum velocity parameters: $V_{1,\text{thiolase}}$, $V_{1,\text{reductase}}$ and $V_{1,\text{synthase}}$. We note that the concentration of hydroxybutyrate is directly inhibited by the parameter $K_{\text{ia,reductase}}$. This inhibitory reaction occurring in the step immediately proceeding the final output of the pathway is why this system is highly sensitive to this parameter. The significance of parameter $K_{\text{ia,reductase}}$ can be further verified from the outputs of AcAcCoA, where a sharp decline caused by the inhibition occurs after reaching its maximum volume, in comparison to hydroxybutyrate which has a smooth decline.

Moreover, a simulation is dedicated to visualize the effects of stiff and sloppy parameters on PHB production. In this example, we consider parameters $V_{1,\text{thiolase}}$ and $K_{\text{p,reductase}}$, as $V_{1,\text{thiolase}}$ is the most stiff parameter in dynamics but its value from estimation/literature is only 0.005, and the proportion of $K_{\text{p,reductase}}$ occupied in sensitivity pie chart is less than 2%, but its value is determined as 16.6. Figure 5.10 shows the synthetic concentration of PHB generated by using the parameter values with various multipli-

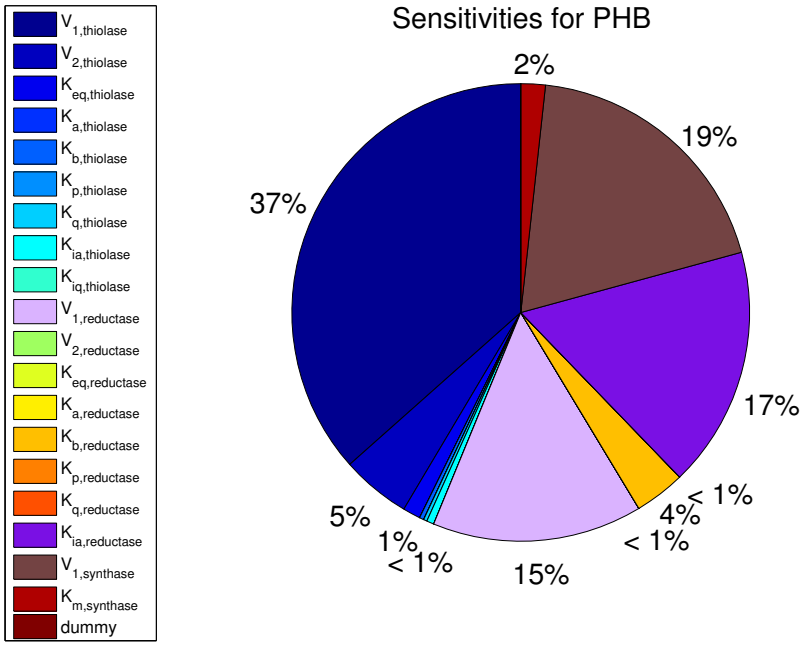


FIGURE 5.9: Result of sensitivity analysis for PHB production pathway.

ers. Apparently, even though the real value of $K_{\text{p,reductase}}$ is 3500 times greater than

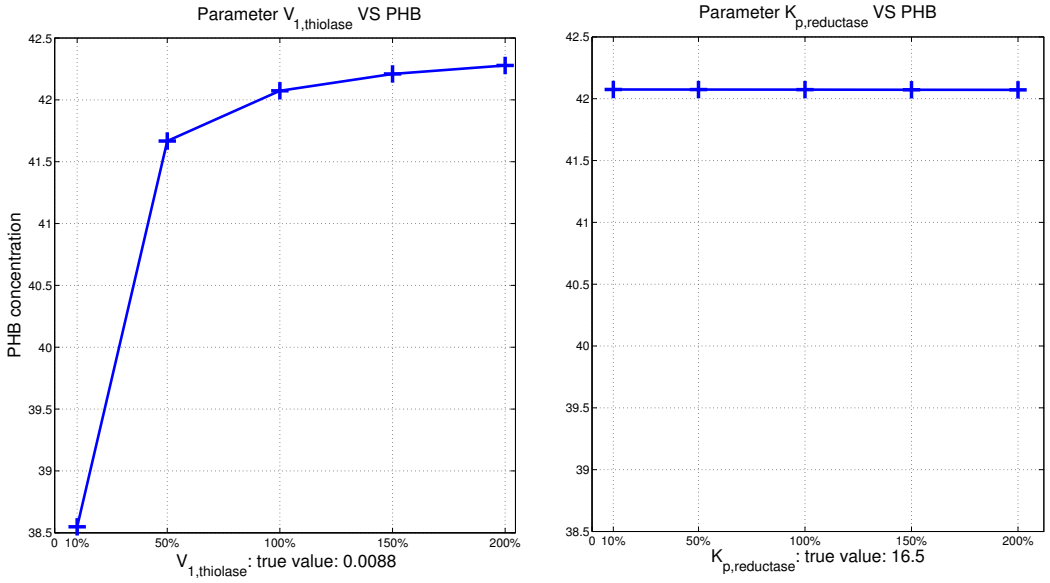
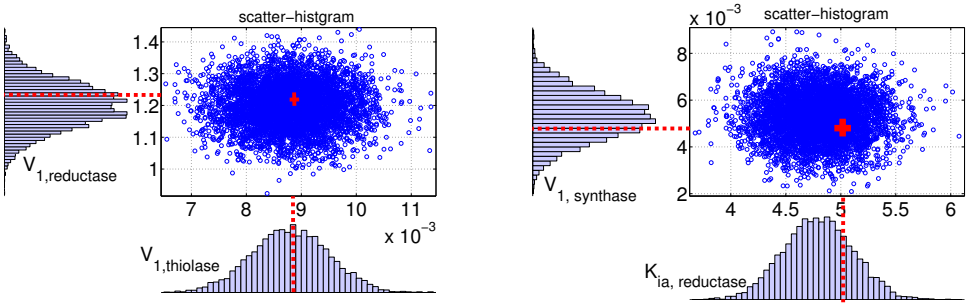


FIGURE 5.10: Maximum concentration of PHB synthesized by using various values of parameter $V_{1,\text{thiolase}}$ and parameter $K_{\text{p,reductase}}$. Options of value changing are 10%, 50%, 100%, 150% and 200%. In simulation, $V_{1,\text{thiolase}} = 0.005$ and $K_{\text{p,reductase}} = 16.5$.

$V_{1,\text{thiolase}}$, the behavioral response to changes of $K_{\text{p,reductase}}$ is negligible, whereas the influence of $V_{1,\text{thiolase}}$ is noticeable.

5.2.3 Parameter estimation

With this insight into the contribution of parameters to the system, we first determine the values of sloppy parameters by a coarse search. In the first run, since the considered all unknown case might be the hardest parameter estimation problem, particles for the sloppy parameters are therefore generated from the informative prior. For instance, the parameter $K_{\text{p,thiolase}}$ that contributes less than 1% to the system outputs and whose value is previously set to 31.4, then the initial particles of this parameter are generated from the distribution given as $\boldsymbol{\theta}_0 \sim \mathcal{U}(29, 32)$. We generate the synthetic data for 25 `hrs` (90000 `min`), with a regular sampling interval of 0.2 `min`, and system output is represented by 450000 sample points, where the coarse and tight tolerances are set to 1,500 and 700 respectively. The additive noise corrupting to observations is generated following a zero mean Gaussian distribution, where the diagonal elements of covariance matrix $\boldsymbol{R}$ is set to variance of solutions of ODEs, *i.e.* $\mathbf{X}$ multiplying a constant. In this simulation, this constant is defined as 0.05. The results of parameter estimation are



(a) $V_{1,\text{thiolase}}$-$V_{1,\text{reductase}}$ (b) $V_{1,\text{synthase}}$-$K_{\text{ia,reductase}}$

FIGURE 5.11: Parameter estimation of four stiff parameters $V_{1,\text{thiolase}}$, $V_{1,\text{reductase}}$, $V_{1,\text{synthase}}$ and $K_{ia,\text{reductase}}$. Results are represented in scatter-histogram graph. The lines in the histograms denote the true values considered in the literature (Leaf and Srienc, 1997). Red crosses are the corresponding points of the true values in the scatter graphs.

given in Figure 5.11, from which the bias estimation is somewhat observed, for instance, the true values of parameters $V_{1,\text{synthase}}$ and $K_{\text{ia,reductase}}$ are adopted as 0.0088 and 5 for generating the observations (van Wegen et al., 2001), and an offset of the mean of the posterior distribution is evident. The discrepancy between the inferences and the adopted values may be caused by the difficulty of all parameters assumed unknown leading to low precision of the estimation for sloppy parameters. However, we also note that the parameter $K_{\text{ia,reductase}}$ can be accurately estimated when it is left as the only unknown parameter of the system.

5.3 Glycolysis pathway

As we stated in complete pathway of PHB shown in Figure 5.1, the original species glucose need to be converted to pyruvate in prior to involve in the PHB sub-pathway. The reactions occurring to form pyruvate from glucose is called glycolysis, for which the schematic graph is described in Figure 5.12.

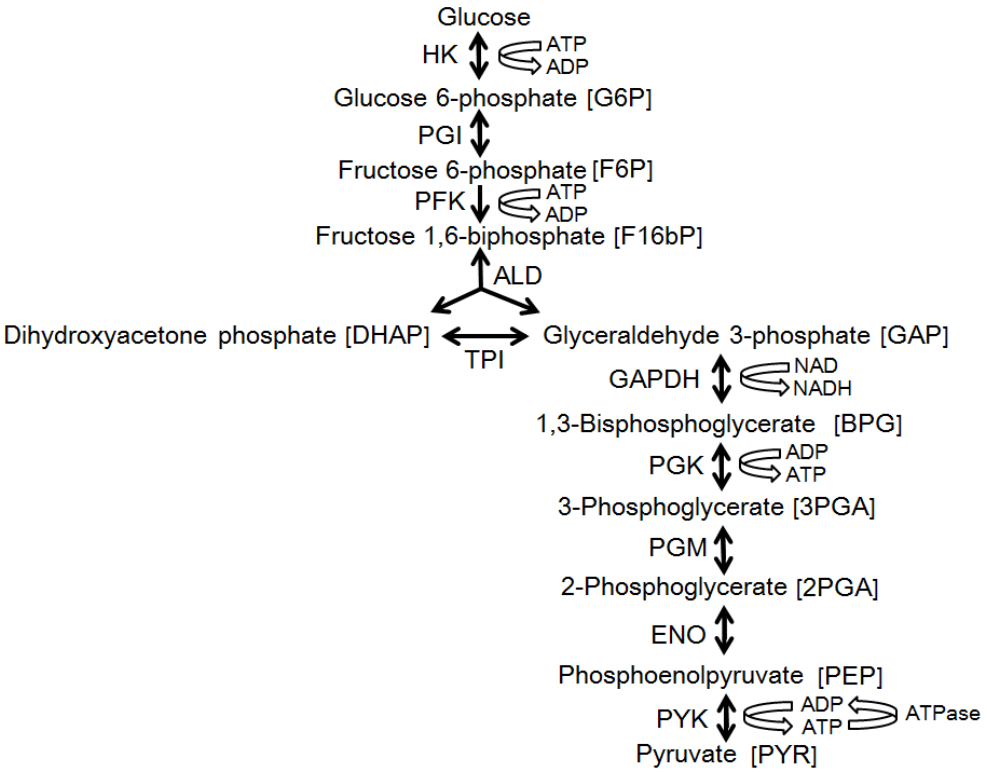


FIGURE 5.12: Schematic graph of the glycolysis pathway. The one way arrow indicates the reaction happens irreversibly, while the two ways arrow implies that the underlying reaction is reversible. The complete name of enzymes shown in graph are HK: hexokinase; PGI: phosphoglucose isomerase; PFK: phosphofructo-kinase-1; ALD: aldolase; TPI: yriose phosphate isomerase; GAPDH: glyceraldehyde 3-phosphate dehydrogenase; PGK: phosphoglycerate kinase; PGM: phosphoglycero-mutase; ENO: enolase; PYK: pyruvate kinase.

The glycolysis is a very general enzymatic pathway, which decomposes the glucose to prepare the specific substrate (in PHB example, Acetyl-CoA is such substrate) for particular reactions via extended systems. Most of reactions occurring in glycolysis are modeled in a similar way for formulating the PHB sub-pathway. Referring to Teusink et al. (2000); Hynne et al. (2001), with the King-Altman method introduced in section 5.1.1, the reactions associated with enzymes PGI and ENO can be easily modeled as the uni-uni Michaelis-Menten kinetics, given as

$$v = V_1 \frac{\frac{[\mathrm{A}]}{K_a}\left(1 - \frac{[\mathrm{P}]}{[\mathrm{A}]K_{eq}}\right)}{1 + \frac{[\mathrm{A}]}{K_a} + \frac{[\mathrm{P}]}{K_p}}, \tag{5.26}$$

while the reactions involve enzymes such as HK, PGK, GAPDH and PYK can be formulated as the bi-bi Michaelis-Menten kinetics without inhibitor, given as

$$v = V_1 \frac{\frac{[\mathrm{A}][\mathrm{B}]}{K_a K_b}\left(1 - \frac{[\mathrm{P}][\mathrm{Q}]}{[\mathrm{A}][\mathrm{B}]K_{eq}}\right)}{\left(1 + \frac{[\mathrm{A}]}{K_a} + \frac{[\mathrm{P}]}{K_p}\right)\left(1 + \frac{[\mathrm{B}]}{K_b} + \frac{[\mathrm{Q}]}{K_q}\right)}. \tag{5.27}$$

Noticing that there is a cyclic interaction occurring among species F16bP, DHAP and GAP, this looped regulatory mechanism causes a serious difficulty in modeling. Recalling the King-Altman method that models biochemical reaction by drawing all possible geometric patterns that form end-product, this complex cyclic interaction might result in explosion of potential pattern numbers. Thus Teusink et al. (2000) suggest that several interactions among intermediate species (e.g. fructose 2,6-bisphosphate), enzyme (aldolase) and energy container (AMP) are assumed to be constant, rather than time-varying. In addition, the effort is also made on regarding complex reactions as a rapid equilibrium, that is using the linear rate function to model sophisticated mechanisms. Following the work (Teusink et al., 2000; Hynne et al., 2001), the rate equation is given as

$$v = V_1 \frac{g_{\mathrm{R}} \lambda_1 \lambda_2 R}{R^2 + LT^2} \tag{5.28}$$

with

$$\lambda_1 = [\mathrm{F6P}]/K_{\mathrm{R,F6P}} \tag{5.29}$$

$$\lambda_2 = [\mathrm{ATP}]/K_{\mathrm{R,ATP}} \tag{5.30}$$

$$R = 1 + \lambda_1 \lambda_2 + g_{\mathrm{R}} \lambda_1 \lambda_2 \tag{5.31}$$

$$T = 1 + c_{\mathrm{ATP}} \lambda_2 \tag{5.32}$$

and

$$\begin{aligned} L = & L_0 \left(\frac{1 + C_{i,\mathrm{ATP}}[\mathrm{ATP}]/K_{\mathrm{ATP}}}{1 + [\mathrm{ATP}]/K_{\mathrm{ATP}}}\right)^2 \left(\frac{1 + C_{i,\mathrm{AMP}}[\mathrm{AMP}]/K_{\mathrm{AMP}}}{1 + [\mathrm{AMP}]/K_{\mathrm{AMP}}}\right)^2 \\ & \left(\frac{1 + C_{i,\mathrm{F26bP}}[\mathrm{F26bP}]/K_{\mathrm{F26bP}} + C_{i,\mathrm{F16bP}}[\mathrm{F16bP}]/K_{\mathrm{F16bP}}}{1 + [\mathrm{F26bP}]/K_{\mathrm{F26bP}} + [\mathrm{F16bP}]/K_{\mathrm{F16bP}}}\right)^2. \end{aligned} \tag{5.33}$$

By using the value of parameters given in the original literature coupled with our PHB sub-pathway, we simulate the concentration of pyruvate that is shown in Figure 5.13. From the graph, we can see that the change of pyruvate concentration becomes transient after a rapid increase in the beginning half hour. Such behavioral response obviously violates our expectation, where it should exist oscillation. The disagreement could be caused by the different organisms adopted for performing experiment, where the glycolysis was carried out in *Saccharomyces cerevisiae* and the PHB was measured in *Bacillus*.

One of the solutions is to estimate the value of parameters in glycolysis by our real

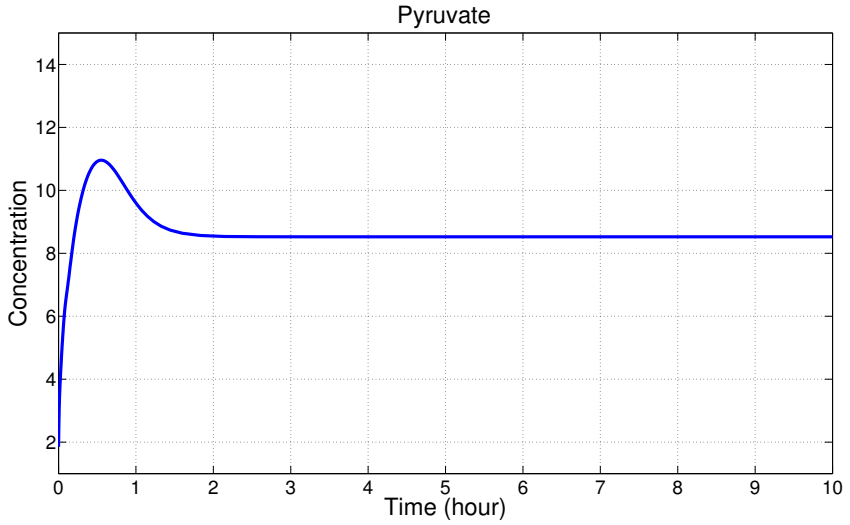


FIGURE 5.13: Simulation of pyruvate obtained by using the model and parameter values given in Teusink et al. (2000).

experimental glucose-PHB data. However, it is due to the curse of dimensionality (the number of parameters in glycolysis is 54 and the overall number of this complete PHB pathway will be 81), a decent inference of parameters of interest is somehow impossible under such limited input-output data. We look forward to estimating parameters precisely after having more experimental data for the individual species in this PHB pathway.

5.4 Discussion

In this chapter, we formulated a biochemical pathway into a set of ordinary differential equations using well-established kinetic laws. Quantitative analysis including sensitivity analysis and parameter estimation was performed to verify the reliability of our model and conclusions were checked with previously work. Even though this illustrative pathway has already been studied, the presented exploration can be considered as a hands-on experience, so that another sophisticated half of this pathway (PHA) can be modeled more thoroughly.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This thesis has shown the application of several probabilistic methods for parameter estimation in computational models of Systems Biology. The parameter estimation algorithms are simulated and their performances, in terms of computational speed and accuracy of estimating the underlying unknown parameter for values, are explored thoroughly. We are able to derive several conclusions from this computational study.

When sequential inference algorithms (extended Kalman, unscented Kalman and particle filter) are applied to the joint estimate of state and parameters, the non-parametric method of particle filtering offers a distinct advantage over the other approximation based parametric methods. By systematically exploring the performance of these methods on the heat shock response model studied by previous authors, we were able to demonstrate reliable convergence to the true underlying parameter values for a range of combination of observed/unobserved states and known/unknown parameter values. In this work, a particular problem in approximated methods (EKF/UKF) is to find out optimal initial conditions. The results reported by the previous authors (e.g. Lillacci and Khammash (2010)) are reproducible only when the conditions are initialized close to their true values. The particle filter, however, due to the interacting nature of the parallel evaluation, often estimate the unknown correctly even when all the particles are set further away from the EKF/UKF initializations. This property makes the PF an attractive estimation approach requiring minimal knowledge of where the true value of the unknown parameter might lie.

We also demonstrate that one-pass algorithms are capable of very efficient computation, achieving the same quality of estimation as batch methods but at a significantly reduced computational expense. While this is not a major advantage with respect to present day data sets in systems biology, we speculate that sequential problems will naturally arise

in biological problems in the near future, for example in modeling at the single-cell level of series of cells passing through a micro fluidics channel.

The family of algorithms centered around approximate Bayesian computation (ABC) were reviewed theoretically and empirically in Chapter 3. This is a powerful methodology for inference and parameter estimation as reported by Toni et al. (2009) among others. Starting from the naive ABC algorithm, and reviewing through the popular methods in the literature, the abilities of approaches at enhancing performance in terms of dependence on initialization, accuracy and computational cost are shown by applying to either the biological models or the statistical models. In the last part of this review, by comparing parameter estimation methods on a particular biological model, an empirical guidance of choosing the algorithm for inference problems in systems biology is given.

The parameters in system could be simultaneously estimated with various uncertainties in their inferences, and this phenomena has been explored by Gutenkunst et al. (2007) who suggested that the set of parameters can be decomposed into having sloppy and stiff properties with respect to the system behavior. We proposed a two stage ABC based inference approach in the light of sensitivity of these parameters, by which the sloppy parameters are assigned values determined by a coarse analysis and the stiff parameters are re-estimated by using a harsh acceptance criterion. This selective allocation of computational budget allows us to achieve a decent balance between accuracy and efficiency. The effectiveness of this proposed method is illustrated by inferring the parameters in four biological models and successfully reproducing system behaviors.

In the final part of this research, described in Chapter 5, we have modeled the fundamentals of chemical reactions that polymerize glucose. This system is different from the previously considered systems such as the heat shock response model and the repressilator model, since the production of this polymer system is governed by the external fed species while others are self-regulatory systems (the heat shock system is not regulatory but it has no external input). This polymer system is captured by three differential equations and nineteen parameters, whose development closely follows the work of Leaf and Srienc (1997), while the novel contributions of our work are carried out by the quantitative analysis including sensitivity analysis and parameter identification. In this work, we assume all parameters are unknown which might be the possible hardest inference problem, and our proposed method described in Chapter 4 successfully recovers the true values of the stiff parameters.

6.2 Future work

As proposed in Chapter 4, the ABC method redistributes the computational budget with respect to the sensitivity of parameter. Unsurprisingly, it may encounter a latent state

which has no distinct sensitivity partition, in which case the selective computational scheme is impossible to provide. Moreover, it could be more reliable to quantitatively define the stiffness or sloppiness rather than observing it intuitively.

From an algorithmic point of view, a possible extension could be to derive advanced ABC methods that may give superior performance. Of particular interest is to use the Riemann manifold Hamiltonian Monte Carlo method in the ABC framework, in which geometric information can efficiently guide the movement of particles to the place thus affecting the less tolerance value. In addition, methods can benefit considerably from GPU computing. Since in the adaptive ABC method adopted in this work, particles are chosen to represent the posterior without necessarily being required to satisfy the acceptance condition and therefore can be distributively calculated over a large number of GPU computing units.

In this work, the modeled polymerization for synthesizing PHB contributes only half of the dual polymer-production pathway, whereas in reality another polymer, namely polyhydroxyalkanoates, is also simultaneously produced by the system. We will formulate the complete pathway in the future, since we are interested in finding a way of selectively producing the particular polymer by switching on/off the corresponding parameter-controlled intermediate species. We will also focus on validating the quantitative analysis of polymer pathway introduced in Chapter 5, including estimating the parameters in glycolysis pathway by using additional experimental data and examining the agreement between synthetic input-output data and the real data.

Appendix A

Inference algorithms

A.1 Sequential inference methods

Inference methods are generally categorized into sequential and batch, where the batch approaches update the inference of state/parameter by re-visiting the distant past data, and which are useful for the off-line problems. Conversely, the sequential approaches also known as the one-pass methods recursively estimate state/parameter by given the data whose arrival is sequential. This class of sequential methods is useful in the real-time applications such as robotics and system control. In the following section, several popular sequential inference methods will be discussed.

A.1.1 Kalman filter

The *Kalman filter* is the most well-known one-pass algorithm whose superior performance depends on the linearity of underlying system and assumption of Gaussian noise corrupting to observations (Kalman, 1960). This parametric approach has been widely used for tracking (Chan et al., 1979) and navigation (Loebis et al., 2004). Algorithm 8 describes the pseudo-code for the Kalman filter. Notation here is the extended state space which consists of the latent states and the unknown parameters, *i.e.* $\mathbf{s} = [x_1, \ldots, x_{\mathrm{n}}, \theta_1, \ldots, \theta_{D_p}]$. Outputs $\mathbf{s}_{t|t}$ and $\mathbf{\Sigma}_{t|t}$ are the estimates to state/parameter and their covariance at time t. The Kalman filter consists of two steps which are prediction and correction. In the prediction phase, the method proposes a temporal state using the underlying dynamical system and previous state. Corrections are made for state coupled with the new coming observations.

[1] In the step, the capital T is the transpose symbol.

Algorithm 8 Kalman filter

`Initialization`

1. Input $\mathbf{s}_{0|0}$, $\mathbf{\Sigma}_{0|0}$ and $\mathbf{y}_{1:T}$.

for $t = 1$ to T **do**

`Prediction`

2. $\mathbf{s}_{t|t-1} = \mathbf{A}\mathbf{s}_{t-1|t-1}$

3[1]. $\mathbf{\Sigma}_{t|t-1} = \mathbf{A}\mathbf{\Sigma}_{t-1|t-1}\mathbf{A}^{\mathrm{T}} + \mathbf{Q}$

`Correction`

4. $\mathbf{K}_t = \mathbf{\Sigma}_{t|t-1}\mathbf{C}^{\mathrm{T}}(\mathbf{C}\mathbf{\Sigma}_{t|t-1}\mathbf{C}^{\mathrm{T}} + \mathbf{R})^{-1}$

5. $\mathbf{s}_{t|t} = \mathbf{s}_{t|t-1} + \mathbf{K}_t(\mathbf{y}_t - \mathbf{C}\mathbf{s}_{t|t-1})$

6. $\mathbf{\Sigma}_{t|t} = \mathbf{\Sigma}_{t|t-1} - \mathbf{K}_t\mathbf{C}\mathbf{\Sigma}_{t|t-1}$

end for

7. Output $\mathbf{s}_{1:t}$.

A.1.2 Extended Kalman filter

While the kalman filter described in Algorithm 8 is the optimal estimator for linear with additive Gaussian noise, many real-world problems do not fall into this category. Non-linearity and non-Gaussian of noise are common in many practical problems such as the systems problems of interest to us. One way of addressing this issue is to use approximation. The *extended Kalman filter* (EKF) is an approach that uses Taylor series expansion of non-linearity and truncate to the first order terms (Jazwinski, 1970; Bar-Shalom et al., 2001). This linearization of functions f(·) and h(·) is always described as

$$\mathbf{F}_t = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{s}}\right|_{\mathbf{s}=\mathbf{s}_{t-1|t-1}}, \quad \mathbf{H}_t = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{s}}\right|_{\mathbf{s}=\mathbf{s}_{t|t-1}}. \tag{A.1}$$

Often truncating to the first order terms is considered good enough, similarly, EKF consists of the prediction and correction steps. The pseudo-code of EKF is given in Algorithm 9 and its derivation is given in section B.3 of Appendix B. An illustrative example is shown in **Example 2.7.**

Algorithm 9 extended Kalman filter

`Initialization`

1. Input $\mathbf{s}_{0|0}$, $\mathbf{\Sigma}_{0|0}$ and $\mathbf{y}_{1:T}$.

for $t = 1$ to T **do**

`Prediction`

2. $\mathbf{s}_{t|t-1} = \mathbf{f}(\mathbf{s}_{t-1|t-1})$

3[2]. $\mathbf{\Sigma}_{t|t-1} = \mathbf{F}_t\mathbf{\Sigma}_{t-1|t-1}\mathbf{F}_t^{\mathrm{T}} + \mathbf{Q}$

`Correction`

4. $\mathbf{K}_t = \mathbf{\Sigma}_{t|t-1}\mathbf{H}_t^{\mathrm{T}}(\mathbf{H}_t\mathbf{\Sigma}_{t|t-1}\mathbf{H}_t^{\mathrm{T}} + \mathbf{R})^{-1}$

5. $\mathbf{s}_{t|t} = \mathbf{s}_{t|t-1} + \mathbf{K}_t(\mathbf{y}_t - \mathbf{h}(\mathbf{s}_{t|t-1}))$

6. $\mathbf{\Sigma}_{t|t} = \mathbf{\Sigma}_{t|t-1} - \mathbf{K}_t\mathbf{H}_t\mathbf{\Sigma}_{t|t-1}$

end for

7. Output $\mathbf{s}_{1:T}$.

Example 2.7. *We consider the univariate non-stationary growth model shown in the Example 2.1 as the case to study the effectiveness of EKF to track nonlinear/non-Gaussian system (Arulampalam et al., 2002). In this example, the time length of data is set to 80 and state is initialized following* $x_0 \sim \mathcal{N}(0, 1)$*. More specifically, the variances of process and observation noises are used as* $\boldsymbol{\sigma}_w^2 = 10$ *and* $\boldsymbol{\sigma}_v^2 = 1$*, respectively. Figure A.1 demonstrates the performance of EKF on tracking the behaviors of dynamics. It is easy to observe a discrepancy between the inferred and true state transitions. It needs to mention that EKF*

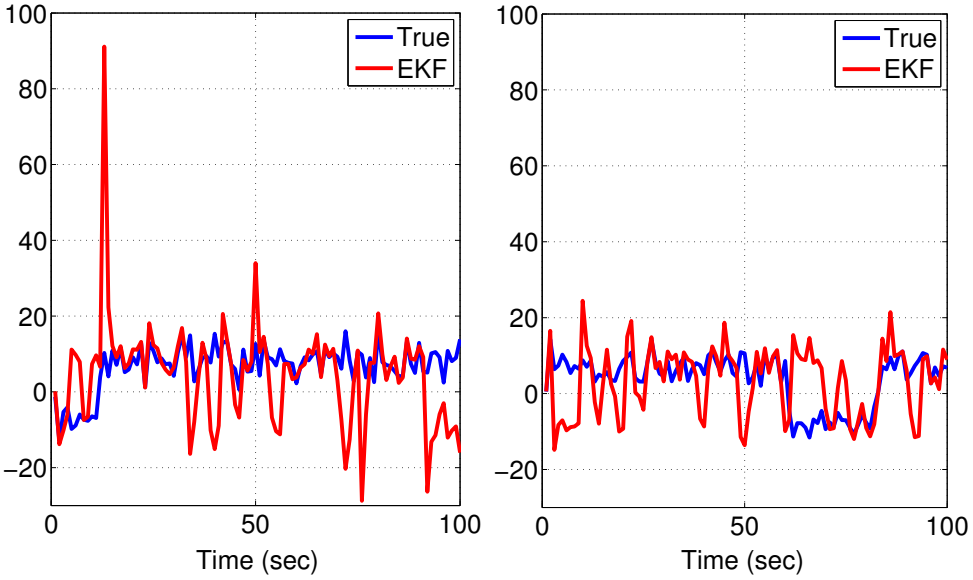


FIGURE A.1: An example of state estimation of the system shown in **Example 2.1** by EKF. In the graphs, the blue solid line is the true state obtained by direct synthesizing data from the system dynamics. The red lines show the inference of state from EKF only given the data of observations. Clearly, an abrupt climb can be seen in the left graph where which can be seen as the unreliable inference. In comparing to the left graph, a relatively precise inference is shown in the right graph. This is because sometimes the linearization neglects the higher-order terms, and in this case a single observation has disturbed tracking by the model. We ran this example ten times, in which the similar abrupt climb happened in five simulations.

A.1.3 Unscented Kalman filter

The distinct feature of EKF is to locally linearize the nonlinear dynamics so that they could be identified by using Kalman filter settings. However, as claimed by Julier et al. (1995), the performance of EKF heavily depends on the ability of the linear (first-order) or quadratic (second-order) linearization on capturing information of dynamics. EKF performs extremely poorly when a considerable amount of information is lost during the transformation. More practically, calculation of the Jacobian matrix is prone to human error and is difficult to debug.

[2]In the step, the capital T is the transpose symbol.

An alternative way of deriving an approximate filter, introduced by Julier et al. (1995), is known as the *unscented Kalman filter* (UKF). The idea here is, instead of approximating the function, we approximate the distribution over parameters by a set of samples. These samples are deterministically drawn for constructing the prior distribution, and then which are propagated through the function of system. These propagated points are known as sigma points, which in association with the new observation are utilized to update the mean and the covariance of the posterior distribution at each time instant.

Let the dimension of the extended state space is $n = D_s + D_p$, with the prior state $\mathbf{s}_{t|t-1}$ and covariance $\mathbf{\Sigma}_{t|t-1}$, and $2n+1$ sigma points $\boldsymbol{\chi}$ selected:

$$\boldsymbol{\chi}_t^0 = \mathbf{s}_{t|t-1}, \tag{A.2}$$

$$\boldsymbol{\chi}_t^i = \mathbf{s}_{t|t-1} + \sqrt{(n+\lambda)\mathbf{\Sigma}_{t|t-1}}, \quad i = 1, \ldots, n \tag{A.3}$$

$$\boldsymbol{\chi}_t^i = \mathbf{s}_{t|t-1} - \sqrt{(n+\lambda)\mathbf{\Sigma}_{t|t-1}}, \quad i = n+1, \ldots, 2n. \tag{A.4}$$

where the prior state $\mathbf{s}_{t|t-1}$ can be regarded as the reference point in *sigma points*, and $\sqrt{(n+\lambda)\mathbf{\Sigma}_{t|t-1}}$ is the regular interval of increment/decrement for other *sigma points*. An illustrative example for producing *sigma points* from 2-dimensional Gaussian random variable is shown in Figure A.2.

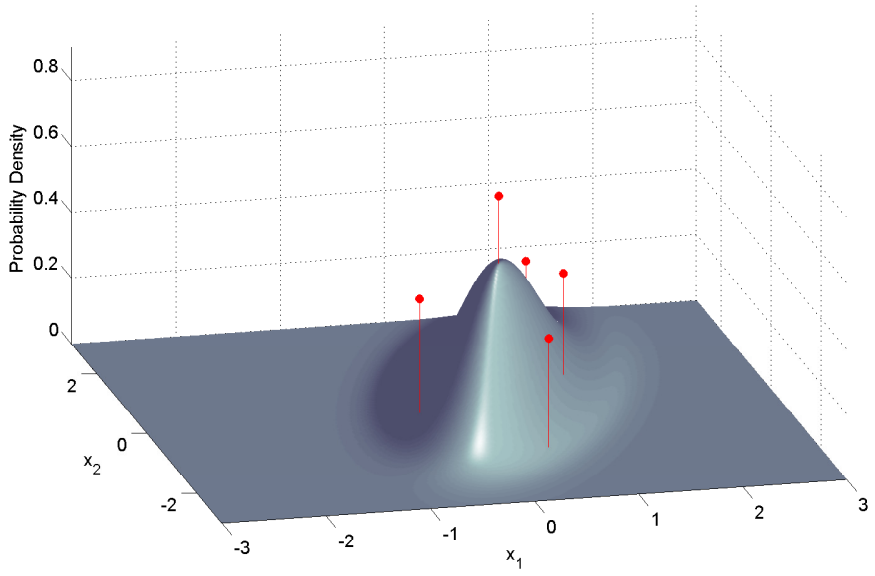


FIGURE A.2: Weighted *sigma points* for a 2-d Gaussian random variables. The second-order statistical information of the distribution is captured by those points.The weights of these *sigma points* are implied by the their heights.

A crucial concept in sampling based methods (UKF belongs to this category) is the weighting scheme, which measures how much the sample is identical to the true underlying state/parameter. The sample which has a high weight indicates that its value is

close to the true parameter/state. UKF determines the weights of sigma points by the following scheme

$$\boldsymbol{w}_{\boldsymbol{\chi}_t^i}^{\mathrm{m}} = \lambda/(n+\lambda),\ i=0 \tag{A.5}$$

$$\boldsymbol{w}_{\boldsymbol{\chi}_t^i}^{\mathrm{c}} = \lambda/(n+\lambda) + (1-\alpha^2+\beta),\ i=0, \tag{A.6}$$

$$\boldsymbol{w}_{\boldsymbol{\chi}_t^i} = 1/\left\{2(n+\lambda)\right\},\ i=1,\ldots,2n;, \tag{A.7}$$

where $\boldsymbol{w}_{\boldsymbol{\chi}_t^i}^{\mathrm{m}}$ and $\boldsymbol{w}_{\boldsymbol{\chi}_t^i}^{\mathrm{c}}$ are two weight vectors for updating the state $\mathbf{s}$ and the covariance matrix $\boldsymbol{\Sigma}$, respectively. They only differ in the first element of vector, while others are identical. λ is a scaling parameter, defined as $\lambda = \alpha^2(n+\kappa)-n$. Parameter α determines the spread of the sigma points in space, the value of which is suggested to be in the range $10^{-1} \leq \alpha \leq 1$ (Julier, 2002). Constant κ is a scaling factor usually set to $3-n$. β is a non-negative weighting factor and is often used to represent the higher order moments of the distribution being used to approximate the posterior (Julier, 2002). Empirically, β is set to 2 (Haykin, 2001).

The generic form of UKF is given as in Algorithm 10. Derivations of UKF are detailed in Van Der Merwe and Wan (2001).

Algorithm 10 Unscented Kalman filter

`Initialization`

1. Input $\mathbf{s}_{0|0}$, $\boldsymbol{\Sigma}_{0|0}$ and $\mathbf{y}_{1:T}$.
for $t = 1$ to T **do**

`Collection of sigma points`

2. Use $\mathbf{s}_{t|t-1}$ and $\boldsymbol{\Sigma}_{t|t-1}$ for picking sigma points $\boldsymbol{\chi}_{t-1}$, via equation (A.2) - (A.4).

`Prediction of states`

3. $\boldsymbol{\chi}_{t|t-1} = \mathbf{f}(\boldsymbol{\chi}_{t-1})$
4. $\mathbf{s}_{t|t-1} = \sum_{i=1}^{2n} \boldsymbol{w}_{\boldsymbol{\chi}}^{\mathrm{m}(i)} \boldsymbol{\chi}_{t|t-1}^{(i)}$
5. $\boldsymbol{\Sigma}_{t|t-1} = \sum_{i=1}^{2n} \boldsymbol{w}_{\boldsymbol{\chi}}^{\mathrm{c}(i)} [\boldsymbol{\chi}_{t|t-1}^{(i)} - \mathbf{s}_{t|t-1}][\boldsymbol{\chi}_{t|t-1}^{(i)} - \mathbf{s}_{t|t-1}]^{\mathrm{T}} + \mathbf{Q}$

`Prediction of observations`

6. $\hat{\boldsymbol{y}}_{t|t-1} = \mathbf{h}(\boldsymbol{\chi}_{t|t-1})$
7. $\mathbf{y}_{t|t-1} = \sum_{i=1}^{2n} \boldsymbol{w}_{\boldsymbol{\chi}}^{\mathrm{m}(i)} \hat{\boldsymbol{y}}_{t|t-1}^{(i)}$

`Correction`

8. $\boldsymbol{\Sigma}_{\mathbf{yy}} = \sum_{i=1}^{2n} \boldsymbol{w}_{\boldsymbol{\chi}}^{\mathrm{c}(i)} [\boldsymbol{y}_{t|t-1}^{(i)} - \hat{\boldsymbol{y}}_{t|t-1}][\boldsymbol{y}_{t|t-1}^{(i)} - \hat{\boldsymbol{y}}_{t|t-1}]^{\mathrm{T}} + \mathbf{R}$
9. $\boldsymbol{\Sigma}_{\mathbf{sy}} = \sum_{i=1}^{2n} \boldsymbol{w}_{\boldsymbol{\chi}}^{\mathrm{c}(i)} [\boldsymbol{\chi}_{t|t-1}^{(i)} - \mathbf{s}_{t|t-1}][\boldsymbol{y}_{t|t-1}^{(i)} - \hat{\boldsymbol{y}}_{t|t-1}]^{\mathrm{T}}$
10. $\mathbf{K}_t = \boldsymbol{\Sigma}_{\mathbf{sy}} \boldsymbol{\Sigma}_{\mathbf{yy}}^{-1}$
11. $\mathbf{s}_{t|t} = \mathbf{s}_{t|t-1} + \mathbf{K}_t(\mathbf{y}_t - \mathbf{y}_{t|t-1})$
12. $\boldsymbol{\Sigma}_{t|t} = \boldsymbol{\Sigma}_{t|t-1} - \mathbf{K}_t \boldsymbol{\Sigma}_{\mathbf{sy}}$
end for
13. Output $\mathbf{s}_{1:T}$.

Example 2.8 *We also employ the univariate non-stationary growth model to examine the performance of UKF on tracking state behavior. Using the same algorithmic settings and initialization as in the case of EKF case, we show results in Figure A.3. Even*

through UKF is incapable of exactly capturing the state behavior, it still outperforms the EKF in term of accuracy.

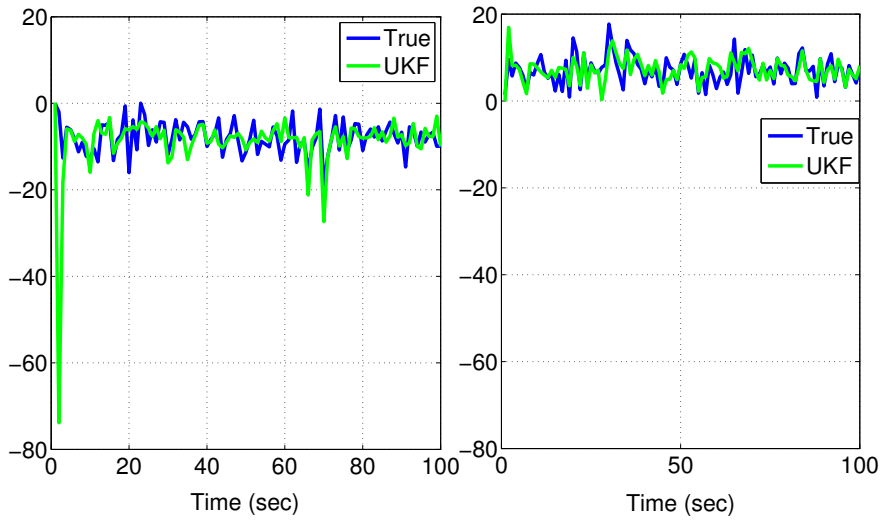


FIGURE A.3: State estimation of the system shown in **Example 2.1** by UKF. In the graph, the blue solid line is the true transition of system, while the green line shows the inference from UKF only given the observations. In comparison to the EKF, as shown in the right graph, UKF tracks the state behaviors better. The superior accuracy is a good reflection of the improvement from the second-order Gaussian approximation. In the simulation of this toy example, UKF usually produces precise inference. Likewise, we also ran this example ten times for UKF, and the failure as shown in the left graph will occasionally happen (3 of 10).

A.1.4 Particle filter

Being a derivative-free method, UKF is not restricted to differentiable dynamics. Compared to the EKF which provides an approximation to the first order accuracy, it is claimed that the UKF is capable of delivering at least second-order accuracy (Julier and Uhlmann, 1997; Van Der Merwe and Wan, 2001). Since the approximation in the UKF is made locally, the approach cannot guarantee that the solution obtained is the global optimum, and this problem is highly possible when the target system is complex.

Quite often, the *particle filter* (PF), also known as the *sequential Monte Carlo* (SMC) method (Liu and Chen, 1998; Doucet et al., 2000, 2001; Arulampalam et al., 2002), is adopted as an alternative to tackle complex inference problems. PF is a *Monte Carlo* approach that uses randomly drawn samples to represent the distribution of unknown state/parameter, instead of performing approximation by minimum number of samples (UKF) or requiring the closed form derivative (EKF).

The role of PF for state and parameter estimation has been studied from several fields over many decades. These include the adaptive estimation of neural networks (Kadirkamanathan and Niranjan, 1993), target tracking from bearing-only measurements (Bar-

Shalom et al., 2001), modeling futures contracts in computational finance (Niranjan, 1997) and to find global minima of artificial neural networks (de Freitas et al., 2000).

Algorithmically, a few versions of PFs have been developed as special cases of the fundamental PF which is also known as the general SIS algorithm and will be introduced below. For instance, Gordon et al. (1993) proposed the sequential importance resampling filter (SIR) as a variant of the basic PF and by which the degeneracy problem is partially addressed. In order to alleviate the influence of the proposal distribution chosen, Pitt and Shephard (1999) derived a novel PF, namely auxiliary sampling importance resampling algorithm, based on SIR associated with a proposal distribution that samples particles in pairs. In the following, we give the derivations of SIS and SIR.

Perfect Monte Carlo method

In *Monte Carlo* approach settings, the idea is that if one is able to randomly generate N independent and identically distributed (i.i.d) samples or particles $\{\mathbf{x}_t^{(i)}; i = 1, \ldots, N\}$ from the posterior distribution $p(\mathbf{x}_t|\mathbf{y}_{1:T})$, then the representation of the posterior is given as

$$p_N(\mathbf{x}_t|\mathbf{y}_{1:T}) = \frac{1}{N}\sum_{i=1}^{N}\delta(\mathbf{x}_t - \mathbf{x}_t^{(i)}), \tag{A.8}$$

where $\delta(\cdot)$ is the Dirac delta function, indicating if sample $\mathbf{x}_t^{(i)}$ is identical to state $\mathbf{x}_t$. Using of the Dirac delta function, the expectation of function $f(\mathbf{x})$ can be approximated as

$$I_N(f) = \int_{\mathbf{x}_t} f(\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:T})d\mathbf{x}_t \approx \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}_t^{(i)}) \tag{A.9}$$

This approximation tends to the real expectation if the law of large numbers is applied, *i.e.*

$$\widehat{I}_N(f) \overset{a.s}{\rightarrow} I_N(f), \text{ as } N \rightarrow +\infty, \tag{A.10}$$

where $\overset{a.s}{\rightarrow}$ denotes almost sure convergence. Moreover, if the real variance of $f(\mathbf{x}_{0:t})$ is less than positive infinity

$$\sigma_f^2 = \mathbb{E}_{p(\mathbf{x}_t|\mathbf{y}_{1:T})}[f^2(\mathbf{x}_t)] - \{\mathbb{E}_{p(\mathbf{x}_t|\mathbf{y}_{1:T})}[f(\mathbf{x}_t)]\}^2 < +\infty. \tag{A.11}$$

then, a central limit theorem holds

$$\sqrt{N}[I_N(f) - I(f)] \overset{N\rightarrow+\infty}{\Rightarrow} \mathcal{N}(0, \sigma_f^2), \tag{A.12}$$

where $\Rightarrow$ denotes convergence in distribution.

Expectation of the dynamics $I(f)$ can be easily obtained and the rate of convergence of the estimation is independent of the dimensionality of $\mathbf{x}_t$ (Doucet et al., 2001). However, in other deterministic numerical integration methods (such as hybrid extended Kalman filter (Lillacci and Khammash, 2010)), the rate of convergence will slow down due to the increase in dimension, a problem known as the *curse of dimensionality* (Bishop, 2006).

Since in general, directly sampling from the posterior distribution $p(\mathbf{x}_t|\mathbf{y}_{1:T})$ is impossible, this perfect *Monte Carlo* sampling is impractical for commonly encountered real-world problems. Consequently, *importance sampling* (IS) has been developed (Geweke, 1989), for which the conceptual idea is to sample particles from an arbitrary distribution, known as the *proposal distribution*, and using a weighted sum of these samples for inference, as derived below.

Importance Sampling Method

Introducing a proposal distribution $q(\mathbf{x}_t|\mathbf{y}_{1:T})$ from which we can conveniently draw samples, we can rewrite the first part of equation A.9 for the expectation $I_N(f)$ as[3],

$$I(f) = \int_{\mathbf{x}_t} f(\mathbf{x}_t) \frac{p(\mathbf{x}_t|\mathbf{y}_{1:T}) q(\mathbf{x}_t|\mathbf{y}_{1:T})}{q(\mathbf{x}_t|\mathbf{y}_{1:T})} d\mathbf{x}_t. \tag{A.13}$$

In this approach, the significance of a sample for representing posterior is quantified by its importance weight, which is defined as $w(\mathbf{x}_t) = \frac{p(\mathbf{x}_t|\mathbf{y}_{1:T})}{q(\mathbf{x}_t|\mathbf{y}_{1:T})}$. Substituting the expression of importance weight into equation A.13, the expectation can be rewritten as

$$I(f) = \int_{\mathbf{x}_t} f(\mathbf{x}_t) \boldsymbol{w}(\mathbf{x}_t) q(\mathbf{x}_t|\mathbf{y}_{1:T}) d\mathbf{x}_t. \tag{A.14}$$

Consequently, if one is able to sample N i.i.d particles from the proposal distribution $\{\mathbf{x}_t^{(i)} \sim q(\mathbf{x}_t|\mathbf{y}_{1:T});\ i = 1, \ldots, N\}$, then a Monte Carlo approximation can be made for $I_N(f)$; that is

$$\widehat{I}_N(f) = \frac{1}{N} \sum_{i=1}^{N} f(\mathbf{x}_{0:t}^{(i)}) w(\mathbf{x}_t^{(i)}) \tag{A.15}$$

By applying the law of large numbers, convergence $\widehat{I}_N(f) \overset{a.s}{\to} I_N(f)$ exists. Additionally, considering the central limit theorem, the convergence rate of $\widehat{I}_N(f)$ will not be influenced by the dimension increase.

Sequential Importance Sampling

Importance sampling, an introduced in the previous section is defined in batch mode, *i.e.* for all the data $1:T$ available together. When data needs to be processed sequentially, we can formulate sequential importance sampling as proposed by Robert and Casella

[3]For sake of interpretation, the proposal distribution is denoted as $q(\mathbf{x}_t|\mathbf{y}_{1:T})$, in general, it can be any arbitrary distribution.

(1999); Liu (2001).

The novelty of SIS is made by adopting the marginal distribution of the proposal distribution at time $t-1$, that facilitates the proposal distribution at time t, given as

$$q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = q(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t}). \tag{A.16}$$

Expanding $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$ according to Bayes's theorem, we have

$$\begin{aligned} p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) &= \frac{p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})}{\int p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})} \\ &\propto p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t}), \end{aligned} \tag{A.17}$$

where the integral at denominator $\int p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})$ is usually treated as a constant. Substitute equations (A.16) - (A.17) into the equation for importance weight

$$\begin{aligned} \boldsymbol{w}(\mathbf{x}_t) &= \frac{p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})}{q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})} \\ &\propto \frac{p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})}{q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t-1})q(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})}. \end{aligned} \tag{A.18}$$

Then, another transformation for $\boldsymbol{w}_{t-1}$ is performed

$$\begin{aligned} \frac{\boldsymbol{w}(\mathbf{x}_{t-1})}{p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})} &= \frac{p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})}{q(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})} \times \frac{1}{p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})} \\ &= \frac{p(\mathbf{y}_{1:t-1}|\mathbf{x}_{0:t-1})p(\mathbf{x}_{0:t-1})}{q(\mathbf{x}_{t-1}|\mathbf{x}_{0:t-2}, \mathbf{y}_{1:t-1})q(\mathbf{x}_{0:t-2}|\mathbf{y}_{1:t-2})} \times \frac{1}{p(\mathbf{y}_{1:t-1}|\mathbf{x}_{0:t-1})p(\mathbf{x}_{0:t-1})} \\ &= \frac{1}{q(\mathbf{x}_{t-1}|\mathbf{x}_{0:t-2}, \mathbf{y}_{1:t-1})q(\mathbf{x}_{0:t-2}|\boldsymbol{y}_{1:t-2})} \\ &= \frac{1}{q(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})}. \end{aligned} \tag{A.19}$$

Further transforming the calculation of importance weight by substituting equations (A.19) into equations (A.18)

$$\boldsymbol{w}(\mathbf{x}_t) = \boldsymbol{w}(\mathbf{x}_{t-1}) \frac{p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})}{q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t})p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})}. \tag{A.20}$$

Since the dynamics of interest is assumed to act as a Markov Chain, therefore, they obey two properties of Markov process

$$1^{st} \textbf{ property:} \qquad p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t}) = \prod_{j=1}^{t} p(\mathbf{y}_j|\mathbf{x}_j), \tag{A.21}$$

$$2^{nd} \textbf{ property:} \qquad p(\mathbf{x}_{0:t}) = \prod_{j=1}^{t} p(\mathbf{x}_j|\mathbf{x}_{j-1}). \tag{A.22}$$

Applying the properties to the equation (A.20), the importance weight calculation becomes

$$\begin{aligned}\boldsymbol{w}_t &= \boldsymbol{w}_{t-1}\frac{\prod_{j=1}^{t} p(\mathbf{y}_j|\mathbf{x}_j)p(\mathbf{x}_j|\mathbf{x}_{j-1})}{\prod_{j=1}^{t-1} p(\mathbf{y}_j|\mathbf{x}_j)p(\mathbf{x}_j|\mathbf{x}_{j-1})q(\mathbf{x}_t|\mathbf{x}_{0:t-1},\mathbf{y}_{1:t})} \\ &= \boldsymbol{w}_{t-1}\frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})}{q(\mathbf{x}_t|\mathbf{x}_{0:t-1},\mathbf{y}_{1:t})}. \end{aligned} \tag{A.23}$$

As the proposal distribution can be any distribution, for the sake of convenience, the state transition distribution $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ is generally used as the proposal distribution. Consequently the importance weight can be recursively calculated

$$\boldsymbol{w}_t \propto \boldsymbol{w}_{t-1}p(\mathbf{y}_t|\mathbf{x}_t). \tag{A.24}$$

Sequential Importance Resampling

SIS provides a recipe for estimating online, but, suffers from a serious deficiency due to the weights of particles become increasingly skewed over many iterations. As a result, only one particle is taken to represent the empirical distribution after a few iterations. This problem is usually known as degeneracy (Doucet et al., 2001; Arulampalam et al., 2002).

A solution to this problem of degeneracy in samples is to resample the population of samples periodically, *i.e.* duplicate samples with hight weights and kill off those with low weights. This idea, introduced by Kitagawa (1998), leads to the *Sequential Importance Resampling* (SIR) in which the current particle $\mathbf{x}^i$ is replaced by $\mathbf{x}^j$ and index j is determined when the cumulative density function of weights up to w^j is greater than a uniform distributed indicator. After each resampling step, the weights are reset to $1/N$ and the previous trajectory of importance weights is discarded. Consequently, the importance weight is further simplified as [4]

$$w_t^{(i)} = p(\mathbf{y}_t|\mathbf{x}_t^{(i)}). \tag{A.25}$$

SIR can be carried out by steps stated in Algorithm 11.

A graphical illustration of the weighting and resampling steps for SIR is shown in Figure A.4. Although the additional resampling step benefits the elimination of the degeneracy effect, it simultaneously results in the *sample impoverishment* problem, since the particles having high weights are repeatedly selected. Many particle filtering techniques have been developed to combat the degeneracy problem. Pitt and Shephard (1999) proposed the auxiliary particle filter and Musso et al. (2001) designed the regularised particle filters, both of which are attempts to deal with this problem.

[4] the procedure of transforming SIS to PF is given in section B.4 of Appendix B

Algorithm 11 Sequential importance resampling

1. `Initialization`, t = 0
for $i = 1, \ldots, N$ **do**
  sample $\boldsymbol{x}_0^i \sim p(\boldsymbol{x}_0)$ and set t = 1
end for
2. `Importance sampling`
for $i = 1, \ldots, N$ **do**
  sample $\tilde{\boldsymbol{x}}_t^i \sim p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^i)$ and set $\tilde{\boldsymbol{x}}_{0:t}^i = (\boldsymbol{x}_{0:t}^i, \tilde{\boldsymbol{x}}_t^i)$
end for
for $i = 1, \ldots, N$ **do**
  evaluate the importance weights $\tilde{\boldsymbol{w}}_t^i = p(\boldsymbol{y}_t|\tilde{\boldsymbol{x}}_t^i)$
end for
Normalize the importance weights
3. `Selection step`
Resample with replacement N particles $(\boldsymbol{x}_{0:t}^i; i = 1, \ldots, N)$ from the set $(\tilde{\boldsymbol{x}}_{0:t}^i; i = 1, \ldots, N)$ according to the importance weights
Set t $\rightarrow$ t+1 and go to step 2.

Example 2.9 *We also consider the univariate non-stationary growth model as an example for testing the ability of SIR for tackling nonlinearity/ non-Gaussian. In this case, the environmental settings are identical to those which are adopted in the EKF and UKF simulations. The results are shown in the Figure A.5, in which SIR illustrates a clear advantage over the two parametric Kalman filters.*

Next, we review several batch inference methods covering from the classical Expectation Maximization estimation method to the most recent Riemann manifold Hamiltonian Monte Carlo approach. The advantages and disadvantages of these batch inference methods will be briefly discussed and illustrated via some toy examples.

A.2 Batch inference methods

As opposed to the sequential inference approaches, the class of batch methods iteratively estimate parameter/state. In this category, all past observations need to be stored and revisited when a new observation appears.

A.2.1 Maximum-likelihood Estimation (MLE)

Maximum-likelihood estimator (MLE) can be traced all the way back to Edgeworth (1908), which finds the fittest value of parameters by maximizing the likelihood function (given the dynamical model $f(\cdot)$ and the observations $\mathbf{y}$). This ML algorithm can be mathematically described as

$$\widehat{\boldsymbol{\theta}}_{\mathrm{ML}} = \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}}{\operatorname{argmax}} \{p(\mathbf{y}|\boldsymbol{\theta})\}. \tag{A.26}$$

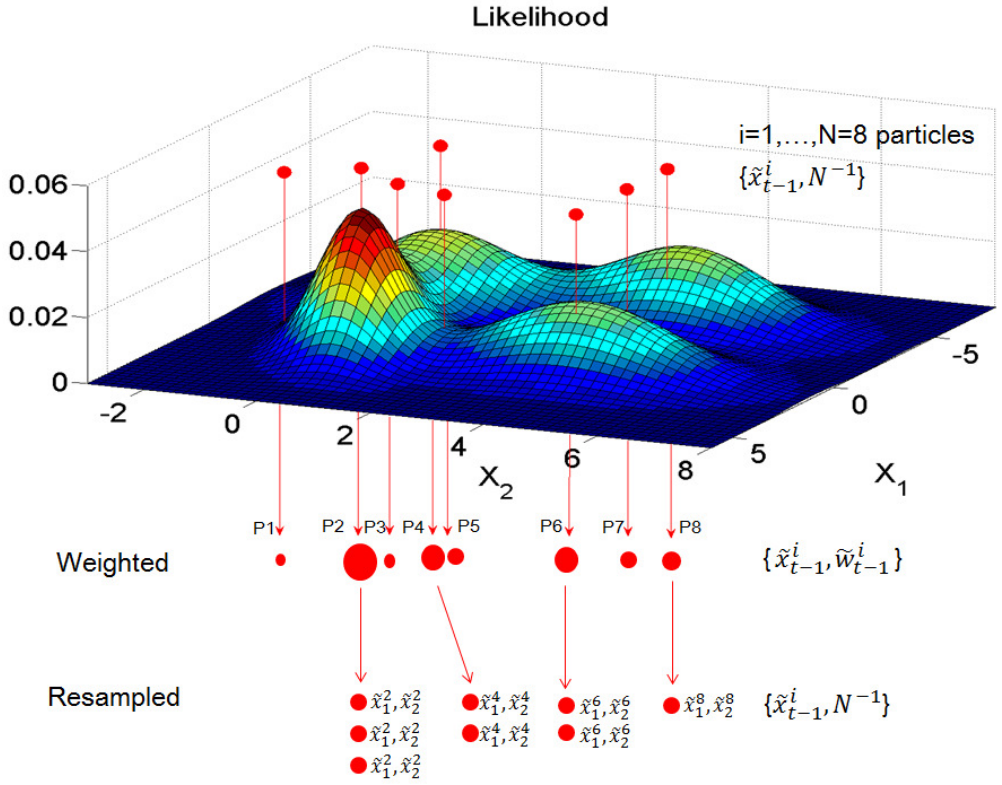


FIGURE A.4: In this illustrative example, SIR begins with the unweighted particle $\{\widetilde{\boldsymbol{x}}_{t-1}^i, N^{-1}\}$ at time instant $t-1$, which approximates distribution $p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-2})$. Particles are then calculated their corresponding weights by using likelihood $p(\boldsymbol{y}_{t-1}|\boldsymbol{x}_{t-1})$ at time $t-1$. This process returns a collection of weighted particles $\{\widetilde{\boldsymbol{x}}_{t-1}^i, \widetilde{\boldsymbol{w}}_{t-1}^i\}$, which provides the approximation of $p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1})$. The resampling step follows the weighting step, in which only the fittest particles will be picked. The weighted measures that negligibly contribute to posterior distribution are neglected at the resampling step. Particles obtained from weighting and resampling steps are both to approximate the posterior distribution $p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1})$.

MLE is often carried out with the natural logarithm of likelihood, denoted as $\ln(p(\mathbf{y}|\boldsymbol{\theta}))$, which provides the algebraic convenience as the advantage. Additionally, it can also avoid the overflow/underflow of calculation in some problems.

Hasenauer et al. (2010) derived a novel MLE-based approach to estimate the distributions of parameters in the heterogeneous cell population model. Effectiveness of the proposed method was assessed on the synthetic data from a model of TNF signal transduction. Weber et al. (2011) applied the MLE to infer the parameters in six differential equations that describe a regulatory mechanism at the trans-Golgi network in mammalian cells (Kramer and Radde, 2010). Andreychenko et al. (2012) employed the MLE to estimate the rate constants of a simple stochastic gene expression model including three reactions and a multi-attractor model which consists of three genes and four parameters. Schelker et al. (2012) improved MLE by first systematically refining the initial guess of states before employing MLE, and success of the proposed approach was tested by applying to JAK-STAT signaling pathway. Rodriguez-Fernandez et al. (2013)

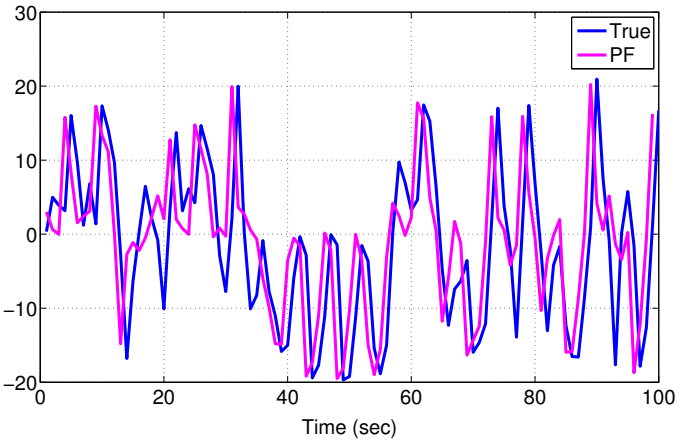


FIGURE A.5: State estimation of system shown in **Example 2.1** by SIR (PF). The blue solid line is the true system state, while the pink line shows the inference from SIR. It is clear that the performance of this non-parametric approach greatly outperforms the Kalman algorithms in terms of accuracy. We also denote that SIR with 1000 particles successfully infer the state dynamics in all ten simulations, while the percentages of EKF and UKF are 50% and 70%, respectively.

deployed MLE to accomplish the simultaneously parameter estimation and model discrimination for a chemical system, by which a regulation of the high affinity K^+ uptake system is characterized.

A.2.2 Expectation-Maximization method

ML is not feasible in certain circumstances, such as when the size of latent state is increases exponentially over time making calculation of the likelihood impossible or expensive, or the likelihood function no closed-form analytical solution. These difficulties motivate the need for a more powerful technique, which is called *expectation maximization* (EM) and maximizes equation A.26 via two steps: the (E)xpectation step and the (M)aximization step.

Since there is no concrete applications in this thesis which are related to EM algorithm, the details are omitted in this section. However, in order to tell a complete story of inference methods, EM algorithm is briefly introduced here. The procedure to carry out the EM algorithm and an illustrative Gaussian mixture model can be found in Appendix B.1.

Let us assume the system has a joint likelihood $p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$ where the $\mathbf{X}$ are observed variables and $\mathbf{Z}$ are the hidden variables, governed by parameter $\boldsymbol{\theta}$. The E-step is to calculate the expectation of joint log-likelihood w.r.t to the posterior distribution of the hidden variables given the old values of parameters $\boldsymbol{\theta}^{\text{old}}$. This expectation is denoted as

$\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}})$, can be mathematically described as

$$\begin{aligned}\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) &= \mathbb{E}_{p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\text{old}})}[\ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})], \\ &= \int_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\text{old}}) \ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) d\mathbf{Z}. \end{aligned} \tag{A.27}$$

Next, the goal of M-step is to maximize the $\mathcal{Q}(\cdot)$ in order to obtain updated parameters estimates

$$\boldsymbol{\theta}^{\text{new}} = \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}}{\text{argmax}}\ \mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}). \tag{A.28}$$

Technically, the way in which the EM algorithm estimate the parameters is by gradually approaching the maximum of the likelihood by increasing the lower bound of the posterior distribution over hidden variables. Details of the proof can be found in Bishop (2006).

Numerous applications in systems biology accomplish parameter estimation and structure identification by utilizing EM algorithm, and a few representative examples are introduced below. Berthoumieux et al. (2012) first assessed the reliability of EM algorithm on estimating parameters of Linlog models which are dedicated to formulate metabolism. EM algorithm was then applied to identify parameters for a model characterizing central carbon metabolism in the *Escherichia coli*. Daigle et al. (2012) developed a novel Monte Carlo EM algorithm which requires no prior knowledge about parameter values, and the authors illustrated the clear advantage of the proposed method over other typical inference approaches in terms of accuracy and computational efficiency, via comparative studies to four biological models from the systems biology literatures. Baldacchino et al. (2012) investigated the performance of the EM method on structure detection and parameter estimation of the nonlinear regression with exogenous inputs model, and empirically claimed that is is capable of yielding precise estimation only given noisy incomplete observations. Vavoulis et al. (2012) combined the EM algorithm with sequential Monte Carlo, in which the expectation of the joint log-likelihood of the latent state variables and the observed variables is instead approximated by a collection of particles. The capability of this EM-SMC algorithm was tested by estimating parameters in Hodgkin-Huxley-type models of single neurons. Liu et al. (2013) introduced a new dynamic framework for describing phenotypic formation in metabolic trait loci and carried out the model identification via EM algorithm.

Intuitively, a major drawback of EM is that the initial guess of the parameter should be appropriately chosen, otherwise, the algorithm may return the local maximum. In order to solve this difficulty, a few 'local' solutions (still within the EM framework), e.g. the algorithm introduced by Daigle et al. (2012), are adopted. Beyond this point, variational Bayesian methods are a means of overcoming this shortcoming of the EM algorithm.

A.3 Markov chain Monte Carlo methods

The popular methods for Bayesian inference when analytical solutions are not possible, or when it is difficult to directly sample from posterior densities of interest are *Markov chain Monte Carlo* (MCMC) method. MCMC methods consider an arbitrary distribution from which samples are randomly drawn, and based on the acceptance probability the samples are being used to estimate the distribution of unknown parameters. Several MCMC methods covering from the most basic *Metropolis-Hastings* (MH) to the advanced *Riemann Manifold Hamiltonian Monte Carlo* (RMHMC) will be introduced in the following sections.

A.3.1 Metropolis-Hastings algorithm

Among the MCMC methods, *Metropolis-Hastings* (MH) algorithm is the most straightforward and underpins the developments of other methods. This method was initially invented by Metropolis and Ulam (1949), and generalized by Hastings (1970).

In the MH algorithm, the collection of samples at the first time instant $\mathbf{x}_0$ is randomly drawn from the prior distribution. Subsequently, at each iteration, values of the samples are suggested empirically by the proposal distribution $q(\cdot)$. These candidate values of samples are accepted/rejected by an acceptance probability. Procedure of MH algorithm is described in Algorithm 12.

Algorithm 12 Metropolis-Hastings algorithm

1. Input $\mathbf{x}_0$, $q(\cdot)$ and MCMC iteration number I_{M}.
for $t = 1$ to I_{M} **do**
2. Draw $\mathbf{x}^* \sim q(\mathbf{x}^*|\mathbf{x}_{t-1})$
3. Draw indicator $u \sim \mathcal{U}(0,1)$
4. **if** $u < \min(1, \frac{p(\mathbf{x}^*)q(\mathbf{x}_{t-1}|\mathbf{x}^*)}{p(\mathbf{x}_{t-1})q(\mathbf{x}^*|\mathbf{x}_{t-1})})$ **then**
$\mathbf{x}_t = \mathbf{x}^*$
5. **else**
$\mathbf{x}_t = \mathbf{x}_{t-1}$
4. **end if**
end for
7. Output $\mathbf{x}_{1:I_{\mathrm{M}}}$.

Example 2.10 *Consider a mixture Gaussian model, which is mathematically described as*

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\mu_k, \sigma_{\mathrm{x}}^2), \tag{A.29}$$

If we consider a simple example for which $K = 2$, $\boldsymbol{\pi} = [0.3,\ 0.7]$, $\boldsymbol{\mu} = [0,\ 10]$ *and* $\boldsymbol{\sigma}_{\mathrm{x}}^2 = [2,\ 4]$. *of interest is to estimate the mean of this mixture Gaussian model,* 1000

data points are generated from equation A.29 by using the command `mvnrnd` of `MATLAB`. *The noisy observations are synthesized by corrupting the noise* $\mathbf{v}_t \sim \mathcal{N}(0, 0.01)$ *to the data points. The random walk scheme is adopted as the transition kernel of MH in this example, given as*

$$\mathbf{x}^* = q(\mathbf{x}_{t-1}) = \mathbf{x}_{t-1} + \boldsymbol{\omega} \sim \mathcal{N}(0, \sigma_k^2), \tag{A.30}$$

where σ_k^2 *is the user-chosen parameter effecting the distance that sample is moved in each transition. Four sets of algorithmic settings (MCMC iteration* I_M *and* σ_k^2*) are conducted for this toy example to introduce the abilities of the MH algorithm. As shown in Figure A.6, it is evident that the performance of MH is highly dependent on the MCMC iterations and the covariance of the transition kernel. That is, an inappropriate covariance of transition function leads to the chain failing to mix, and biased estimation may be made when not enough MCMC iterations have been performed.*

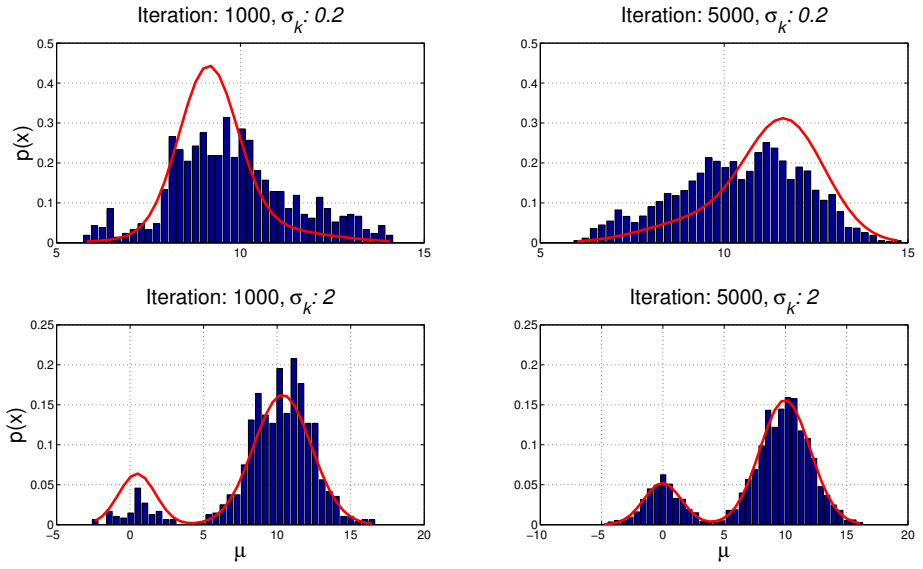


FIGURE A.6: Illustration of MH for the mixture Gaussian model with different algorithmic settings.

MH algorithm is a straightforward solution to discover the posterior. However, the major disadvantages of this approach is that samples can be highly correlated and carefully parametric tuning is needed in order to achieve a desirable acceptance rate, motivating the development of other advanced algorithms.

A.3.2 Gibbs sampling method

Gibbs sampling method, a special case of MH algorithm, enhances its capabilities by making use of a particular choice of $q(\mathbf{x}^*|\mathbf{x}^{t-1})$ (for sake of clear description, the time index is denoted as the superscript), originally introduced by Geman and Geman (1984);

Gelfand and Smith (1990). Consider the same distribution $p(\mathbf{x})$ with variables vector $\mathbf{x} = [x_1, \ldots, x_n]^{\mathrm{T}}$. In each step of the Gibbs sampling, the system state is updated component-wise, rather than as an entire vector. That is, the value of each variable is drawn from the distribution conditional on the values of the other variables, denoted as $x_i \sim p(x_i|\mathbf{x}_{-i})$. The subscript $-i$ is a compact way for describing the indices $[1, \ldots, i-1, i+1, \ldots, n]$. This sampling scheme is cyclically carried out through all the variables in the system.

Gibbs sampling is a special case of MH algorithm, since the acceptance probability of proposed moves is always set to one. Specifically, let us assume the i^{th} state at the previous time step is moved from $\mathbf{x}_i^{t-1}$ to $\mathbf{x}_i^*$ with the transition probability $q(\mathbf{x}_i^*|\mathbf{x}^{t-1}) = p(\mathbf{x}_i^*|\mathbf{x}_{-i}^{t-1})$, then the acceptance probability of $\mathbf{x}_i^*$ becomes

$$\begin{aligned} A(\mathbf{x}^*, \mathbf{x}^{t-1}) &= \frac{p(\mathbf{x}^*)q(\mathbf{x}^{t-1}|\mathbf{x}^*)}{p(\mathbf{x}^{t-1})q(\mathbf{x}^*|\mathbf{x}^{t-1})} \\ &= \frac{p(x_i^*, \mathbf{x}_{-i}^*)p(x_i^{t-1}|\mathbf{x}_{-i}^*)}{p(x_i^{t-1}, \mathbf{x}_{-i}^{t-1})p(x_i^*|\mathbf{x}_{-i}^{t-1})} \\ &= \frac{p(x_i^*|\mathbf{x}_{-i}^*)p(\mathbf{x}_{-i}^*)p(x_i^{t-1}|\mathbf{x}_{-i}^*)}{p(x_i^{t-1}|\mathbf{x}_{-i}^{t-1})p(\mathbf{x}_{-i}^{t-1})p(x_i^*|\mathbf{x}_{-i}^{t-1})}. \end{aligned} \tag{A.31}$$

If we take the previous sample $\mathbf{x}_{-i}^{t-1}$ as the value for perturbed sample $\mathbf{x}_{-i}^*$, we can further derive the acceptance probability as

$$A(\mathbf{x}^*, \mathbf{x}^{t-1}) = \frac{p(x_i^*|\mathbf{x}_{-i}^{t-1})p(\mathbf{x}_{-i}^{t-1})p(x_i^{t-1}|\mathbf{x}_{-i}^{t-1})}{p(x_i^{t-1}|\mathbf{x}_{-i}^{t-1})p(\mathbf{x}_{-i}^{t-1})p(x_i^*|\mathbf{x}_{-i}^{t-1})} = 1. \tag{A.32}$$

That is, each MH move is always accepted. Generally, Gibbs sampling can be described as Algorithm 13.

Algorithm 13 Gibbs sampling method

1. Input $\mathbf{x}_0$ and MCMC iteration number I_{M}.
for $t = 1$ to I_{M} **do**
2. Draw $x_1^t \sim p(x_1|x_2^{t-1}, x_3^{t-1}, \ldots, x_n^{t-1})$
3. Draw $x_2^t \sim p(x_2|x_1^t, x_3^{t-1}, \ldots, x_n^{t-1})$
$\vdots$
4. Draw $x_i^t \sim p(x_i|x_1^t, x_2^t, \ldots, x_{i-1}^t, x_{i+1}^{t-1}, \ldots, x_n^{t-1})$
$\vdots$
5. Draw $x_n^t \sim p(x_n|x_1^t, x_2^t, \ldots, x_{n-1}^t)$
end for
7. Output $\mathbf{x}_{1:I_{\mathrm{M}}}$.

Example 2.11 *Assume we have a linear system which is denoted as*

$$\mathbf{y} = \boldsymbol{\theta}\mathbf{X} + \boldsymbol{\omega}, \ \mathrm{X} \sim \mathcal{N}(0, 1), \tag{A.33}$$

where $\mathbf{X} \in \mathbb{R}^{2\times 100}$ *and* $\mathbf{X} \sim \mathcal{U}(0,1)$, $\boldsymbol{\theta} = [7,\ 1.8]$ *and* $\boldsymbol{\omega} \sim \mathcal{N}(0,1)$. *Of interest is to estimate* $\boldsymbol{\beta}$ *given the noisy observations* $\mathbf{y} \in \mathbb{R}^{1\times 100}$. *The MCMC iteration number* I_M *is set to 1500, the sample is initially generated from the uniform distribution* $\mathcal{U}(0,10)$ *and the proposal density for moving the sample is used as* $q(\boldsymbol{\theta}^*) = \boldsymbol{\theta}_{t-1} + \boldsymbol{v} \sim \mathcal{N}(0, \boldsymbol{\sigma}_k^2)$, *and* $\boldsymbol{\sigma}_k^2 = 0.2$. *Figure A.7 shows estimates to the posterior distribution of parameters* $p(\boldsymbol{\theta}|\mathbf{y})$, *for MH algorithm and Gibbs sampling algorithm, respectively. Both methods produce accurate estimations. The only difference is that MH algorithm converges to the target distribution along an arbitrary trajectory, whereas moves in Gibbs is always parallel to one of the axes.*

This example also shows that Gibbs sampling method always accepts each sample, compared to MH which accepts the sample 66% of the time in 1200 iterations. That is, more iterations are needed if MH is required to produce the same number of accepted samples as Gibbs sampling method.

It can easily be seen that Gibbs sampling algorithm alleviates the acceptance rate issue of MH algorithm, however, unsurprisingly, the dependence of local moves introduces the opportunity for Gibbs sampling to get stuck in a local mode and be unable to represent the posterior.

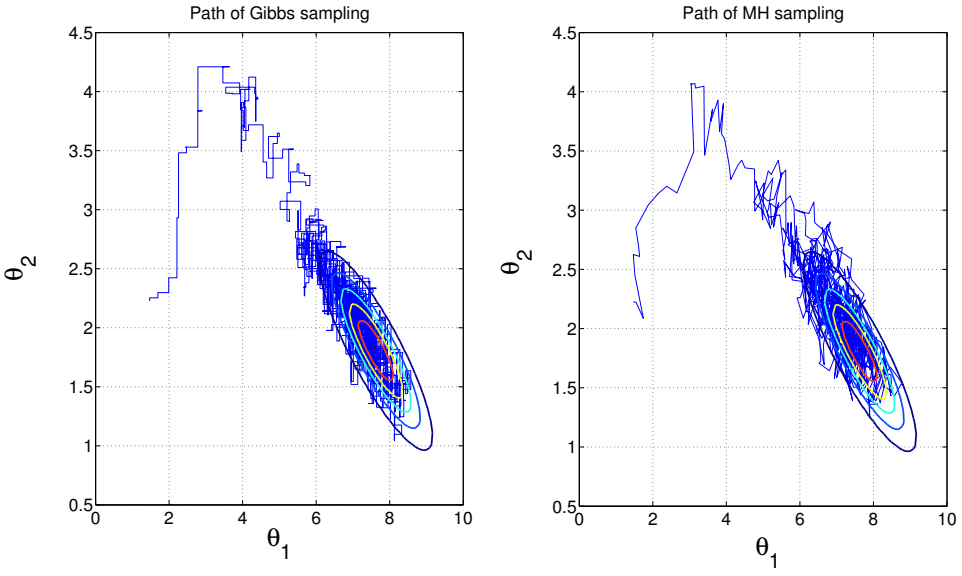


FIGURE A.7: Realizations of two chains constructed by Gibbs sampler (*left*) and MH (*right*) for estimating the parameters of a linear system. The likelihood $p(\mathbf{y}|\boldsymbol{\theta})$ is shown as the *ellipse* contours in graph.

Gibbs sampling has been applied ti a range if real-world statistical problems (Casella and George, 1992). For instance, Zeger and Karim (1991); Dellaportas and Smith (1993) casted the generalized linear model in the Bayesian framework and achieve computational efficiency by using Gibbs sampling; Diebolt and Robert (1994) assessed the effectiveness of Gibbs sampling on evaluating the posterior distribution of the hidden data of mixture models; in systems biology, Geyer and Thompson (1995); Churchill and

Lazareva (1999) employed Gibbs sampling to restore the sequence of states visited by the hidden Markov chain using the DNA sequence as the observed outputs. Lange et al. (1992) casted the absolute number of CD4 T-cells which is regarded as a marker of disease progression for persons infected with HIV to the nonlinear models, and analyzed the data by identifying the parameters of models by using Gibbs sampling.

A.3.3 Riemann manifold Hamiltonian Monte Carlo method

Aforementioned conventional MCMC methods are required to tweak the algorithmic settings in order to achieve good inference performance. Specifically, a large movement made in each MCMC step offers a higher chance for a sample to escape from the trapped region. Nevertheless if the sample has already been in a region of high probability, it is more likely to be rejected to the detriment of the acceptance rate. In contrast, a small move increases the acceptance rate but more iterations are required to explore the entire space and the retrieved samples are highly correlated.

Information geometry has been combined with MCMC to solve this difficulty, in which the moves of MCMC samples are driven by gradient of dynamics. (Duane et al., 2011) proposed the first genuine physics-inspired MCMC, known as *Hamiltonian Monte Carlo* (HMC), which adopted a Hamiltonian dynamical system to facilitate the transition of samples. Geometric information adds to the appeal of HMC by guiding its exploration to enhance its efficiency. Nevertheless, in the empirical investigation (Neal, 2010), HMC just partially alleviates the dependence on the tuning process, since its transition kernel needs to define a user-specific constant to propose the move of samples, associating with the gradient of dynamics. An advanced method has recently been invented by Girolami and Calderhead (2011), namely *Riemann manifold Hamiltonian Monte Carlo method* (RMHMC), which considerably enhances the effectiveness of MCMC method by using geometric information automatically without requiring manual refinement. Because RMHMC is not directly relevant to this thesis, but is introduced for completeness. Details of the algorithm and a comparative study can be found in Appendix B.2.

RMHMC adopts the geometric information for guiding the exploration, and has shown great potential in parameter estimation. Calderhead and Girolami (2011) highlighted the use of RMHMC on the high dimensional parameter estimation problem, and its effectiveness is illustrated by applying to parameter inference of the cell signaling pathways and the enzymatic circadian control system. Yuan et al. (2012) successfully analyzed the abnormalities of heat beat with RMHMC, in combination with a state-space point process model. Dondelinger et al. (2013) pre-processed the experimental data by modeling with a Gaussian process so as to boost the efficiency of RMHMC. The effectiveness of this approach was examined by studying two benchmark ODE systems, in which results obtained were on a par with the original RMHMC in terms of accuracy, yet required less computational expense.

A.4 Kalman and Particle Filters

In the following section, we introduce the general framework of state-space models with partial observations from the biological systems. Three popular one-pass inference methods are preliminarily studied on the incomplete noisy synthetic dataset, where the potentials of methods to identify the kinetic parameter are thoroughly explored. A comparative study between sequential and batch methods is further carried out to deliver empirical guidance for choosing an inference algorithm.

A.5 Estimating state and parameter

In systems biology, the process of interest is often characterized by a set of ordinary differential equations (ODEs), which capture changes to a system with respect to time and solutions of which help to explain behavior at the system level. As mentioned in Chapter 2, we generally consider nonlinear state-space models of the biological systems in which the dynamics are deterministic and observations noisy. More specifically, the nonlinear state-space models adapted for systems biology are given as

$$\dot{\mathbf{x}}_t = f(\mathbf{x}_t, \boldsymbol{\theta}), \tag{A.34}$$

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \int_{t-1}^{t} f(\mathbf{x}, \boldsymbol{\theta}_t) d\tau, \tag{A.35}$$

$$\mathbf{y}_t = h(\mathbf{x}_t) + \boldsymbol{v}_t, \tag{A.36}$$

where state vector $\mathbf{x}_t$ may consist of concentrations of different molecular species at time t, and $\mathbf{y}_t$ quantifies the noisy observations relating to $\mathbf{x}_t$ via the output function $h(\cdot)$. $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_p\}$ is the parameter vector of dynamics. It is necessary to recall that $\boldsymbol{v}_t$ is the zero mean Gaussian noise corrupting observations, and its covariance matrix is denoted by $\boldsymbol{R}$. This $\boldsymbol{R}$ is a positive definite matrix, quantifying the quality of observations. Additionally, $\boldsymbol{R}$ is defined as a diagonal matrix due to the noise corrupting to each observation is assumed to be uncorrelated. Setting of $\boldsymbol{R}$ governs the performance of inference methods, investigation on this subject is given in section A.6.3.

Quite often, the hidden states and the unknown kinetic parameter values, give rise to difficulties in throughly exploring the system. Biochemical experiments conducted *in virtro* for the latent state variable, and the elaborate hand-tuning of parameter values until satisfactory results are achieved, are conventional solutions. However, the measurement taken *in vitro* might not be a good reflection of those *in vivo* and the unreliability of tweaking process necessitates the use of probabilistic inference tools for identifying system, either state variable or kinetic parameter.

Following the method suggested by Sitz et al. (2002), the unknown parameters are

treated as the additional states of system. In order to estimate parameters, an artificial dynamics is imposed to the unknowns and the random walk scheme is usually chosen, which is given as

$$\boldsymbol{\theta}_t = k(\boldsymbol{\theta}_{t-1}) = \boldsymbol{\theta}_{t-1} + \boldsymbol{\omega}_t^{\theta} \tag{A.37}$$

where $k(\cdot)$ is the function to perturb the parameters, and $\boldsymbol{\omega}_t^{\theta}$ is the zero mean Gaussian with covariance matrix $\boldsymbol{\sigma}_k^2$. The random walk scheme is often used as the artificial dynamics due to its simplicity.

Several probabilistic inference approached such as Maximum-Likelihood (ML), Expectation-Maximization (ML) and Markov chain Monte Carlo (MCMC) have been previously applied to identify the biological systems, either only states or states and parameters simultaneously. In the following sections, we focus on the use of EKF, UKF and PF, being known as the popular sequential methods, to infer the states and parameters of a specific biological system. The introduction of EKF, UKF and PF can be found in section A.1.2, A.1.3 and A.1.4 of chapter 2. A comparative study of performance of these three approaches is carried out, and the empirical suggestion of using considered methods are proposed according to the comparison.

A.5.1 Estimating single hidden state or unknown parameter

In this study, we consider cellular response to heat shock as a model system to illustrate the effectiveness of three sequential inference approaches. El-Samad et al. (2006) describe a heat shock response model in the bacterium *Escherichia coli*, with three differential equations. The model consists of genes encoding molecular chaperones, transcribed in response to a sudden change of environmental temperature, transcription factor σ^{32}, which, via binding to RNA polymerase enables the transcriptional regulation of heat shock proteins. σ^{32}, which is rarely present under normal temperatures in the range (30 °C-37 °C), rapidly accumulates at elevated temperatures activating the transcription of heat shock genes, leading to a different equilibrium.

The variation in total numbers of the relevant proteins are lumped in the model as two terms whose dynamics is described, along with the numbers of unfolded proteins as follows

$$\begin{aligned}
\dot{D}_t &= K_d \frac{S_t}{1 + \frac{K_s D_t}{1 + K_u U_f}} - \alpha_d D_t \\
\dot{S}_t &= \eta(t) - \alpha_0 S_t - \alpha_s \frac{\frac{K_s D_t}{1 + K_u U_f}}{1 + \frac{K_s D_t}{1 + K_u U_f}} S_t \\
\dot{U}_f &= K(t)[P_t - U_f] - [K(t) + K_{fold}] D_t.
\end{aligned} \tag{A.38}$$

Here, D_t represents the number of molecules of chaperones, S_t, the number of molecules of σ^{32} and U_f describes the total number of unfolded proteins. P_t and K_{fold} are the total concentration of proteins in the cell and the coefficient for the folding process, respectively. $K(t)$ and $\eta(t)$ are constants assuming different values at different steady state temperatures.

El-Samad et al. (2006) proposed an advanced algorithm based on the Lyapunov function to analyze the robustness of biological circuits, and gave a quantitative analysis of the heat shock model by applying the proposed algorithm on the system. Petre et al. (2011) simplified the original model by identifying the reaction of behavioral response to heat shock and proposed a mathematical validation of the simple model. Lillacci and Khammash (2010) used this heat shock model as an illustration to examine the capability of their proposed inference algorithm for parameter estimation. In this work, a validated test is combined to UKF and the parameters of heat shock model are precisely estimated.

Since the heat shock system has been previously employed to test the effectiveness of inference approach, and the advantage of the proposed method is claimed based on the investigation (Lillacci and Khammash, 2010). Consequently, we consider this system as a reference to carry out the comparison between the previously studied algorithm (Lillacci and Khammash, 2010) and other approaches (EKF and PF). The synthetic data is simulated by solving the heat shock model by making use of `MATLAB`'s `ODE45` function to integrate the differential equations, generating data over a time length of 200 min and sampled at regular intervals of 0.2 min, giving 1000 samples points representing the observations. In the simulation, the states found hidden in observations are reflected through the observation function $h(\cdot)$. As an example, if the state D_t in the heat shock system is hidden in the observations, such inaccessibility can be mimicked by defining a binary diagonal observation matrix, given as

$$h(\cdot) = \boldsymbol{G} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{A.39}$$

The simplest scenario we considered is one in which two of the three states and all parameters were assumed known. This corresponds to the case of a well understood biological system, some aspects of which are experimentally observable, subject to additive measurement noise, whereas others are inaccessible. The objective is to precisely estimate the hidden state given only the input-output dynamics.

A.5.2 Extended Kalman filter

As shown in the pseudo-code of EKF (Algorithm 9), the analytical solutions of Jacobian matrix of system transition and observation function are required, these first-order partial derivatives are given as

$$\boldsymbol{F}_t = \begin{bmatrix} \frac{\partial f_{D_t}}{\partial_{D_t}} & \frac{\partial f_{D_t}}{\partial_{S_t}} & \frac{\partial f_{D_t}}{\partial_{U_f}} \\ \frac{\partial f_{S_t}}{\partial_{D_t}} & \frac{\partial f_{S_t}}{\partial_{S_t}} & \frac{\partial f_{S_t}}{\partial_{U_f}} \\ \frac{\partial f_{U_f}}{\partial_{D_t}} & \frac{\partial f_{U_f}}{\partial_{S_t}} & \frac{\partial f_{U_f}}{\partial_{U_f}} \end{bmatrix}, \quad \boldsymbol{H}_t = \begin{bmatrix} \frac{\partial h_{D_t}}{\partial_{D_t}} & \frac{\partial h_{D_t}}{\partial_{S_t}} & \frac{\partial h_{D_t}}{\partial_{U_f}} \\ \frac{\partial h_{S_t}}{\partial_{D_t}} & \frac{\partial h_{S_t}}{\partial_{S_t}} & \frac{\partial h_{S_t}}{\partial_{U_f}} \\ \frac{\partial h_{U_f}}{\partial_{D_t}} & \frac{\partial h_{U_f}}{\partial_{S_t}} & \frac{\partial h_{U_f}}{\partial_{U_f}} \end{bmatrix}. \tag{A.40}$$

EKF is set to start from the initial prior state vector $\widehat{\mathbf{x}}_0 = [0,\ 0,\ 0]$ and the diagonal elements of error covariance $\widehat{P}_0$ are 25, 25, 25 and 1×10^{-2}, respectively. The process noise covariance matrix is set to an approximately zero diagonal matrix, 5×10^{-6}, while the diagonal elements in covariance matrix for observation noise $\boldsymbol{R}$ are initially 0.01 times variance of synthetic state data $\mathbf{X}$. This multiplier of variance is varied in the following section to examine the effect of noise level on the performance of inference algorithms. The results of inferring with EKF in the single hidden state case are given

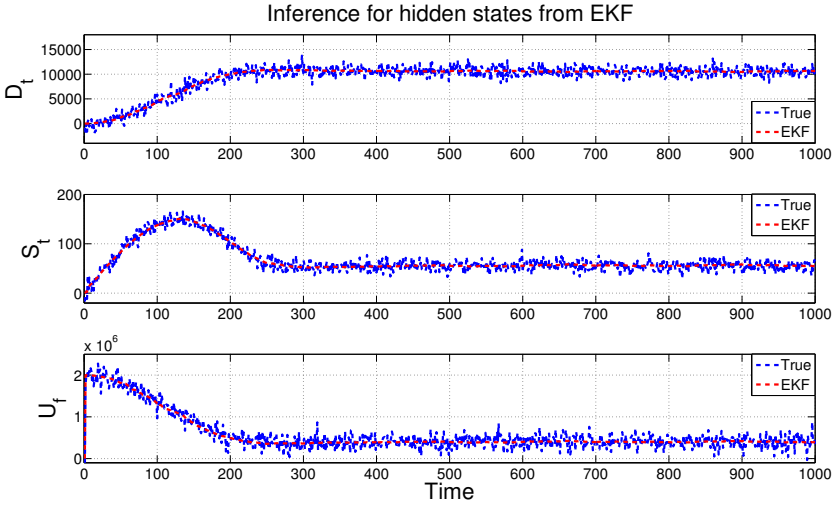


FIGURE A.8: State estimation of heat shock model by the EKF algorithm, in which only one state is assumed to be unknown. Each row shows a particular hidden state when assuming the other two are known. In order to compare, the 'true' states generated by directly solving the system transition are shown as the blue dash lines.

in Figure A.8. As shown in the graphs, the behaviors of this transient system have been accurately tracked by EKF, when only a single state is unobserved.

In the upcoming simulation, EKF is used for estimating the parameters of the system, where only one parameter is assumed to be unknown while the remaining five are known Moreover, all states are observed in the simulation. The inferences are shown in Figure A.9. A good convergence to the true value can be observed in three cases, and failures occur in the estimation for parameters α_s, K_d and K_u. Information on these three

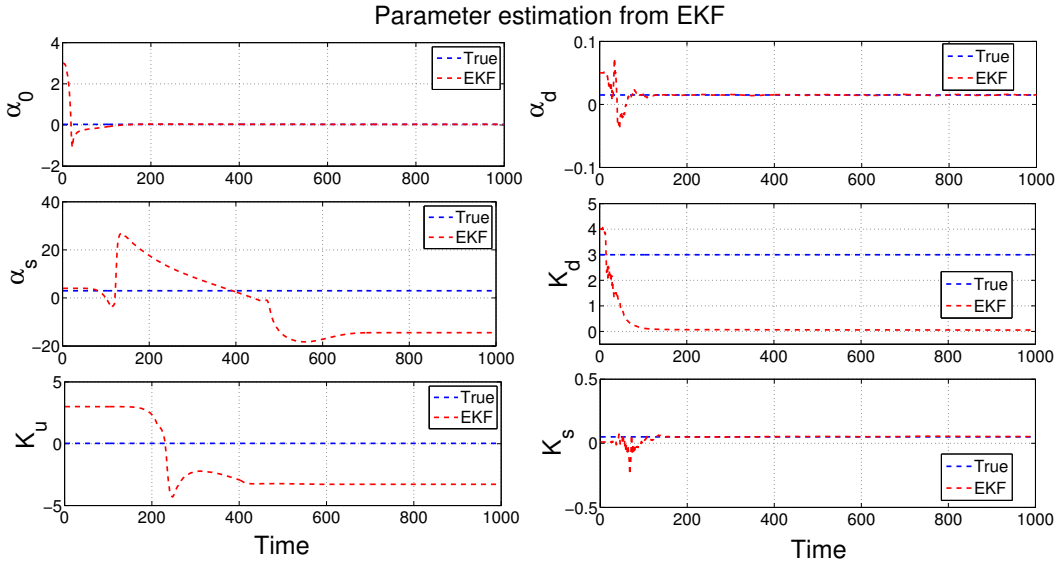


FIGURE A.9: Estimations of single unknown parameter of the heat shock model using EKF algorithm. Each graph shows result of the particular unknown parameter with assuming the remaining five parameters and three states in system are known. For comparison, the true values of parameters provided from literatures are shown as the blue dash lines.

parameters is lost due to the truncation of high-order terms in *Taylor series expansion* is neglected during the consecutive linearizations. This example serves to illustrate the limitation of EKF caused by the linearization, which make it possibly incapable of capturing the behavioral information of highly complex systems.

A.5.3 Unsecented Kalman filter

Likewise, we first test the performance of UKF on inferring the single hidden state or parameter within the identical dataset and algorithmic settings, *i.e.* the initial guess and covariance matrix. The results of UKF inferring the single hidden state are shown in Figure A.10.

It can be seen from Figure A.10, as expected, a good ability for tracking system behavior is clearly shown. Moreover, comparing these parameter estimation results from the UKF (shown in Figure A.11) with those for the EKF shown in Figure A.9, the UKF outperforms the EKF on parameter estimation, since the UKF successfully reaches the true values of parameters α_s, K_u and K_d, whereas failures appear for the EKF.

A.5.4 Particle filter

The appeal of PF lies in its potential to introduce a Markov chain transition kernel $K(\mathbf{x}^*_{0:t}|\mathbf{x}_{0:t})$, with invariant distribution $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$ to perturb the particles toward the high density areas, i.e. $\int K(\mathbf{x}^*_{0:t}|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = p(\mathbf{x}^*_{0:t}|\mathbf{y}_{1:t})$ (Andriue et al., 2001).

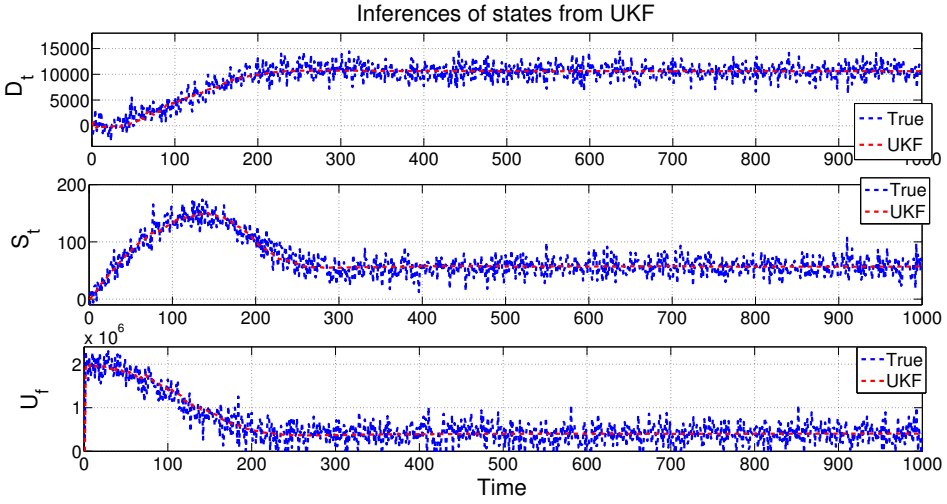


FIGURE A.10: State inference of heat shock model by UKF. Each row shows the particular hidden state with assuming other two in system are known. For comparison purpose, the true states produced by directly solving ODEs are shown as the blue dash lines.

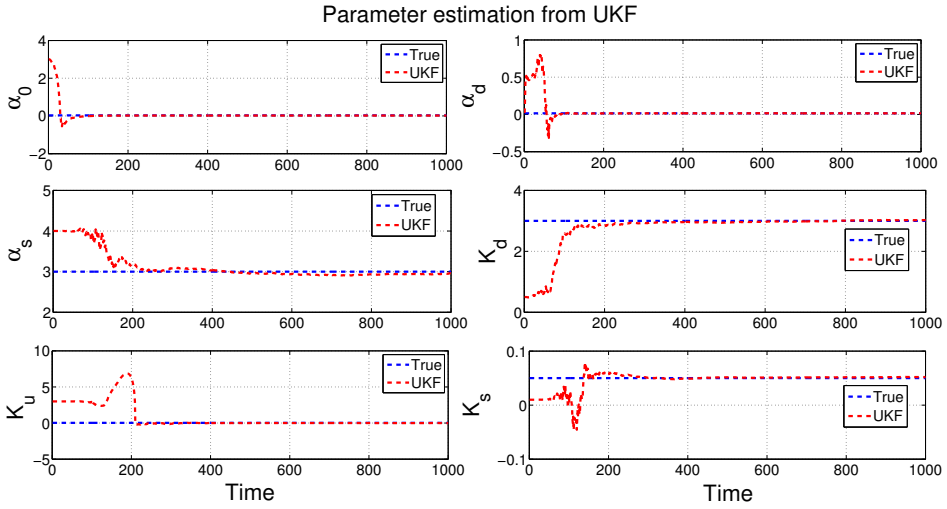


FIGURE A.11: Estimations of the single unknown parameter of heat shock model using UKF algorithm, assuming all states are observable and the remaining five are known. The true values of parameters provided from literatures are shown as the blue dash lines.

From a practical perspective, in order to carry out a tracking behavior which helps convergence from an arbitrary initial guess to the true value, random walk given as equation A.37 is often adopted for this transition kernel. There is an extensive discussion in the literature on the ratio between the assumed noise variance of the random walk and the noise variance of the observations being the determinant of convergence in SMC setting (Kitagawa, 1998; Bar-Shalom et al., 2001; Li et al., 2004).

However, this random walk may sometimes lead to divergence of the posterior. Liu (2001) suggested an alternative state/parameter evolution, where a kernel smoothing with shrinkage is utilized for achieving estimate convergence. Updating of state/param-

eter in each iteration is controlled by either a specific direct or the variance-dependent step size. This scheme is adopted in our work to perform the evolution of state/parameter, and its mathematical description is given below

$$\begin{aligned} \mathbf{x}_{t+1} &= a\mathbf{x}_t + (1-a)\bar{\mathbf{x}}_t + \boldsymbol{\omega}_t^x \\ a &= \frac{3\delta - 1}{2\delta}, \end{aligned} \tag{A.41}$$

where δ is a discount factor in [0 1] and $\bar{\mathbf{x}}_t$ is the mean of particles at time instance t. $\boldsymbol{\omega}_t^x \sim \mathcal{N}(0, h^2\boldsymbol{V}_t)$ is the additive noise corrupting parameter evolution. where $h^2 = 1-a^2$ and $\boldsymbol{V}_t$ is the covariance matrix of particles at time instance t. A comparison between the traditional random walk and this advanced transition kernel is carried out and represented in chapter 4.

The capability of PF for estimating a hidden state or an unknown parameter is also assessed by applying to the previously studied case. Unsurprisingly, as seen in Figure A.12 and Figure A.13, PF accurately captures system dynamics and achieves good convergence to the true value of the parameter.

Consequently, we conclude that EKF struggles with the loss of information caused by linearization and produces inaccurate estimates. UKF performs similarly to PF on system identification in terms of accuracy. PF, benefiting from its sampling strategy, is clearly observed the superior performance on inference. The additional complexity caused by particles is likely, however, to come at an extra computational cost.

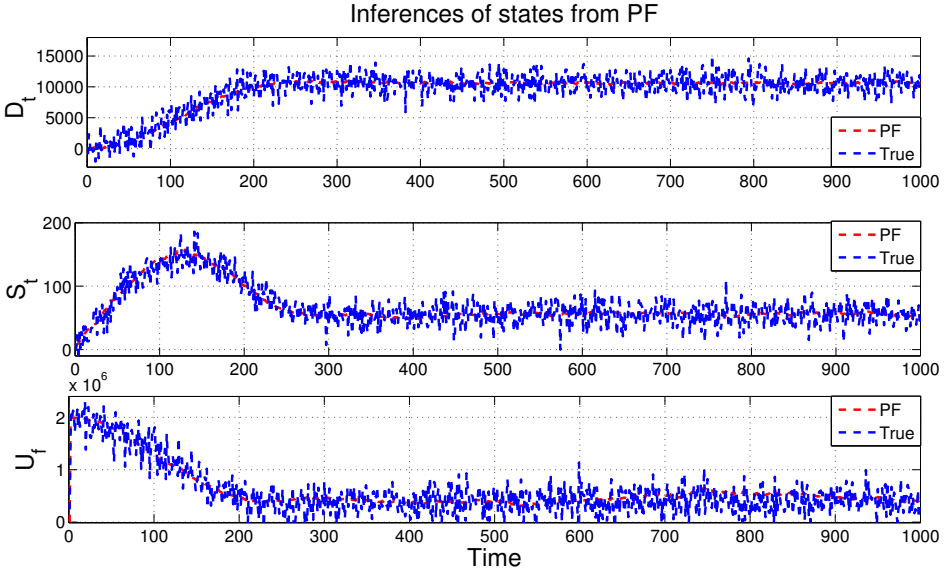


FIGURE A.12: State estimation of heat shock model by using PF algorithm, in which only one state is unknown, while other two of three are observable. Each row shows the particular hidden state. Behavior of the latent state generated by directly solving the system dynamics are shown as the blue dash lines.

In the simple case, the estimates from UKF and PF are very much the same, while EKF

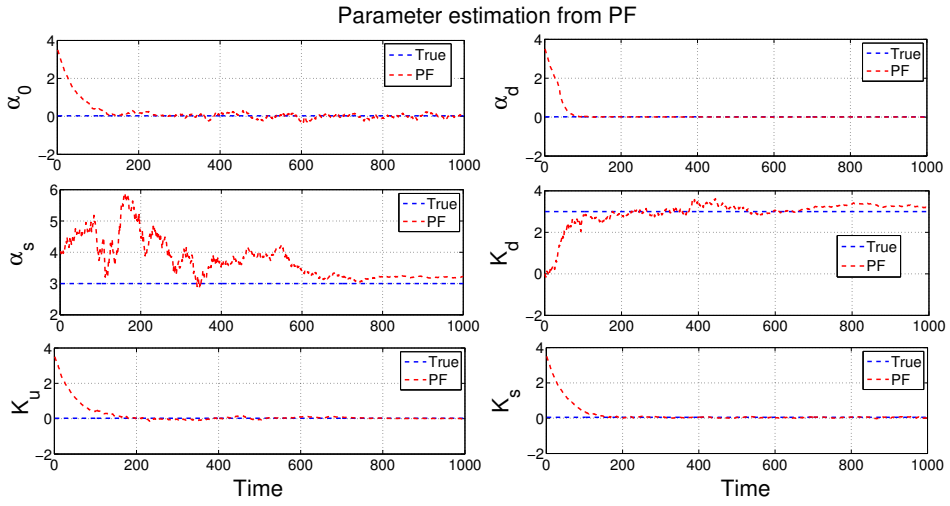


FIGURE A.13: Single unknown parameter estimations of heat shock model by PF algorithm, where all states are observable and five of six parameters are given. The true values of parameters from the literatures are shown as the blue dash lines.

may fail to converge to the true value of parameter. In the following section, we are progressively making the problem harder, and attempt to discriminate the capability of UKF and PF on parameter estimation.

A.6 Influences of initial condition and regime of data on Kalman and particle filterings

Intuitively, the conceptual idea of UKF and PF is identical to each other (based on sampling), however, they only differ in the number of particles drawn and the way to pick particles. When the system of interest is highly nonlinear, UKF is limited by the deterministic sampling scheme and the untunable number of samples used. In comparison to UKF, PF takes the advantage of its capability of randomly drawing the particles and specifying the number of particles based on the complexity of the system of interest.

In the previously studied case, no clear difference in the ability for system identification appear. We further examine the algorithms on inference without providing such favorable environmental settings.

A.6.1 Estimating a single unknown parameter with unfavorable prior and various observed states

In this set of simulations, we keep the regime of dataset the same including the time length and sampling interval. In addition, all states are assumed to be observable in

system outputs. Algorithms are carried out only with differences in initialization, in which the EKF and UKF are set to be the closest to the truth among all the particles used as initial samples of the PF.

By considering this favorable setting for Kalman filter techniques, algorithms are employed to identify the system, either by simultaneously estimating the parameter K_s and state S_t or by only inferring the parameter by given complete noise-free observations. Results are shown in Figure A.14.

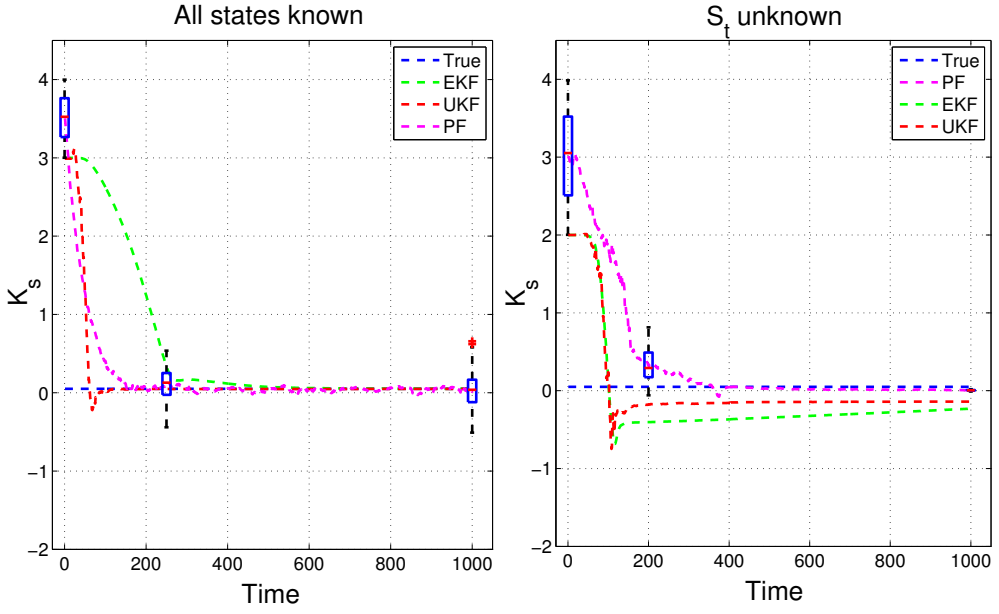


FIGURE A.14: Estimations of a single parameter K_s from all states are observable and two observed states with different initializations. The boxes show the distribution of samples of the particle filter at 1^{st}, 250^{th} and 1000^{th} points in time.

As shown in Figure A.14, three algorithms perform similarly in the complete observation case. Interestingly, when parameter K_s and state S_t are simultaneously estimated, even though the EKF and UKF are provided a highly favorable algorithmic setting, they fail to recover the correct value of the unknown, while the PF is still able to find the correct solution. Having a relatively high nonlinearity in its expression, the hidden state S_t leads to a lower probability for parametric approaches to find the correct solution. In order to verify this claim, we further examine the methods on simultaneously estimating K_s and another two states. Estimates of K_s with different hidden state are shown in Figure A.15, from which we can easily observe that the decline in nonlinearity positively effects the performance of Kalman filtering techniques.

By making the inference task progressively harder, we next examine the performances of algorithms in estimating one parameter simultaneously, with all possible combinations of two states being left unknown. Complete results are shown in Figure A.16. We note the general trend of EKF performing worse than the other two. Quite often, the UKF algorithm gains the higher rates for converging to the true values when the hidden state

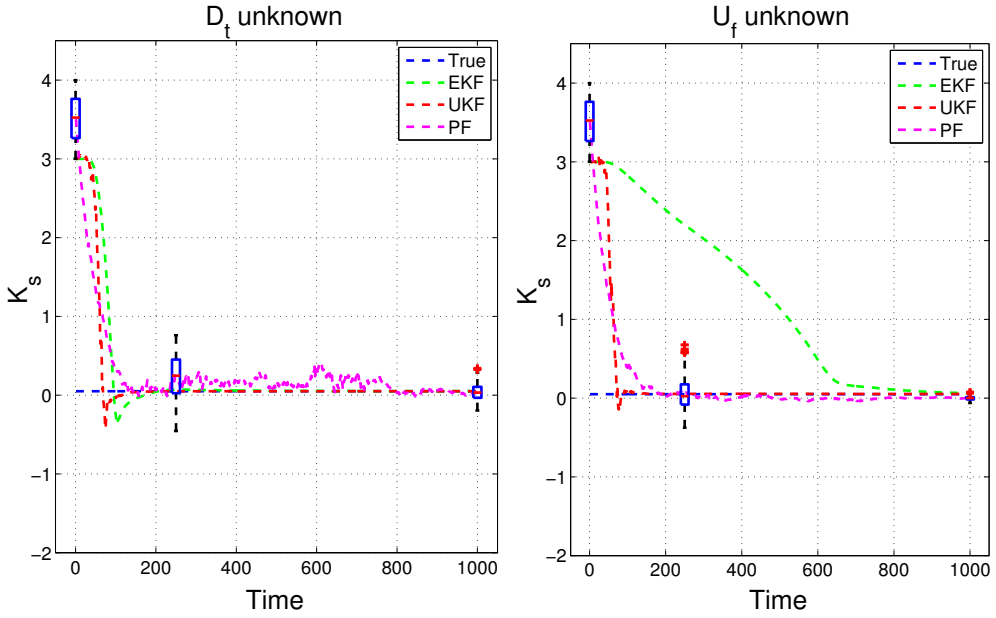


FIGURE A.15: Estimations of a single parameter K_s from the unfavorable initializations, assuming states D_t or U_f are unknown. The boxes show the distribution of samples of the particle filter at initial, two hundredth and final points in time.

variable is formulated by the simple equation. For instance, in the cases that assume D_t or U_f are unobserved, it is easier for UKF to find the correct solution of unknown parameters, in comparison to S_t is the hidden state. PF, as expected, is found to be more resilient than either EKF or UKF in converging to the true.

Consequently, we conclude that the PF has the greatest potential to recover the correct value from either the unfavorable prior distribution or the incomplete observation. For the kalman methods, the capability of inference is considerably driven by the nonlinearity of hidden state variable, however, a general superiority of UKF over EKF is always expected.

In addition, as seen in the graphs, some particular parameters, for instance α_0 and α_s, are easier for parametric methods to converge to the true value than others. This is expected because the conditional posterior probabilities over these particular parameters are unimodal, and close to Gaussian, so Kalman filtering techniques have a good chance of achieving promising solutions. When the parameter is lumped in models with a high nonlinearity, the posterior distributions are often multi-modal and non-Gaussian, therefore non-parametric particle filtering is better suited. This identifiability is related to the sensitivity of parameters in model and has been quantitatively defined by Gutenkunst et al. (2007), this problem also appears in the *approximate Bayesian computation* methods introduced in chapter 3.

In this single unknown parameter case, we explore all the possible combinations and find in 20 of the 42 simulations, PF has an advantage over EKF and UKF, whereas in the remaining 22 it performs no worse than them. Table A.1 summarizes the success/failure

by studying these graphs.

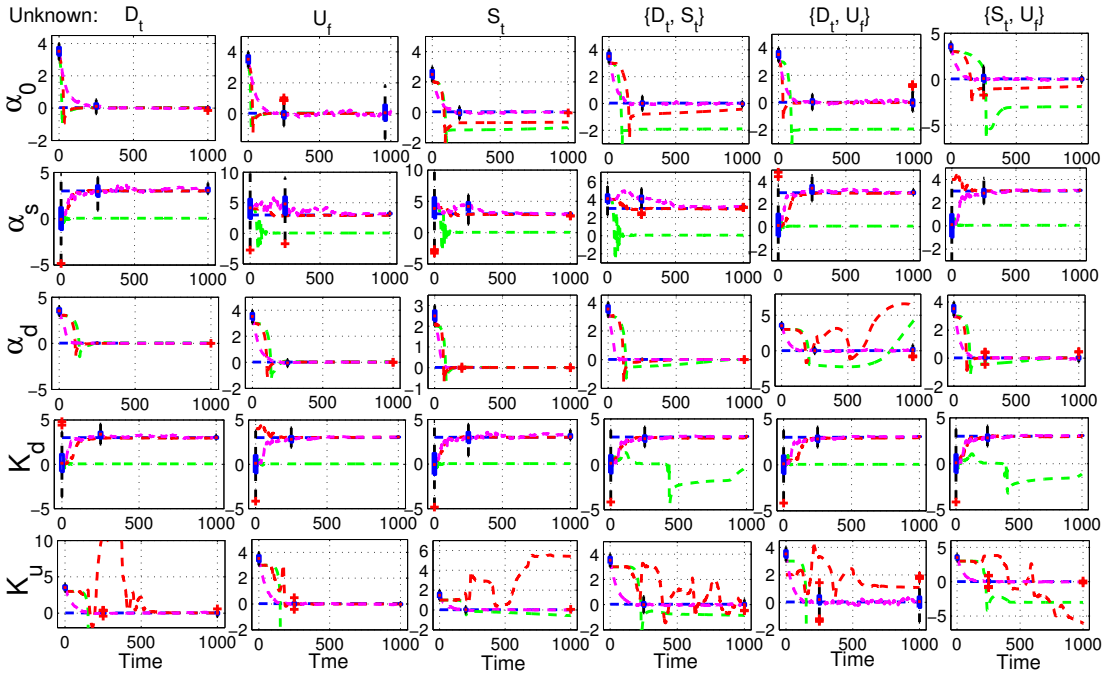


FIGURE A.16: Estimations of all parameters in the single unknown cases. Starting from the unfavorable initializations, assuming one or two states are inaccessible in the observations. The boxes show the distribution of samples of the particle filter at 1^{th}, 250^{th} and 1000^{th} points in time. *Rows* of graphs are categorized by parameter. *Columns* show the latent states in simulation. The results of EKF, UKF and PF are indicated by green, red and purple dash lines, respectively.

A.6.2 Estimating multiple parameters

Three sequential inference algorithms are further applied to estimate two parameters and infer the state S_t of the heat shock response system. Estimations of K_s and K_d are shown in Figure A.17 as a representative example and the complete results are shown in Appendix C. By studying all graphs, in all possible combinations of two unknowns, PF wins the battle among three sequential algorithms in 13 of 15 cases, except two simulations for $\{K_d, K_u\}$ and $\{\alpha_s, K_d\}$.

Quite naturally, more failures appear in algorithms using a unfavorable prior distribution. This is to be expected because the problem is then significantly harder than the single unknown parameter case. A comparison of various initial conditions from which convergence is reached for the three sequential filters is carried out, where the results are shown in Figure A.18. We find that the PF is more resilient than either EKF or UKF in converging to the true solution.

For completeness, we also consider the extreme case of all parameters being unknown by given two noisy observations (D_t and U_f). Results are given in Figure C.1 of Appendix

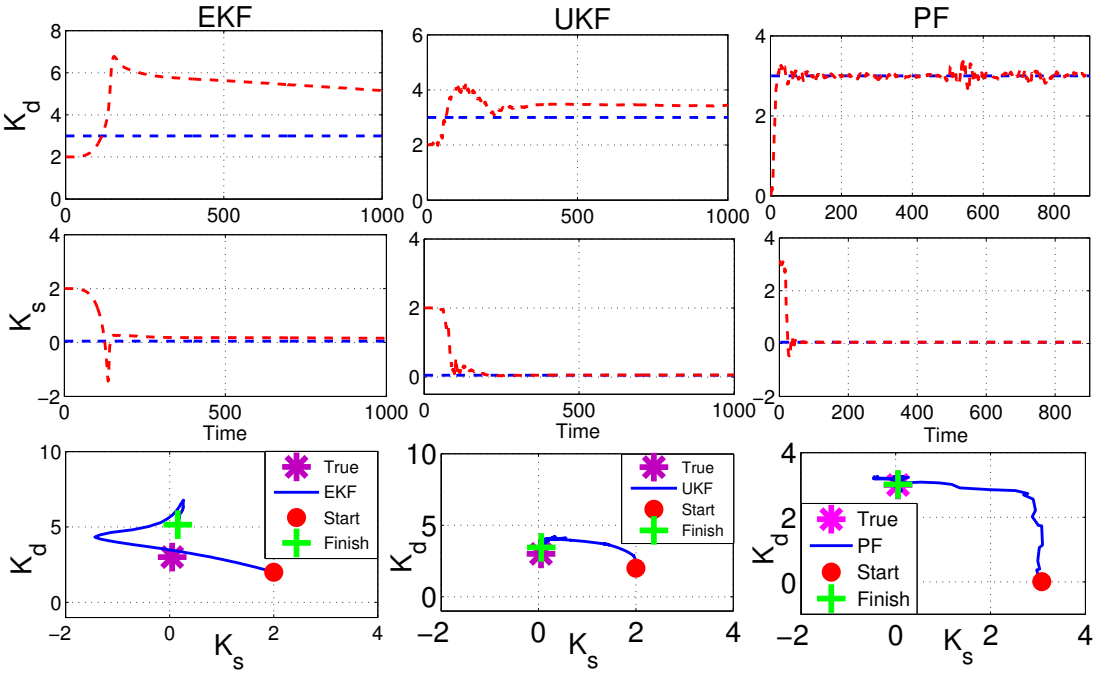


FIGURE A.17: Estimations of K_s and K_d in the two unknown parameters space, assuming state S_t is hidden in system outputs. *First* column: Results obtained from the EKF. Trajectory of tracking behaviors is shown in the left-hand side of the bottom row. *Second* and *Third* columns: Graphs of estimations produced by UKF and PF, respectively.

C. EKF and UKF were found to be incapable of estimating in such scenarios, whereas PF was able to correctly estimate four of six unknowns. The two failures occurred in the inferences for K_d and α_d, which is probably due to their significant influences on charactering the hidden state S_t.

A.6.3 Effects of data regimes

Several questions, such as how would methods perform when fast data sampling is considered, what is the minimal time window preventing algorithms from breaking down, and what is the effect of noise on inference, can be posed. Such empirical investigations, fortunately, is possible to gain if the mathematical expression of the system is given.

In order to explore the effect of time length, we carry out three sequential filters using data generated with various time windows, i.e. 40, 60, 80, 100, 120, 140, 160, 180, 250 and 350 `min`. By studying the simulations shown as Figure C.2 - C.4 in Appendix C, when the time length is less than 80 `min` (i.e. 120 data points for representing observations with sampling at 0.2 `min`), Kalman filtering algorithms break down due to the insufficient system information provided. Due to it being highly case dependent, it is not possible to offer general guidance on choosing this setting

Sampling interval, as the critical factor of operating regime, is sometimes a limitation

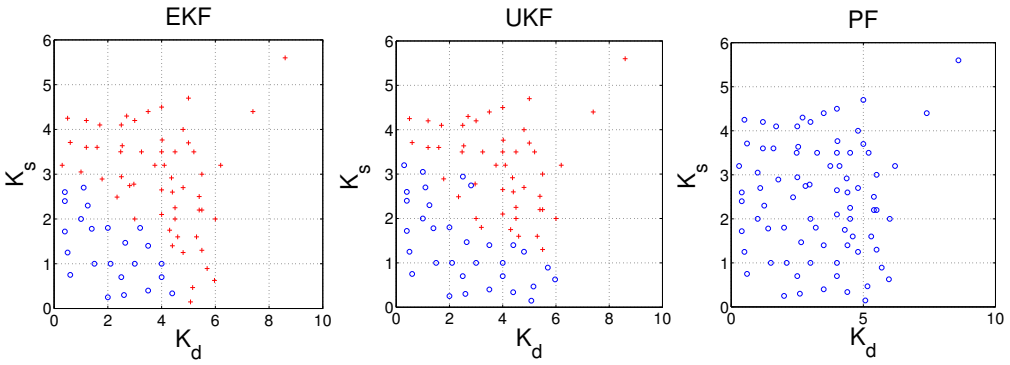


FIGURE A.18: A comparison of simultaneously estimating K_s and K_d starting from various conditions, from which convergence is reached for the three different sequential filters. If the method fail to produce precise estimation, then the underlying prior is denoted by red '+', when the success occurs, the starting point of this simulation is described by blue 'o'.

in performance (e.g. fast sampling of data point is a bottleneck for cellular metabolism as the intracellular turn-over is less than the sampling rate (Villas-Boas et al., 2007)), as a result of which, the investigation of effect of data regime could be more meaningful. Apart from the regular setting of sampling interval which is taken from the previous literature (Lillacci and Khammash, 2010) and defined as 0.2 `min`, several reasonable values including 0.1, 0.25, 0.5 and 1 `min`, are adopted for synthesizing dataset. The results of this study can be found at Figure C.5 - C.7 in Appendix C the benefits of faster or slower sampling intervals are summarized in Table A.2, from which we can find rapid sampling may be useful in some scenarios.

Intuitively, noise should be an issue for performing inference approaches. Simulation to quantitatively analyze the influence of noise are carried out by varying the multiplier as 0.0001, 0.001, 0.01, 0.05, 0.1 and 1. The estimations of parameter K_d from multiplier chosen as 0.0001, 0.05 and 1 are shown in Figure A.19. It is clearly observed that all algorithms find the correct value with approximately zero noise, while failures are found in three sequential filters when the observations are massively noisy. Nevertheless, as seen in the middle column of Figure A.19, PF outperforms other parametric methods in terms of handling noisy observations. Complete results are shown in Figure C.8 - C.10 of Appendix C

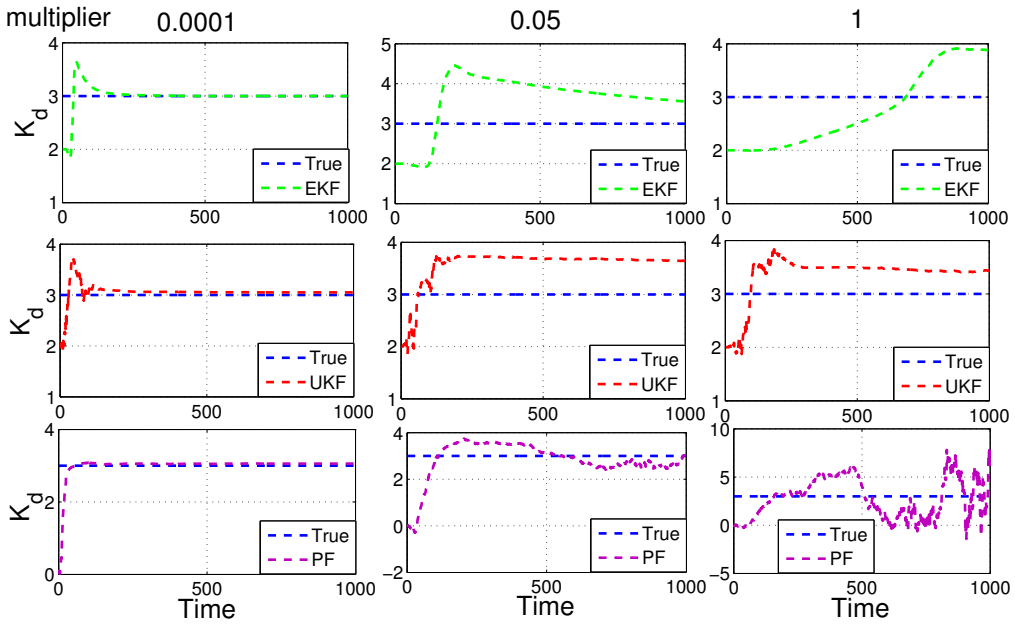


FIGURE A.19: Estimation of parameter K_d in the two unknowns (K_d and K_s) from three sequential algorithms. *columns* are response to the noise variance mutilplier 0.0001, 0.05 and 1, respectively.

A.7 The advantage of sequential approaches

The Kalman and PF algorithms adopted in this work are sequential approaches. Their use should be considered in the context of Bayesian inference methods that operate on batch data (Vyshemirsky and Girolami, 2008; Wilkinson, 2009), for example as in the heat shock model where all the data are available. In such cases, the sequential models, being one-pass algorithms offer a computational advantage. We illustrate this by comparing the PF and a *Metropolis-Hastings* sampler (MH). To ensure a fair comparison, the MH was carried out in the simplest environmental setting, that is to infer the single unknown parameter given completely noise-free observations. Results of these estimations and comparisons of the computational cost are shown in Figure A.20. Unsurprisingly, the MH successfully recovers the unknown in all cases, however, since the MH re-visits the entire dataset at each iteration, PF is more efficient than MH with increasing numbers of iterations.

In addition, the heat shock response system has been used for demonstrating the capability of *maximum-likelihood estimator* (MLE) for parameter estimation by El-Samad et al. (2006), and we are therefore motivated to deliver an empirical investigation to quantify the effectivenesses of these two methods. While the noise is set reasonably, MLE outperforms PF in terms of accuracy and computational efficiency, regardless the combination of unknown parameter and hidden state. We then examine two algorithms by using various initializations and generating the additive noise from a mixture-Gaussian distribution, given as $\boldsymbol{v}_t \sim 0.3 \cdot \mathcal{N}(10, 5^2) + 0.7 \cdot \mathcal{N}(25, 10^2)$. Results are shown in Figure A.20, where the negative effect of mixture Gaussian noise on MLE is evident. However,

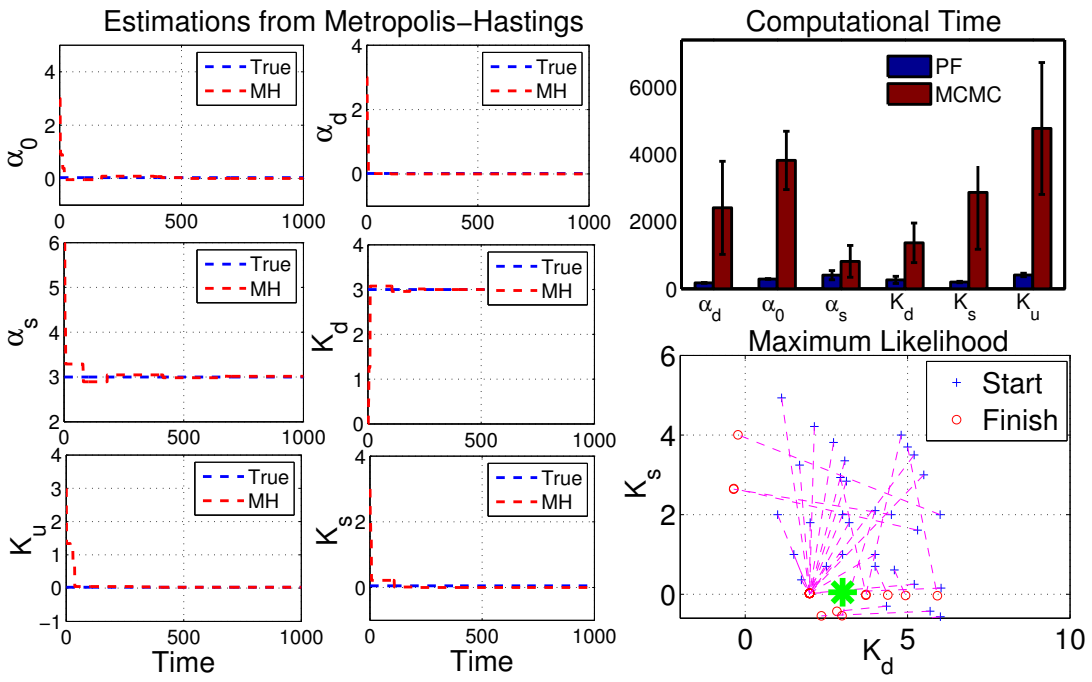


FIGURE A.20: *First and second* panels: estimations of parameters within single unknown space from Metropolis-Hastings algorithm. *top corner*: comparison of computational costs between a batch method and PFs in estimating single unknown parameters. *bottom corner*: performance of a deterministic optimization approach. From various initial conditions, when simultaneously estimating two unknown parameters with mixture Gaussian observation noise, the maximum-likelihood method fails to converge (true values denoted by green '×') whereas the PF was able to find the correct solution in the posterior mean in about half of these and the results are shown in Figure C.11 of Appendix C.

PF is able to find the correct solutions in about half of its attempts.

A.8 Discussion

This work demonstrates the effectiveness of the method of particle filtering in state and parameter estimation of deterministic biological systems from noisy observations. We have shown this via a comparative study between extended Kalman, unscented Kalman and particle filtering applied to the heat shock response system using single and multiple parameter estimations. While previous authors (Sun et al., 2008; Lillacci and Khammash, 2010) have argued that the Kalman filter itself is capable of such estimations, our critical appraisal shows that this is only possible when the initial guess of the state and/or state parameters are very close to the true values. Convergence is not achieved when the initial conditions differ significantly from the corresponding true values. The particle filter, on the other hand, is able to converge to true values even when all the particles are initially set to values far away from the underlying true values, providing a powerful, yet simple to implement, way of tackling difficult inference problems in systems biology. We further showed in this work that when the complexity

of the problem is gradually increased (i.e. the number unknown parameters/states to be inferred), the Kalman filter algorithms failed well before the particle filter did. Even in the extreme case of all parameters being unknown, the PF manages to find correct estimates for four of the six cases. This suggests that the non-parametric approach of the particle filter, by virtue of being able to systematically propagate uncertainties while exploring the space over a wide range, is a powerful methodology to tackle such difficult problems.

We also note that the identifiability of parameter/state caused by the sensitivity property emerges as an issue in Kalman algorithms and particle filter. The quantitative analysis of identifiability inspires our development of a new inference method to overcome this difficulty and details are introduced in chapter 4.

A.9 ABC-Regression algorithm

The tradeoff made for acceptance rate and precision limits the widespread use of ABC-rejection. Beaumont (2003) introduced the modification of standard rejection ABC method associated with a local regression adjustment. This so-called ABC-regression method and ABC-rejection appear fundamentally in collecting samples for approximating the posterior distribution. With making use of the local correction, ABC-regression is allowed to afford a relatively large tolerance ϵ. The innovation of this algorithm is in characterizing the relationship between the collection of accepted samples and their corresponding discrepancies via a linear regression model, given as

$$\widehat{\boldsymbol{\theta}} = \mathbf{D}\boldsymbol{\beta} + \boldsymbol{\omega}, \tag{A.42}$$

where $\widehat{\boldsymbol{\theta}} \in \mathbb{R}^{N_{reg} \times D_p}$ are the accepted samples from the rejection scheme. $\boldsymbol{\beta} \in \mathbb{R}^{q \times D_p}$ are the regression coefficients and q is the dimension of the summary statistics which is introduced below. $\boldsymbol{\omega} \in \mathbb{R}^{N_{reg} \times D_p}$ denotes the unobserved random variables and $\mathbf{D} \in \mathbb{R}^{N_{reg} \times q}$ is the matrix indicating the discrepancy between the pseudo-observations and the true dataset, defined as

$$\mathbf{D} = \begin{bmatrix} s^*_{1,1} - s_1 & s^*_{1,2} - s_2 & \dots & s^*_{1,q} - s_q \\ s^*_{2,1} - s_1 & s^*_{2,2} - s_2 & \dots & s^*_{2,q} - s_q \\ s^*_{3,1} - s_1 & s^*_{3,2} - s_2 & \dots & s^*_{3,q} - s_q \\ \vdots & \vdots & \vdots & \vdots \\ s^*_{n,1} - s_1 & s^*_{n,2} - s_2 & \dots & s^*_{n,q} - s_q \end{bmatrix}. \tag{A.43}$$

We note that this discrepancy matrix is calculated using the summary statistics $\boldsymbol{s} \in \mathbb{R}^{q \times 1}$, instead of the raw data $\mathbf{x}$. This is because the ABC methods were motivated initially to tackle the inference problem in population genetics, a field often making use of very large datasets. When each individual point in dataset is used to evaluate

the discrepancy matrix, the computational complexity becomes unaffordable. Summary statistics[5], capturing dynamical features as much as possible with minimum effort, are instead used to calculate the matrix $\mathbf{D}$.

The linear regression model shown in equation A.42 was taken by Beaumont (2003) to correct the result from the rejection scheme, in which the updated inference is adopted for the term $\boldsymbol{\omega}$ in model and the previous result is corrected by

$$\boldsymbol{\theta} = \widehat{\boldsymbol{\theta}} - \mathbf{D}^{\mathrm{T}}\boldsymbol{\beta} \tag{A.44}$$

$$= \widehat{\boldsymbol{\theta}} - (\mathbf{S}^{*} - \mathbf{S})^{\mathrm{T}}\boldsymbol{\beta}, \tag{A.45}$$

where $\mathbf{S} \in \mathbb{R}^{q \times N_{reg}}$ is the summary statistics matrix of the true dataset, which is produced by replicating from the vector $\mathbf{s}$. The value of the regression coefficients matrix $\boldsymbol{\beta}$ is often determined via the least-squares estimator, in which the sum of squared residuals (i.e. the unobserved random variables $\boldsymbol{\omega}$) is given as

$$\begin{aligned} SS(\boldsymbol{\beta}) &= \sum_{n=1}^{N_{reg}} \hat{\theta}_n - (\boldsymbol{s}^{*} - \boldsymbol{s})'\boldsymbol{\beta} \\ &= (\widehat{\boldsymbol{\theta}} - \mathbf{D}\boldsymbol{\beta})^{\mathrm{T}}(\widehat{\boldsymbol{\theta}} - \mathbf{D}\boldsymbol{\beta}), \end{aligned} \tag{A.46}$$

then the regression coefficients $\boldsymbol{\beta}$ can be estimated by first differentiating $SS(\boldsymbol{\beta})$ with respect to $\boldsymbol{\beta}$, given as

$$\frac{\partial SS(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = -2(\widehat{\boldsymbol{\theta}} - \mathbf{D}\widehat{\boldsymbol{\theta}})\mathbf{D}^{\mathrm{T}} = 0. \tag{A.47}$$

Then setting the equation A.47 to zero, we have

$$\boldsymbol{\beta} = (\mathbf{D}^{\mathrm{T}}\mathbf{D})^{-1}\mathbf{D}^{\mathrm{T}}\widehat{\boldsymbol{\theta}}, \tag{A.48}$$

solution of $\boldsymbol{\beta}$ is obtained.

Moreover, in practice, errors (components in $\mathbf{D}$) are not always equally allocated around the 'best-fitting line' for the entire $\widehat{\boldsymbol{\theta}}$, violating model assumptions of homoscedasticity[6]. To handle heteroscedasticity[7], in ABC-regression, the sample from rejection-sampling method $\widehat{\theta}_i$ is weighted by the Epanechnikov kernel, defined as

$$w_i = K_{\epsilon}(d_i) = \begin{cases} c\epsilon^{-1}(1 - (d_i/\epsilon)^2) & d_i \leq \epsilon \\ 0 & d_i \geq \epsilon \end{cases} \tag{A.49}$$

[5] In systems biology, statistical measures such as mean, median, mode, standard deviation and skewness are widely used. In population genetics, more information can be used as summary statistics. For example, the number and frequency of segregating sites, number of population pairs, Shanon's index and variance of allele length in each population.

[6] Homoscedasticity means that the distance between each data point and 'best-fitting line' is identical.

[7] Heteroscedasticity means that the distance between each data point and 'best-fitting line' is subject to change.

where d_i is the discrepancy between $\boldsymbol{s}_i^*$ and $\boldsymbol{s}_s$, c is the normalizing constant and ϵ is the tolerance value. Other kernel functions, for instance, the Gaussian kernel could be adopted for weighting samples. The advantage of Epanechnikov kernel is due to an initial smooth decrease, falling sharply to zero as d_i increases, therefore, few small non-zero values are assigned to weights (Fan and Zhang, 1999; Beaumont, 2003).

The weighted least squares is naturally extended from the original least squares estimator, which determines the regression coefficients matrix $\boldsymbol{\beta}$ by minimizing the sum of squared residuals associated with their corresponding weights, given as

$$\boldsymbol{\beta} = (\mathbf{D}^{\mathrm{T}}\mathbf{W}\mathbf{D})^{-1}\mathbf{D}^{\mathrm{T}}\mathbf{W}\boldsymbol{\theta}, \tag{A.50}$$

where the i^{th} diagonal element of the weights matrix $\mathbf{W}$ is taken by $K_\epsilon(d_i)$. ABC-regression can be carried out with the steps shown in Algorithm 14.

Algorithm 14 ABC-regression

`Input`: $\pi(\theta)$, ϵ, $\mathbf{x}_0$, N_{reg}, $\mathbf{s}$, $\eta(\cdot)$, $\rho(\cdot,\cdot)$ and $f(\cdot,\cdot)$.
`Output`: $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_{N_{reg}}\}$

n=1
Repeat
1. Draw $\theta^* \sim \pi(\theta)$
2. Synthesize $\mathbf{X}^* \sim f(\mathbf{x}_0, \theta^*)$
3. Calculate summary statistics $\mathbf{s}^* = \eta(\mathbf{X}^*)$
4. Evaluate discrepancy $d^* = \rho(\mathbf{s}^*, \mathbf{s})$
5. **if** $d^* \leq \epsilon$ **then**
6. $\quad\widehat{\theta}_n = \theta^*$, $\mathbf{S}_n^* = \mathbf{s}^*$ and $\mathbf{d}_n = d^*$
7. $\quad n = n + 1$
8. **end if**

until $n = N_{reg}$

9. Evaluate weight vector $\boldsymbol{w} : 1 \times N_{reg}$ using equation A.49
10. Calculate regression coefficient matrix $\boldsymbol{\beta} : q \times D_p$ according to equation A.50
11. Replicate $\mathbf{s} : q \times 1 \rightarrow \mathbf{S} : q \times N_{reg}$
12. Compute $\boldsymbol{\theta} = \widehat{\boldsymbol{\theta}} - (\mathbf{S}^* - \mathbf{S})^{\mathrm{T}}\boldsymbol{\beta}$

Example 4.2 *Let ABC-regression work through the deterministic Lotka-Volterra model as a fair comparison, to illustrate its advantage over ABC-rejection. We examine the performances of these two algorithms with different values of of ϵ and combinations of summary statistics. Mean, variance and median are captured as the summary statistics for the dataset; By making use ϵ as 20, 2 and 0.2, we progressively make the acceptance of particle harder. The results obtained using summary statistics are shown in Figure A.21. It is easy to observe that the estimates from ABC-rejection roughly center around the true values, and when the tolerance becomes tighter the precision of inference from rejection scheme is increased greatly. Moreover, considering the identical tolerance $\epsilon = 20$, the accuracy of inference from ABC-rejection is lower than the results shown in Figure 3.2. This is due to the discrepancy being evaluated based on the summary*

statistics in this example, instead of the raw dataset adopted in the previous study. Surprisingly, the expected improvement from regression adjustment is only evident in the case where mean is taken as the summary statistics. When the mean, variance and median are adopted simultaneously for summary statistics, ABC-regression performs worse than ABC-rejection, regardless of the value of tolerance ϵ. This is known as the curse of dimensionality (Bishop, 2006). The accuracy and reliability of ABC-regression decreases rapidly with increasing number of summary statistics, therefore, such problem is the major hindrance for the successful inference by ABC-regression (Joyce and Marjoram, 2008; Beaumont, 2010). In addition, the informative prior distribution is considered, where samples are generated from $\mathcal{U}(0.2, 0.9)$, while the true values of parameters are 0.5. We also note that, in Lotka-Volterra model, the periodicity of species is irrelevant to parameter values and dependence only appears on the magnitude. Consequently, the gain of choosing mean as summary statistics can not be expected generally.

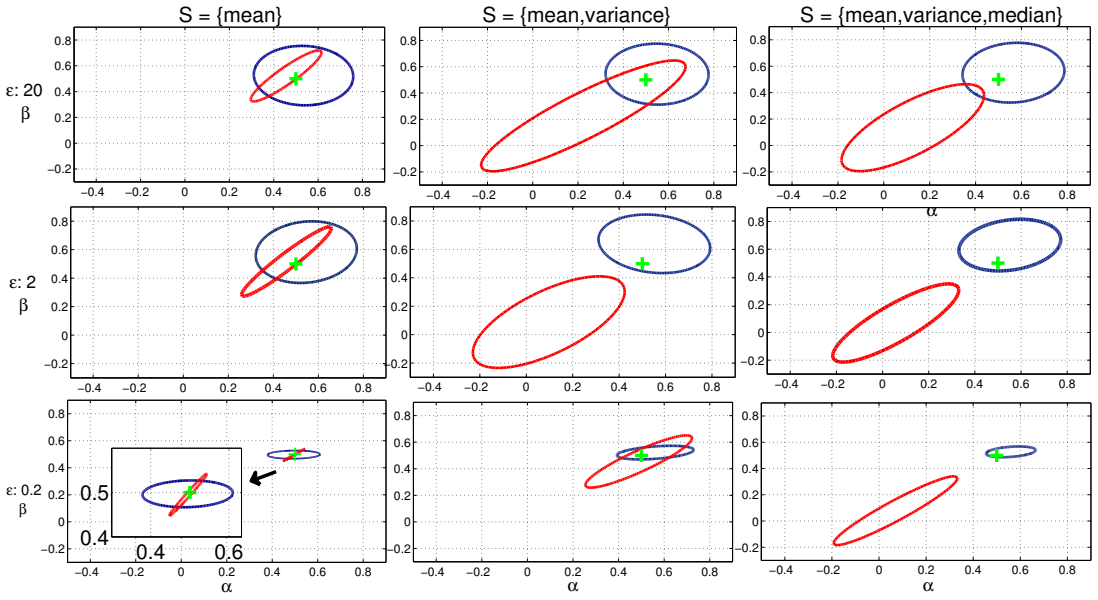


FIGURE A.21: Illustrations of the the posterior distribution $p(\boldsymbol{\theta}|\mathbf{X})$ from ABC-rejection and ABC-regression. The panels at *columns* show $p(\boldsymbol{\theta}|\mathbf{X})$ under three combinations of summary statistics: $\mathbf{s} = \{\text{mean}\}$, $\{\text{mean}, \text{variance}\}$ and $\{\text{mean}, \text{variance}, \text{median}\}$. The panels at *rows* denote the estimations under three tolerance levels: $\epsilon = \{20,\ 2,\ 0.2\}$. The results from ABC-rejection are shown in *(blue ellipse)* and *(red ellipse)* contours denote ABC-regression, point of the true parameter values is indicated as the *green cross*.

TABLE A.1: Results of all combinations in the single unknown case

Parameter						State			Success			Clear Advantage of PF
K_s	K_u	K_d	α_0	α_d	α_s	D_t	S_t	U_f	EKF	UKF	PF	
?	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	No
✓	?	✓	✓	✓	✓	✓	✓	✓	×	✓	✓	Yes
✓	✓	?	✓	✓	✓	✓	✓	✓	×	✓	✓	Yes
✓	✓	✓	?	✓	✓	✓	✓	✓	✓	✓	✓	No
✓	✓	✓	✓	?	✓	✓	✓	✓	✓	✓	✓	Yes
✓	✓	✓	✓	✓	?	✓	✓	✓	×	✓	✓	No
?	✓	✓	✓	✓	✓	?	✓	✓	✓	✓	✓	No
✓	?	✓	✓	✓	✓	?	✓	✓	×	✓	✓	Yes
✓	✓	?	✓	✓	✓	?	✓	✓	×	✓	✓	No
✓	✓	✓	?	✓	✓	?	✓	✓	✓	✓	✓	No
✓	✓	✓	✓	?	✓	?	✓	✓	✓	✓	✓	No
✓	✓	✓	✓	✓	?	?	✓	✓	×	✓	✓	No
?	✓	✓	✓	✓	✓	✓	?	✓	✓	✓	✓	Yes
✓	?	✓	✓	✓	✓	✓	?	✓	×	×	✓	Yes
✓	✓	?	✓	✓	✓	✓	?	✓	×	✓	✓	Yes
✓	✓	✓	?	✓	✓	✓	?	✓	×	×	✓	Yes
✓	✓	✓	✓	?	✓	✓	?	✓	✓	✓	✓	No
✓	✓	✓	✓	✓	?	✓	?	✓	×	✓	✓	Yes
?	✓	✓	✓	✓	✓	✓	✓	?	✓	✓	✓	No
✓	?	✓	✓	✓	✓	✓	✓	?	×	✓	✓	Yes
✓	✓	?	✓	✓	✓	✓	✓	?	×	✓	✓	No
✓	✓	✓	?	✓	✓	✓	✓	?	✓	✓	✓	No
✓	✓	✓	✓	?	✓	✓	✓	?	✓	✓	✓	No
✓	✓	✓	✓	✓	?	✓	✓	?	×	✓	✓	No
?	✓	✓	✓	✓	✓	?	?	✓	×	×	✓	Yes
✓	?	✓	✓	✓	✓	?	?	✓	×	×	✓	Yes
✓	✓	?	✓	✓	✓	?	?	✓	×	✓	✓	Yes
✓	✓	✓	?	✓	✓	?	?	✓	×	✓	✓	Yes
✓	✓	✓	✓	?	✓	?	?	✓	✓	✓	✓	No
✓	✓	✓	✓	✓	?	?	?	✓	×	✓	✓	Yes
?	✓	✓	✓	✓	✓	?	✓	?	✓	✓	✓	No
✓	?	✓	✓	✓	✓	?	✓	?	×	×	✓	Yes
✓	✓	?	✓	✓	✓	?	✓	?	×	✓	✓	No
✓	✓	✓	?	✓	✓	?	✓	?	×	✓	✓	No
✓	✓	✓	✓	?	✓	?	✓	?	×	×	✓	Yes
✓	✓	✓	✓	✓	?	?	✓	?	×	✓	✓	No
?	✓	✓	✓	✓	✓	✓	?	?	✓	×	✓	Yes
✓	?	✓	✓	✓	✓	✓	?	?	×	×	✓	Yes
✓	✓	?	✓	✓	✓	✓	?	?	×	✓	✓	No
✓	✓	✓	?	✓	✓	✓	?	?	×	×	✓	Yes
✓	✓	✓	✓	?	✓	✓	?	?	✓	✓	✓	No
✓	✓	✓	✓	✓	?	✓	?	?	×	✓	✓	No

TABLE A.2: Clear advantage of algorithms caused by sampling interval

sampling time (min)	EKF	UKF	PF
0.1	✓	×	×
0.25	×	×	✓
0.5	×	×	×
1	×	×	×

Appendix B

Details of EM and RMHMC

In this Appendix, we introduce the algorithmic details of Expectation-Maximization (EM) and Riemann Manifold Hamiltonian Monte Carlo (RMHMC). Even though either of these algorithms have not been directly used in our work, however, this introduction is a solid complement of literature review presented in chapter 2. Two corresponding examples are also shown.

B.1 Expectation Maximization (EM) method

We start with the joint likelihood, $p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$, of observables $\mathbf{X}$ and hidden variable $\mathbf{Z}$ parameterized by $\boldsymbol{\theta}$. The E-step is to calculate the expectation of log-joint likelihood w.r.t to the posterior distribution of the hidden variables given the old values of parameters $\boldsymbol{\theta}^{\text{old}}$. This expectation is denoted as $\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}})$, given as

$$\begin{aligned}\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) &= \mathbb{E}_{p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\text{old}})}[\ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})], \\ &= \int_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\text{old}}) \ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) d\mathbf{Z}. \end{aligned} \tag{B.1}$$

Quite straightforward, the M-step is to maximize the $\mathcal{Q}(\cdot)$ so that obtain the updated parameters estimates

$$\boldsymbol{\theta}^{\text{new}} = \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}}{\text{argmax}}\ \mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}). \tag{B.2}$$

The general EM algorithm can estimate parameter from an initial guess through the cycling iterations, and the procedure can be summarized as the Algorithm 15.

Algorithm 15 Expectation-Maximization method

1. Input initial guess of parameter $\boldsymbol{\theta}_0$ and iteration number I_{EM}.
   Repeat following steps until the convergence criterion is satisfied:
2. **E-step** Evaluate the $\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\mathrm{old}})$ following equation B.1.
3. **M-step** Propose the estimate of parameter: $\boldsymbol{\theta}^{\mathrm{new}} = \mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\mathrm{old}})$.
4. Update the log likelihood $\ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}^{\mathrm{new}})$ and set $\boldsymbol{\theta}^{\mathrm{old}} = \boldsymbol{\theta}^{\mathrm{new}}$
5. Output $\boldsymbol{\theta}^{\mathrm{new}}$.

Example 2.9 *Consider a Gaussian mixture model as an illustrative example (Bishop, 2006), which has the joint likelihood as*

$$\begin{aligned} p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) &= p(\mathbf{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) p(\mathbf{Z}|\boldsymbol{\pi}) \\ &= \prod_{n=1}^{N} \prod_{k=1}^{K} \pi_k^{z_{nk}} \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Lambda}^{-1})^{z_{nk}}, \end{aligned} \tag{B.3}$$

where $\mathbf{x}_n$ is the observed data, z_{nk} is the latent variable of system and implies if the data point $\mathbf{x}_n$ belongs to the kth Gaussian component. As z_{nk} is a binary indicator, thus $\sum_{k=1}^{K} z_{nk} = 1$. π_k is known as the mixing coefficients and explains how the mixture Gaussian distribution is proportionally constructed by the individual component, holding the property $\sum_{k=1}^{K} \pi_k$. $\boldsymbol{\pi}_k$ and $\boldsymbol{\Sigma}_k$ are the mean and precision matrix of kth Gaussian component.

The direct calculation of the complete-data likelihood is intractable, since variables $\mathbf{Z}$ are hidden in system. Therefore, in the E-step, the expected value of $\mathbf{Z}$ under its posterior distribution is used, instead of the expectation of complete-data likelihood. This expectation of each z_{nk} can be described as

$$\begin{aligned} \mathbb{E}_{p(z_{nk}|\mathbf{x}_n)}[z_{nk}] &= \frac{z_{nk} p(z_{nk}|\mathbf{x}_n)}{\sum_{j=1}^{j=K} p(z_{nj}|\mathbf{x}_n)} \\ &= \frac{z_{nk} p(\mathbf{x}_n|z_{nk}) p(z_{nk})}{\sum_{j=1}^{j=K} p(\mathbf{x}_n|z_{nj}) p(z_{nj})} \\ &= \frac{z_{nk} [\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)]^{z_{nk}}}{\sum_{j=1}^{j=K} [\pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)]^{z_{nj}}} \\ &= \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{j=K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \\ &= \gamma(z_{nk}), \end{aligned} \tag{B.4}$$

where $\gamma(z_{nk})$ is known as the responsibility that explains the contribution of kth Gaussian component to the observation $\mathbf{x}_n$. By making use of the responsibility $\gamma(z_{nk})$, the expectation of the complete-data log likelihood w.r.t the latent variables $\mathbf{Z}$ is given by

$$\mathbb{E}_{\mathbf{Z}}[\ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})] = \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(z_{nk}) \{\ln \pi_k + \ln \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\}. \tag{B.5}$$

In the M-step, with combined use of the multivariate Gaussian distribution and setting the derivative of log likelihood in equation B.5 w.r.t the mean $\boldsymbol{\mu}_k$ to zero, we can derive the equation given as following

$$\begin{aligned} 0 &= -\gamma(z_{nk})\boldsymbol{\Sigma}_k(\mathbf{x}_n - \boldsymbol{\mu}_k) \\ &= -\sum_{n=1}^{N} \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{j=K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \boldsymbol{\Sigma}_k(\mathbf{x}_n - \boldsymbol{\mu}_k), \end{aligned} \tag{B.6}$$

therefore, the mean $\boldsymbol{\mu}_k$ can be updated by

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{j=K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \mathbf{x}_n, \text{ where } N_k = \sum_{n=1}^{N} \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{j=K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}. \tag{B.7}$$

Similarly, we can obtain $\boldsymbol{\Sigma}_k$ and π_k as given

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^{N} \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{j=K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^{\mathrm{T}} \tag{B.8}$$

$$\pi_k = \frac{N_k}{N}. \tag{B.9}$$

We illustrate the EM algorithm by applying to a 2-dimensional Gaussian mixture example, in which 600 data points are generated by using command `mvnrnd` *in* `MATLAB` *where the mixing coefficients vector is defined as $\boldsymbol{\pi} = [0.2\ 0.4\ 0.4]$. The mean and covariance matrix for each Gaussian distribution are defined as*

$$\begin{aligned} &\boldsymbol{\mu} = \begin{bmatrix} 0 & 2 & 5 \\ -0.5 & 3 & 5 \end{bmatrix}, \\ &\boldsymbol{\Sigma} = [\boldsymbol{\Sigma}_1; \boldsymbol{\Sigma}_2; \boldsymbol{\Sigma}_3], \text{ where } \boldsymbol{\Sigma}_1 = \begin{bmatrix} 1 & 1.5 \\ 1.5 & 3 \end{bmatrix}; \boldsymbol{\Sigma}_2 = \begin{bmatrix} 0.3 & 0.5 \\ 0.5 & 4 \end{bmatrix}; \boldsymbol{\Sigma}_3 = \begin{bmatrix} 5 & 1.5 \\ 1.5 & 0.6 \end{bmatrix}, \\ &\boldsymbol{\pi} = [0.2, 0.4, 0.4]. \end{aligned} \tag{B.10}$$

Figure B.1 the results from EM algorithm applying to this artificial dataset at the different iterations. As shown in graphs, when the mixing coefficients are known, the arbitrary initial guess of mean and covariance matrix can eventually converge to their true values after 13 iterations.

B.2 Riemann manifold Hamiltonian Monte Carlo (RMHMC) method

In chapter 2, we have discussed two popular MCMC methods, *i.e.*, *Metropolis-Hastings* and *Gibbs methods*, where we have shown that these two algorithms are the high-

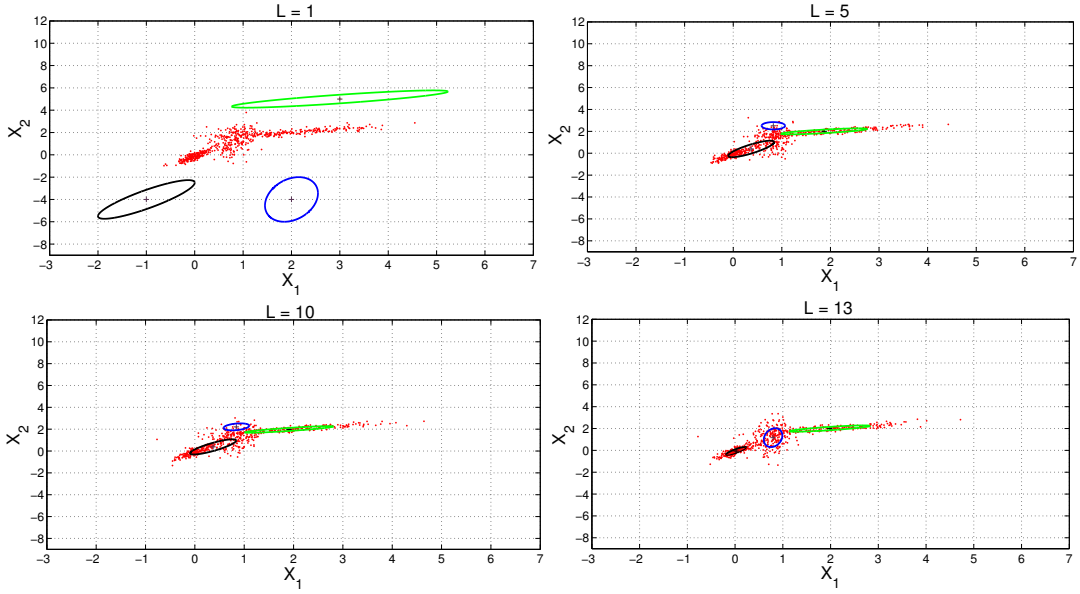


FIGURE B.1: Illustration of EM algorithm for estimating the means of a Gaussian mixture model. Sample points drawn from the joint distribution $p(\mathbf{X}, \mathbf{Z})$ are shown as the *red dots*. The component Gaussians differentiated by EM are shown as the *green*, *blue* and *purple ellipse* circles. In the first iteration, the initial guesses are arbitrarily set to be far away from samples. EM algorithm finally achieves to the convergence after 13 iterations.

demanding algorithmic tuning methods. Specifically, a large movement made in each MCMC step (covariance matrix of Markov transition kernel) offers higher chance for sample to escape from the trapped region, on the other hand, the sample is more likely to be rejected sacrificing on the acceptance rate if it is in a region of high probability. In contrast, a small move increases the acceptance rate, however, more iterations are required to explore the entire space and the retrieved samples are highly correlated.

The information geometry was combined with MCMC to solve this difficulty, in which the moves of MCMC samples are driven by the gradient of dynamics. (Duane et al., 2011) proposed the first genuine physics combined MCMC, known as *Hamiltonian Monte Carlo* (HMC), which adopted a Hamiltonian dynamical system to facilitate the transition of sample. Geometric information adds to the appeal of HMC by guiding its exploration to enhance its efficiency. Nevertheless, from the empirical investigation (Neal, 2010), such method still desperately relies on the tuning process to address the inference problem, especially for the systems in highly dimensional. An advanced method has recently been invented by Girolami and Calderhead (2011) , namely *Riemann manifold Hamiltonian Monte Carlo method* (RMHMC), which considerably enhances the effectiveness of MCMC method by utilizing the geometry information in an automatic manner. RMHMC is briefly introduced below.

Quite similar to the previous presented case, of interest is to sample from a distribution $p(\mathbf{x})$ with the variables vector $\mathbf{x} \in \mathbb{R}^D$ and denote the logarithm of $p(\mathbf{x})$ as $\mathcal{L}(\mathbf{x})$. An auxiliary variable $\mathbf{p} \in \mathbb{R}^D$ with distribution $p(\mathbf{p}) \sim \mathcal{N}(0, \mathbf{M})$ is further employed, then

the negative joint log-density of $p(\mathbf{x}, \mathbf{p})$ can be written as

$$H(\mathbf{x}, \mathbf{p}) = -\mathcal{L}(\mathbf{x}) + \frac{1}{2}\log((2\pi)^D|\mathbf{M})|) + \frac{1}{2}\mathbf{p}^T\mathbf{G}(\mathbf{x})^{-1}\mathbf{p}. \tag{B.11}$$

From the original literature of HMC (Duane et al., 2011), $H(\mathbf{x}, \mathbf{p})$ can be regarded as the Hamiltonian dynamics, consisting of the sum of a potential energy function $-\mathcal{L}(\mathbf{x})$ at the position $\mathbf{x}$, a kinetic energy term $\frac{1}{2}\mathbf{p}^T\mathbf{G}(\mathbf{x})^{-1}\mathbf{p}$ with the auxiliary variable $\mathbf{p}$ (generally called momentum variable) and a mass matrix $\mathbf{M}$. Generally, the capability of HMC is driven by the mass matrix, and as the empirical suggestion, the identity matrix is often used as $\mathbf{M}$ (Girolami and Calderhead, 2011). Unsurprisingly, if the dimensionality of $\mathbf{x}$ is high, HMC will perform poorly and finding an appropriate $\mathbf{M}$ becomes somehow impossible.

Girolami and Calderhead (2011) proposed the advanced HMC by adaptively adjusting the transition of $\mathbf{x}$ based on the current position of the state $\mathbf{x}$. Specifically, the momentum variable $\mathbf{p}$ is distributed as a function of state $\mathbf{x}$, that is $\mathbf{p} \sim \mathcal{N}(0, \mathbf{G}(\mathbf{x}))$. And $p(\mathbf{x})$ is defined on a Riemann manifold instead of Euclidean space, in a way the mass matrix $\mathbf{G}(\mathbf{x})$ is possible to be used for a position-specific metric tensor. Consider an observations $\mathbf{y}$, the latent variables $\mathbf{x}$ and the joint density $p(\mathbf{y}, \mathbf{x})$, then the expected Fisher information matrix is used to define the position-specific metric:

$$\begin{aligned}\mathbf{G}(\mathbf{x})_{i,j} &= -\mathbb{E}_{p(\mathbf{y}|\mathbf{x})}[\frac{\partial^2}{\partial x_i \partial x_j}\log\, p(\mathbf{y}|\mathbf{x})], \\ &= \mathbb{E}_{p(\mathbf{y}|\mathbf{x})}[\frac{\partial}{\partial x_i}\log\, p(\mathbf{y}|\mathbf{x})\frac{\partial}{\partial x_j}\log\, p(\mathbf{y}|\mathbf{x})].\end{aligned} \tag{B.12}$$

The concept of utilizing the expected Fisher information matrix as the metric tensor on Riemann manifold was first derived by Rao (1945), and which has been intensively employed in the modern statistical inference (Murray and Rice, 1993; Amari and Nagaoka, 2000).

With combined use of the expected Fisher information matrix, the Hamiltonian dynamics can be given as

$$\frac{dx_i}{d\tau} = \frac{\partial \boldsymbol{H}}{\partial p_i} = \{\mathbf{G}(\mathbf{x})^{-1}\mathbf{p}\}_i, \tag{B.13}$$

$$\frac{dp_i}{d\tau} = -\frac{\partial \boldsymbol{H}}{\partial x_i} = \frac{\partial \mathcal{L}}{\partial x_i} - \frac{1}{2}\mathrm{tr}(\mathbf{G}(\mathbf{x})^{-1}\frac{\partial \mathbf{G}(\mathbf{x})}{\partial x_i}) + \frac{1}{2}\mathbf{G}(\mathbf{x})^{-1}\frac{\partial \mathbf{G}(\mathbf{x})}{\partial x_i}\mathbf{G}(\mathbf{x})^{-1}\mathbf{p}. \tag{B.14}$$

The partial differential equations are solved by the generalized leapfrog integrator. Additionally, as the constant mass matrix $\mathbf{M}$ is replaced by the position-specific metric tensor $\mathbf{G}(\mathbf{x})$, solutions of equations (B.13) - (B.14) provide the way to automatically move the state and momentum variable $(\mathbf{x}, \mathbf{p}) \rightarrow (\mathbf{x}^*, \mathbf{p}^*)$ with remaining the proper-

ties of volume preservation and reversibility:

$$\mathbf{p}(\tau+\frac{\varepsilon}{2}) = \mathbf{p}(\tau) - \frac{\varepsilon}{2}\nabla_{\mathbf{x}}\boldsymbol{H}(\mathbf{x}(\tau), \mathbf{p}(\tau+\frac{\varepsilon}{2})), \tag{B.15}$$

$$\mathbf{x}(\tau+\varepsilon) = \mathbf{x}(\tau) + \frac{\varepsilon}{2}(\nabla_{\mathbf{p}}\boldsymbol{H}(\mathbf{x}(\tau), \mathbf{p}(\tau+\frac{\varepsilon}{2})) + \nabla_{\mathbf{p}}\boldsymbol{H}(\mathbf{x}(\tau+\varepsilon), \mathbf{p}(\tau+\frac{\varepsilon}{2}))), \tag{B.16}$$

$$\mathbf{p}(\tau+\varepsilon) = \mathbf{p}(\tau+\frac{\varepsilon}{2}) - \frac{\varepsilon}{2}\nabla_{\mathbf{x}}\boldsymbol{H}(\mathbf{x}(\tau+\tau), \mathbf{p}(\tau+\frac{\varepsilon}{2})). \tag{B.17}$$

Consequently, to interpret the RMHMC conveniently, the updates of state and auxiliary variables are rewritten in terms of Gibbs sampling as

$$\mathbf{p}^*|\mathbf{x}^{t-1} \sim p(\mathbf{p}^*|\mathbf{x}^{t-1}) = \mathcal{N}(\mathbf{p}^*|0, \mathbf{G}(\mathbf{x}^{t-1})), \tag{B.18}$$

$$\mathbf{x}^*|\mathbf{p}^* \sim p(\mathbf{x}^*|\mathbf{p}^*). \tag{B.19}$$

Similarly, the pair $\mathbf{x}^*, \mathbf{p}^*$ is accepted or rejected with the probability

$$\mathrm{A}(\mathbf{x}^*, \mathbf{x}^{t-1}) = \min(1, \exp(-H(\mathbf{x}^*, \mathbf{p}^*) + H(\mathbf{x}^{t-1}, \mathbf{p}^{t-1}))). \tag{B.20}$$

Example 2.12 *Consider a dynamical system which is formed as (Girolami and Calderhead, 2011)*

$$p(\mathbf{x}|\boldsymbol{\theta}) \sim \mathcal{N}(\mathrm{g}(\boldsymbol{\theta}), \boldsymbol{\Sigma}_{\mathbf{x}}^{-1}),\ \mathrm{g}(\boldsymbol{\theta}) = \theta_1 + \theta_2^2, \tag{B.21}$$

$$p(\boldsymbol{\theta}) \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1}), \tag{B.22}$$

where the diagonal elements of covariance matrix $\boldsymbol{\Sigma}_{\mathbf{x}}^{-1}$ is set to 2 and $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1}$ is used as an identity matrix. HMC and RMHMC are employed to estimate the probability distribution of parameter given the sequence of observations $p(\boldsymbol{\theta}|\mathbf{x})$. For launching HMC, the mass matrix $\mathbf{M}$ is set to be symmetric and whose diagonal elements are generated from the zero mean Gaussian distribution and all off-diagonal entries are zero. Figure B.2(a) demonstrates the results from HMC and RMHMC. In this case, the joint probability density $p(\mathbf{x}, \boldsymbol{\theta})$ is shaped as a banana by considering the mean as one, i.e. $\mathrm{g}(\boldsymbol{\theta}) = 1$. It can be easily seen that RMHMC outperforms HMC in term of efficiency, however, both methods are able to thoroughly visit the space under this simple scenario. In order to differ the ability of handling the complex circumstance between these two methods, $p(\mathbf{x}, \boldsymbol{\theta})$ is formed as a multimodal distribution via taking $\theta_1 = 1$ and $\theta_2 = 2$. No sample is accepted by HMC with remaining the same algorithmic settings, and Figure B.2(b) shows that RMHMC precisely estimates the posterior distribution.

B.3 Derivation of extended Kalman filter

The Kalman filter (KF) relying on the Gaussian approximation provides an estimator, through recursive minimization of the covariance of approximation error, to infer the

hidden state efficiently. Ideally, the original KF algorithm would like to tackle the linear and Gaussian problem. Given the nonlinearity of biological systems, superior performance of KF cannot be expected and this limitation motivates the *extended Kalman filter* (EKF).

Similar to KF, EKF consists of two steps: prediction and correction. In the prediction step, priors estimates of states and error covariance matrix $\mathbf{\Sigma}$ at the current time instant, are produced by taking the estimates from the previous time instant through the transition model. In correction step, in incorporating the current observation, the priors are refined to propose the posterior estimates.

We describe the procedure of EKF in a more rigorous mathematical framework. Let us assume the posterior estimation at time t follows a Gaussian distribution with mean μ_{x_t} and covariance $\mathbf{\Sigma}_{x_t}$, i.e. $p(\mathbf{x}_t|\mathbf{Y}_t) \sim \mathcal{N}(\boldsymbol{\mu}_{x_t}, \mathbf{\Sigma}_{x_t})$. From the perspective of Kalman filtering, the mean $\boldsymbol{\mu}_{x_t}$ is seen as the difference between the real state $\mathbf{x}_t$ and the inferred variable $\hat{\mathbf{x}}_{t|t}$, while covariance matrix $\mathbf{\Sigma}_{x_t}$ is interpreted as the confidence in accuracy of obtained estimations, denoted as $\boldsymbol{P}_{t|t}$. Consequently, taking the Kalman filter expression, the posterior estimation can be further written as

$$p(\mathbf{x}_t|\mathbf{Y}_t) \sim \mathcal{N}(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t}, \boldsymbol{P}_{t|t}), \tag{B.23}$$

where $\mathbf{Y}_t = \{\mathbf{y}_1, \ldots, \mathbf{y}_t\}$ is all observations up to time t. In order to compute the prior estimate at next time step $t+1$, from the Bayes law, the prior estimation (also the conditional probability density function of $\mathbf{x}_{t+1}$ by given $\mathbf{Y}_t$) can be formulated as

$$p(\mathbf{x}_{t+1}|\mathbf{Y}_t) = \int p(\mathbf{x}_{t+1}|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{Y}_t)d\mathbf{x}_t \tag{B.24}$$

$$= \int p(\mathbf{x}_{t+1} - f(\mathbf{x}_t))p(\mathbf{x}_t|\mathbf{Y}_t)d\mathbf{x}_t. \tag{B.25}$$

The extension of equation B.24 to equation B.25 due to the nonlinearity of the system of interest is given as

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t) + \omega_t^x, \tag{B.26}$$

where ω_t^x is the Gaussian noise with zero mean and $\boldsymbol{Q}_t$ covariance. $p(\mathbf{x}_{t+1}|\mathbf{x}_t)$ in equation B.25 is therefore expressed as

$$\begin{aligned} p(\mathbf{x}_{t+1}|\mathbf{x}_t) &= p(\mathbf{x}_{t+1} - f(\mathbf{x}_t)) \\ &= \frac{1}{(2\pi)^{n/2}\left|\boldsymbol{Q}_t\right|^{1/2}} \exp\{-\frac{1}{2}\left[\mathbf{x}_{t+1} - f(\mathbf{x}_t)\right]^{\mathrm{T}} \boldsymbol{Q}_t^{-1} \left[\mathbf{x}_{t+1} - f(\mathbf{x}_t)\right]\}. \end{aligned} \tag{B.27}$$

Since the nonlinearity caused by $f(\mathbf{x}_t)$ violates the Gaussian assumption of equation B.27. The innovation made by EKF is to linearize the system dynamics $f(\mathbf{x}_t)$ in the presence of *Taylor series expansion*. In this linearization, only the first-order deriva-

tive terms around the posterior estimation $\hat{\mathbf{x}}_{t|t}$ remain while the higher-order terms are neglected, such process can be denoted as

$$\begin{aligned} f(\mathbf{x}_t) &= f(\hat{\mathbf{x}}_{t|t}) + \nabla f_x|_{\hat{\mathbf{x}}_{t|t}} \cdot (\mathbf{x}_t - \hat{\mathbf{x}}_{t|t}) \\ &= \underbrace{f(\hat{\mathbf{x}}_{t|t}) - \nabla f_x|_{\hat{\mathbf{x}}_{t|t}} \cdot \hat{\mathbf{x}}_{t|t}}_{s_t} + \nabla f_x|_{\hat{\mathbf{x}}_{t|t}} \cdot \mathbf{x}_t, \end{aligned} \tag{B.28}$$

where $\nabla f_x|_{\hat{\mathbf{x}}_{t|t}}$ is the Jacobian matrix of system dynamics $f(\cdot)$, given as

$$\nabla f_x|_{\hat{\mathbf{x}}_{t|t}} = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_{t|t}}. \tag{B.29}$$

Taking this linearization to the nonlinear system model, equation B.26 can be rewritten as

$$\begin{aligned} \mathbf{x}_{t+1} &= f(\mathbf{x}_t) + \omega_t^x \\ &= \nabla f_x|_{\hat{\mathbf{x}}_{t|t}} \cdot \mathbf{x}_t + \underbrace{f(\hat{\mathbf{x}}_{t|t}) - \nabla f_x|_{\hat{\mathbf{x}}_{t|t}} \cdot \hat{\mathbf{x}}_{t|t}}_{s_t} + \omega_t^x \\ &= \nabla f_x|_{\hat{\mathbf{x}}_{t|t}} \cdot \mathbf{x}_t + s_t + \omega_t^x. \end{aligned} \tag{B.30}$$

Accordingly, the transition of state variable $\mathbf{x}_t$ is updated linearly. Substituting equation B.28 and equation B.23 into equation B.25, the prior estimation can be rewritten as

$$\begin{aligned} p(\mathbf{x}_{t+1}|\mathbf{Y}_t) &= \int p(\mathbf{x}_{t+1}|\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{Y}_t) d\mathbf{x}_t \\ &= \int p(\mathbf{x}_{t+1} - f(\mathbf{x}_t)) p(\mathbf{x}_t|\mathbf{Y}_t) d\mathbf{x}_t \\ &= \int \mathcal{N}(\mathbf{x}_{t+1} - \nabla f_x|_{\hat{\mathbf{x}}_{t|t}} \cdot \mathbf{x}_t - s_t, \boldsymbol{Q}_t) \cdot \mathcal{N}(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t}, \boldsymbol{P}_{t|t}) d\mathbf{x}_t \end{aligned} \tag{B.31}$$

For the sake of simplicity, we adopt a variable transformation here

$$\mathbf{z}_t = \nabla f_x|_{\hat{\mathbf{x}}_{t|t}} \cdot \mathbf{x}_t. \tag{B.32}$$

Then the mean and covariance matrix (actually, since the covariance matrix is diagonal, therefore, such calculation is for variance) of this variable $\mathbf{z}_t$ are given as

$$\mathbb{E}[\mathbf{z}_t] = \mathbb{E}[\nabla f_x|_{\hat{\mathbf{x}}_{t|t}} \cdot \mathbf{x}_t] = \nabla f_x|_{\hat{\mathbf{x}}_{t|t}} \mathbb{E}[\mathbf{x}_t] = \nabla f_x|_{\hat{\mathbf{x}}_{t|t}} \cdot \hat{\mathbf{x}}_{t|t} \tag{B.33}$$

$$\begin{aligned} \mathrm{var}(\mathbf{z}_t) &= \mathbb{E}[\mathbf{z}_t \mathbf{z}_t^{\mathrm{T}}] = \mathbb{E}[(\nabla f_x|_{\hat{\mathbf{x}}_{t|t}} \cdot \mathbf{x}_t)(\nabla f_x|_{\hat{\mathbf{x}}_{t|t}} \cdot \mathbf{x}_t)^{\mathrm{T}}] \\ &= (\nabla f_x|_{\hat{\mathbf{x}}_{t|t}}) \mathbb{E}[\mathbf{x}_t \mathbf{x}_t^{\mathrm{T}}] (\nabla f_x|_{\hat{\mathbf{x}}_{t|t}})^{\mathrm{T}} \\ &= (\nabla f_x|_{\hat{\mathbf{x}}_{t|t}}) \boldsymbol{P}_{t|t} (\nabla f_x|_{\hat{\mathbf{x}}_{t|t}})^{\mathrm{T}} \end{aligned} \tag{B.34}$$

Expanding the distribution $\mathcal{N}(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t}, \boldsymbol{P}_{t|t})$ associated with the similar variable trans-

formation[1], we have

$$\begin{aligned}
p(\mathbf{x}_t|\mathbf{Y}_t) &= \mathcal{N}(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t}, \boldsymbol{P}_{t|t}) = \mathcal{N}(\widetilde{\mathbf{x}}_t - \widetilde{\hat{\mathbf{x}}}_{t|t}, \widetilde{\boldsymbol{P}}_{t|t}) \\
&= \frac{1}{(2\pi)^{n/2}\left|\widetilde{\boldsymbol{P}}_{t|t}\right|^{1/2}} \exp\{-\frac{1}{2}(\nabla f_x|_{\hat{\mathbf{x}}_{t|t}} \cdot \mathbf{x}_t - \nabla f_x|_{\hat{\mathbf{x}}_{t|t}} \cdot \hat{\mathbf{x}}_{t|t})^{\mathrm{T}} \\
&\qquad\qquad \times \widetilde{\boldsymbol{P}}_{t|t}^{-1}(\nabla f_x|_{\hat{\mathbf{x}}_{t|t}} \cdot \mathbf{x}_t - \nabla f_x|_{\hat{\mathbf{x}}_{t|t}} \cdot \hat{\mathbf{x}}_{t|t})\} \\
&= \frac{1}{(2\pi)^{n/2}\left|\widetilde{\boldsymbol{P}}_{t|t}\right|^{1/2}} \exp\{-\frac{1}{2}(\nabla f_x|_{\hat{\mathbf{x}}_{t|t}} \cdot \mathbf{x}_t - \nabla f_x|_{\hat{\mathbf{x}}_{t|t}} \cdot \hat{\mathbf{x}}_{t|t})^{\mathrm{T}} \\
&\qquad\qquad \times ((\nabla f_x|_{\hat{\mathbf{x}}_{t|t}})\boldsymbol{P}_{t|t}(\nabla f_x|_{\hat{\mathbf{x}}_{t|t}})^{\mathrm{T}})^{-1} \\
&\qquad\qquad \times (\nabla f_x|_{\hat{\mathbf{x}}_{t|t}} \cdot \mathbf{x}_t - \nabla f_x|_{\hat{\mathbf{x}}_{t|t}} \cdot \hat{\mathbf{x}}_{t|t})\} \\
&= \left|\nabla f_x|_{\hat{\mathbf{x}}_{t|t}}\right| \cdot \frac{1}{(2\pi)^{n/2}\left|(\nabla f_x|_{\hat{\mathbf{x}}_{t|t}})\boldsymbol{P}_{t|t}(\nabla f_x|_{\hat{\mathbf{x}}_{t|t}})^{\mathrm{T}}\right|^{n/2}} \\
&\times \exp\{-\frac{1}{2}(\nabla f_x|_{\hat{\mathbf{x}}_{t|t}} \cdot \mathbf{x}_t - \nabla f_x|_{\hat{\mathbf{x}}_{t|t}} \cdot \hat{\mathbf{x}}_{t|t})^{\mathrm{T}} \\
&\qquad \times ((\nabla f_x|_{\hat{\mathbf{x}}_{t|t}})\boldsymbol{P}_{t|t}(\nabla f_x|_{\hat{\mathbf{x}}_{t|t}})^{\mathrm{T}})^{-1}(\nabla f_x|_{\hat{\mathbf{x}}_{t|t}} \cdot \mathbf{x}_t - \nabla f_x|_{\hat{\mathbf{x}}_{t|t}} \cdot \hat{\mathbf{x}}_{t|t})\}. \quad \text{(B.35)}
\end{aligned}$$

Therefore, from this expression, we can claim that

$$\begin{aligned}
p(\mathbf{x}_t|\mathbf{Y}_t) &= \mathcal{N}(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t}, \boldsymbol{P}_{t|t}) \\
&= \left|\nabla f_x|_{\hat{\mathbf{x}}_{t|t}}\right| \mathcal{N}(\nabla f_x|_{\hat{\mathbf{x}}_{t|t}} \cdot \mathbf{x}_t - \nabla f_x|_{\hat{\mathbf{x}}_{t|t}} \cdot \hat{\mathbf{x}}_{t|t}, (\nabla f_x|_{\hat{\mathbf{x}}_{t|t}})\boldsymbol{P}_{t|t}(\nabla f_x|_{\hat{\mathbf{x}}_{t|t}})^{\mathrm{T}}). \quad \text{(B.36)}
\end{aligned}$$

Substituting equation B.36 into the prior estimation, then it can be rewritten as

$$\begin{aligned}
p(\mathbf{x}_{t+1}|\mathbf{Y}_t) &= \int \mathcal{N}(\mathbf{x}_{t+1} - \nabla f_x|_{\hat{\mathbf{x}}_{t|t}} \cdot \mathbf{x}_t - s_t, \boldsymbol{Q}_t) \\
&\qquad \times \mathcal{N}(\nabla f_x|_{\hat{\mathbf{x}}_{t|t}} \cdot \mathbf{x}_t - \nabla f_x|_{\hat{\mathbf{x}}_{t|t}} \cdot \hat{\mathbf{x}}_{t|t}, (\nabla f_x|_{\hat{\mathbf{x}}_{t|t}})\boldsymbol{P}_{t|t}(\nabla f_x|_{\hat{\mathbf{x}}_{t|t}})^{\mathrm{T}}) d(\nabla f_x|_{\hat{\mathbf{x}}_{t|t}} \cdot \mathbf{x}_t) \\
&= \mathcal{N}(\mathbf{x}_{t+1} - s_t, \boldsymbol{Q}_t) * \mathcal{N}(\mathbf{x}_{t+1} - \nabla f_x|_{\hat{\mathbf{x}}_{t|t}} \cdot \hat{\mathbf{x}}_{t|t}, (\nabla f_x|_{\hat{\mathbf{x}}_{t|t}})\boldsymbol{P}_{t|t}(\nabla f_x|_{\hat{\mathbf{x}}_{t|t}})^{\mathrm{T}}, \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(B.37)}
\end{aligned}$$

here $*$ represents the convolution of two distributions. After the calculation, the expression of prior estimation is given as

$$\begin{aligned}
p(\mathbf{x}_{t+1}|\mathbf{Y}_t) &= \mathcal{N}(\mathbf{x}_{t+1} - \nabla f_x|_{\hat{\mathbf{x}}_{t|t}} \cdot \hat{\mathbf{x}}_{t|t} - s_t, \boldsymbol{Q}_t + (\nabla f_x|_{\hat{\mathbf{x}}_{t|t}})\boldsymbol{P}_{t|t}(\nabla f_x|_{\hat{\mathbf{x}}_{t|t}})^{\mathrm{T}}) \\
&= \mathcal{N}(\mathbf{x}_{t+1} - f(\mathbf{x}_{t|t}), \boldsymbol{Q}_t + (\nabla f_x|_{\hat{\mathbf{x}}_{t|t}})\boldsymbol{P}_{t|t}(\nabla f_x|_{\hat{\mathbf{x}}_{t|t}})^{\mathrm{T}}). \quad \text{(B.38)}
\end{aligned}$$

We therefore are able to conclude that if $p(\mathbf{x}_t|\mathbf{Y}_t)$ is a Gaussian distribution with $\hat{\mathbf{x}_{t|t}}$ mean and $\boldsymbol{P}_{t|t}$ covariance matrix, then the prior estimation of state at the next time step $p(\mathbf{x}_{t+1}|\mathbf{Y}_t)$ can be delivered by making use of the linearization of system model $f(\cdot)$ around $\hat{\mathbf{x}_{t|t}}$. Given the equation B.38 and $\boldsymbol{Q}_t = 0$, the prior of state estimation and the

[1] $\widetilde{\mathbf{x}}_t = \nabla f_x|_{\hat{\mathbf{x}}_{t|t}} \cdot \mathbf{x}_t$, $\widetilde{\hat{\mathbf{x}}}_{t|t} = \nabla f_x|_{\hat{\mathbf{x}}_{t|t}} \cdot \hat{\mathbf{x}}_{t|t}$

error covariance matrix are given as

$$\hat{\mathbf{x}}_{t+1|t} = f(\mathbf{x}_{t|t}) \tag{B.39}$$

$$\boldsymbol{P}_{t+1|t} = (\nabla f_x|_{\hat{\mathbf{x}}_{t|t}})\boldsymbol{P}_{t|t}(\nabla f_x|_{\hat{\mathbf{x}}_{t|t}})^{\mathrm{T}}. \tag{B.40}$$

For the sake of simplicity, in the pseudo-code and the details of implementations, the Jacobian matrix ∇f_x is denoted as $\boldsymbol{F}$. With a similar derivation, the posterior estimation is produced by updating the prior associated with the observations at time $t+1$, which are given as

$$\boldsymbol{K}_{t+1} = \boldsymbol{P}_{t+1|t}\boldsymbol{H}_{t+1}^{\mathrm{T}}\left[\boldsymbol{H}_{t+1}\boldsymbol{P}_{t+1|t}\boldsymbol{H}_{t+1}^{\mathrm{T}} + \boldsymbol{R}_{t+1}\right]^{-1} \tag{B.41}$$

$$\hat{\mathbf{x}}_{t+1|t+1} = \hat{\mathbf{x}}_{t+1|t} + \boldsymbol{K}_{t+1} \cdot \{\mathbf{y}_{t+1} - h(\hat{\mathbf{x}}_{t+1|t})\} \tag{B.42}$$

$$\boldsymbol{P}_{t+1|t+1} = \{\boldsymbol{I} - \boldsymbol{K}_{t+1}\boldsymbol{H}_{t+1}\}\boldsymbol{P}(t+1|t), \tag{B.43}$$

where $\boldsymbol{H}_{t+1}$ is Jacobian matrix of observation function $h(\cdot)$ around the $\hat{\mathbf{x}}_{t+1|t}$, specified as

$$\boldsymbol{H}_{t+1} = \nabla h_x|_{\hat{\mathbf{x}}_{t+1|t}} = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_{t+1|t}}. \tag{B.44}$$

$\boldsymbol{K}_{t+1}$ is known as the Kalman gain, a measure of how much EKF should adjust the prior estimate in response to the new observation. Algorithmically, a small covariance $\boldsymbol{R}_{t+1}$ implies that new observations are the high quality, as a result of which, performing a relatively large update for posterior is reliable. In contrast, when $\boldsymbol{H}_{t+1}\boldsymbol{P}_{t+1|t}\boldsymbol{H}_{t+1}^{\mathrm{T}}$ is smaller than the $\boldsymbol{R}_{t+1}$, meaning the prior has already achieved a high accuracy and only a tiny refinement is caused by the new observation. The complete block of pseudo-code of EKF is given in Algorithm 9.

B.4 Derivation of sequential importance resampling

Particle filter (PF), one of the *sequential Monte Carlo* methods, allows for the online approximation of the target distribution via a set of randomly drawn 'particles'. This method plays a significant role in solving real-time problems, in which the data arrival is sequential. Moreover, the 'sequential' approach may well implying the evolution of PF focuses more on information from the recent past rather than the one from the distant past.

In the framework of PF, we are given information on the system of interest as

$$\textbf{initial distribution:} \quad p(\mathbf{x}_0) \quad \text{for } t = 0 \tag{B.45}$$

$$\textbf{transition model:} \quad p(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t-1}) \quad \text{for } t \geq 1 \tag{B.46}$$

$$\textbf{observation distribution:} \quad p(\mathbf{y}_t|\mathbf{x}_{0:t}, \mathbf{y}_{1:t-1}) \quad \text{for } t \geq 1. \tag{B.47}$$

Here we denote the states and the observations up to time t as $\mathbf{x}_{0:t} \triangleq \{\mathbf{x}_0, \ldots, \mathbf{x}_t\}$ and $\mathbf{y}_{1:t} \triangleq \{\mathbf{y}_1, \ldots, \mathbf{y}_t\}$, respectively. These expressions are here formulated in generality. Under the Markov assumption, the transition and observation models are simplified as $p(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t-1}) = p(\mathbf{x}_t|\mathbf{x}_{t-1})$ and $p(\mathbf{y}_t|\mathbf{x}_{0:t}, \mathbf{y}_{1:t-1}) = p(\mathbf{y}_t|\mathbf{x}_t)$. More details can be found out in section A.1.4 of chapter 2.

Of interest is to recursively estimate the *filtering distribution* $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ and the expectations of the posterior distribution $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$

$$\boldsymbol{I}(f_t) = \mathbb{E}_{p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})}[f_t(\mathbf{x}_{0:t})]. \tag{B.48}$$

Given N particles $\{\mathbf{x}_{0:t-1}^i\}_{i=1}^N$ at time $t-1$, which can precisely approximate the distribution $p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})$, then, PF paves a way to update N particles $\{\mathbf{x}_{0:t}^i\}_{i=1}^N$ for approximating the posterior $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$. In most cases, direct sampling from the posterior is impossible, thus, PF considers an alternative distribution for which sampling is possible, namely the proposal distribution $q(\mathbf{x}_{0:t})$, to help particles update from $\mathbf{x}_{0:t-1}$ to $\mathbf{x}_{0:t}$. This proposal distribution $q(\cdot)$ should be designed with care, due to its critical role in 'encouraging' the frequent selection of particles with important values.

More specifically, in a general SMC framework, the current particles $\{\mathbf{x}_{0:t-1}^i\}_{i=1}^N$ are used to compute the new set of particles through the proposal distribution, denoted as

$$q(\hat{\mathbf{x}}_{0:t}|\mathbf{y}_{1:t}) = \int q(\hat{\mathbf{x}}_{0:t}|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t}) p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1}) d\mathbf{x}_{0:t-1}. \tag{B.49}$$

Unfortunately, in most cases, this integral is intractable. In PF, a Markov approximation is adopted to deal this difficulty, and details are also provided in the section A.1.4 of chapter 2. Updating made by the proposal distribution is then defined as

$$q(\hat{\mathbf{x}}_{0:t}|\mathbf{y}_{1:t}) = q(\hat{\mathbf{x}}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t}) p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1}). \tag{B.50}$$

In PF, the importances of particles $\hat{\mathbf{x}}_{0:t}$ need to be weighted by the scheme given below

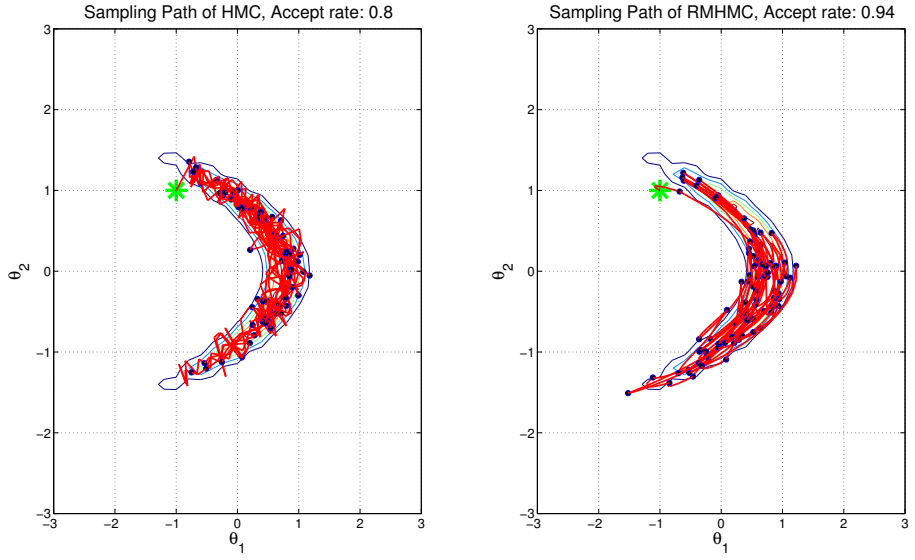
$$\begin{aligned} \boldsymbol{w}_t &= \frac{q(\hat{\mathbf{x}}_{0:t}|\mathbf{y}_{1:t})}{p(\hat{\mathbf{x}}_{0:t}|\mathbf{y}_{1:t})} \\ &= \frac{p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t})}{p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})} \times \frac{p(\hat{\mathbf{x}}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t})}{q(\hat{\mathbf{x}}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t})} \\ &\propto \frac{p(\mathbf{y}_t|\hat{\mathbf{x}}_t) p(\hat{\mathbf{x}}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t})}{q(\hat{\mathbf{x}}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t})}. \end{aligned} \tag{B.51}$$

Various PFs, such as auxiliary particle filter (Pitt and Shephard, 1999) and 'likelihood' particle filter (Arulampalam et al., 2002), have developed around different possible proposal distributions. In the generic PF introduced in this work, the optimal proposal distribution is used as

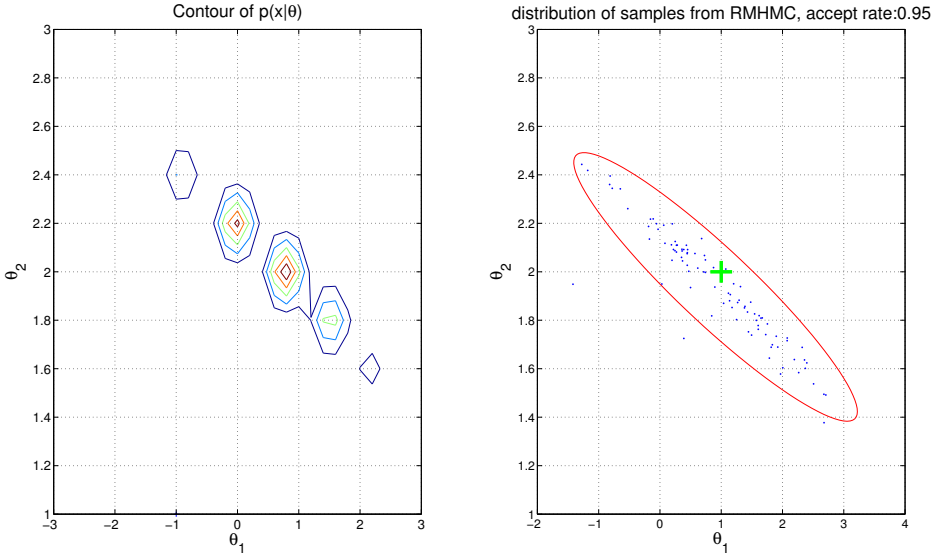
$$q(\hat{\mathbf{x}}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t}) = p(\hat{\mathbf{x}}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t-1}). \tag{B.52}$$

By doing so, the calculation of importance weights is simplified as

$$\boldsymbol{w}_t = p(\mathbf{y}_t|\hat{\mathbf{x}}_t). \tag{B.53}$$



(a)

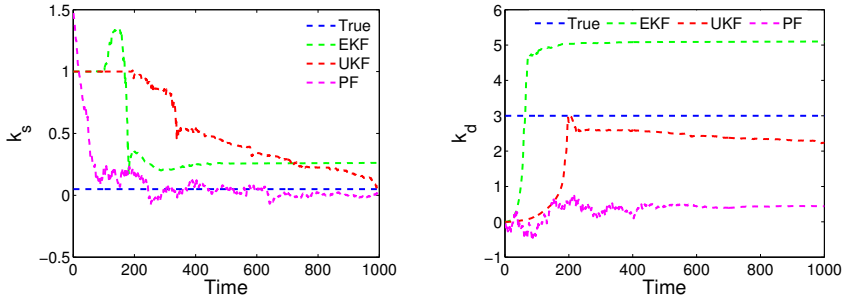


(b)

FIGURE B.2: (a): Illustrations of HMC (*left*) and RMHMC (*right*) for estimating the invariant mean of a Gaussian distribution shown in equation B.21. The trajectories in both cases represent 100 samples. The *banana* shape is the log joint density $p(\mathbf{x}, \boldsymbol{\theta})$. (b): Contour of the log joint density $p(\mathbf{x}, \boldsymbol{\theta})$ is shown in the *left* graph, in which $\theta_1 = 1$ and $\theta_2 = 2$. *Red ellipse* shown in *right* graph represents the distribution of samples obtained from RMHMC and *green* ('+') indicates the true values of $\boldsymbol{\theta}$. This implementation is modified from the code given by Professor Mark Girolami in EPSRC/RSS GRADUATE TRAINING PROGRAMME 2012.

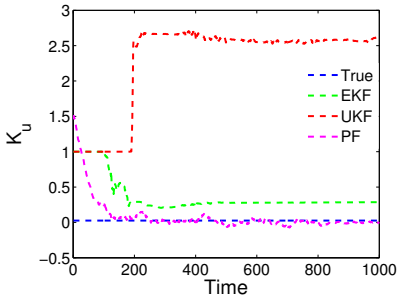
Appendix C

Supplementary graphs for Heat Shock study

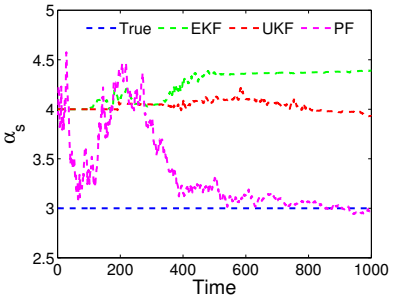


(a) K_s

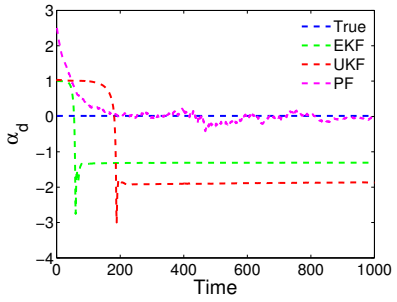
(b) K_d



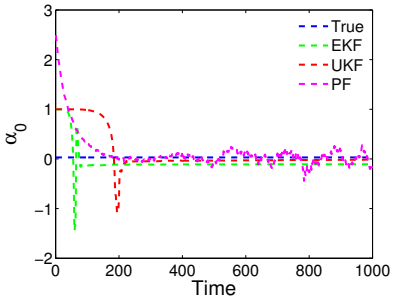
(c) K_u



(d) α_s

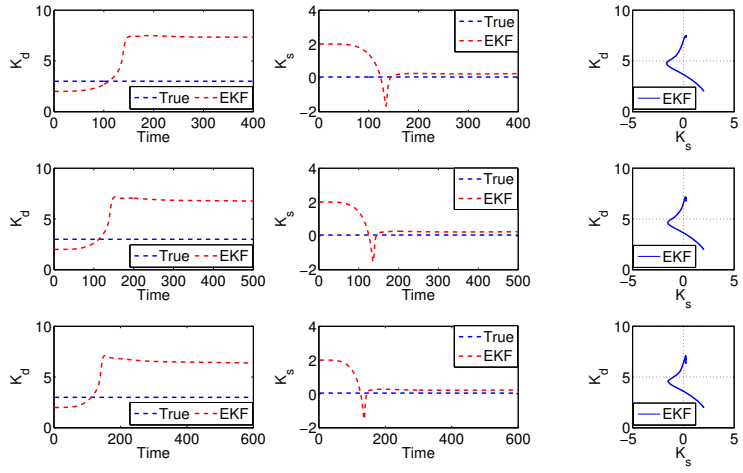


(e) α_d

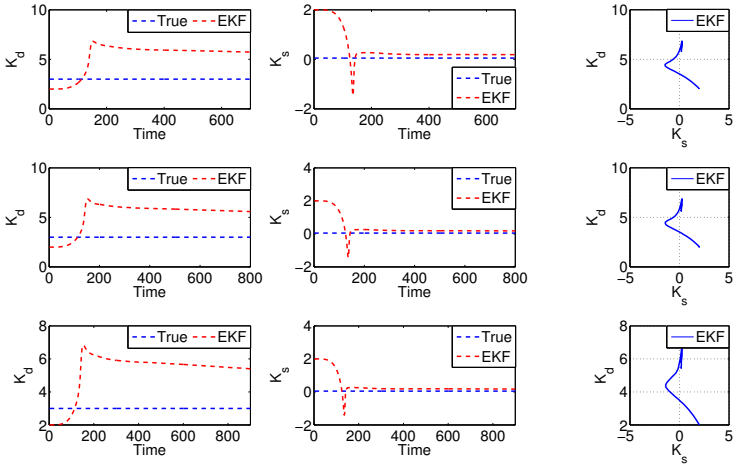


(f) α_0

FIGURE C.1: Simultaneous inference of all six model parameters from three noisy state observations.

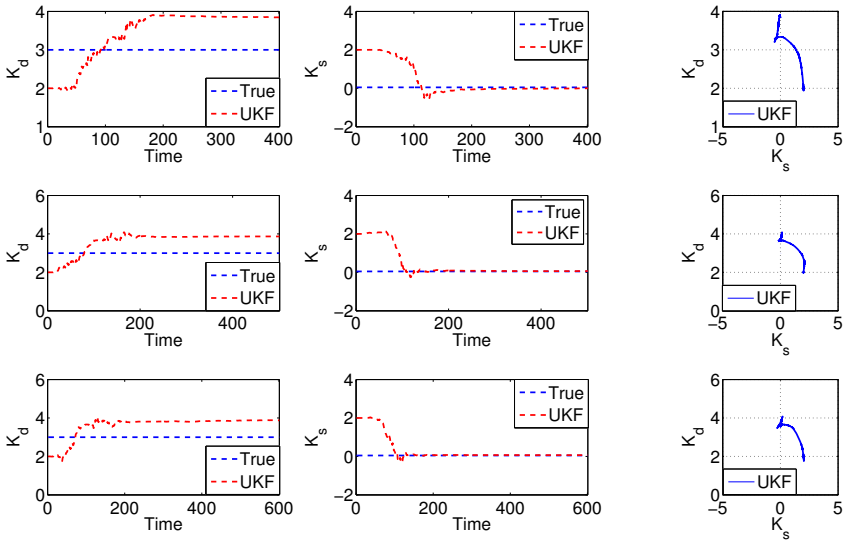


(a) 80s, 100s and 120s

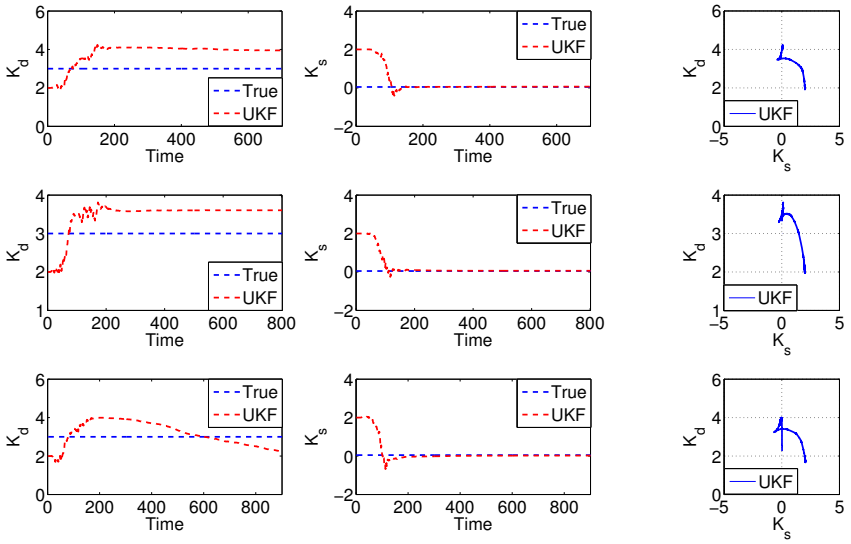


(b) 140s, 160s and 180s

FIGURE C.2: Top row : Estimation of K_s and K_d with 80s, 100s and 120s data length from EKF. Bottom row : Estimation of K_s and K_d with 140s, 160s and 180s data length from EKF. from EKF.

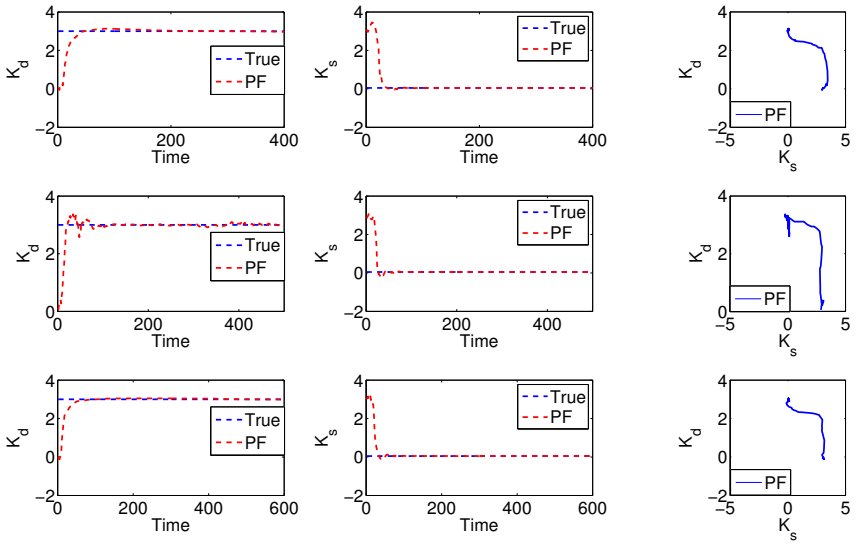


(a) 80s, 100s and 120s

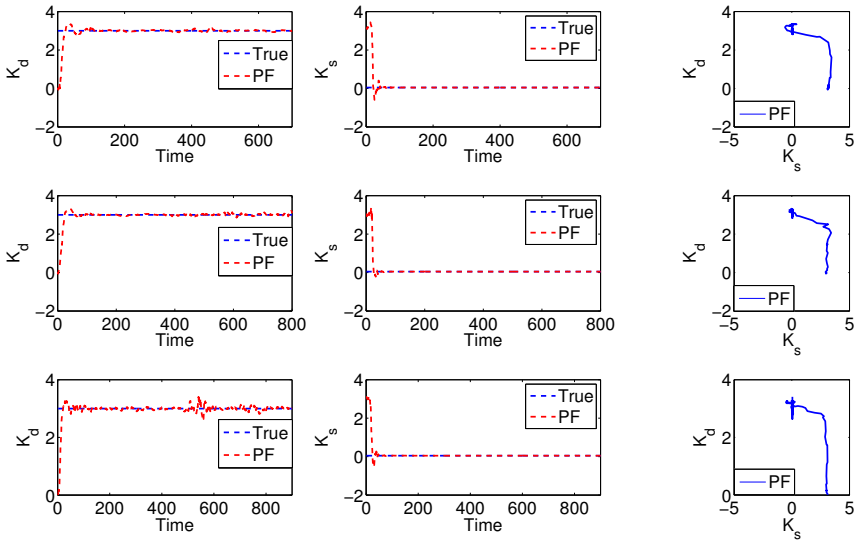


(b) 140s, 160s and 180s

FIGURE C.3: Top row : Estimation of K_s and K_d with 80s, 100s and 120s data length from UKF. Bottom row : Estimation of K_s and K_d with 140s, 160s and 180s data length from UKF.

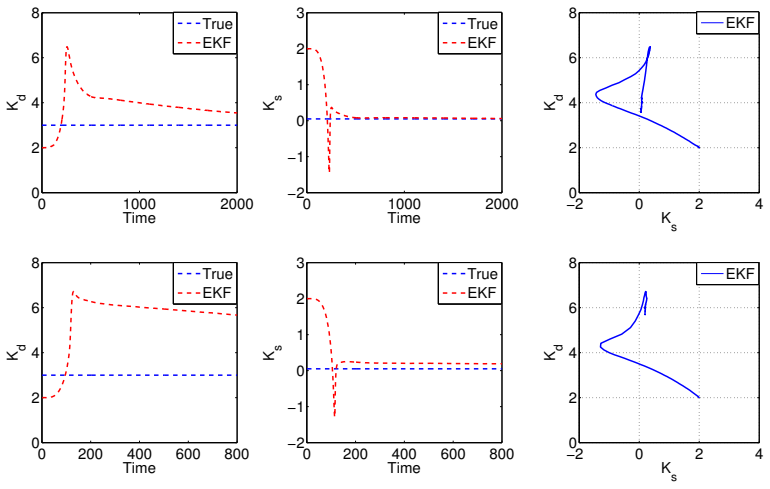


(a) 80s, 100s and 120s

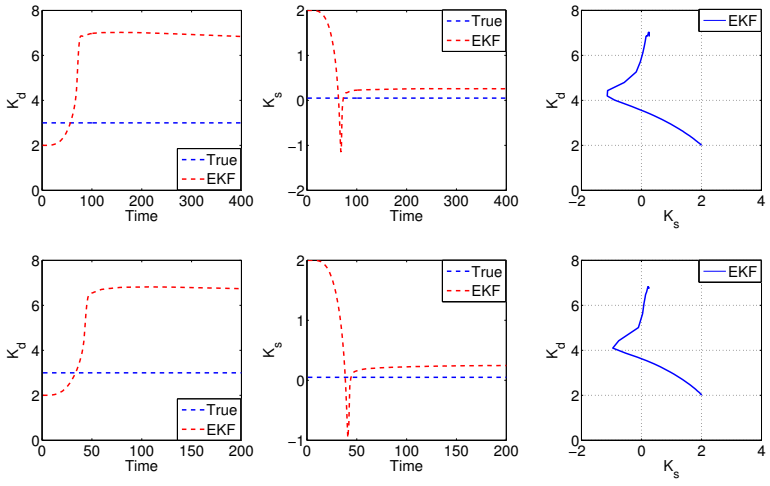


(b) 140s, 160s and 180s

FIGURE C.4: Top row : Estimation of K_s and K_d with 80s, 100s and 120s data length from PF. Bottom row : Estimation of K_s and K_d with 140s, 160s and 180s data length from PF.

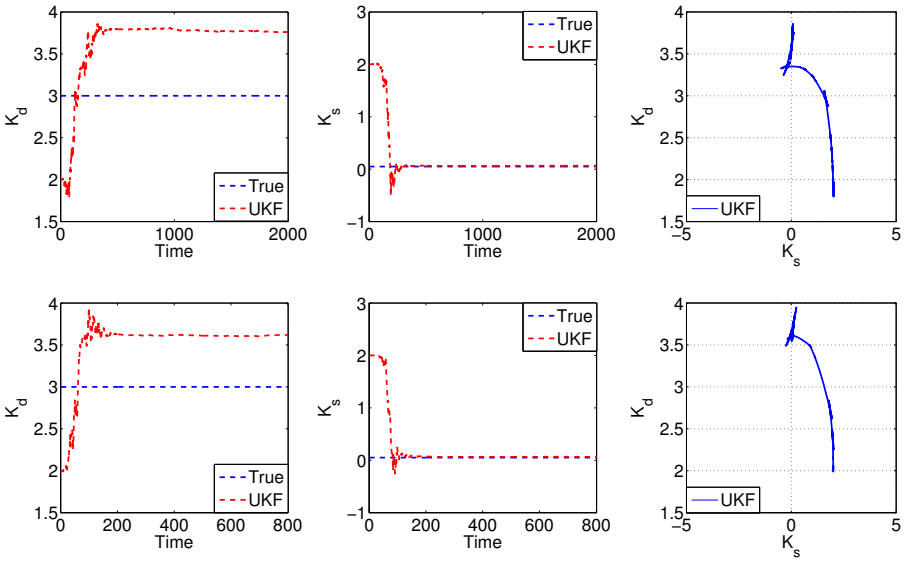


(a) 0.1s and 0.25s

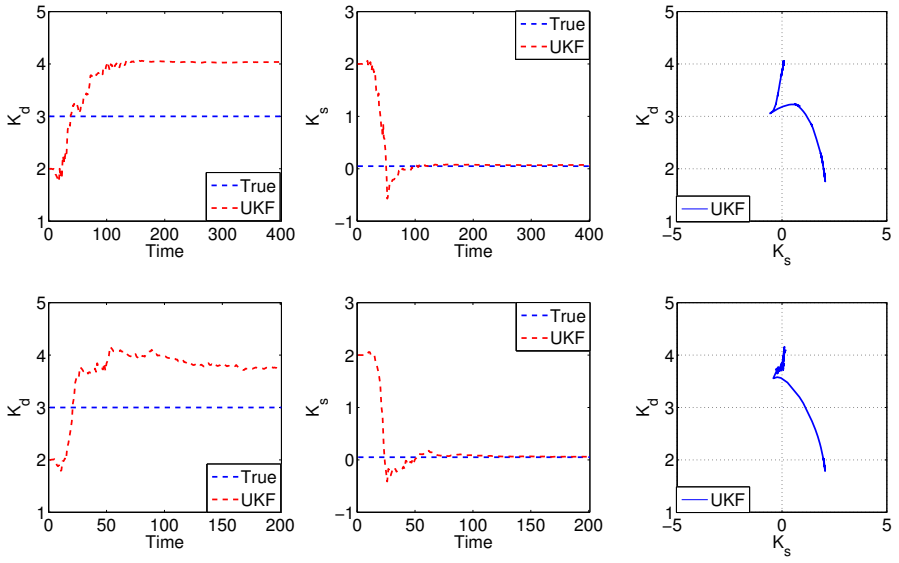


(b) 0.5s and 1s

FIGURE C.5: Top row : Estimation of K_s and K_d with 0.1s and 0.25s sampling intervals from EKF. Bottom row : Estimation of K_s and K_d with 0.5s and 1s sampling intervals from EKF.

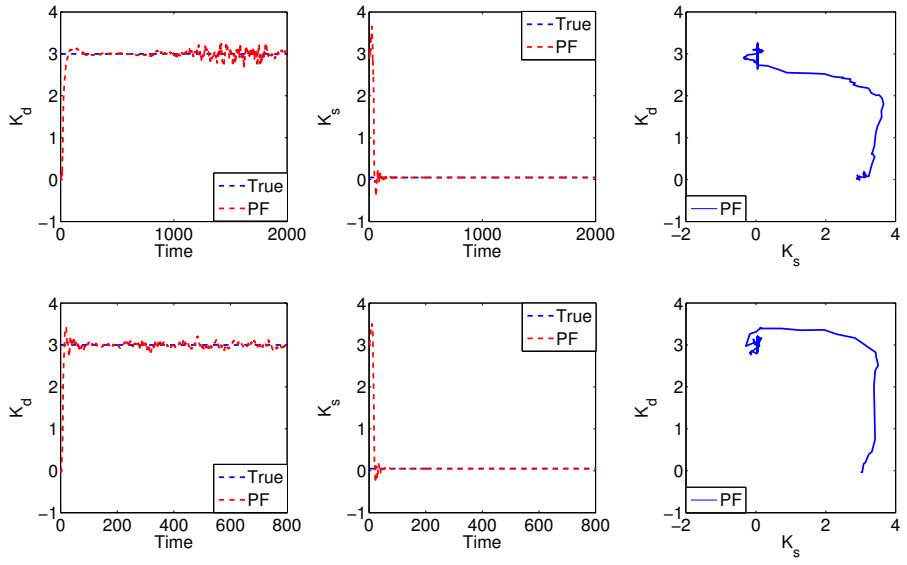


(a) 0.1s and 0.25s

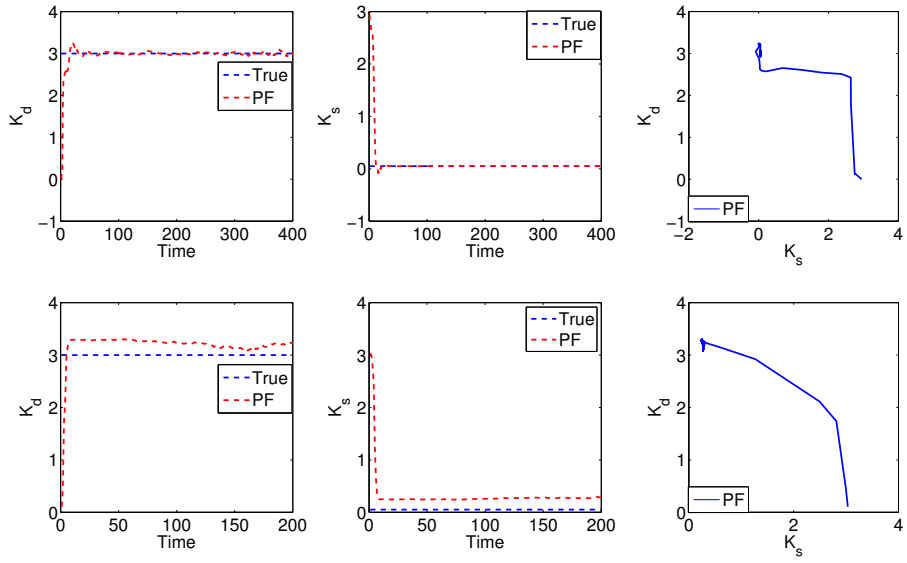


(b) 0.5s and 1s

FIGURE C.6: Top row : Estimation of K_s and K_d with 0.1s and 0.25s sampling intervals from UKF. Bottom row : Estimation of K_s and K_d with 0.5s and 1s sampling intervals from UKF.

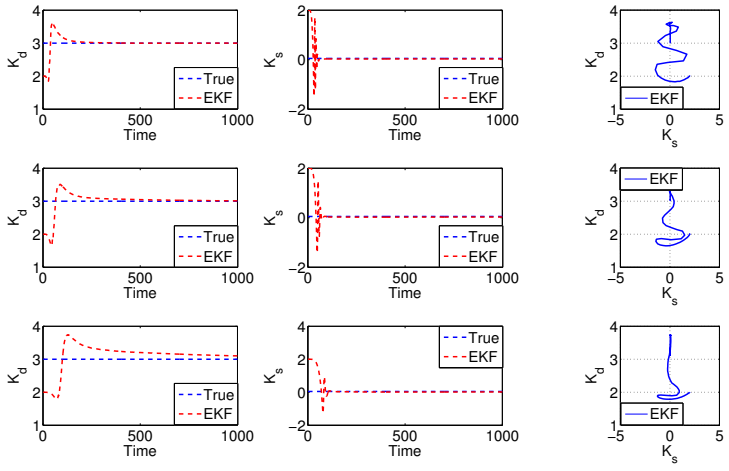


(a) 0.1s and 0.25s

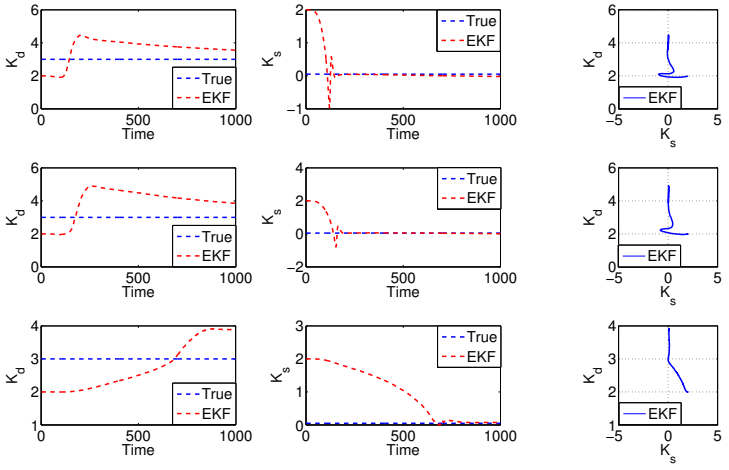


(b) 0.5s and 1s

FIGURE C.7: Top row : Estimation of K_s and K_d with 0.1s and 0.25s sampling intervals from PF. Bottom row : Estimation of K_s and K_d with 0.5s and 1s sampling intervals from PF.

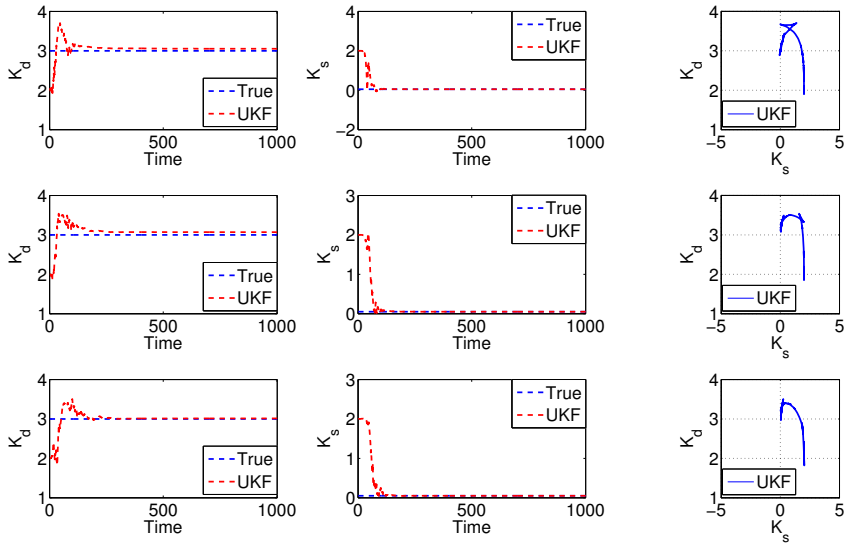


(a) 0.0001, 0.001 and 0.01

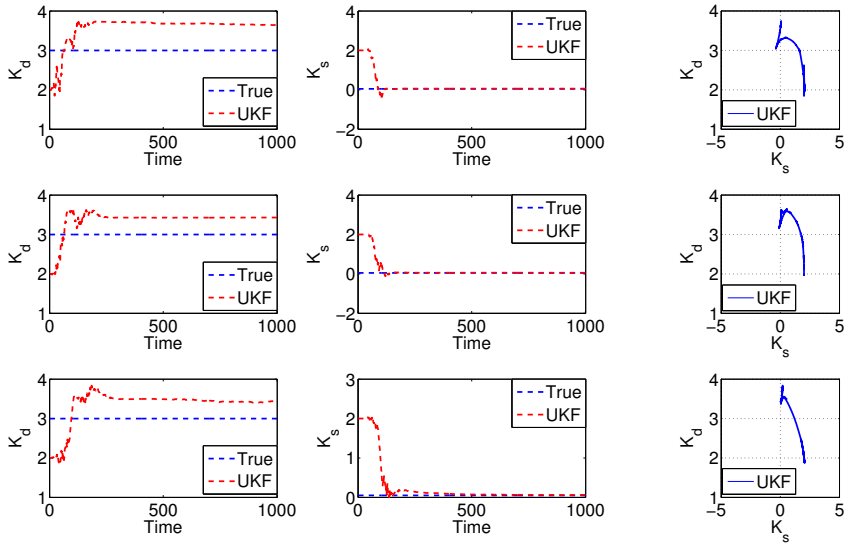


(b) 0.05, 0.1 and 1

FIGURE C.8: Top row : Estimation of K_s and K_d with 0.0001, 0.001 and 0.01 multipliers from EKF. Bottom row : Estimation of K_s and K_d with 0.05, 0.1 and 1 multipliers from EKF.

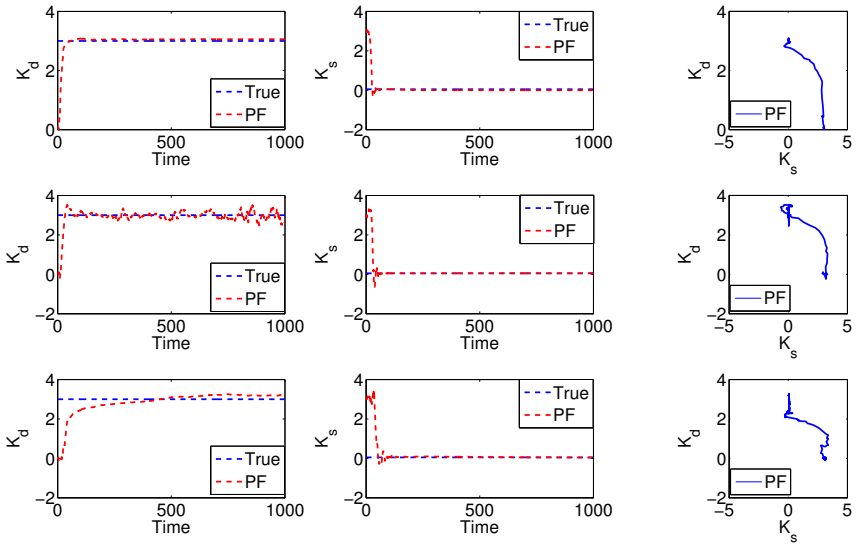


(a) 0.0001, 0.001 and 0.01

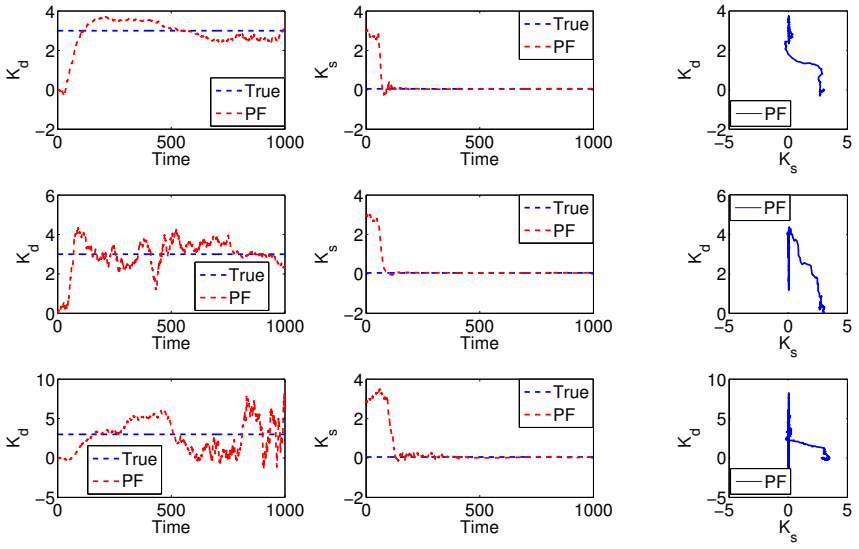


(b) 0.05, 0.1 and 1

FIGURE C.9: Top row : Estimation of K_s and K_d with 0.0001, 0.001 and 0.01 multipliers from UKF. Bottom row : Estimation of K_s and K_d with 0.05, 0.1 and 1 multipliers from UKF.



(a) 0.0001, 0.001 and 0.01



(b) 0.05, 0.1 and 1

FIGURE C.10: Top row : Estimation of K_s and K_d with 0.0001, 0.001 and 0.01 multipliers from PF. Bottom row : Estimation of K_s and K_d with 0.05, 0.1 and 1 multipliers from PF.

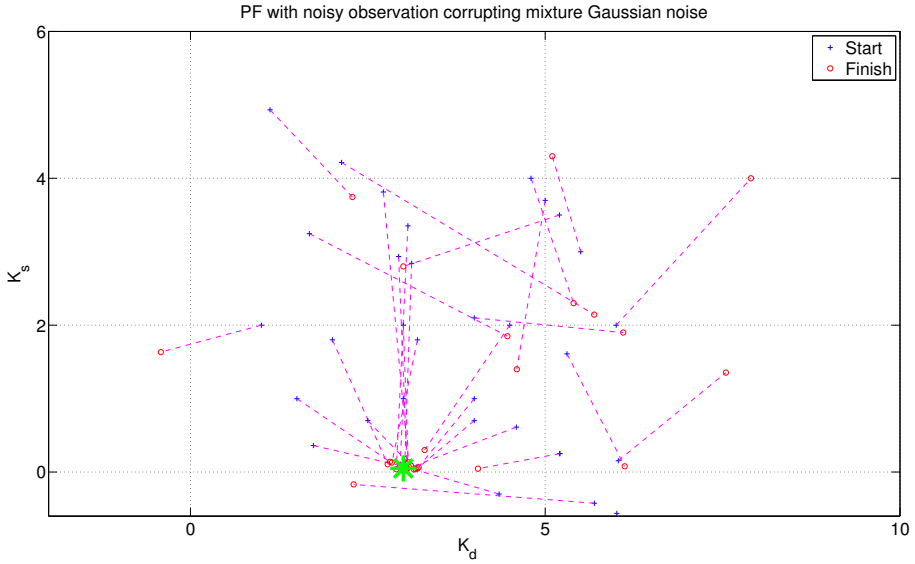


FIGURE C.11: The performance of PF on inferring two parameters K_d and k_s from the noisy observations which are generated by corrupting the Mixture Gaussian noise.

Appendix D

Supplementary information of ABC Coupled with Sensitivity Analysis Method

D.1 Details of the cell cycle system

The formula of cell cycle consists of six ordinary differential equations with twenty parameters (Jacquet et al., 2003), of which the details are described below

$$\begin{aligned}
\frac{\mathrm{dM}}{dt} &= -V_1 + V_2 + k_2 MN \\
\frac{\mathrm{dM}^*}{dt} &= V_1 - V_2 - k_1 M^* \\
\frac{\mathrm{dMN}^*}{dt} &= k_1 M^* + V_3 + V_4 \\
\frac{\mathrm{dMN}}{dt} &= V_4 - V_3 - k_2 MN \\
\frac{\mathrm{dXA}}{dt} &= V_5 - V_6 \\
\frac{\mathrm{dRA}}{dt} &= V_7 - V_8
\end{aligned} \tag{D.1}$$

With

$$
\begin{aligned}
V_1 &= V_{KS}\left(\frac{M}{K_1+M}\right)\\
V_2 &= V_P\left(\frac{M^*}{K_2+M^*}\right)\\
V_3 &= V_{KSN}\left(\frac{MN}{K_3+MN}\right)\\
V_4 &= V_{PN}\left(\frac{RA}{K_{a1}+RA}\right)\left(\frac{MN^*}{K_4+MN^*}\right)\\
V_5 &= V_{KX}\left(\frac{MN^*}{K_{a2}+MN^*}\right)\left(\frac{1-XA}{K_5+1-XA}\right)\\
V_6 &= V_{PX}\left(\frac{XA}{K_6+XA}\right)\\
V_7 &= V_{KR}\times XA\left(\frac{1-RA}{K_7+1-RA}\right)\\
V_7 &= V_{PR}\left(\frac{RA}{K_8+RA}\right)
\end{aligned}
\tag{D.2}
$$

where the values of parameters are as following: $V_P = 0.3$, $V_{KSN} = 0.5$, $V_{KS} = 0.5$, $V_{PN} = 2$, $V_{KX} = 1.3$, $V_{PX} = 0.6$, $V_{KR} = 1.6$, $V_{PR} = 0.9$, $k_1 = 6.6$, $k_2 = 5$, $K_i = 0.01(i = 1, \ldots, 8)$ and $K_{a1} = K_{a2} = 0.2$. Initial conditions of states are set as $M = XI = RI = 1$, and $M^* = MN = MN^* = XA = RA = 0$.

D.2 Cell Cycle System

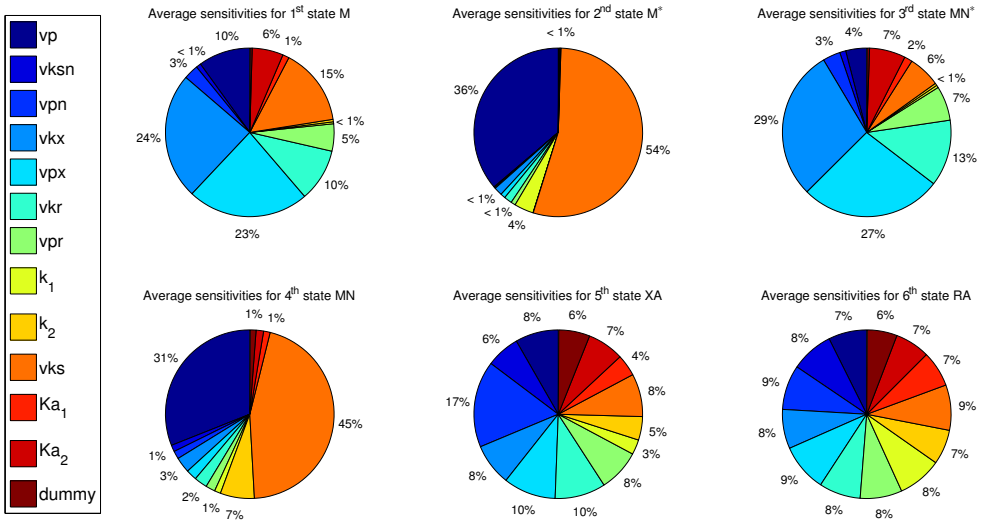


FIGURE D.1: The average sensitivity status of parameters in states M, M^*, MN^*, MN, XA and RA.

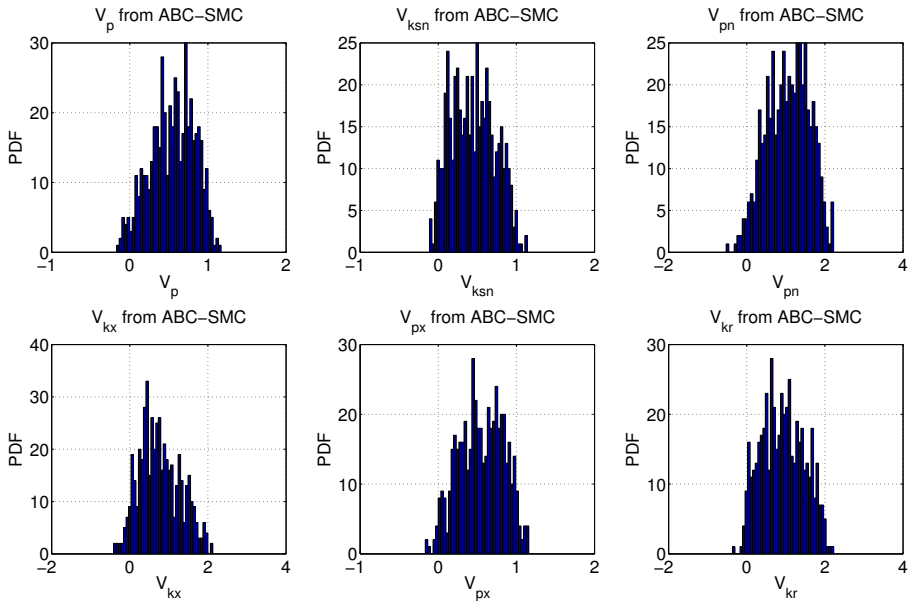


FIGURE D.2: Histogram graphs show the estimations of parameters V_p, V_{ksn}, V_{pn}, V_{kx}, V_{px} and V_{kr} from the original ABC-SMC. True values: $V_p = 0.3$, $V_{ksn} = 0.5$, $V_{pn} = 2$, $V_{kx} = 1.3$, $V_{px} = 0.6$ and $V_{kr} = 1.6$.

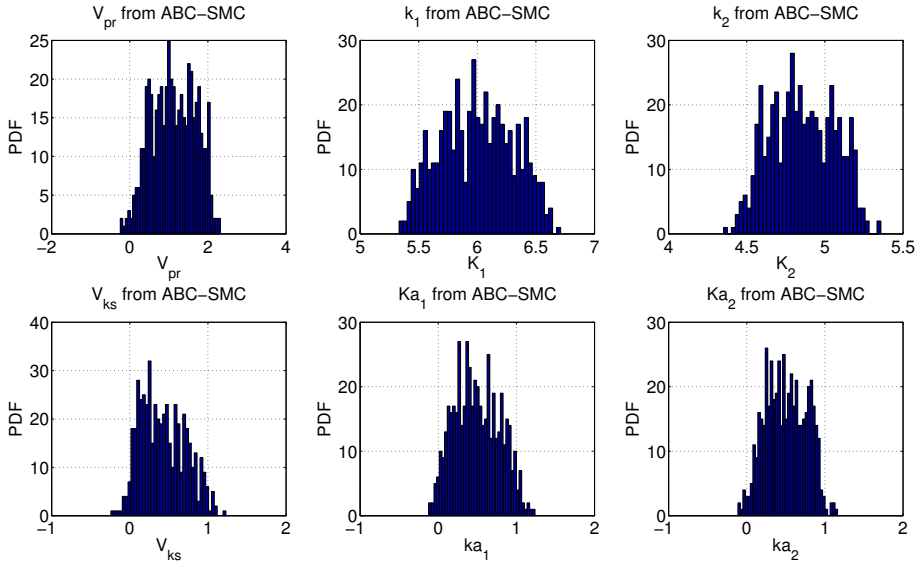


FIGURE D.3: Histogram graphs show the estimations of parameters V_{pr}, k_1, k_2, V_{ks}, Ka_1 and Ka_2 from the original ABC-SMC. True values: $V_{pr} = 0.9$, $k_1 = 6.6$, $k_2 = 5$, $V_{ks} = 0.5$, $Ka_1 = 0.2$ and $Ka_2 = 0.2$.

D.3 Implementation details

Delay-driven oscillatory system: Numerically integrating the delay differential equations above using `MATLAB`'s `dde23` package, we generated a dataset of 1000 minutes in length at a sampling interval of 0.2 min. With the periodicity of this oscillation being 120 min, this covers about 10 cycles. We progressively increase the complexity of the problem in

each of the following examples; the dataset of the first considered example is generated without corrupting noise, meaning noise-free. The parameters of the model were set to the following values following Monk (2003): the initial state vector $\mathrm{x}_0 = [3, 100]$, delay $\tau = 18.5\,\mathrm{min}$ decay constants $\mu_m = 0.03\,\mathrm{min}^{-1}$, $\mu_p = 0.03\,\mathrm{min}^{-1}$ and terms in the autoregulation function $p_0 = 100$ and $n = 5$. The number of samples N is chosen as 500 and prior distributions for generating the first samples are $\mu_m \sim \mathcal{N}(0, 0.001)$, $\mu_p \sim \mathcal{N}(0, 0.001)$, $p_0 \sim \mathcal{N}(95, 10)$ and $n \sim \mathcal{N}(2, 2)$. The diagonal elements of covariance matrix $\boldsymbol{Q}$ for transition kernel $k(\cdot)$ is $[5, 0.05, 0.05, 0.1]$. The final tolerance ϵ_T is set to 3000 for the first run, and is narrowed down to 1000 for the second parse. Implementation details for the sensitivity analysis are given as: the number of search curves $\mathrm{N_r}$ is 5; number of samples used for each curve $\mathrm{N_{se}}$ is 2049; the maximum number of Fourier coefficients M is 4; the samples for parameters after Fourier expansion are assumed to satisfy the distributions as $\mu_m \in \mathcal{U}(0.001, 0.1)$, $\mu_p \in \mathcal{U}(0.001, 0.1)$, $p_0 \in \mathcal{U}(1, 10)$, $n \in \mathcal{U}(20, 150)$ and dummy variable $\in \mathcal{U}(1, 10)$.

Represillator: The initial state vector is $\boldsymbol{x}_0 = [0, 2, 0, 1, 0, 3]$. The initial particles are generated from the distribution $\boldsymbol{\alpha}_{0,0} \sim \mathcal{U}(-2, 10)$, $\boldsymbol{n}_0 \sim \mathcal{U}(0, 10)$, $\boldsymbol{\beta}_0 \sim \mathcal{U}(-5, 20)$ and $\boldsymbol{\alpha}_0 \sim \mathcal{U}(500, 2500)$, for which the true values are $\alpha_0 = 1$, $n = 2$, $\beta = 5$, $\alpha = 1000$; the number of particles used N is chosen as 1500; the weighting factor α is 0.99 (notice that α presented here is for running the weights calculation for ABC-SMC algorithm, and is different from the system parameter α); the integer factor M is 10; the final tolerance ϵ_T is set to 200 for the first parse and becomes 40 for the second; the discount factor δ for transition kernel $k(\cdot)$ is 0.98; the threshold for resampling scheme is $N/2$; initial guesses for states are chosen as $\boldsymbol{X}_0 \sim \mathcal{U}(0, 1)$. The implementation details of the sensitivity analysis are given as: the number of search curves $\mathrm{N_r}$ is 5; the number of samples $\mathrm{N_{se}}$ used for each curve is 2049; the maximum number of Fourier coefficients M is 4; the samples for parameters after Fourier expansion are assumed to satisfy the distributions $\alpha_0 \in \mathcal{U}(0.5, 2)$, $n \in \mathcal{U}(1, 5)$, $\beta \in \mathcal{U}(3, 7)$, $\alpha \in \mathcal{U}(700, 1200)$, and dummy variable $\in \mathcal{U}(1, 10)$. The way to generate the synthetic data is the same as ABC-SMC.

Heat Shock: Implement details follow the previous work Liu and Niranjan (2012) including the time length, regular sampling interval and corrupted noise for observations. For running the inference algorithm, the prior distributions for generating samples for parameters $k_d \sim \mathcal{N}(3, 1)$, $\alpha_d \sim \mathcal{N}(0, 0.1)$, $\alpha_0 \sim \mathcal{N}(0, 0.1)$, $\alpha_s \sim \mathcal{N}(3, 1)$, $k_u \sim \mathcal{N}(0, 0.1)$, $k_u \sim \mathcal{N}(0, 0.1)$; The true values of parameters are $[k_d,\ \alpha_d,\ \alpha_0,\ \alpha_s,\ k_s,\ k_u] = [3,\ 0.015,\ 0.03,\ 3,\ 0.05,\ 0.0254]$. Integer number M is 10; Number of used samples $N = 1500$; the weighting factor α is 0.99; Threshold for resampling is $N/2$; the discount factor δ for transition kernel $k(\cdot)$ is 0.99; Ideal tolerance for the first inference iteration is $\epsilon_T = 10$, while the ideal tolerance for the second inference iteration is $\epsilon_T = 2$. The prior distributions for generating samples for states $\boldsymbol{X} \sim \mathcal{N}(0, 2)$. For the second inference iteration, the number of samples increases to $N = 2500$; Integer factor M is 20;

The implementation details of the sensitivity analysis are given as:the number of search

curves N_r is 5; the number of samples N_{se} used for each curve is 2049; the maximum number of Fourier coefficients M is 4; the samples for parameters after Fourier expansion are assumed to satisfy the distributions $k_d \in \mathcal{U}(1,5)$, $\alpha_d \in \mathcal{U}(0.001, 0.1)$, $\alpha_0 \in \mathcal{U}(0.001, 0.1)$, $\alpha_s \in \mathcal{U}(1,5)$, $k_s \in \mathcal{U}(0.001, 0.1)$, $k_u \in \mathcal{U}(0.001, 0.1)$ and dummy variable $\in \mathcal{U}(0.1, 6)$.

Cell Cycle: The implementation details of ABC-SMC for the most sensitive parameters (i.e. V_p, V_{px}, V_{ks} and V_{kx} are the most sensitive parameters of this system) are given as: The initial state vector for synthesizing the real dataset is $\boldsymbol{x}_0 = [1, 0, 0, 0, 0, 0]$. The time length for generating dataset is 200 minutes, sampling at regular interval of 0.2 minutes and no decaying needs to be considered, i.e. 1000 sample points are used for representing the system outputs. The diagonal elements in covariance matrix for observation noise $\boldsymbol{\omega}$ are 0.05 times variance of synthetic state dataset $\boldsymbol{X}$. The initial distributions of ABC-SMC for synthesizing six state outputs are all following $\hat{\boldsymbol{X}}_{\mathbf{0}} \sim \mathcal{N}(0.5, 0.25)$; the prior distributions for generating the samples of unknown parameters are $\boldsymbol{V}_{ks,0} \sim \mathcal{U}(-10, 10)$, $\boldsymbol{V}_{kx,0} \sim \mathcal{U}(-5, 5)$, $\boldsymbol{V}_{px,0} \sim \mathcal{U}(-10, 10)$ and $\boldsymbol{V}_{p,0} \sim \mathcal{U}(-5, 5)$; the number of samples used N is 2500; the integer factor M is 25; the weighting factor α is 0.99; the final tolerance ϵ_T is 0.0005 the discount factor δ for transition kernel $k(\cdot)$ is 0.97; the threshold for resampling scheme is $N/2$.

Bibliography

U. Alon. *An introduction to systems biology: design principles of biological circuits.* Chapman & Hall/CRC, London, 2006.

S. Amari and H. Nagaoka. *Methods of information geometry.* Oxford University Press, Oxford, 2000.

A. Andreychenko, L. Mikeev, D. Spieler, and V. Wolf. Approximate maximum likelihood estimation for stochastic chemical kinetics. *EURASIP Journal on Bioinformatics and Systems Biology*, 1:1–14, 2012.

C. Andrieu, A. Doucet, S. S. Singh, and V. B. Tadic. Particle methods for change detection, system identification, and control. *Proceedings of the IEEE*, 92(3):423–438, 2004.

C. Andrieu and T. Johanes. A tutorial on adaptive Markov chain Monte Carlo. *Stat. Comput.*, 18:343–373, 2008.

C. Andriue, N. De Freitas, A. Doucet, and M. I. Jordan. An introduction to MCMC for machine learning. *Machine Learning*, 50:5–43, 2001.

M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2), 2002.

M. Ashyraliyev, J. Jaeger, and J. G. Blom. Parameter estimation and determinability analysis applied to *Drosophila* gap gene circuits. *BMC Systems Biology*, 2, 2008.

C. T. H. Baker, G. A. Bocharov, J. M. Ford, P. M. Lumb, S. J. Norton, C. A. H. Paul, T. Junt, P. Krebs, and B. Ludewig. Computational approaches to parameter estimation and model selection in immunology. *Journal of Computational and Applied Mathematics*, 184:50–76, 2005.

T. Baldacchino, S. R. Anderson, and V. Kadirkamanathan. Structure detection and parameter estimation for NARX models in a unified EM framework. *Automatica*, 48 (5):857–865, 2012.

Y. Bar-Shalom and T. E. Fortmann. *Tracking and data association.* Academic Press Professional, Inc., 1987.

Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation, Tracking and Navigation: Theory, Algorithms and Software*. New York: John Wiley & Sons, 2001. ISBN 0-471-41655-X.

M. Barenco, D. Tomescu, D. Brewer, R. Callard, J. Stark, and M. Hubank. Ranked prediction of p53 targets using hidden variable dynamic modeling. *Genome Biology*, 7(25), 2006.

D. Battogtokh and J. J. Tyson. Bifuracation analysis of a model of the budding yeast cell cycle. *Chaos*, 14(3):653–661, 2004.

M. J. Beal, F. Falciani, Z. Ghahramani, C. Rangel, and D. L. Wild. A Bayesian approach to reconstructing genetic regulatory networks with hidden factors. *Bioinformatics*, 21: 349–356, 2005.

M. Beaumont, C. P. Robert, J. M. Marin, and J. M. Cornuet. Adaptivity for abc algorithms: the abc-pmc scheme. *Biometrika*, 94(4):983–990, 2009.

M. A. Beaumont. Estimation of population growth or decline in genetically monitored populations. *Genetics*, 164:1139–1160, 2003.

M. A. Beaumont. Approximate Bayesian computation in evolution and ecology. *Annu. Rev. Ecol. Evol. Syst*, 41:379–406, 2010.

W. Becker, J. Rowson, J. E. Oakley, A. Yoxall, G. Manson, and K. Worden. Bayesian sensitivity analysis of a model of the aortic valve. *Journal of biomechanics*, 44(8): 1499–1506, 2011.

A. A. Berryman. The Origins and Evolution of Predator-Prey Theory. *Ecology*, 73(5): 1530–1535, 1992.

S. Berthoumieux, M. Brilli, H. de Jong, D. Kahn, and E. Cinquemani. Identification of metabolic network models from incomplete high-throughput datasets. *Bioinformatics*, 27(13):86–95, 2012.

C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag, New York, 2006.

S. M. Blower and H. Dowlatabadi. Sensitivity and uncertainty analysis of complex models of disease transmission: an HIV model, as an example. In *International Statistical Review*, pages 229–243, 1994.

S. M. Blower, H. B. Gershengorn, and R. M. Grant. A tale of two futures: HIV and antiretroviral therapy in San Francisco. *Science*, 287(5453):650–654, 2000.

D. M. Bortz and P. W. Nelson. Model selection and mixed-effects modeling of HIV infection dynamics. *Bull. Math. Biol.*, 68:2005–2025, 2006.

B. Calderhead and M. Girolami. Statistical analysis of nonlinear dynamical systems using differential geometric sampling methods. *Interface Focus*, 1(6):821–835, 2011.

O. Cappé, A. Guillin, J.-M. Marin, and C. Robert. Population Monte Carlo. *J. Comput. Graph. Statist.*, 13(4):907–929, 2004.

G. Casella and E. I. George. Explaining the Gibbs sampler. *The American Statistician*, 46(3):167–174, 1992.

Y. T. Chan, A. G. C. Hu, and J. B. Plant. A Kalman filter based tracking scheme with input estimation. *Aerospace and Electronic Systems, IEEE Transactions on*, AES-15 (2):237–244, 1979.

K. C. Chen, A. Csikasz-Nagy, B. Gyorffy, J. Val, B. Novak, and J. J. Tyson. Kinetic analysis of a molecular model of the budding yeast cell cycle. *Molecular biology of the cell*, 11(1):369–391, 2000.

B. L. P. Cheung, B. A. Riedner, G. Tononi, and B. Van Veen. Estimation of cortical connectivity from EEG using state-space models. *Biomedical Engineering, IEEE Transactions on*, 57(9):2122–2134, 2010.

S. Chib, F. Nardari, and N. Shephard. Markov Chain Monte Carlo methods for stochastic volatility models. *Journal of Econometrics*, 108(2):281–316, 2002.

G. A. Churchill and B. Lazareva. Bayesian restoration of a hidden Markov chain with applications to DNA sequencing. *Journal of Computational Biology*, 6(2):261–277, 1999.

W. W. Cleland. The kinetics of enzyme-catalyzed reactions with two or more substrates or products: I. Nomenclature and rate equations. *Biochim. Biophys*, 67(104-137), 1963.

F. R. Cross. Two redundant oscillatory mechanisms in the yeast cell cycle. *Dev. Cell*, 4:741–752, 2003.

R. I. Cukier, C. M. Fortuin, K. E. Shuler, A. G. Petschek, and J. H. Schaibly. Study of sensitivity of coupled reaction systems to uncertainties in rate coefficients. 1. theory. *Journal of Chemical Physics*, 59(8):3873–3878, 1973.

B. J. Daigle, M. K. Roh, L. R. Petzold, and J. Niemi. Accelerated maximum likelihood parameter estimation for stochastic biochemical systems. *BMC bioinformatics*, 13(1): 68, 2012.

J. T. Davis, H-H. Chen, R. M. Moore, Y. Nishitani, S. Masamune, A. J. Sinskey, and C. T. Walsh. Biosynthetic thiolase from *Zoogloea ramigera*: Inactivation with haloacetyl CoA analogs. *J. Biol. Chem*, 262:90–96, 1987.

J. F. de Freitas, M. Niranjan, A. H. Gee, and A. Doucet. Sequential Monte Carlo methods to train neural network models. *Neural Computation*, 12(4):955–993, 2000.

P. Del Moral, A. Doucet, and A. Jasra. Sequential Monte Carlo samplers. *J. R. Stat. Soc*, 68:411–436, 2006.

P. Del Moral, A. Doucet, and A. Jasra. An adaptive sequential Monte Carlo method for approximate bayesian computation. *Stat. Comput*, 22(5):1009–1020, 2012.

P. Dellaportas and A. F. Smith. Bayesian inference for generalized linear and proportional hazards models via Gibbs sampling. *Applied Statistics*, 42:443–443, 1993.

M. Desai and P. Ormerod. Richard Goodwin: A Short Appreciation. *The Economic Journal*, 108(450):1431–1435, 1998.

M. Dewar, V. Kadirkamanathan, M. Opper, and G. Sanguinetti. Parameter estimation and inference for stochastic reaction-diffusion systems: application to morphogenesis in D. melanogaster. *BMC Systems Biology*, 4(1):21, 2010.

R. P. Dickinson and R. J. Gelinas. Sensitivity analysis of ordinary differential equation systems-a direct method. *Journal of computational physics*, 21(2):123–143, 1976.

J. Diebolt and C. P. Robert. Estimation of finite mixture distributions through Bayesian sampling. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 363–375, 1994.

F. Dondelinger, D. Husmeier, S. Rogers, and M. Filippone. parameter inference using adaptive gradient matching with Gaussian processes. In *In Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, pages 216–228, 2013.

A. Doucet, N. De Freitas, and N. Gordon. An introduction to sequential Monte Carlo methods. In *Sequential Monte Carolo Methods in Practice*, pages 3–13, 2001.

A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and computing*, 10(3):197–208, 2000.

S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth. Hybrid Monte Carlo. *Physics Letters B*, 195(2):123–214, 2011.

F. Y. Edgeworth. On the probable errors of frequency-constants. *Journal of the Royal Statistical Society*, 71:499–512, 1908.

H. El-Samad, S. Prajna, A. Papachristodoulou, J. Doyle, and M. Khammash. Advanced methods and algorithms for biological networks analysis. *Proceedings of the IEEE*, 94: 832–853, 2006.

S. J. Elledge. Cell Cycle Checkpoints: Preventing an Identity Crisis. *Science*, 274(5293): 1664–1672, 1996.

M. B. Elowitz and S. Leibler. A synthetic oscillatory network of transcriptional regulators. *Nature*, 403:335–338, 2000.

J. Fan and W. Zhang. Statistical estimation in varying coefficient models. *Ann. Stat*, 27:1491–1518, 1999.

D. A. Fell and H. M. Sauro. Metabolic control and its analysis. *European Journal of Biochemistry*, 148(3):555–561, 1985.

Y. Fomekong-Nanfack, M. Postma, and J. A. Kaandorp. Inferring drosophila gap gene regulatory network: a parameter sensitivity and perturbation analysis. *BMC Systems Biology*, 3(1):94, 2009. ISSN 1752-0509.

A. G. Fredrickson. Formulation of structured growth models. *Biotechnol. Bioeng*, 18: 1481–1486, 1976.

A. E. Gelfand and A. F. M. Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85:398–409, 1990.

A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin. *Bayesian data analysis*. CRC press, 3th edition, 2013.

S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-6(6):721–741, 1984.

T. U. Gerngross, K. D. Snell, O. P. Peoples, A. J. Sinskey, E. Csuhai, S. Masamune, and J. Stubbe. Overexpression and purification of the soluble polyhydroxyalkanoate synthase from *Alcaligenes eutrophus*: Evdence for a required posttranslational modification for catalytic activity. *Biochemistry*, 33:9311–9320, 1994.

J. Geweke. Bayesian inference in econometric models using Monte Carlo integration. *Econometrica*, 24:1317–1399, 1989.

C. J. Geyer and E. A. Thompson. A new approach to the joint estimation of relationship from DNA fingerprint data. In *Population management for survival and recovery*, pages 245–260, New York, 1995. Columbia University Press.

M. Ghil and K. Ide. Data assimilation in meteorology and oceanography: theory and practice. *J. Meteor. Soc. Jpn*, 75:111–496, 1997.

M. Girolami and B. Calderhead. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214, 2011.

A. Golightly and D. J. Wilkinson. Bayesian inference for stochastic kinetic models using a diffusion approximation. *Biometrics*, 61:781–788, 2005.

A. K. Gombert and J. Nielsen. Mathematical modelling of metabolism. *Journal of Biotechnology*, 11:180–186, 2001.

N. J. Gordon, D. J. Salmond, and A. F. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *In IEE Proceedings F (Radar and Signal Processing)*, 140(2):107–113, 1993.

R. N. Gutenkunst, J. J. Waterfall, F. P. Casey, K. S. Brown, C. R. Myers, and J. P. Sethna. Universally Sloppy Parameter Sensitivities in Systems Biology Models. *PLoS Comput. Biol.*, 3(10):189–197, 2007.

J. Hasenauer, S. Waldherr, N. Radde, M. Doszczak, P. Scheurich, and F. Allgöwer. A maximum likelihood estimator for parameter distributions in heterogeneous cell populations. *Procedia Computer Science*, 1(1):1655–1663, 2010.

W. K. Hastings. Monte Carlo sampling methods using Markov chain and their Application. *Biometrika*, 57:97–109, 1970.

S. S. Haykin. *Kalman filtering and neural networks.* Wiley, New York, 2001.

G. W. Haywood, A. J. Anderson, L. Chu, and E. A. Dawes. The tole of NADH- and NADPH-linked acetoacetyl-CoA reductases in the poly-3-hydroxybutrate synthesizing organism *Alcaligenes eutrophus*. *FEMS Microbiol. Lett*, 52(10):259–264, 1988.

R. Heinrich and T. A. Rapoport. A Linear Steady-State Treatment of Enzymatic Chains. *European Journal of Biochemistry*, 42(1):97–105, 1974.

A. L. Hodgkin and A. F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology*, 4(117): 500–544, 1952.

J. Hull and A. White. The pricing of options on assets with stochastic volatilities. *Journal of Finance*, 42:281–300, 1987.

F. Hynne, S. Danø, and P. G. Sørensen. Full-scale model of glycolysis in *saccharomyces cerevisiae*. *Biophysical Chemistry*, 94:121–163, 2001.

A. E. C. Ihekwaba, D. S. Broomhead, R. L. Grimley, N. Benson, and D. B. Kell. Sensitivity analysis of parameters controlling oscillatory signaling in the NF-κB pathway: the roles of IKK and IκBBα. *Systems Biology*, 1(1):93–103, 2004.

R. L. Iman, J. E. Campbell, and J. C. Helton. An approach to sensitivity analysis of computer models. I- Introduction, input, variable selection and preliminary variable assessment. *Journal of quality technology*, 13:174–183, 1981.

B. P. Ingalls, B. P. Duncker, D. R. Kim, and B. J. McConkey. System Level Modeling of the Cell Cycle Using Budding Yeast. *Cancer Information*, 3:357–370, 2007.

M. Jacquet, G. Renault, S. Lallet, J. D. Mey, and A. Goldbeter. Oscillatory nucleocytoplasmic shuttling of the general stress response transcriptional activators msn2 and msn4 in *Saccharomyces cerevisiae*. *Cell Biology*, 161(3):497–505, 2003.

B. Jayawardhana, D. B. Kell, and M. Rattray. Bayesian inference of the sites of perturbations in metabolic pathways via Markov chain Monte Carlo. *Bioinformatics*, 24(9): 1191–1197, 2008.

A. H. Jazwinski. *Stochastic processes and filtering theory.* Academic Press, 1970.

P. Joyce and P. Marjoram. Approximately sufficient statistics and Bayesian computation. *Stat. Appl. Genet. Mol. Biol.*, 7(1):26, 2008.

S. J. Julier. The scaled Unscented Transformation. *In Proceedings of the American Control Conference*, pages 4555–4559, 2002.

S. J. Julier and J. K. Uhlmann. New extension of the Kalman filter to nonlinear systems. In *AeroSense'97*, pages 182–193. International Society for Optics and Photonics, 1997.

S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte. A new approach for filtering nonlinear systems. *Proceedings of the American Control Conference*, D(82):1628–1632, 1995.

H. A. Kacser and J. Burns. The control of flux. *In Symp. Soc. Exp. Biol.*, 27:65–104, 1973.

V. Kadirkamanathan and M. Niranjan. A function estimation approach to sequential learning with neural networks. *Neural Computation*, 5:954–975, 1993.

R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME-Journal of Basic Engineering*, D(82):35–45, 1960.

P. Kennedy. *A guide to econometrics.* MIT press., New York, 2003.

B. Kessler and B. Witholt. Factors involved in the regulatory network of polyhydroxyalkanoate metabolism. *Journal of Biotechnology*, 86(2):97–104, 2001.

K. A. Kim, S. L. Spencer, J. G. Albeck, J. M. Burke, P. K. Sorger, and S. Gaudet. A systematic calibration of a cell signaling network model. *BMC bioinformatics*, 11(1): 202, 2010.

E. L. King and C. Altman. A schematic method of deriving the rate laws for enzyme-catalyzed reactions. *The Journal of physical chemistry*, 60(10):1375–1378, 1956.

A. Kiparissides, S. S. Kucherenko, A. Mantalaris, and E. N. Pistikopoulos. Global sensitivity analysis challenges in biological systems modeling. *Industrial & Engineering Chemistry Research*, 48(15):7168–7180, 2009.

G Kitagawa. A self-organizing state-space model. *J. Am. Stat. Assoc*, 93(443):1203–1215, 1998.

G. Kitagawa and W. Gersch. *Smoothness Priors Analysis of Time Series.* Springer-Verlag, New York, 1996.

H. Kitano. Systems biology: A brief overview. *Science*, 295(5560):1662–1664, 2002.

M. Komorowski, M. J. Costa, D. A. Rand, and M. P. Stumpf. Sensitivity, robustness, and identifiability in stochastic chemical kinetics models. *Proceedings of the National Academy of Sciences*, 108(21):8645–8650, 2011.

A. Kramer and N. Radde. Towards experimental design using a Bayesian framework for parameter identification in dynamic intracellular network models. *Procedia Computer Science*, 1:1645–1653, 2010.

N. Lange, B. P. Carlin, and A. E. Gelfand. Hierarchical Bayes models for the progression of HIV infection using longitudinal CD4 T-cell numbers. *Journal of the American Statistical Association*, 87(419):615–626, 1992.

N. D. Lawrence, G. Sanguinetti, and M. Rattray. Modelling transcriptional regulation using Gaussian processes. In *Advances in Neural Information Processing Systems 19*, page 785, Vancouver, 2006. MIT press.

T. A. Leaf and F. Srienc. Metabolic Modeling of Polyhydroxybutyrate Biosynthesis. *Biotechnology and bioengineering*, 57:557–570, 1997.

S. J. Lee, D. J. Lee, and H. S. Oh. Technological forecasting at the Korean stock market: a dynamic competition analysis using Lotka-Volterra model. *Technological Forecasting and Social Change*, 72(8):1044–1057, 2005.

J-C. Leloup and A. Goldbeter. Toward a detailed computational model for the mammalian circadian clock. *Proceedings of the National Academy of Sciences*, 100(12): 7051–7056, 2003.

R. Li, R. Goodall, and V. Kadirkamanathan. Estimation of parameters in a linear state space model using a Rao-Blackwellised particle filter. In *IEE Proc. - Control Theory Appl.*, pages 727–738, 2004.

G. Lillacci and M. Khammash. Parameter estimation and model selection in computational biology. *PLoS Computational Biology*, 6(3):696–713, 2010.

G. Liu, L. Kong, Z. Wang, C. Wang, and R. Wu. Systems mapping of metabolic genes through control theory. *Advanced drug delivery reviews*, 65(7):918–928, 2013.

J Liu and M West. Combined parameter and state estimation in simulation-based filtering. In *Sequential Monte Carlo Methods in Practice*, pages 197–217, 2001.

J. S. Liu. *Monte Carlo strategies in scientific computing.* Springer, New York, 2001.

J. S. Liu and R. Chen. Sequential Monte Carlo methods for dynamical systems. *J. Amer. Statist. Assoc*, 93:1032–1044, 1998.

X Liu and M. Niranjan. State and parameter estimation of the heat shock response system using Kalman and particle filters. *Bioinformatics*, 28(11):1501–1507, 2012.

Y. Y. Liu, J. J. Slotine, and A. L. Barabási. Controllability of complex networks. *Nature*, 437:167–173, 2011.

D. Loebis, R. Sutton, J. Chudley, and W. Naeem. Adaptive tuning of a Kalman filter via fuzzy logic for an intelligent AUV navigation system. *Control Engineering Practice*, 12(12):1531 – 1539, 2004.

A. J. Lotka. *Elements of physical biology.* MD: Williams & Wilkins Co, Baltimore, 1925.

S. Marino, I. B. Hogue, C. J. Ray, and D. E. Kirschner. A methodology for performing global uncertainty and sensitivity analysis in systems biology. *J Theor Biol*, 254: 178–196, 2008.

P. Marjoram, J. Molitor, V. Plagnol, and S. Tavaré. Markov chain Monte Carlo without likelihoods. *Proc Natl Acad Sci*, 100(26):15324–15328, 2003.

H. H. McAdams and A. Arkin. It's a noisy business! Genetic regulation at the nanomolar scale. *Trends Genet*, 15:65–69, 1999.

M. D. McKay, R. J. Beckman, and W. J. Conover. Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, 1979.

P. Mendes and D. B. Kell. Non-linear optimization of biochemical pathways: applications to metabolic engineering and parameter estimation. *Bioinformatics*, 14(10):869–883, 1998.

N. Metropolis and S. Ulam. The Monte Carlo method. *Journal of the American Statistical Association*, 44(247):335–341, 1949.

L. Michaelis and M. L. Menten. Die Kinetik der Invertinwirkung. *Biochem Z*, 49: 333–369, 1913.

C. G. Moles, P. Mendes, and J. R. Banga. Parameter estimation in biochemical pathways: A comparison of global optimization methods. *Genome Research*, 13:2467 – 2474, 2003.

N. A. M. Monk. Oscillatory Expression of Hes1, p53 and NF-κB Driven by Transcriptional Time Delays. *Current Biology*, 13:1409–1413, 2003.

T. G. Muller, D. Faller, J. Timmer, I. Swameye, O. Sandra, and U. Klingmüller. Tests for cycling in a signalling pathway. *J. R. Stat. Soc. Ser. C*, 53(557):58–62, 2004.

T. G. Muller and J. Timmer. Modeling the non-linear dynamics of cellular signal transduction. *Int. J. Bifurcat. Chaos.*, 14:2069–2079, 2004.

M. K. Murray and J. W. Rice. *Differential Geometry and Statistics.* Chapman and Hall, New York, 1993.

C. Musso, N. Oudjane, and F. LeGland. Improving regularized particle filter. In *Sequential Monte Carlo Methods in Practice*, pages 197–217, 2001.

K. Nakamura, R. Yoshida, M. Nagasaki, S. Miyano, and T. Higuchi. Parameter estimation of *in silico* biological pathways with particle filtering towards a petascale computing. *Pacific Symposium on Biocomputing*, 14:227–238, 2009.

R. M. Neal. MCMC using Hamiltonian dynamics. In *Handbook of Markov Chain Monte Carlo*. Boca Raton, 2010.

R. K. Niedenthal, L. Riles, M. Johnston, and J. H. Hegemann. Green fluorescent protein as a marker for gene expression and subcellular localization in budding yeast. *Yeast*, 12:773–786, 1996.

M. Niranjan. Sequential tracking in pricing financial options using model based and neural network approaches. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems*, pages 960–966, 1997.

Denis. Noble. Caradiac action and pacemaker potentials based on the Hodgkin-Huxley equations. *Nature*, 188:495–497, 1960.

G. A. Parker and C. D. Johnson. Decoupling linear dynamical systems using disturbance accommodation control theory. In *System Theory, 2009. SSST 2009. 41st Southeastern Symposium on*, pages 199–204, 2009.

I. Petre, A. Mizera, C. Hyder, A. Meinander, A. Mikhailov, R. Morimoto, L. Sistonen, J. Eriksson, and R. Back. A simple mass-action model for the eukaryotic heat shock response and its mathematical validation. *Natural Computing*, 10(1):595–612, 2011.

M. Pitt and N. Shephard. Filtering via simulation: Auxiliary particle filters. *J. Amer. Statist. Assoc*, 94(446):590–599, 1999.

Y. Poirier, C. Somerville, L. A. Schechtman, M. M. Satkowski, and I. Noda. Synthesis of high-molecular-weight poly([r]-(-)-3-hydroxybutyrate) in transgenic *arabidopsis thaliana* plant cells. *Int. J. Biol. Macromol*, 17(1):7–12, 1995.

J. K. Pritchard, M. T. Seielstad, A. Perez-Lezaun, and M. W. Feldman. Population growth of human y chromosomes: a study of y chromosome microsatellites. *Molecular Biology and Evolution*, 16:1791–1798, 1999.

M. Quach, N. Brunel, and F. d'Alché 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.

H. Rabitz, M. Kramer, and D. Dacol. Sensitivity analysis in chemical kinetics. *Annual review of physical chemistry*, 34(1):419–461, 1983.

C. R. Rao. Information and accuracy attainable in the estimation of statistical parameters. *Bull. Calc. Math. Soc.*, 37:81–91, 1945.

H. E. Rauch, C. T. Striebel, and F. Tung. Maximum likelihood estimates of linear dynamic systems. *JAIAA journal*, 3(8):1445–1450, 1965.

C. Reder. Metabolic control theory: a structural approach. *Journal of Theoretical Biology*, 135(2):175–201, 1988.

C. P. Robert and G. Casella. *Monte Carlo Statistical Methods.* Springer-Verlag, New York, 1999.

D. V. Roberts. *Enzyme Kinetics.* Cambridge University Press, Cambridge, 1977.

M. Rodriguez-Fernandez, M. Rehberg, A. Kremling, and J. R. Banga. Simultaneous model discrimination and parameter estimation in dynamic models of cellular systems. *BMC systems biology*, 7(1):1–14, 2013.

S. Rogers and M. Girolami. *A first course in machine learning.* CRC Press, 2011.

S. Roweis and Z. Ghahramani. A unifying review of linear Gaussian models. *Neural Computation*, 11:305–345, 1999.

M. Saisana, A. Saltelli, and S. Tarantola. Uncertainty and sensitivity analysis techniques as tools for the quality assessment of composite indicators. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 168(2):307–323, 2005.

A. Saltelli. Making best use of model evaluations to compute sensitivity indices. *Computer Physics Communications*, 145(2):280–297, 2002.

A. Saltelli and R. Bolado. An alternative way to compute Fourier amplitude sensitivity test (FAST). *Computational Statistics &Data Analysis*, 26(4):445–460, 1998.

A. Saltelli, K. Chan, and E. M. Scott. *Sensitivity analysis Wiley series in probability and statistics.* Willey, New York, 2000.

A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J Cariboni, D. Gatelli, M. Saisana, and S. Tarantola. *Global sensitivity analysis: the primer.* Wiley, New York, 2008.

A. Saltelli, M. Ratto, S. Tarantola, and F. Campolongo. Sensitivity analysis for chemical models. *Chemical reviews*, 105(7):2811–2828, 2005.

A. Saltelli, S. Tarantola, and K. P. S. Chan. A quantitative model-independent method for global sensitivity analysis of model output. *Technometrics*, 41(1):39–56, 1999.

S. Sanei and J. Chamber. *EEG signal processing*. Wiley-Interscience, 2007.

G. Sanguinetti, N. D. Lawrence, and M. Rattray. Probabilistic inference of transcription factor concentrations an gene-specific regulatory activities. *Bioinformatics*, 22:2775–3781, 2006.

M. Schelker, A. Raue, J. Timmer, and C. Kreutz. Comprehensive estimation of input signals and dynamics in biochemical reaction networks. *Bioinformatics*, 28(18):529–534, 2012.

M. Secrier, T. Toni, and M. P. H. Stumpf. The ABC of reverse engineering biological signalling systems. *Mol. BioSyst.*, 5:1925–1935, 2009.

L. Shang, D. D. Fan, M. Kim, J. Choi, and H. N. Chang. Modeling of poly(3-hydroxybutyrate) production by high cell density fed-batch culture of *ralstonia eutropha*. *Biotechnology and Bioprocess Engineering*, 12(4):417–423, 2007.

N. Shephard. Statistical aspects of ARCH and stochastic volatility. In D. R. Cox, D. V. Hinkley, and O. E. Barndorff-Nielsen, editors, *Time Series Models with Econometric, Finance and Other Applications*, pages 1–67, London, 1996. UK: Chapman & Hall.

L. C. Shi and B. L. Lu. Off-line and on-line vigilance estimation based on linear dynamical system and manifold learning. In *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, pages 6587–6590, 2010.

S. A. Sisson, Y. Fan, and M. M. Tanaka. Sequential Monte Carlo without likelihoods. *Proc. Natl. Acad. Sci*, 104:1760–1765, 2007.

S. A. Sisson, Y. Fan, and M. M. Tanaka. Correction: Sequential Monte Carlo without likelihoods. *Proc Natl Acad Sci*, 54:e1760, 2009.

A. Sitz, U. Schwarz, J. Kurths, and H. U. Voss. Estimation of parameters and unobserved components for nonlinear systems from noisy time series. *Physical Review E*, 66 (016210), 2002.

I. Y. M. Sobol'. On sensitivity estimation for nonlinear mathematical models. *Mathematical Modeling and Comput*, 1(4):407–414, 1990.

J. Srividhya and M. S. Gopinathan. A simple time delay model for eukaryotic cell cycle. *Journal of theoretical biology*, 241(3):617–627, 2006.

X. Sun, L. Jin, and M. Xiong. Extended Kalman filter for estimation of parameters in nonlinear state-space models of biochemical networks. *PLoS ONE*, 3(11):e3758, 11 2008.

I. Swameye, T. G. Müller, J. T. Timmer, O. Sandra, and U. Klingmüller. Identification of nucleocytoplasmic cycling as a remote sensor in cellular signaling by databased modeling. *Proceedings of the National Academy of Sciences*, 100(3):1028–1033, 2003.

S. Tavaré, D. J. Balding, R. C. Griffiths, and P. Donnelly. Inferring coalescence times from dna sequence data. *Genetics*, 145:505–518, 1997.

B. Teusink, F. Baganz, H. V. Westerhoff, and S. G. Oliver. Metabolic control analysis as a tool in the elucidation of the function of novel genes. *Methods Microbiol*, 26: 297–336, 1998.

B. Teusink, J. Passarge, C. A. Reijenga, E. Esgalhado, C. C. van der Weijden, M. Schepper, M. C. Walsh, B. M. Bakker, K. van Dam, H. V. Westerhoff, and J. L. Snoep. Can yeast glycolysis be understood in terms of *in vitro* kinetics of the constituent enzymes? Testing biochemistry. *Eur. J. Biochem*, 267:5313–5329, 2000.

T. Toni, D. Wlech, N. Strelkowa, A. Ipsen, and M. P. H. Stumpf. Approximate Bayesian computation scheme for parameter inference and model seletion in dynamical systems. *J. R. Soc*, 6:187–202, 2009.

E. Triantaphyllou and A. Sánchez. A sensitivity analysis approach for some deterministic multi-criteria decision-making methods. *Decision Sciences*, 28(1):151–194, 1997.

H-C. Tseng, Y. Zhou, Y. Shen, and L-H. Tsai. A survey of cdk5 activator p35 and p25 levels in Alzheimer's disease brains. *FEBS Letters*, 523:58–62, 2002.

J. J. Tyson. Modeling the cell division cycle: cdc2 and cyclin interactions. *Proceedings of the National Academy of Sciences*, 88(16):7328–7332, 1991.

J. J. Tyson and B. Novak. Regulation of the eukaryotic cell cycle: molecular antagonism, hysteresis, and irreversible transitions. *Journal of Theoretical Biology*, 210(2):249–263, 2001.

R. Van Der Merwe and E. A. Wan. The square-root unscented Kalman filter for state and parameter-estimation. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, volume 6, pages 3461–3464. IEEE, 2001.

R. J. van Wegen, S-Y. Lee, and A. P. J. Middelberg. Metabolic and Kinetic Analysis of Poly(3-Hydroxybutyrate) Production by Recombinant *Escherichia coli*. *Biotechnology and Bioengineering*, 74(1):70–81, 2001.

D. V. Vavoulis, V. A. Straub, J. A. Aston, and J. Feng. A self-organizing state-space-model approach for parameter estimation in Hodgkin-Huxley-type models of single neurons. *PLoS computational biology*, 8(3):e1002401, 2012.

S. G. Villas-Boas, J. Nielsen, J. Smedsgaard, M. A. Hansen, and U. Roessner-Tunali. *Metabolome analysis: an introduction*, volume 24. John Wiley & Sons, 2007.

V. Volterra. Variazioni e fluttuazioni del numero d'individui in soecie animali conviventi. *Men. R. Acad. Naz. dei Lincei*, 2:31–113, 1926.

L. Von Bertalanffy. *General system theory: Foundations, development, applications.* George Braziller, NewYork, 1968.

V. Vyshemirsky and M. A. Girolami. BioBayes: A software package for Bayesian inference in systems biology. *Bioinformatics*, 24(17):1933–1934, 2008.

J. D. F. Wadsworth, S. Joiner, A. F. Hill, T. A. Campbell, M. Desbruslais, P. J. Luthert, and J. Collinge. Tissue distribution of protease resistant prion protein in variant Creutzfeldt-Jakob disease using a highly sensitive immunoblotting assay. *LANCET*, 359:171–180, 2001.

J. D. Wall. A comparison of estimators of the population recombination rate. *Mol. Biol. Evol*, 17:156–163, 2000.

F. Wang and S. Y. Lee. Production of poly(3-hydroxybutyrate) by fed-batch culture of filamentation-suppressed recombinant *Escherichia coli. Appl Environ Microbiol*, 63: 4765–4769, 1997.

P. Weber, J. Hasenauer, F. Allgöwer, and N. Radde. Parameter estimation and identifiability of biological networks using relative data. In *In Proceedings of the 18th IFAC World Congress*, pages 11648–11653, 2011.

J. Wiedenmann, S. Ivanchenko, F. Oswald, F. Schmitt, C. Röcker, A. Salih, K-D. Spindler, and G. U. Nienhaus. EosFP, a fluorescent marker protein with UV-inducible green-to-ren fluorescence conversion. *PNAS*, 101:15905–15910, 2004.

C. K. Wikle, M. Berliner, and N. Cressie. Hierarchical Bayesian space-time models. *Environ. Ecolog. Statist.*, 5:43–50, 1998.

D. J. Wilkinson. Stochastic modeling for quantitative description of heterogeneous biological systems. *Nat. Rev. Genet*, 10:122–133, 2009.

A. P. Wlaschin, C. T. Trinh, R. Carlson, and F. Srienc. The fractional contributions of elementary modes to the metabolism of *escherichia coli* and their estimation from reaction entropies. *Metabolic Engineering*, 8:338–352, 2006.

J. Wodzinska, K. D. Snell, A. Rhomberg, A. J. Sinskey, K. Biemann, and J. Stubbe. Polyhydroxybutyrate synthase: Evidence for covalent catalysis. *J. Am. Chem. Soc.*, 118:6319–6320, 1996.

H. H. Wong, R. J. Van Wegen, J. I. Choi, Lee. S. Y., and A. P. J. Middelberg. Metabolic analysis of poly(3-hydroxyburate) production by recombinant *Escherichia coli. J. Microbiol Biotechnol*, 9:593–603, 1999.

L. Wu, W. V. Winden, W. V. Gulik, and J. J. Heijnen. Application of metabolome data in functional genomics: A conceptual strategy. *Metab. Eng*, 7:302–310, 2005.

J. Yang, V. Kadirkamanathan, and S. A. Billings. In *vivo* intracellular metabolite dynamics estimation by sequential Monte Carlo filter. In *Computational Intelligence and Bioinformatics and Computational Biology, 2007. CIBCB '07. IEEE Symposium on*, pages 387 –394, april 2007.

K. Yuan, M. Niranjan, and M. Girolami. Markov chain Monte Carlo methods for state-space models with point process observations. *Neural Computation*, 24(6):1462–1486, 2012.

S. L. Zeger and M. R. Karim. Generalized linear models with random effects; a Gibbs sampling approach. *Journal of the American statistical association*, 86(413):79–86, 1991.

C. Zhan and L. Yeung. Parameter estimation in systems biology models using spline approximation. *BMC Systems Biology*, 5(1):14, 2011.

H. Zhang, A. V. Holden, I. Kodama, H. Honjo, M. Lei, T. Varghese, and M. R. Boyett. Mathematical models of action potentials in the periphery and center of the rabbit sinoatrial node. *American Journal of Physiology - Heart and Circulatory Physiology*, 279(1):397–421, 2000.

L. X. Zhang and E. L. Boukas. Stability and stabilization of Markovian jump linear systems with partly unknown transition probabilities. *Automatica*, 45:463–468, 2009.

Y. Zheng and A. Rundell. Comparative study of parameter sensitivity analyses of the TCR-activated Erk-MAPK signalling pathway. *IEE Proceedings-Systems Biology*, 153 (4):201–211, 2006.

Z. Zi. Sensitivity analysis approaches applied to systems biology models. *IET Syst. Biol.*, 5(6):336–346, 2011.