Statistical Learning of Some Complex Systems: From Dynamic Systems to Market Microstructure

Statistical Learning of Some Complex Systems: From Dynamic Systems to Market Microstructure

A dissertation presented
by

Xiao Tong

to
The Department of Statistics
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy
in the subject of

Statistics

Harvard University
Cambridge, Massachusetts
May 2013

Professor S.C. Samuel Kou **Xiao Tong**

Statistical Learning of Some Complex Systems: From Dynamic Systems to Market Microstructure

Abstract

A complex system is one with many parts, whose behaviors are strongly dependent on each other. There are two interesting questions about complex systems. One is to understand how to recover the true structure of a complex system from noisy data. The other is to understand how the system interacts with its environment. In this thesis, we address these two questions by studying two distinct complex systems: dynamic systems and market microstructure. To address the first question, we focus on some nonlinear dynamic systems. We develop a novel Bayesian statistical method, Gaussian Emulator, to estimate the parameters of dynamic systems from noisy data, when the data are either fully or partially observed. Our method shows that estimation accuracy is substantially improved and computation is faster, compared to the numerical solvers. To address the second question, we focus on the market microstructure of hidden liquidity. We propose some statistical models to explain the hidden liquidity under different market conditions. Our statistical results suggest that hidden liquidity can be reliably predicted given the visible state of the market.

Contents

Acknowledgments

Five years ago I was overjoyed about coming to Harvard for my PhD. I was not fully aware that this new path would have several ups and downs. Research is extremely different from the common college experience, where knowledge is being taught, while in research we seek to push the boundaries of our knowledge. Research is more like a voyage of discovery into unknown lands. I feel extremely lucky to have Professor Samuel Kou as my advisor to keep my ship in the right direction. Without his constant inspiration and careful guidance my journey would have been completely different.

I also owe a great deal to my other committee members. Professor Tirthankar Dasgupta guided me in experimental design and computer experiment. I am very honored to have him in my committee. I also want to thank Professor Xiao-Li Meng for his guidance in the Statistics department and generous help on serving on my committee. Moreover, I also want to thank Mr. Steven Finch, who proofread my thesis and gave me very useful advice.

I also want to show my sincere gratitude to all great scholars with whom I worked. Working with Dr. Ruihong Huang and Professor Nikolaus Hautsch at Humboldt-Universität zu Berlin on the market microstructure of hidden liquidity is always intriguing. Their invitation to Germany was generous and made my dream come true. I also enjoyed the enlightening discussion with my colleague and friend, Samuel Wong, on dynamic systems. I also want to thank other faculty members within statistics department, Prof. Jun Liu, Carl Morris, Donald Rubin, Joseph Blitzstein, Yoonjung Lee, Edo Airoldi and Stephen Blyth for their interesting courses and seminars that shape my studies and research.

I would like to thank my friends, colleagues and staffs at the Department of Statistics at Harvard University. I enjoyed all the moments in the department. The soccer games we played in the indoor soccer league and weekends, as well as the games and drinks we had, will always be part of my memory.

I would like to thank all the teachers, professors and scholars from whom I have learned in my life. I received my elementary and high-school education in China while went to western-style college and graduate school. The fascinating blend of Chinese culture and western influence was deeply rooted in my heart, where each culture accepts from the other. Therefore, I quoted from one of my favorite Chinese philosophers, Lao Tzu, for each chapter in my thesis to show my appreciation and how Chinese philosophy can be connected to the western scientific approach.

I also want to thank Bayern München and their fans worldwide. It is Bayern that makes my Ph.D. life never alone. *One day FC Bayern, always FC Bayern.*

Last but most importantly, I wish to thank my parents, my grandparents and fiancée Pan, for their consistent help, love and support during my PhD study. This dissertation is dedicated to them.

To my parents, my grandparents and Pan

Chapter 1

Introduction

Non-linearity begets completeness;
Misjudgment creates linearity.

- Lao Tzu

This thesis presents work on statistical learning of two complex systems: dynamic systems and market microstructure. Complex systems are ones with a large effective number of variables that interact in complicated and nonlinear ways. People often treat a dynamic system as a physical system with particles or genes as the main participants in the complex system. On the other hand, market microstructure is a social complex system with human beings as the main participants in the complex system.

We first study dynamic systems. The study of dynamic systems can be traced back to an early era of modern science. Physicists had developed various differential equations to model dynamic systems. For example, Newton's laws of motion are fundamental tools to understand the dynamics of a particle or a small body. Linear

differential equations such as Newton's, Maxwell's and Schroedinger's can adequately describe the reality of the physical world since they are basically equations of forces in a vacuum. However, linear differential equations fail to capture the dynamics of biological processes due to the inherent complexity of biology. Nonlinearity becomes a fundamental property of biological systems. Nonlinearity has been witnessed ubiquitously in many biological systems due to the fact that it is fundamental in generating structural changes in complex phenomenon in biology. As Lao Tzu, a famous Chinese philosopher, put it: Nonlinearity begets completeness; Misjudgment creates linearity. We focus mainly on the nonlinear dynamic systems emerged in biology and neuroscience. In particular, we are interested in the *inverse* problem of dynamic systems, which is the statistical inference of the inner structure of dynamic systems from observed data.

Dynamic systems are often described by a set of linear or nonlinear ordinary differential equations, which we may denote by

$$\frac{dx(\mathbf{t})}{d\mathbf{t}} = f(x, \mathbf{t}|\boldsymbol{\Theta})$$

where the vector $x(\mathbf{t})$ contains the values of the system outputs at time $\mathbf{t} \in [\mathbf{0}, \mathbf{T}]$, and Θ is a vector of parameters which may not be known from experimental data, theoretical considerations or other sources of information. From the physical or biological experiments whose underlying processes are usually described by differential equations, experimental data are recorded at discrete time points. Due to some measurement error, these data may be noisy. Suppose the dynamic system has N components, and for each observable system component $i, 1 \leq i \leq N$, and recorded time points

$0 < t_1 < t_2 < ... < t_n$, we obtain data $y_i(t_j), 1 \leq j \leq n$, where $y_i(t_j) = x_i(t_j) + e_{ij}$. Here, e_{ij} is the measurement error, assumed for now to be iid and normal with mean zero and variance σ^2. The motivation is to estimate the parameters Θ of the dynamic system, given those collected noisy data $\{y_i(t_j)\}$.

Chapters 2 and 3 focus on the statistical learning of dynamic systems. Current methods for estimating parameters in dynamic systems from noisy data are computationally intensive and rely heavily on the numerical solutions of underlying differential equations. In Chapter 2, we propose a new Bayesian scheme, which creates an artificial system driven by a Gaussian process to approximate the dynamic system. We introduce an auxiliary variable that connects this artificial Gaussian system to the real dynamic system and further design a sampling scheme enabling our artificial Gaussian system to emulate the real dynamic system as close as desired. In Chapter 3, we further impose a hierarchical structure to model the means of different components in our Gaussian system, providing an efficient and accurate estimate of the trajectory of the missing components when dealing with partially observed data. We illustrate this method, named as Gaussian Emulator, by numerical examples ranging from neuroscience to system biology, in both complete and partially observed cases, resulting in a dramatic saving of computational time and fast convergence while still retaining a precise estimation accuracy.

While dynamic systems can now point to a solid record of scientific accomplishments, improving our understanding of processes from neuroscience to biology, market microstructure is far less understood. Chapter 4 is dedicated to understanding this complex social system, which is much more complicated than many physical systems

since human beings, unlike particles or genes, often under- or over-react to information when they make economic decisions. So any attempt to use differential equations to capture intrinsic details of market microstructure would be unavoidably brittle. Therefore, data-driven statistical learning would be a more proper tool utilized to understand this complex system.

Chapter 4 focuses on the statistical learning of market microstructure, especially the microstructure of hidden liquidity. Recent assessment of electronic limit order markets shows a growing use of undisclosed orders. We provide *first* evidence on how trading against hidden liquidity provided by these orders can improve transaction prices for small retail investors and large institutional investors. We propose statistical models to conduct statistical inference on the price improvement due to the possible existence of hidden liquidity by studying ModelView data on (hidden) limit order book information as well as TotalView data on disclosed order activities at NASDAQ. The analysis of a cross-section of stocks shows that market conditions reflected by the (visible) bid-ask spread, (visible) depth, recent trading signals, executed hidden volumes, and aggressive orders updated by low-frequency traders affect the price improvement. Our empirical evidence indicates that price improvement increases when traders compete for the provision of hidden liquidity, while it decreases when they protect themselves against adverse selection if that risk becomes very high. Our statistical models show the dynamic and interactive effects of some market conditions on the price improvement due to hidden liquidity. Overall, our empirical results suggest that price improvements by hidden liquidity are reliably predictable given the visible state of the market.

Chapter 2

The Precursor of Gaussian Emulator: an Approximate Bayesian Approach for Inference of Dynamic Systems

Receding, it is described as far away.
Being far away, it is described as turning back.
- Lao Tzu

2.1 Introduction

2.1.1 Background

Dynamic systems are used in modeling diverse behaviors in a wide variety of sciences, engineering and economics. Scientists often use differential equations to model the dynamic systems based on their understanding of the systems. However, the parameters of differential equations are unknown. It is crucial to infer those parameters since they often have important scientific interpretations.

In the last several decades, extensive research has been conducted on estimating the parameters of deterministic dynamic systems. Biegler et al. [1986] proposed a nonlinear least squares (NLS) method to tackle this problem: first, we select a trial set of parameters and use a numerical method (such as Euler discretization and Runge-Kutta) to approximate the solution given the parameters and initial conditions, obtaining $\hat{x}_i(t_j|\Theta)$ for observed component $i = 1, 2, ..., N$ and time $j = 1, ..., n$. Then we compute $\hat{\Theta}$ to minimize

$$\sum_{i=1}^{N}\sum_{j=1}^{n}(y_i(t_j) - \hat{x}_i(t_j|\Theta))^2.$$

This method depends heavily on the numerical solutions of differential equations, which could be extremely slow if the systems are very stiff. Also, this method assumes that the initial conditions to differential equations are known, which could be unrealistic.

To circumvent the numerical solutions of differential equations, Ramsay et al. [2007] proposed a generalized smoothing method by expressing $x_i(\mathbf{t})$ in terms of a basis

function expansion, $x_i(\mathbf{t}) = \sum_{\mathbf{k=1}}^{\mathbf{K_i}} \mathbf{c_{ik}}\phi_{\mathbf{ik}}(\mathbf{t})$, where the number K_i of basis functions (splines) in vector ϕ_i is chosen so as to allow enough flexibility in the behavior of $x_i(\mathbf{t})$ to satisfy the dynamic system. In their profiling procedure, the nuisance parameters are defined to be *implicit* functions $\hat{c}_i(\Theta, \sigma; \lambda)$ of the structural parameters so that once Θ and σ are changed, an *inner* fitting criterion $J(\hat{c}|\Theta, \sigma, \lambda)$ is reoptimized with respect to $\hat{c}$ alone:

$$J(\hat{c}|\Theta, \sigma, \lambda) = \sum_{i=1}^{N}\sum_{j=1}^{n} \log L(y_i|x(t_j)) - \lambda \sum_{i=1}^{N} \int (\dot{x}_i(\mathbf{t}) - \mathbf{f_i}(\mathbf{X}(\mathbf{t})|\mathbf{\Theta}))^{\mathbf{2}}\mathbf{dt}.$$

The estimating function $\hat{c}_i(\Theta, \sigma; \lambda)$ is *regularized* by incorporating a penalty term in J that controls the extent to which $\hat{x}_i(\mathbf{t}) = \sum_{\mathbf{k=1}}^{\mathbf{K_i}} \mathbf{\hat{c}_{ik}}\phi_{\mathbf{ik}}(\mathbf{t})$ does not satisfy the differential equation exactly. In other words, the penalty term is defined using the ODE model, which forces the nonparametric basis to satisfy the ODE model and penalizes the roughness of the nonparametric basis. The amount of regularization is controlled by smoothing parameters λ. A data fitting criterion $H(\Theta, \sigma|\lambda)$, which can be taken to be negative log-likelihood, is then optimized with respect to the structural parameters alone. Their method on some test examples shows that they can estimate parameters of interest successfully.

Another line of frequentists' approach is to view the dynamic system as a state-space model. Estimating parameters for state space models is rather easy if the parameters are time-varying random variables, which could be proceeded by using standard techniques for filtering and smoothing. However, difficulty arises when the true parameters are not to vary with time. Ionides et al. [2006] proposed a new method that is based on a sequence of filtering operations which are shown to converge to a

maximum likelihood parameter estimate. Their approach showed how time-varying parameter algorithms may be harnessed for use in statistical inference in the fixed-parameters in dynamic systems.

Chen and Wu [2008] proposed a two-stage estimator for time-varying parameters in dynamic systems. The first stage is to use a standard local polynomial regression method to estimate $x_i(\mathbf{t})$ at a given time t_0. Based on Taylor expansion, $x_i(t_j), j = 1, ..., n$, is approximated locally by

$$x_i(t_j) \approx \beta_0(t_0) + \beta_1(t_j - t_0) + ... + \beta_p(t_j - t_0)^p$$

for t_j in a neighborhood of a given time point t_0. Let $\beta = (\beta_0(t_0), \beta_1(t_1), ..., \beta_n(t_n))$, then the local polynomial estimator $\hat{\beta}$ can be estimated by minimizing a locally weighted least squares criterion

$$\sum_{j=1}^{n} [y_i(t_j) - (\beta_0(t_0) + \beta_1(t_j - t_0) + ... + \beta_p(t_j - t_0)^p)]^2 K_h(t_j - t_0)$$

where K is a kernel function and h is a bandwidth. The second stage is to substitute estimators $\hat{x}_i(t_j)$ into the ODEs, and again apply local polynomial regressions to estimate time-varying parameters. Some innovative features of their methods include that they do not solve the differential equations numerically. Moreover their methods are regression-based approaches and thus can avoid the high computational cost and possible numerical errors caused by numerical evaluation of nonlinear differential equations.

In parallel to frequentists' approaches, Gelman et al. [1996] suggested a possible

Bayesian approach by letting the observed y_i at time t_j follow

$$y_i(t_j)|\Theta, \sigma^2 \sim N(\hat{x}_i(t_j|\Theta), \sigma^2) \tag{2.1}$$

where $\hat{x}_i(t_j|\Theta)$ denotes the numerical solution of the ODE system given the set of parameters. If we choose a prior π for Θ and σ^2, then the posterior distribution of Θ and σ^2 can be easily obtained:

$$p(\Theta, \sigma^2|\{y_i(t_j)\}) \propto \pi(\Theta, \sigma^2) \prod_{i,j} N(\hat{x}_i(t_j|\Theta), \sigma^2).$$

We use the Metropolis-Hastings algorithm to update Θ and draw from its posterior density.

Huang et al. [2006] subsequently extended this Bayesian approach with mixed-effects modeling techniques to estimate both population and individual parameters of human immunodeficiency virus (HIV) dynamic systems under a framework of the hierarchical Bayesian nonlinear (mixed-effects) model. A common difficulty encountered by such Bayesian methods is that the sampling scheme relies heavily on numerical solutions, which could be very computationally expensive.

2.1.2 Ideal sampling under Bayesian framework

Under the Bayesian framework, in order to circumvent the numerical solutions to differential equations, we view $x_i(\mathbf{t})$ and Θ as two separate random variables, in which we can propose a proper prior on $x_i(\mathbf{t})$. In this case, we can do exact sampling as follows:

- sample σ^2 from $p(\sigma^2|\{y_i(\mathbf{t})\}) \sim \pi(\sigma^2)\prod_i p(y_i(\mathbf{t})|\sigma^2)$
- sample $x_i(\mathbf{t})$ from $p(x_i(\mathbf{t})|\{y_i(\mathbf{t})\}, \sigma^2)$
- sample Θ from $p(\Theta|\{x_i(\mathbf{t})\}, \{y_i(\mathbf{t})\}, \sigma^2)$

As a first step, we must evaluate $p(y_i(\mathbf{t})|\sigma^2)$, which is equal to

$$\begin{aligned} & \int p(y_i(\mathbf{t}), x_i(\mathbf{t})|\sigma^2)dx_i(\mathbf{t}) \\ = & \int p(y_i(\mathbf{t})|x_i(\mathbf{t}), \sigma^2)p(x_i(\mathbf{t})|\sigma^2)dx_i(\mathbf{t}) \end{aligned}$$

Even though $p(y_i(\mathbf{t})|x_i(\mathbf{t}), \sigma^2)$ follows a normal distribution, $p(x_i(\mathbf{t})|\sigma^2)$ might not be a normal distribution. As a result, the integral is computationally intractable, which makes the rest of the sampling scheme infeasible. This motivates us to devise an approximation sampling scheme that would result in accurate estimation and fast computation.

2.1.3 The approximate Bayesian approach

Our innovative approach is, instead of working with the dynamic system alone, we create an artificial system that is a mirror of the dynamic system. The true solutions of the dynamic system have their counterparts in the artificial system. We put a Gaussian process prior on the counterparts in this artificial system to make the computational difficulty from the ideal sampling case feasible. We then use this artificial system to approximate the true dynamic system as closely as possible. As a result, we replace the true solution of dynamic systems by the counterparts in the artificial

system, and design an efficient MCMC sampling scheme for ODE parameters. This approximation method allows for more computationally efficient and more convenient inference of ODE parameters and still retains great estimation accuracy.

The remainder of this chapter is organized as follows: Section 2.2 introduces two test bed examples: FitzHugh-Nagumo equations widely used in neuroscience and a repressilator model in system biology. These two models are used for the purpose of illustration throughout the Chapter 2 and Chapter 3. Section 2.3 introduces the approximate Bayesian inference. Section 2.4 demonstrates our methods to the two simulation examples motivated by the two systems in Section 2.2. Section 2.5 extends our approximate Bayesian framework to the case where only partial components of dynamic systems have been observed, along with numerical illustrations. The chapter concludes with a discussion in Section 2.6.

2.2 Two Illustrative Examples

2.2.1 FitzHugh-Nagumo equations

The FitzHugh-Nagumo equations were developed by FitzHugh [1961] and Nagumo et al. [1962] to model the behavior of spike potentials in the giant axon of squid neurons:

$$\begin{aligned} \frac{dV}{dt} &= c(V - \frac{V^3}{3} + R), \\ \frac{dR}{dt} &= -\frac{1}{c}(V - a + bR). \end{aligned}$$

The system describes the reciprocal dependences of the voltage V across an axon membrane and a recovery variable R summarizing outward currents. As Wilson [1999] pointed out, solutions to the FitzHugh-Nagumo equations do exhibit that are common to elements of biological neural networks. It was also used by Ramsay et al. [2007] as a test example for their generalized smoothing method.

The parameters of interest are $\theta = \{a, b, c\}$ and we assign values (0.2, 0.2, 1) and initial conditions $(V, R) = (-1, 1)$ respectively. While the R-equation can be viewed as a simple linear system $\frac{dR}{dt} = -\frac{b}{c}R$ with linear inputs V and a, the V-equation is non-linear. V exhibits nearly exponential increase when V is small so that $\frac{dV}{dt} \approx cV$. However, as V passes $\pm\sqrt{3}$, the influence of -$V^3/3$ would pull V back towards 0. Therefore, the solutions would alternate between smooth evolution and the sharp changes in direction as shown in Figure 2.1.

Another concern in dynamic systems modeling is the irregular shape of the fitted surface. For example, Figure 2.1 (right) displays the likelihood surface of the simulated data of FitzHugh-Nagumo equations given the ODE parameters and initial conditions above. The features of this surface include some sharp change in behavior.

2.2.2 Modeling transcriptional regulatory network

In transcriptional regulatory networks, for example, a transcription process (including transcript elongation, splicing, processing, and export from the nucleus to the cytoplasm) may be modeled as a nonlinear ODE, see Cosentino and Bates [2011]:

$$\frac{dM}{dt} = -M(t) + g(P(t))$$

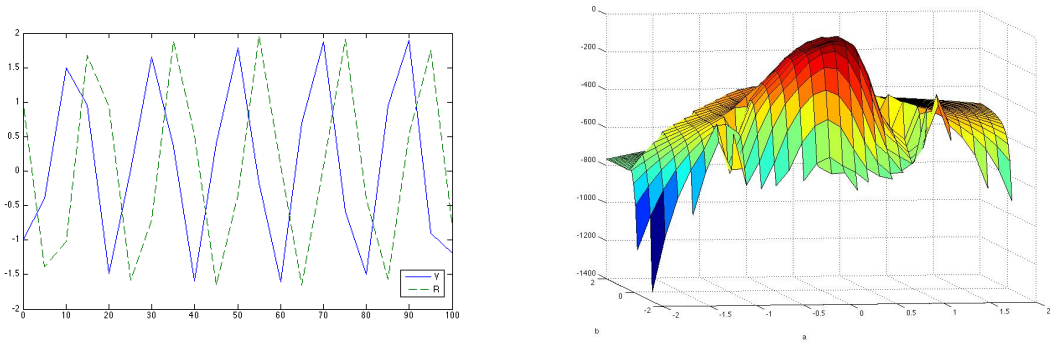

Figure 2.1: **Left:** The solutions of FitzHugh-Nagumo equations with parameter values $a = 0.2, b = 0.2$ and $c = 1.0$ and initial conditions $(V, R) = (-1, 1)$. **Right:** The likelihood surface of FitzHuge-Nagumo model.

where $M(t)$ and $P(t)$ represent the concentrations of mRNA and protein respectively. The function g describes how TFs (transcriptional factors) regulate the transcription of one gene, and experimental evidence suggests the response of mRNA to TF concentrations has a nonlinear Hill curve form, see Elowitz and Leibler [2000]. Here we take the nonlinear functional form to be $g(P(t)) = \frac{a}{1+P(t)^b}$, where b is a Hill coefficient. A translation process may be considered as a linear ODE:

$$\frac{dP}{dt} = c(M(t) - P(t))$$

where c denotes the ratio of the protein decay rate to the mRNA decay rate.

Elowitz and Leibler [2000] designed an artificial genetic network to understand the functioning of the network of the interacting biomolecules. They proposed three transcriptional repressor systems, where each gene transcribes the repressor protein for the next gene in the loop, to build an oscillating network. The resulting oscillations are fairly consistent with the experiment results. In the repressilator network graph

in Figure 2.2, the first repressor protein, *lacI* from *E.coli*, inhibits the transcription of the second repressor gene, *tetR* from the tetracycline-resistance transposon Tn10, whose protein product in turn inhibits the expression of a third gene, *cI* from phage. Finally, *cI* inhibits *lacI* expression, completing the cycle.

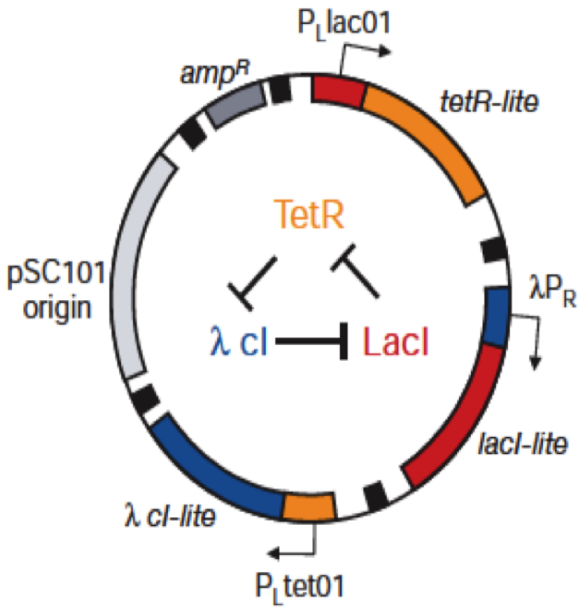

Figure 2.2: The network graph for three genes' interaction from Elowitz and Leibler (2000).

Three repressor-protein concentrations, p_i, and their corresponding mRNA concentrations, m_i(where i is $lacI, tetR$ or cI) were treated as continuous dynamic variables. The kinetics of the system are determined by six coupled first-order differential

equations:

$$\frac{dm_i}{dt} = -m_i + \frac{a}{1 + p_j^b}$$
$$\frac{dp_i}{dt} = c(m_i - p_i)$$

$i = (lacl, tetR, cl)$ and $j = (cl, lacl, tetR)$,

2.3 Approximate Bayesian Inference

The original dynamic system can be represented graphically in Figure 2.3. For notation ease, we use $Y = (y_1(\mathbf{t}), ..., y_N(\mathbf{t}))$ and $X = (x_1(\mathbf{t}), ..., x_N(\mathbf{t}))$ respectively.

2.3.1 Creating a Gaussian system

Imagine we are in an artificial system, which is the mirror of this dynamic system in the sense that the it has the counterpart of X, Y and σ^2 from the dynamic system. While the parameter of interests Θ is the main determinant of X in the dynamic system, a hyperparameter α plays the similar role in the artificial system. Let $y_{G,i}(\mathbf{t}), x_{G,i}(\mathbf{t}), i = 1, ..., N$ and σ_G^2 be the counterparts of $y_i(\mathbf{t}), x_i(\mathbf{t})$ and σ^2 in this artificial system. Similarly, we use $Y_G = (y_{G,1}(\mathbf{t}), ..., y_{G,N}(\mathbf{t}))$ and $X_G = (x_{G,1}(\mathbf{t}), ..., x_{G,N}(\mathbf{t}))$ respectively. The graphical representation of the artificial system is shown in Figure 2.4. The motivation of creating such an artificial system is to make all the steps of ideal sampling work in this system. In particular, the artificial system needs to be flexible and capable of approximating a wide range of functions and processes, and also allow fast computation. However, we must be aware that

this artificial system has nothing to do with the dynamic system so far. In the later sections, we will connect this artificial system to the real dynamic system and design an efficient sampling scheme to estimate the parameter Θ.

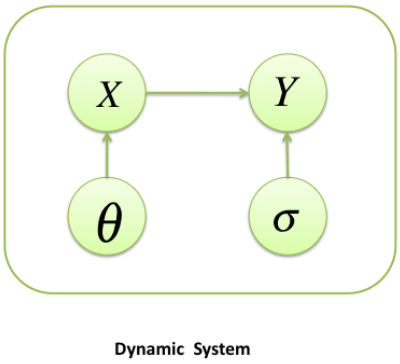

Figure 2.3: The graphical representation of a dynamic system. Θ is the parameter of interests.

Assume $y_{G,i}(\mathbf{t}) \sim N(x_{G,i}(\mathbf{t}), \sigma_G^2)$. We further assume that $y_{G,1}(\mathbf{t}), ..., y_{G,N}(\mathbf{t})$ are independent conditioning on X_G. In order to make this artificial system approximate a wide range of functions and processes, we have to pick up a proper prior for $x_{G,i}(\mathbf{t})$, and the Gaussian process is an ideal candidate. In probability theory and statistics, a Gaussian process is a stochastic process whose realizations consist of random variables associated with every point in an interval of time (or a region of space) such that each has a normal distribution. By focusing on processes that are *Gaussian*, it turns out that the computations required for inference and learning become relatively easy. For example, if a random process is modeled as a Gaussian process, the dis-

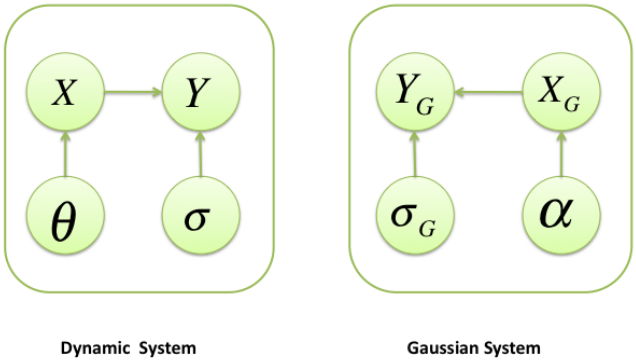

Figure 2.4: The graphical representation of the artificial system. X_G, Y_G and σ_G^2 are the counterparts of X, Y and σ^2 from the dynamic system. The artificial system needs to be flexible and capable of approximating a wide range of functions and processes, and also allow fast computation.

tributions of various derived quantities can be obtained explicitly. Theoretical and practical developments over the last decade have made Gaussian processes a serious competitor for statistical inference and learning especially in the Bayesian paradigm. The wide use of Gaussian processes can been found in spatial statistics (see Stein [1999]), computer experiment (see Fielding and Liong [2011]) and machine learning literature (see Rasmussen and Williams [2004]).

Back to our artificial system, we impose a Gaussian process prior on $x_{G,i}(\mathbf{t})$ with hyperparameters α_i so the prior distribution of $x_{G,i}(\mathbf{t}) = (x_{G,i}(t_1), ..., x_{G,i}(t_n))$ follows a multivariate normal distribution with mean $\mu_{G,i}(\mathbf{t})$, an n-by-1 vector, and n-by-n covariance matrix C_{α_i}. We call this artificial system a "Gaussian system".

Then the posterior distribution of $x_{G,i}(\mathbf{t})$ is

$$p_{\alpha_i}(x_{G,i}(\mathbf{t})|y_{G,i}(\mathbf{t}), \sigma_G^2) \sim N(\mu_i, \Sigma_i) \tag{2.2}$$

where $\mu_i = \mu_{G,i}(\mathbf{t}) + C_{\alpha_i}(C_{\alpha_i} + \sigma^2 I_n)^{-1}(y_{G,i}(\mathbf{t}) - \mu_{G,i}(\mathbf{t}))$ and $\Sigma_i = \sigma_G^2 C_{\alpha_i}(C_{\alpha_i} + \sigma^2 I_n)^{-1}$.

Further, the conditional distribution of $\dot{x}_{G,i}(\mathbf{t})$, defined as $\frac{dx_{G,i}}{dt}$ evaluated at $t = \mathbf{t}$, is easily obtained:

$$p_{\alpha_i}(\dot{x}_{G,i}(\mathbf{t})|y_{G,i}(\mathbf{t}), \sigma_G^2) \sim N(a_i, b_i) \tag{2.3}$$

Here $a_i = \dot{\mu}_{G,i}(\mathbf{t}) + C'_{\alpha_i}(C_{\alpha_i} + \sigma_G^2 I_n)^{-1}(y_{G,i}(\mathbf{t}) - \mu_{G,i}(\mathbf{t}))$ and $b_i = C''_{\alpha_i} - C'_{\alpha_i}(C_{\alpha_i} + \sigma_G^2 I_n)^{-1} C'_{\alpha_i}$, where C'_{α_i} =Cov$(\dot{x}_{G,i}(\mathbf{t}), x_{G,i}(\mathbf{t}))$, C''_{α_i} =Cov$(\dot{x}_{G,i}(\mathbf{t}), \dot{x}_{G,i}(\mathbf{t}))$ provided that they exist. The derivation of (2.2) and (2.3) can be seen in the Appendix A.

2.3.2 Covariance functions of Gaussian processes

It is obvious that the functional forms of a_i and b_i are determined by the covariance functions for Gaussian process priors, namely C_{α_i}. Clearly, a covariance function is a crucial ingredient of any Gaussian process. One assumption for choosing covariance functions is that points with inputs $\mathbf{t}$ which are close are likely to have similar target values $x(\mathbf{t})$. For example, one of the most widely-used covariance functions is the *squared exponential (SE)* covariance function, which has the form

$$C_{\alpha_i}(t_p, t_q) = \sigma_{C,i}^2 e^{-\frac{r^2}{2\alpha_i'^2}}$$

where $\alpha_i = (\sigma^2_{C,i}, \alpha'_i)$, and $\sigma^2_{C,i}$ denotes the signal variance and hyperparameter α'_i denotes the length-scale for the component i, and $r = |t_p - t_q|$. Expressions for C'_{α_i} and C''_{α_i} for SE covariance function are derived in Appendix A. This covariance function is infinitely differentiable, which means that a Gaussian process with this covariance function has mean square derivatives of all orders and thus is very smooth. Stein [1999] argues that such strong smoothness assumptions are unrealistic for modeling many physical processes and instead recommends the *Matern* class.

The *Matern class* of covariance functions is given by

$$C_{\alpha_i} = \sigma^2_{C,i} \frac{2^{1-v}}{\Gamma(v)} (\frac{\sqrt{2v}r}{\alpha'_i})^v K_v (\frac{\sqrt{2v}r}{\alpha'_i})$$

where K_v is a modified Bessel function, see Stein [1999]. Notice that when $v \to \infty$, we obtain the SE covariance function. Stein [1999] named this the Matern class after the work of *Matern*. The process with a Matern covariance function is k-times mean square differentiable if and only if $v > k$. So in order to ensure that C'_{α_i} and C''_{α_i} exist, we would take $v > 2$. Rasmussen and Williams [2004] pointed out that the Matern covariance functions become especially simple when v is half-integer: $v = p + 1/2$, where p is a non-negative integer. For example, when $v = 3/2, 5/2$, covariance functions have the form:

$$\begin{aligned} C_{v=3/2,\alpha_i} &= \sigma^2_{C,i}(1 + \frac{\sqrt{3}r}{\alpha_i}) \exp(-\frac{\sqrt{3}r}{\alpha_i}) \\ C_{v=5/2,\alpha_i} &= \sigma^2_{C,i}(1 + \frac{\sqrt{5}r}{\alpha_i} + \frac{5r^2}{3\alpha_i^2}) \exp(-\frac{\sqrt{5}r}{\alpha_i}) \end{aligned}$$

These covariance functions are illustrated in Figure 2.5.

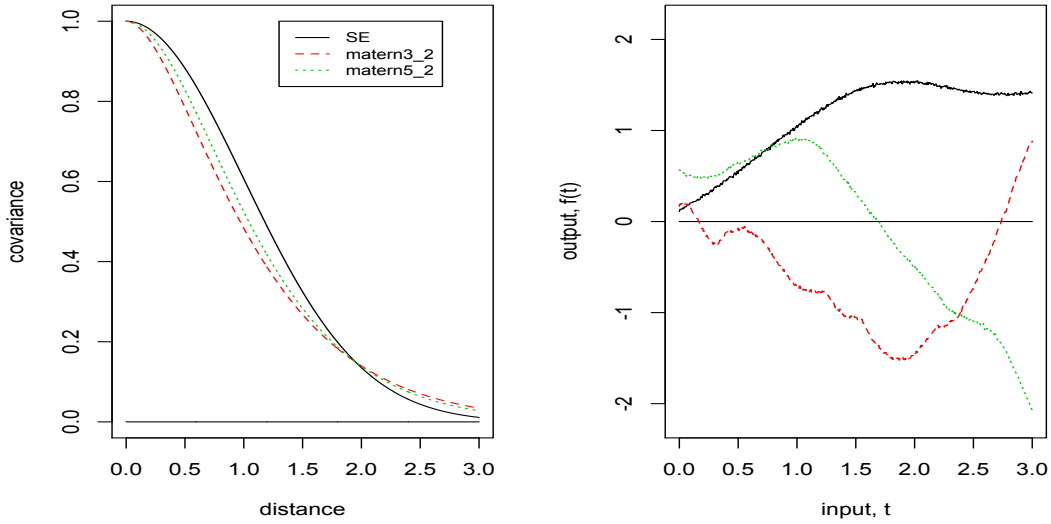

Figure 2.5: **Left:** covariance functions. **Right:** random functions drawn from Gaussian processes with Matern covariance functions for different values of v, with $\sigma_C^2 = 1$ and $l = 1$. The sample functions on the right were obtained using a discretization of the t-axis of 500 equally-spaced points.

Rasmussen and Williams [2004] argued that it is probably hard to distinguish between values of $v \geq 7/2$ if the training data is very noisy without explicit prior knowledge about the existence of high order derivatives. Since Matern class with $v = 3/2$ is just once differentiable, we choose Matern class with $v = 5/2$ for our Gaussian system. Expression for C'_{α_i} and C''_{α_i} for Matern class with $v = 5/2$ are derived in the Appendix A.

In summary, a great advantage of the Gaussian process is its simplicity. A Gaussian process with a simple covariance function such as Matern class with a specified degree of freedom may require just two parameters (length-scale parameter and variance parameter) to be fitted, which can then capture some complex structure of

training data and make posterior prediction conveniently. This is another reason that we are so keen to utilize Gaussian processes to overcome the computational difficulty encountered in a Bayesian approach for estimating parameters in complex dynamic systems.

2.3.3 Bridging the Gaussian system to the dynamic system

Introducing an auxiliary variable

The introduction of a Gaussian process prior allows us to sample $x_{G,i}(\mathbf{t})$ from its posterior distribution (2.2) conditional on data and hyperparameters. Now we would like to connect this Gaussian system to the real dynamic system by introducing an auxiliary variable so that it has analytic conditional distributions on both systems. In dynamic systems, conditioning on $x_i(\mathbf{t})$ and Θ is equivalent to conditioning on the derivatives of $x_i(\mathbf{t})$. Therefore, a natural choice for the auxiliary variable would be a noisy version of the derivatives of $x_i(\mathbf{t})$: $z_i(\mathbf{t}) = \dot{x}_i(\mathbf{t}) + \sqrt{\frac{1}{2}\gamma}\epsilon$, where γ is a constant and ϵ follows a standard normal distribution. Conditional on $x_i(\mathbf{t})$ and Θ from dynamic systems, we have

$$p_\gamma(z_i(\mathbf{t})|x_i(\mathbf{t}), \Theta) \sim N(f_i(x_i(\mathbf{t}), \Theta), \frac{1}{2}\gamma I_n) \tag{2.4}$$

Similarly, in the Gaussian system, we define $z_{G,i}(\mathbf{t}) = \dot{x}_{G,i}(\mathbf{t}) + \sqrt{\frac{1}{2}\gamma}\epsilon$. From (2.3), we get

$$p_\gamma(z_{G,i}(\mathbf{t})|y_{G,i}(\mathbf{t}), \alpha, \sigma_G^2) \sim N(a_i, b_i + \frac{1}{2}\gamma I_n) \tag{2.5}$$

We can now see how this auxiliary variable builds a bridge between the two systems. In Section 2.6.1, we provide another perspective to see why this auxiliary variable Z would be the *only* choice to connect these two systems, by comparing our method to the variational EM algorithm. Our idea is to use the artificial Gaussian system, which is computationally tractable, to emulate the real dynamic system optimally in the sense that the "distance" of these two systems is as close as possible. Figure 2.6 illustrates a graphical representation of two systems, which are connected by this auxiliary variable. Notice that we can choose arbitrary value of γ, so γ gives us flexibility on how well we require our Gaussian system to approximate the real dynamic system. A large γ sets a loose requirement for this approximation while a small γ requires a strict approximation. Later on, we can create a cascading sequence of MCMC chains according to a decreasing sequence of γ.

Defining a measure for the "distance" between two systems

It is however unclear which measure should be used for the "distance" between these two systems. Here is one candidate measure to use in our approximate Bayesian inference. On the one hand, from the Gaussian system, we have

$$p(z_{G,i}(\mathbf{t})|x_{G,i}(\mathbf{t}), y_{G,i}(\mathbf{t}), \alpha_i, \sigma_G^2) = N(\tilde{\mu}_i, \tilde{\Sigma}_i + \frac{1}{2}\gamma I_n) \tag{2.6}$$

where $\tilde{\mu}_i = \mu_{G,i}(\mathbf{t}) + C'_{\alpha_i} C_{\alpha_i}^{-1}(X_{G,i}(\mathbf{t}) - \mu_{G,i}(\mathbf{t}))$ and $\tilde{\Sigma}_i = C''_{\alpha_i} - C'_{\alpha_i} C_{\alpha_i}^{-1} C'_{\alpha_i}$. The derivation of (2.6) is in Appendix A. On the other hand, from the dynamic system, we have $p(z_i(\mathbf{t})|x_i(\mathbf{t}), \Theta) = N(f_i, \frac{1}{2}\gamma I_n)$, where f_i denotes $f_i(x_i(\mathbf{t}), \Theta)$. Since

$$Cov(y_i(\mathbf{t}), z_i(\mathbf{t})|x_i(\mathbf{t}), \Theta, \sigma^2) = 0$$

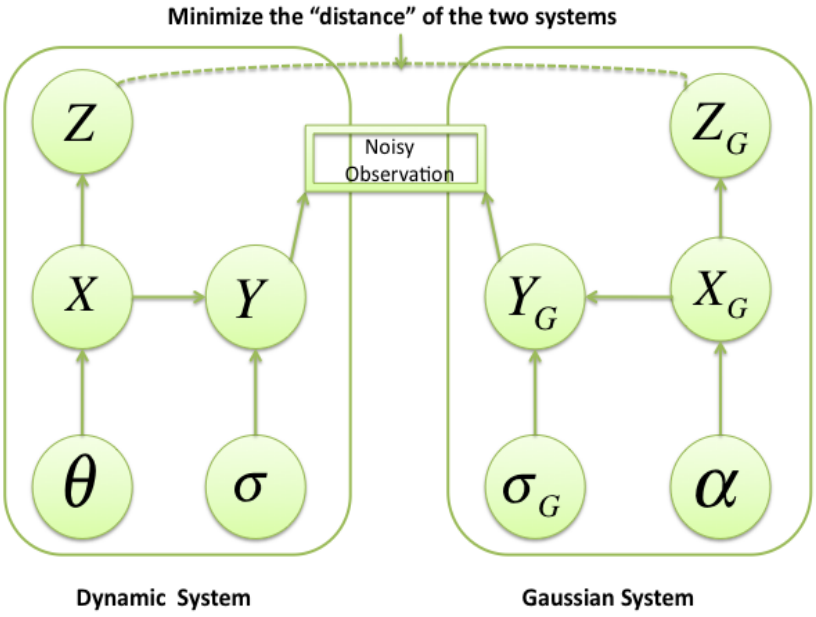

Figure 2.6: The graphical model represents the connection between the Gaussian system and the dynamic system, where the dotted lines represent a transfer of information between the systems.

then we have

$$p(z_i(\mathbf{t})|x_i(\mathbf{t}), y_i(\mathbf{t}), \Theta, \sigma^2) = N(f_i, \frac{1}{2}\gamma I_n) \tag{2.7}$$

The measure we define for the "distance" between the two system is the Kullback-Leibler (K-L) divergence of (2.6) and (2.7). We apparently cannot minimize this "distance" in one step, hence we will do it iteratively. At each step, given Θ and a known covariance structure, we choose hyperparameters $\{\alpha_i\}$ such that the K-L divergence of (2.6) and (2.7) is minimized. Since (2.6) and (2.7) are both normal, the K-L divergence of (2.6) and (2.7) has an analytic form (see Appendix A). At

each step, the optimal $\{\alpha_i\}$ obtained from minimization of this "distance" makes the Gaussian system mimic the dynamic system as closely as possible. As a result, we can replace $\sigma^2, x_i(\mathbf{t})$ in the dynamic system by $\sigma_G^2, x_{G,i}(\mathbf{t})$ sampled from the Gaussian system. This leads us to propose the following sampling scheme:

Starting with initial value $\{\alpha_i^*\}$, we do as follows:

- sample σ_G^2 from $p(\sigma_G^2|\{y_{G,i}(\mathbf{t})\}, \{\alpha_i^*\}) = \pi(\sigma_G^2)\prod_i N_{y_{G,i}(\mathbf{t})}(\mu_{G,i}, (\sigma_G^2 I_n + C_{\alpha_i^*}))$
- sample $x_{G,i}(\mathbf{t})$ from (2.2) and use in $\sigma_G^2, x_{G,i}(\mathbf{t})$ to be $\sigma^2, x_i(\mathbf{t})$ in the dynamic system
- sample Θ from $p(\Theta|\{x_i(\mathbf{t})\}, \{y_i(\mathbf{t})\}, \sigma^2)$
- calculate $\{\alpha_i\}$ such that K-L divergence of (2.6) and (2.7) is minimized
- update $\{\alpha_i^*\}$ by this set of $\{\alpha_i\}$

Iterate these steps until Θ converge.

Evaluate $p(\Theta|\{x_i(\mathbf{t})\}, \{y_i(\mathbf{t})\}, \sigma^2)$

Now we want to sample Θ from $p(\Theta|\{x_i(\mathbf{t})\}, \{y_i(\mathbf{t})\}, \sigma^2)$. For notational ease, let $Z = (z_1(\mathbf{t}), z_2(\mathbf{t}), ..., z_N(\mathbf{t}))$. Notice that

$$\begin{aligned} p(\Theta|X, Y, \sigma^2) &= \int p(\Theta, Z|X, Y, \sigma^2)dZ \\ &= \int p(Z|X, Y, \sigma^2)p(\Theta|Z, X, Y, \sigma^2)dZ \\ &\propto \int p(Z|X, Y, \sigma^2)p(\Theta|X, Y, \sigma^2)p(Z|\Theta, X, Y, \sigma^2)dZ \end{aligned}$$

and that the integrand involves $p(\Theta|X, Y, \sigma^2)$ itself; thus we propose an iterative way to solve this integral. Let $p_0(\Theta|X, Y, \sigma^2)$ be the prior for Θ. We create a sequence of

$\{p_m(\Theta|X,Y,\sigma^2)\}$ to approximate $p(\Theta|X,Y,\sigma^2)$. At step m, we have

$$\begin{aligned} p_m(\Theta|X,Y,\sigma^2) &\propto \int p(Z|X,Y,\sigma^2)p_{m-1}(\Theta|X,Y,\sigma^2)p(Z|\Theta,X,Y,\sigma^2)dZ \\ &\propto p_{m-1}(\Theta|X,Y,\sigma^2)\int p(Z|X,Y,\sigma^2)p(Z|\Theta,X,Y,\sigma^2)dZ \end{aligned}$$

Let $G(\Theta,X,Y)=\int p(Z|X,Y,\sigma^2)p(Z|\Theta,X,Y,\sigma^2)dZ$. Since $p(Z|X,Y,\sigma^2)$ is a marginal distribution of $p(Z|\Theta,X,Y,\sigma^2)$ over Θ, $G(\Theta,X,Y)$ can be written as:

$$\begin{aligned} &\int\Big(\int p_{m-1}(\Theta'|X,Y,\sigma^2)p(Z|\Theta',X,Y,\sigma^2)d\Theta'\Big)p(Z|\Theta,X,Y,\sigma^2)dZ \\ =\ &\int\int p_{m-1}(\Theta'|X,Y,\sigma^2)\Big(\int p(Z|\Theta',X,Y,\sigma^2)p(Z|\Theta,X,Y,\sigma^2)dZ\Big)d\Theta' \end{aligned}$$

Let $H(\Theta',\Theta,X,Y,\sigma^2)=\int p(Z|\Theta',X,Y,\sigma^2)p(Z|\Theta,X,Y,\sigma^2)dZ$. Since $p(Z|\Theta,X,Y,\sigma^2)$ is a normal distribution, $H(\Theta',\Theta,X,Y,\sigma^2)$ has an analytic form which is equal to

$$\prod_{i=1}^{N}\exp(-\frac{1}{2}(f_i(\Theta',X)-f_i(\Theta,X))^T(\gamma I_n)^{-1}(f_i(\Theta',X)-f_i(\Theta,X)))$$

Hence, $G(\Theta,X,Y)=\int p_{m-1}(\Theta'|X,Y,\sigma^2)H(\Theta',\Theta,X,Y,\sigma^2)d\Theta'$. We use a Monte Carlo method to approximate $G(\Theta,X,Y)$ simply by using the previous Monte Carlo sample Θ^{m-1} to substitute into $H(\Theta^{m-1},\Theta,X,Y,\sigma^2)$, yielding

$$\begin{aligned} p_n(\Theta|X,Y,\sigma^2) &\propto p_{m-1}(\Theta|X,Y,\sigma^2)H(\Theta^{(m-1)},\Theta,X,Y,\sigma^2) \\ &\propto ... \propto p_0(\Theta|X,Y,\sigma^2)\prod_{j=0}^{m-1}H(\Theta^{(j)},\Theta,X,Y,\sigma^2) \end{aligned} \tag{2.8}$$

We call this approximation method *Approx 1.0*. It takes $O(m^2)$ for m iterations, which is still computationally expensive.

From $O(m^2)$ to $O(m)$

We now leverage the Gaussian system to speed up the evaluation of $p_m(\Theta|X, Y, \sigma^2)$ based on *Approx 1.0*. Recall that we use a Gaussian system to approximate the true dynamic system, under which $x_i(\mathbf{t}), \mathbf{y_i}(\mathbf{t}), \mathbf{z_i}(\mathbf{t})$ can be substituted by $x_{G,i}(\mathbf{t}), \mathbf{y_{G,i}}(\mathbf{t}), \mathbf{z_{G,i}}(\mathbf{t})$. Then, $p(Z|X, Y, \sigma^2)$ can be approximated by $p(Z_G|X_G, Y_G, \sigma_G^2)$, which has the form (2.6). In this case, the integrand of $G(\Theta, X, Y)$ can be approximated by a product of two normal density functions, denoted by $H'(\Theta, X)$, which has a closed-form expression:

$$\prod_{i=1}^{N} \exp(-\frac{1}{2}(f_i - \tilde{\mu}_i)^T(\tilde{\Sigma}_i + \frac{1}{2}\gamma I_n)^{-1}(f_i - \tilde{\mu}_i)) \tag{2.9}$$

Therefore $p_m(\Theta|X, Y, \sigma^2)$ can be reduced as $p_0(\Theta|X, Y, \sigma^2)H'^m(\Theta, X)$, which only takes $O(m)$. We call this approximation method *Approx 2.0*. The Gaussian system kills two birds with one stone: it not only provides an excellent surrogate model for the real dynamic system, but also greatly facilitates the computation of the approximation to $p_m(\Theta|X, Y, \sigma^2)$.

2.3.4 A proposed sampling scheme

Now, with everything at hand, we can put all pieces together and conduct a complete sampling scheme.

With a fixed γ, we start with initial value $\{\alpha_i^*\}$. At step m, we do as follows:

- sample σ_G^2 from $p(\sigma_G^2|\{y_{G,i}(\mathbf{t})\}, \{\alpha_i^*\}) = \pi(\sigma_G^2)\prod_{i=1}^{N} N_{y_{G,i}(\mathbf{t})}(\mu_{G,i}, (\sigma_G^2 I + C_{\alpha_i^*}))$

- sample $x_{G,i}(\mathbf{t})$ from (2.2) and use in $\sigma_G^2, x_{G,i}(\mathbf{t})$ to be $\sigma^2, x_i(\mathbf{t})$ in the dynamic system
- sample $\Theta^{(m)}$ from $p_m(\Theta|X, Y, \sigma^2)$ using Metropolis-Hasting algorithm.
- calculate $\{\alpha_i\}$ such that K-L divergence of (2.6) and (2.7) is minimized
- update $\{\alpha_i^*\}$ by this set of $\{\alpha_i\}$

2.4 Numerical examples

2.4.1 Fitting the FitzHugh-Nagumo equations

We set up simulated data by adding Gaussian error with standard deviation $\sqrt{2}$ to the solution for parameters $\{a, b, c\} = \{0.2, 0.2, 1\}$ and initial conditions $\{V, R\} = \{-1, 1\}$ at times 0, 5,...,100. Though this dynamic system consists only of 2 equations and 3 parameters, it displays a highly nonlinear likelihood surface given the simulated data in Figure 2.1 (right). So this model provides an excellent test for our method.

We start with initial values $(a, b, c) = (1, 1, 3)$. From the graph in Figure 2.1 (right), we must travel through many bumpy areas before we climb to the peak at $(a, b, c) = (0.2, 0.2, 1)$, which means that the naive MCMC with numerical solutions may take time to eventually converge to the true values, see Figure 2.7. To implement our approximate Bayesian methods, we choose $\mu_{G,i}$ to be zero vector, and the covariance function to be the Matern family in the Gaussian system, and run the MCMC chain by setting $\gamma = 2$, which is shown in Figure 2.8. When we minimize K-L distance to obtain $\{\alpha_i^*\}$, we specify a region for possible values for each α_i^*, over

which we perform the optimization. In this case, we choose [0, 10] for α_1 and α_2. The comparison of MCMC samples by using *Approx 1.0* and *Approx 2.0* is reflected in Figure 2.9. We see that the density plots for three parameters by using *Approx 1.0* and *Approx 2.0* are very similar. However, *Approx 2.0* (*Approx 2.0* takes less than 30 mins to finish 5,000 iterations) takes much less computational time than *Approx 1.0* (*Approx 1.0* takes 1.5 hours to finish 5,000 iterations). To calibrate how well our Gaussian system approximates the dynamic system, we calculate the 95% confidence region of X_G and find that most of the X values across time are within that region, see Figure 2.10. This demonstrates that our Gaussian system is capable of emulating the dynamic system quite well.

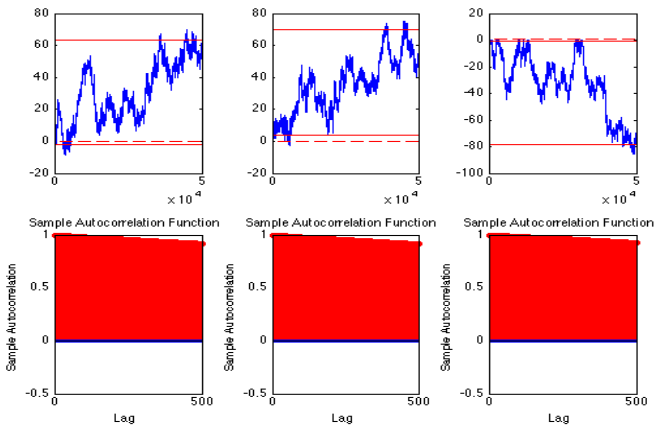

Figure 2.7: Numerical Solvers for the FN model for the fully observed case. The 95% confidence regions are represented by two solid red lines and the true solutions are represented by dashed lines. **Left:** simulated results for parameter a. **Middle:** simulated results for parameter b. **Right:** simulated results for parameter c.

Another salient property of our approximate Bayesian approach is that it reduces the computational time dramatically. Our method costs less than 5 minutes to com-

plete one chain with 5,000 iterations. However, it takes more than 30 minutes to complete one chain with 5,000 iterations by using MCMC with numerical solvers by Gelman *et al* (1996) and still cannot reach the true solution due to irregular likelihood surface, provided that the initial conditions are unknown, as we compare the histogram of MCMC samples by our method and numerical solver in Figure 2.11. Our method does not require initial conditions for differential equations nor does it require solving the ODE systems numerically.

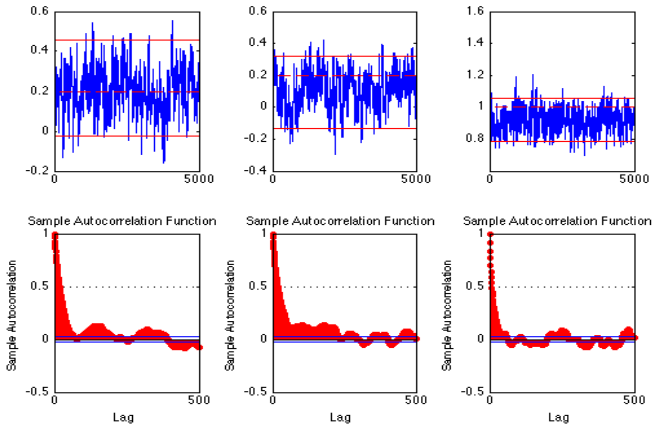

Figure 2.8: Simulation Results for *Approx 2.0* with Matern covariance. The 95% confidence regions are represented by two solid red lines and the true solutions are represented by dashed lines.

2.4.2 Fitting the Repressilator Model

We are interested in the parameters (a, b, c) in our repressilator model in (2.2). While MCMC with numerical solutions for 50,000 iterations performs poorly shown in Figure 2.12, our approximate Bayesian approach provides better estimation and faster convergence shown in Figure 2.13. In our setting, we choose $\gamma = 10$ and

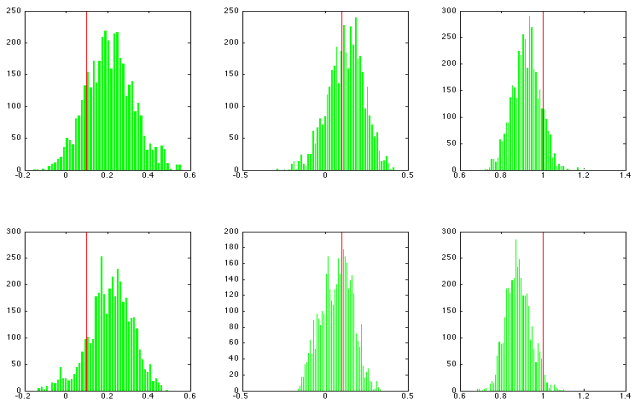

Figure 2.9: Simulation Results for *Approx 2.0* (the first row) versus *Approx 1.0* (the second row) with $\gamma = 2$. The true solutions are represented by a solid black line.

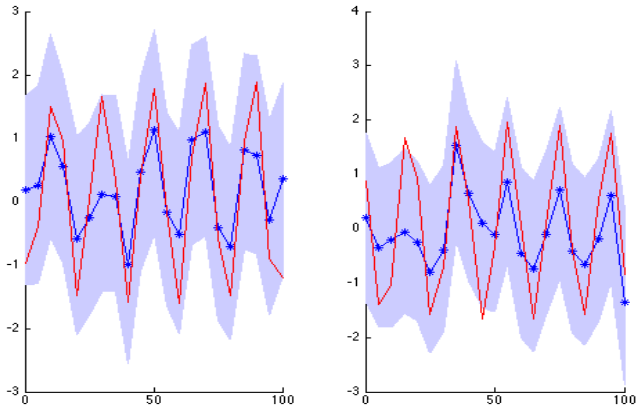

Figure 2.10: Calibrate our Gaussian system for the FN model for the fully observed case. The grey area represents 95% confidence regions, the blue line represents median of Monte Carlo samples, and the red line represents the numerical solutions. **Left:** simulated results for V. **Right:** simulated results for R.

Matern covariance functions, and it gives us very fast convergence as we can see that the autocorrelations of MCMC samples for the three parameters decay very fast. Therefore, we do not have to construct the subsequent chain to speed up the

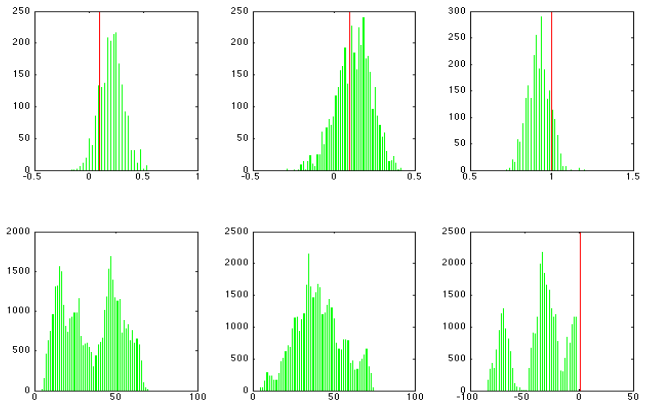

Figure 2.11: Comparison of our approximate Bayesian method and the numerical solver for the FN model for the fully observed case. The red vertical line represents the numerical solutions. The first row is the histogram of the samples from Gaussian emulator and the second row is the histogram of the samples from numerical solver.

convergence. The density plot is shown Figure 2.14 indicates that our Gaussian system approximates the dynamic system fairly well. Our Gaussian systems calibrate the true dynamic system well as the medians of X_G for six components are very close to the true numerical solutions across the time, shown in Figure 2.15. In terms of computational time, our method takes less than an hour to complete one chain with 5,000 iterations while MCMC with numerical solvers take more than 4 hours to finish 5,000 iterations, given that the initial conditions are known.

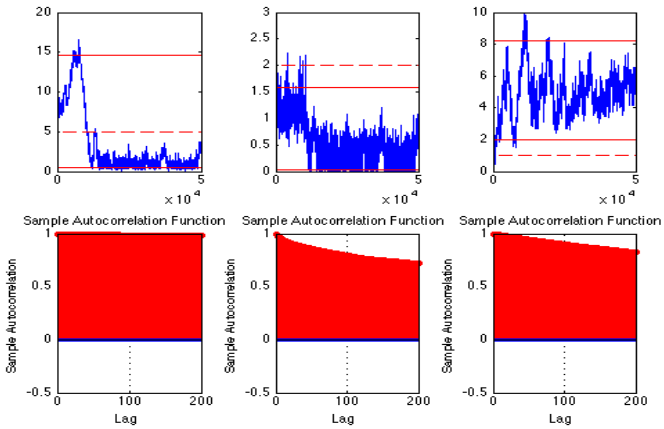

Figure 2.12: Numerical Solvers for the repressilator model for the fully observed case. The 95% confidence regions are represented by two solid red lines and the true solutions are represented by dashed lines. **Left:** simulated results for parameter a. **Middle:** simulated results for parameter b. **Right:** simulated results for parameter c.

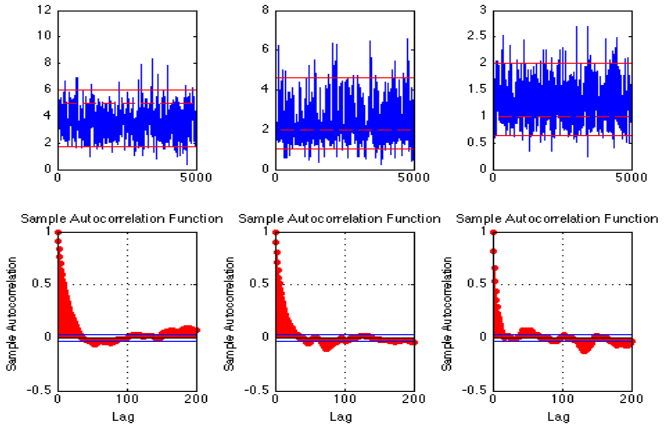

Figure 2.13: Simulation Results for the Repressilator model. The 95% confidence regions are represented by two solid red lines and the true solutions are represented by dash lines.

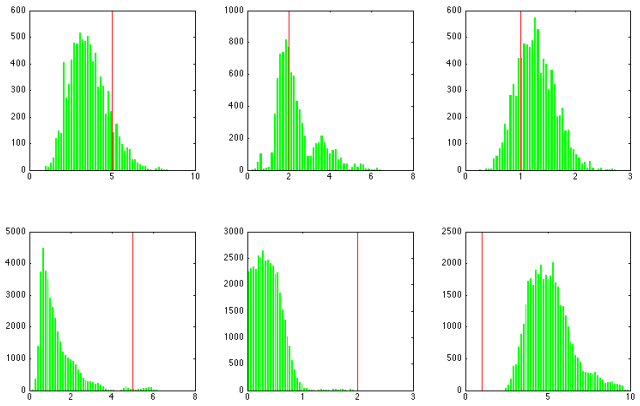

Figure 2.14: Comparison of our approximate Bayesian approach and the numerical solver for the Repressilator model for the fully observed case. The red vertical line represents the numerical solutions. The first row is the histogram of the samples from our method and the second row is the histogram of the samples from numerical solver.

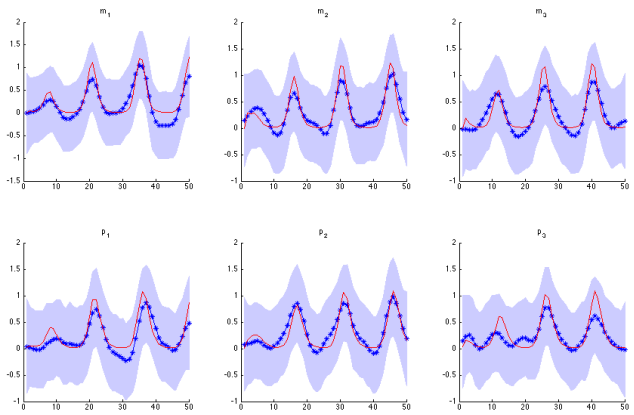

Figure 2.15: Simulated results for the Repressilator model for the fully observed case. The grey area represents 95% confidence regions, the blue line represents median of Monte Carlo samples, and the red line represents the numerical solutions. The first row (from left to right): m_1, m_2, m_3 and the second row (from left to right): p_1, p_2, p_3.

2.5 Partially Observed Case

2.5.1 Extending our approximate Bayesian approach to partially observed case

Frequently, due to some measurement difficulty, we often encounter that some components of the dynamic systems are unable to be observed in the experiments. Therefore, how to estimate parameters in the dynamic systems in the absence of the observed data for some components becomes an issue. We would extend our approximate Bayesian approach to address this issue. For notational ease, let N_o be the number of the observed components and N_m be the number of the missing components, and, therefore, $N = N_o + N_m$. We define $\{x_{miss}(\mathbf{t})\}$ to be the collection of unobserved components and $\{x_{obs}(\mathbf{t})\}$ to be the collection of observed components: $\{x_{com}(\mathbf{t})\} = \{x_{miss}(\mathbf{t}), x_{obs}(\mathbf{t})\}$. In this case, we sample Θ and $\{x_{miss}(\mathbf{t})\}$ from $p(\Theta, \{x_{miss}(\mathbf{t})\}|\{x_{obs}(\mathbf{t})\}, \{y(\mathbf{t})\}, \sigma^2)$

$$\propto p(\{x_{miss}(\mathbf{t})\}|\{x_{obs}(\mathbf{t})\}, \{y(\mathbf{t})\}, \sigma^2)p(\Theta|\{x_{com}(\mathbf{t})\}, \{y(\mathbf{t})\}, \sigma^2)$$

. It is, however, hard to compute $p(\{x_{miss}(\mathbf{t})\}|\{x_{obs}(\mathbf{t})\}, \{y(\mathbf{t})\}, \sigma^2)$ directly. In the same spirit, we sample Θ and $\{x_{miss,G}(\mathbf{t})\}$ from $p(\Theta, \{x_{miss,G}(\mathbf{t})\}|\{x_{obs,G}(\mathbf{t})\}, \{y_G(\mathbf{t})\}, \sigma_G^2) \propto$

$$p(\{x_{miss,G}(\mathbf{t})\}|\{x_{obs,G}(\mathbf{t})\}, \{y_G(\mathbf{t})\}, \sigma_G^2)p(\Theta|\{x_{com,G}(\mathbf{t})\}, \{y_G(\mathbf{t})\}, \sigma_G^2)$$

For dynamic systems, we witnessed the dependence of one component on another in 2.1(left). This gives us a hint to enhance the modeling of our Gaussian system when

some components are not observed. In our parallel Gaussian system, we generalize our previous modeling approach to the extent that we impose a covariance structure between the components not observed and the components that are observed. In this case, let $X_{M,G} = \{x_{miss,G}(\mathbf{t})\}, X_{O,G} = \{x_{miss,G}(\mathbf{t})\}, X_{F,G} = (X_{M,G}, X_{O,G})$. We model $X_{F,G}$ as a Gaussian process:

$$X_{F,G} = \begin{pmatrix} X_{M,G} \\ \\ X_{O,G} \end{pmatrix} \sim N\left(\begin{pmatrix} 0_{N_o n} \\ \\ 0_{N_m n} \end{pmatrix}, \begin{pmatrix} C_{\alpha_M} & C_{\alpha_C} \\ \\ C_{\alpha_C} & C_{\alpha_O} \end{pmatrix} \right), \tag{2.10}$$

where C_{α_C} denotes the cross covariance between $X_{M,G}$ and $X_{O,G}$. We use Kronecker type structure to model C_{α_C} as follows: let A be the covariance between fields. In this case, $A(i,j) = \sigma_{C,i}\sigma_{C,j}$ for $1 \leq i \leq N_o$ and $1 \leq j \leq N_m$, where i is the index for the observed component and j is the index for the missing component. B is the *correlation* function for a given field and here we use a Matern family with $\sigma_C = 1$ and hyperparameter α_C to model B. Then,

$$C_{\alpha_C} = A \otimes B. \tag{2.11}$$

Furthermore, We define $Z_{F,G} = \dot{X}_{F,G} + \frac{1}{2}\gamma I_{Nn}$ in the Gaussian system and $Z_F = \dot{X_F} + \frac{1}{2}\gamma I_{Nn \times Nn}$ in the dynamic system. In our Gaussian system,

$$p_\alpha(Z_{F,G}|X_{F,G}, \sigma_G^2) = N(\mu_{F,G}, \Sigma_{F,G} + \frac{1}{2}\gamma I_{Nn}) \tag{2.12}$$

where

$$\mu_{F,G} = \begin{pmatrix} C'_{\alpha_M} & C'_{\alpha_C} \\ C'_{\alpha_C} & C'_{\alpha_O} \end{pmatrix} \begin{pmatrix} C_{\alpha_M} & C_{\alpha_C} \\ C_{\alpha_C} & C_{\alpha_O} \end{pmatrix}^{-1} X_{F,G}$$

$$\Sigma_{F,G} = \begin{pmatrix} C''_{\alpha_M} & C''_{\alpha_C} \\ C''_{\alpha_C} & C''_{\alpha_O} \end{pmatrix} - \begin{pmatrix} C'_{\alpha_M} & C'_{\alpha_C} \\ C'_{\alpha_C} & C'_{\alpha_O} \end{pmatrix} \begin{pmatrix} C_{\alpha_M} & C_{\alpha_C} \\ C_{\alpha_C} & C_{\alpha_O} \end{pmatrix}^{-1} \begin{pmatrix} C'_{\alpha_M} & C'_{\alpha_C} \\ C'_{\alpha_C} & C'_{\alpha_O} \end{pmatrix}$$

.

And in the dynamic system,

$$p(Z_F|X_F, \Theta) = N(f_F, \frac{1}{2}\gamma I_{Nn}) \tag{2.13}$$

Therefore, the "distance" between our Gaussian system and dynamic system in the partially observed case is the K-L divergence of (2.12) and (2.13). Now recall

$$p(\Theta, X_{M,G}|X_{O,G}, Y_G, \sigma_G^2) \propto p(X_{M,G}|X_{O,G}, Y_G, \sigma_G^2) p(\Theta|X_{F,G}, Y_G, \sigma_G^2) \tag{2.14}$$

Since

$$\begin{pmatrix} X_{M,G} \\ X_{O,G} \\ Y_G \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} C_{\alpha_M} & C_{\alpha_C} & C_{\alpha_C} \\ C_{\alpha_C} & C_{\alpha_O} & C_{\alpha_O} \\ C_{\alpha_C} & C_{\alpha_O} & C_{\alpha_O} + \sigma^2 I_n \end{pmatrix} \right) \tag{2.15}$$

it follows that

$$X_{M,G}|X_{O,G}, Y_G, \sigma_G^2 \sim N(\mu_M, \Sigma_M) \tag{2.16}$$

where $\mu_M = C_{\alpha_O}^{-1} C_{\alpha_C} X_{O,G}$ and $\Sigma_M = C_{\alpha_M} - C_{\alpha_C} C_{\alpha_O}^{-1} C_{\alpha_C}$

Similarly, we can approximate $p(\Theta|X_{F,G}, Y_G, \sigma_G^2)$ by a sequence of $\{p_m(\Theta|X_{F,G}, Y_G, \sigma_G^2)\}$:

we define

$$H''(\Theta, X_{F,G}, Y_G) \propto \exp(-\frac{1}{2}(f_F - \mu_{F,G})^T(\Sigma_{F,G} + \gamma I_{Nn})^{-1}(f_F - \mu_{F,G})). \tag{2.17}$$

Then

$$p_m(\Theta|X_{F,G}, Y_G, \sigma_G^2) \propto p_0(\Theta|X_{F,G}, Y_G, \sigma_G^2)H''^m(\Theta, X_{F,G}, Y_G). \tag{2.18}$$

We now combine those equations above, yielding the analytic form of $p(\Theta, X_{M,G}|X_{O,G}, Y_G, \sigma_G^2)$.

A complete sampling scheme for partially observed data:

With a fixed γ, we start with initial value $\{\alpha^*\}$. At step n, we do as follows:

- sample σ_G^2 from $p(\sigma_G^2|\{Y_G, \{\alpha^*\}) = \pi(\sigma_G^2)N_{Y_G}(\mu_G, (\sigma_G^2 I + C_{\alpha_O^*}))$
- sample $X_{O,G}$ from (2.2) and use $\sigma_G^2, X_{O,G}$ to be σ^2, X_O in the dynamic system
- calculate $p_n(\Theta, X_{M,G}|X_{O,G}, Y_G, \sigma_G^2) \propto p(X_{M,G}|X_{O,G}, Y_G, \sigma_G^2)p_n(\Theta|X_{F,G}, Y_G, \sigma_G^2)$
- sample $\Theta^{(n)}$ and $X_{M,G}^{(n)}$ from $p_n(\Theta, X_{M,G}|X_{O,G}, Y_G, \sigma_G^2)$ using Metropolis-Hasting algorithm
- calculate $\{\alpha\}$ such that K-L divergence of (2.12) and (2.13) is minimized
- update $\{\alpha^*\}$ by this set of $\{\alpha\}$

2.5.2 Numerical Examples

Component V is unobserved in FitzHugh-Nagumo equations

We assume that the component, V, is not observed in the FitzHugh-Nagumo equations. Our numerical solver performs poorly for the partially observed case, shown in

Figure 2.16. The MCMC samples for both parameters and missing components are shown in Figure 2.17 and Figure 2.18, with $\gamma = 2$. The results show that the true values of three parameters and true values for the unobserved components are within 95% confidence region of MCMC samples. The density plots of MCMC samples are shown in Figure 2.19, in which the coverage is wider than that for the complete observed case in Figure 2.10 due to less information (some components are unobserved).

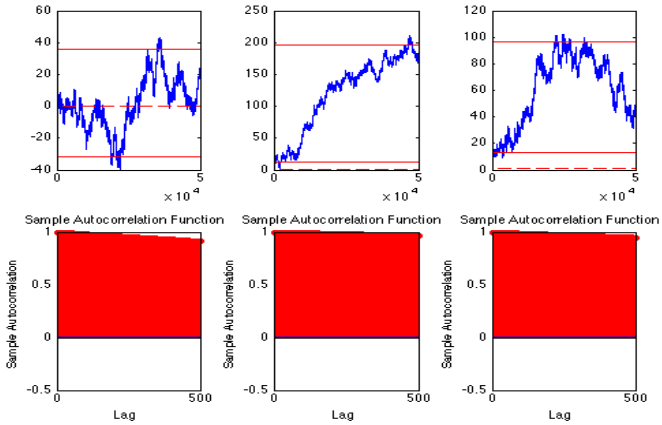

Figure 2.16: Numerical Solvers for the FN model for the partially observed case. The 95% confidence regions are represented by two solid red lines and the true solutions are represented by dashed lines. **Left:** simulated results for parameter a. **Middle:** simulated results for parameter b. **Right:** simulated results for parameter c.

Protein components are unobserved in the repressilator model

In biological experiments, scientists use DNA microarrays to measure the expression levels of genes or to genotype multiple regions of a genome. The level of mRNA is easy to measure by microarrays or gene chips. Although protein microarrays may use similar detection methods as DNA Microarrays, a problem is that protein concentrations in a biological sample may be many orders of magnitude different from that for

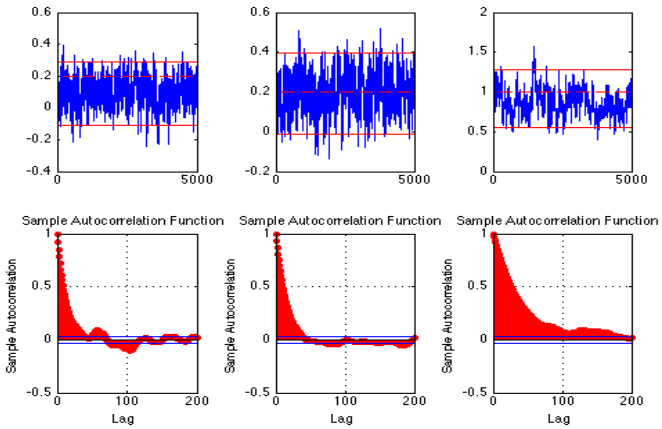

Figure 2.17: Simulated results for the FN model for the partially observed case. The 95% confidence regions are represented by two solid red lines and the true solutions are represented by dashed lines. **Left:** simulated results for parameter a. **Middle:** simulated results for parameter b. **Right:** simulated results for parameter c.

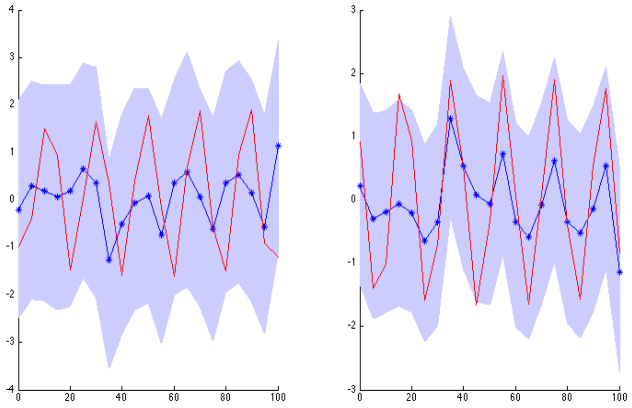

Figure 2.18: Calibrate our Gaussian systems for the FN model for the partially observed case. The grey area represents 95% confidence regions, the blue line represents median of Monte Carlo samples, and the red line represents the numerical solutions. **Left:** simulated results for the missing component V. **Right:** simulated results for the observed component R.

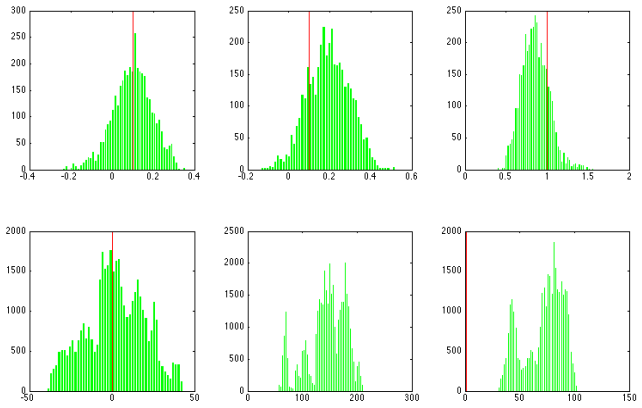

Figure 2.19: Comparison of our approximate Bayesian method and the numerical solver for the FN model for the partially observed case. The red vertical line represents the numerical solutions. The first row is the histogram of the samples from Gaussian emulator and the second row is the histogram of the samples from numerical solver.

mRNAs. This results in the case in which the levels of proteins are not recorded or observed. In this case, we assume that the levels for proteins, p_1, p_2, and p_3, are not observed in the repressilator model.

In terms of the covariance structure between mRNAs and proteins, if we model it as a full covariance matrix proposed in (2.11), then we have 42 hyperparameters that we have to optimize over, which could become a computational burden in our method. Alternatively, while we retain the independent assumption within the mRNA components and the protein components respectively, we can propose a *sparse* covariance matrix between mRNAs and proteins. The linear model between m_i and p_i, governed by the linear ODE, indicates a strong correlation between m_i and p_i. Therefore, we will impose a covariance matrix between m_i and p_i as we did in (2.11), and zero covariance matrix between m_i and p_j, where $i \neq j$. In this case, we have only 18

hyperparameters that we have to optimize over, a significant reduction from the case of having a full covariance matrix.

Our numerical solver performs poorly for the partially observed case, shown in Figure 2.20. However, our approximate Bayesian approach outperforms the numerical solver, supported by the simulated results for both parameters and missing components in Figure 2.21 and Figure 2.22, with $\gamma = 10$. The comparison of histograms of MCMC samples generated by our approximate Bayesian approach and numerical solvers respectively, shown in Figure 2.23, further confirms that our methods did a good job.

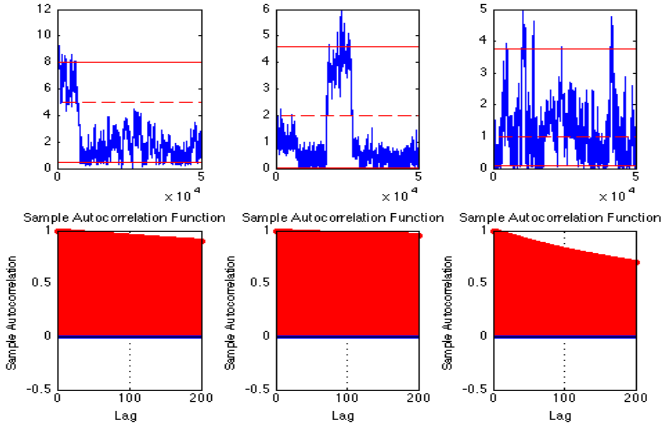

Figure 2.20: Numerical Solvers for the repressilator model for the partially observed case. The 95% confidence regions are represented by two solid red lines and the true solutions are represented by dashed lines. **Left:** simulated results for parameter a. **Middle:** simulated results for parameter b. **Right:** simulated results for parameter c.

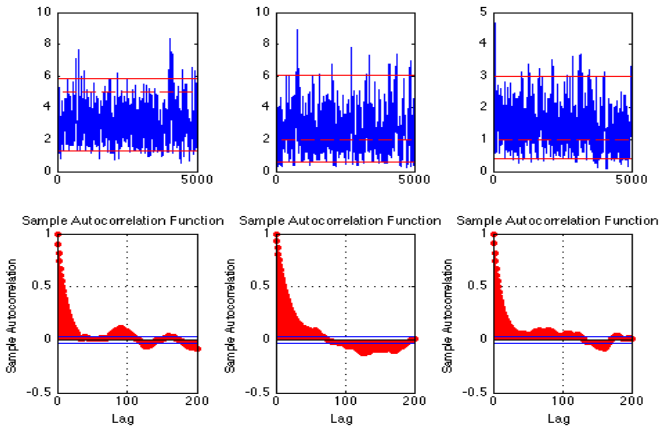

Figure 2.21: Simulated results for the Repressilator model for the partially observed case. The 95% confidence regions are represented by two solid red lines and the true solutions are represented by dashed lines. **Left:** simulated results for parameter a. **Middle:** simulated results for parameter b. **Right:** simulated results for parameter c.

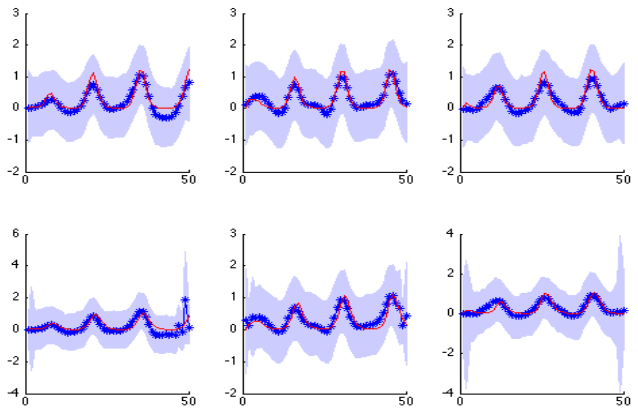

Figure 2.22: Calibrate our Gaussian systems for the Repressilator model for the partially observed case. The grey area represents 95% confidence regions, the blue line represents median of Monte Carlo samples, and the red line represents the numerical solutions. The first row (from left to right): m_1, m_2, m_3 and the second row for missing components (from left to right): p_1, p_2, p_3.

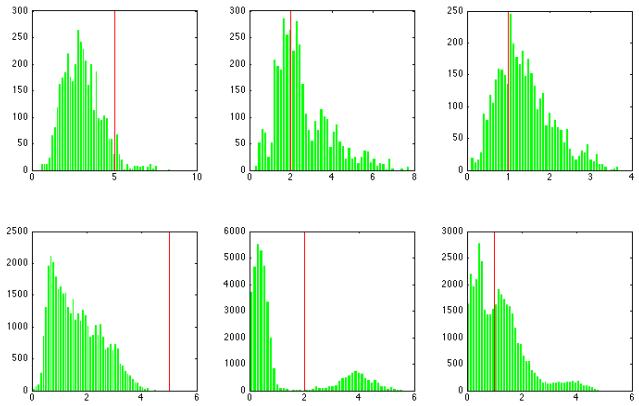

Figure 2.23: Comparison of our approximate Bayesian method and the numerical solver for the Repressilator model for the partially observed case. The red vertical line represents the numerical solutions. The first row is the histogram of the samples from our approximate method and the second row is the histogram of the samples from numerical solver.

2.6 Discussion

2.6.1 Is it just another version of variational EM algorithm?

Some readers might suspect that our algorithm is just another version of the variational EM algorithm. Thus, before we delve into variational EM algorithm, let us review the original Expectation-Maximization (EM) algorithm by Dempster et al. [1977]. The EM algorithm alternates between an E step, which infers posterior distributions over hidden variables given a current parameter setting, and an M step, which maximizes the log likelihood $l(\Theta)$ with respect Θ given the sufficient statistics from the E step. In our case, the hidden variables include X and Z, so we define

$\tilde{X} = \{X, Z\}$. Such updates can be derived using the lower bound:

$$\begin{aligned} l(\Theta) &= \ln \int p(\tilde{X}, Y|\Theta) d\tilde{X} \\ &= \ln \int \frac{p(\tilde{X}, Y|\Theta)}{q(\tilde{X})} q(\tilde{X}) d\tilde{X} \\ &\geq \int \ln \frac{p(\tilde{X}, Y|\Theta)}{q(\tilde{X})} q(\tilde{X}) d\tilde{X} \\ &= \int \ln p(\tilde{X}, Y|\Theta) q(\tilde{X}) d\tilde{X} - \int \ln q(\tilde{X}) q(\tilde{X}) d\tilde{X} \\ &\equiv F(q(\tilde{X}), \Theta) \end{aligned} \tag{2.19}$$

where we use Jensen's inequality which follows from the fact that the ln function is concave. $F(q(\tilde{X}), \Theta)$ is a lower bound on $l(\Theta)$. Defining the *energy* of a global configuration $(\tilde{X}, Y)$ to be $-\ln p(\tilde{X}, Y|\Theta)$, the lower bound $F(q(\tilde{X}), \Theta)$ is the negative of a quantity known in statistical physics as the *free energy*: the expected energy under $q(\tilde{X})$ minus the entropy of $q(\tilde{X})$.

Therefore, at each iteration of EM algorithm, the E step maximizes $F(q(\tilde{X}), \Theta)$ with respect to $q(\tilde{X})$ and M step does so with respect to Θ. Mathematically speaking, with iteration number t, starting from some initial parameters $\Theta^{(0)}$, the update equation is

$$\begin{aligned} \text{E Step: } q(\tilde{X})^{(t+1)} &\leftarrow \text{argmax}_{q(\tilde{X})} F(q(\tilde{X}), \Theta^{(t)}) \\ \text{M Step: } \Theta^{(t+1)} &\leftarrow \text{argmax}_{\Theta} F(q^{(t+1)}(\tilde{X}), \Theta) \end{aligned}$$

From the E step, it turns out that the maximum over $q(\tilde{X})$ of the bound is obtained

by setting

$$q(\tilde{X})^{(t+1)} = p(\tilde{X}|Y, \Theta^{(t)})$$

at which the bound becomes an equality. Unfortunately, in many cases, the complicated posterior distribution of hidden variables might lead to an intractable form of $F(q(\tilde{X}), \Theta)$. In a variational approach (Bishop [2006]), we can constrain the posterior distribution to be of a particular tractable form, for example, factorized over the hidden variable $\tilde{X} = \{X, Z\}$. Using calculus of variations we can still optimize $F(q(\tilde{X}), \Theta)$ as a functional of constrained distributions $q(\tilde{X})$. The M step is conceptually identical to that in the original EM algorithm, except that it is based on sufficient statistics calculated with respect to the constrained posterior instead of the exact posterior.

We can re-write the lower bound $F(q(\tilde{X}), \Theta)$ as

$$\begin{aligned} F(q(\tilde{X}), \Theta) &= \int \ln \frac{p(\tilde{X}, Y|\Theta)}{q(\tilde{X})} q(\tilde{X}) d\tilde{X} \\ &= \int \ln p(Y|\Theta) q(\tilde{X}) d\tilde{X} + \int \ln \frac{p(\tilde{X}|Y, \Theta)}{q(\tilde{X})} q(\tilde{X}) d\tilde{X} \\ &= \int \ln p(Y|\Theta) q(\tilde{X}) d\tilde{X} - \int \ln \frac{q(\tilde{X})}{p(\tilde{X}|Y, \Theta)} q(\tilde{X}) d\tilde{X} \end{aligned} \tag{2.20}$$

Thus the E step of the variational EM algorithm is equivalent to minimizing the following the quantity:

$$\int \ln \frac{q(\tilde{X})}{p(\tilde{X}|Y, \Theta)} q(\tilde{X}) d\tilde{X} \equiv \mathrm{KL}[q(\tilde{X})||p(\tilde{X}|Y, \Theta)] \tag{2.21}$$

which is the Kullback-Leibler divergence between the variational distribution $q(\tilde{X})$

and the exact hidden variable posterior $p(\tilde{X}|Y,\Theta)$. One can choose $q(\tilde{X})$ to be in a particular parameterized family:

$$q(\tilde{X}) = q(\tilde{X}|\lambda)$$

where λ are *variational parameters*. In essence, the E-step of the variational EM algorithm is conducted via minimization of K-L divergence with respect to a set of variational parameters λ.

Our methods bear some resemblance to the variational EM algorithm: choose a particular parameterized family (Gaussian), and minimize K-L divergence of this constrained family and the true posterior distribution with respect to the hyper-parameters. However, in the variational EM setting, we have to know the exact form of $p(\tilde{X}|Y,\Theta)$, which is the product of $p(Z|X,Y,\Theta)$ and $p(X|Y,\Theta)$. We know $p(Z|X,Y,\Theta)$ follows a normal distribution, but we do not know the form of $p(X|Y,\Theta)$. In particular, the only analytic distribution we have in the dynamic system is $p(Z|X,Y,\Theta)$. As a result, we cannot conduct E step of the variational EM algorithm due to the unknown form of $p(\tilde{X}|Y,\Theta)$. In other words, our methods are *not* the variational EM algorithm. Obtaining an explicit form of K-L divergence in (2.21) under the variational EM framework is not possible. This suggests that our construction of a Gaussian system and our proposal of "distance" between the two systems might be the only way to make all computational steps tractable.

2.6.2 The choice of γ

As discussed in the previous section, we can choose an arbitrary value of γ to realize the sample scheme above. So how to pick up γ becomes an issue in our method. A heuristic guidance would be starting with a relatively large γ. If the convergence of our MCMC samples with that γ is relatively fast, we can accept the value of that γ. However, if the convergence is rather slow, we can create multiple MCMC chains by varying the value of γ if we view γ as a controlling variable,. Suppose we create a first MCMC chain by the sample scheme with $\gamma = \gamma_0$. When γ is relatively large, a looser requirement for our approximation to the real dynamic system gives the Markov chain much more freedom to explore the sampling space. We then move to the next chain by letting $\gamma = \gamma_1$, where $\gamma_1 < \gamma_0$. In this case, the requirement for approximation becomes more strict. Define $\tilde{\Theta} = \{\Theta, x(\mathbf{t}), \alpha, \sigma^2\}$. We randomly draw a set of parameters from the previous chain as our starting point of MCMC chain with $\gamma = \gamma_1$. At parameter update step t, let $\tilde{\Theta}_t$ be the current set of parameters. We do as follows:

- (local move) with probability $1 - p$, perform a regular MCMC step
- (cross move) with probability p, draw a set of parameters from the previous chain, say $\tilde{\Theta}_{trial}$, and accept it with probability

$$r = \frac{P_{\gamma_1}(\tilde{\Theta}_{trial}|y((\mathbf{t})))P_{\gamma_0}(\tilde{\Theta}_t|y((\mathbf{t})))}{P_{\gamma_1}(\tilde{\Theta}_t|y((\mathbf{t})))P_{\gamma_0}(\tilde{\Theta}_{trial}|y((\mathbf{t})))}$$

  otherwise keep $\tilde{\Theta}_t$ as the next sample.

The cross move can be viewed as a global type of move: it proposes samples that occur anywhere in the space. The local move, however, draws samples near the previous draw in the Markov Chain. Hence the creation of this subsequent chain blends these two moves to take advantage of both strengths. The ratio of how often each type of move occurs is determined by p. In practice, we found that the sampling performance is not very sensitive to the choice of p for $20\% \leq p \leq 50\%$, which is also confirmed by Kou et al. [2012].

We can construct $M+1$ such chains, using similar methods, monitored by $\gamma_M < \gamma_{M-1} < ... < \gamma_0$. We stop the chain when γ_M is small. The construction of multiple MCMC chains speeds up convergence of subsequent chains with the help of previous chains. This is motivated by equi-energy sampler (Kou et al. [2006]), in which a sequence of distributions indexed by a temperature ladder is created and the flat distributions help the rough ones to be sampled faster.

Take FN model for an example. If we are not satisfied with the decay rate of the autocorrelations of our MCMC samples with $\gamma = 2$ in Figure 2.8, we can construct a subsequent chain with $\gamma = 1$ by allowing it to swap with MCMC samples generated by the previous chain with $\gamma = 2$. As a result, the autocorrelations of MCMC samples decrease significantly with the help of previous chains, which is shown in Figure 2.24.

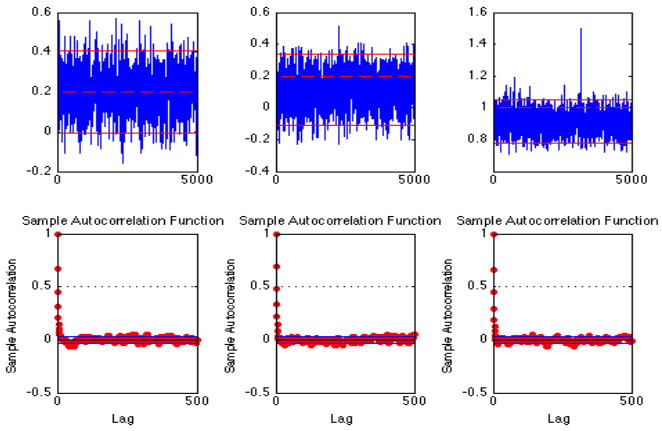

Figure 2.24: The simulated result for FN model with $\gamma = 1$, amid the swap with the MCMC samples generated by the previous chain with $\gamma = 2$

Chapter 3

Gaussian Emulator: a Full Gibbs Sampler Scheme for Inference of Dynamic Systems

To be constantly aware of the models is known as black virtue.

- Lao Tzu

3.1 The Duality of the Two Systems

The approximate approach for sampling $p(\Theta|\{x_i(\mathbf{t})\}, \{y_i(\mathbf{t})\}, \sigma^2)$ in Chapter 1 is equivalent to minimizing H' if we put a flat prior on Θ. Notice that the form of H' is exactly one form of K-L divergence we defined early on. Recall that the hyperparameters $\{\alpha_i\}$ are the minimizer of the "distance" given the parameter Θ. Now Θ can be obtained by minimizing the "distance" given the hyperparameters $\{\alpha_i\}$. This duality between the Gaussian system and dynamic system, as well as the

duality between $\{\alpha_i\}$ and Θ, allows us to generate some variations of the preceding sampling scheme.

Variation 1: The first variation of the preceding sampling scheme is to calculate $\{\alpha_i\}$ and Θ together such that the "distance" is minimized.

Variation 2: If we treat the Gaussian system as our target system and our dynamic system as our approximation system, then we can calculate Θ such that the "distance" is minimized, then calculate $\{\alpha_i\}$ such that the updated "distance" (with new Θ plug-in) is minimized.

Variation 3: In the Gaussian system, we can obtain $\{\alpha_i\}$ by cross-validation, that is,

$$\{\alpha_i\} = \text{argmin} \sum_{i=1}^{N} \sum_{j=1}^{n} [(Y_{G,i}(t_j) - \hat{\mu}_i(t_j))^2 + \hat{\sigma}_i^2(t_j)]$$

where $\hat{\mu}_i(t_j) = E(Y_{G,i}(t_j)|Y_G \backslash Y_{G,i}(t_j))$ and $\hat{\sigma}_i^2(t_j) = Var(Y_{G,i}(t_j)|Y_G \backslash Y_{G,i}(t_j))$. Then we calculate Θ such that the distance is minimized.

3.2 A Coherent Sampling Framework: Gaussian Emulator

Can we devise a coherent sampling scheme such that all the variations can be embedded into that framework? One potential drawback of this sampling scheme is that optimization steps might slow down the whole sampling scheme when the dynamic system grows large. So instead of optimizing over an objective function,

we will view the objective function as an *energy* function. In this case, we have the objective function of cross-validation as an *energy* function for $\{\alpha_i\}$ and σ_G^2, defined as follows:

$$U(\{\alpha_i\}, \sigma_G^2|Y_G) = \sum_{i=1}^{N}\sum_{j=1}^{n}[(Y_{G,i}(t_j) - \hat{\mu}_i(t_j))^2 + \hat{\sigma}_i^2(t_j)]. \tag{3.1}$$

Then the posterior distribution of $\{\alpha_i\}$ and σ_G^2 is

$$P(\{\alpha_i\}, \sigma_G^2|Y) \propto e^{-\frac{U(\{\alpha_i\},\sigma_G^2|Y)}{T_1}} \tag{3.2}$$

where T_1 is a temperature.

Similarly, we have the "distance" as an *energy* function for Θ. Let

$$KL = \prod_i \exp(-\frac{1}{2}(f_i - \tilde{\mu}_i)^T(\tilde{\Sigma}_i + \frac{1}{2}\gamma I_n)^{-1}(f_i - \tilde{\mu}_i)). \tag{3.3}$$

Then we have

$$P(\Theta|X, Y, \sigma^2, \{\alpha_i\}) \propto e^{-\frac{KL}{T_2}} \tag{3.4}$$

where T_2 is a temperature.

This framework would embed all the variations of the preceding sampling scheme and enable us to perform a full Gibbs Sampler. We call it a *Gaussian Emulator*.

Gaussian Emulator: a Gibbs sampler scheme

- Sample $\{\alpha_i\}, \sigma_G^2$ from $P(\{\alpha_i\}, \sigma_G^2|Y_G, X_G, \Theta)$
- Sample X_G from $P(X_G|\{\alpha_i\}, \sigma_G^2, Y, \Theta)$

- Substitute X_G, σ_G^2 in the Gaussian system to be X, σ^2 in the dynamic system
- Sample Θ from $P(\Theta|X, Y, \sigma^2, \{\alpha_i\})$

Denote $l = -\sum_i \log P_{\alpha_i}(x_{G,i}(\mathbf{t})|y_{G,i}(\mathbf{t}), \sigma_G^2)$, then $P(X_G|\{\alpha_i\}, \sigma_G^2, Y_G) \propto \exp(-l)$ and l is the *energy* function for X_G in this case. $P(\{\alpha_i\}, \sigma_G^2|Y_G, X_G, \Theta)$ and $P(X_G|\{\alpha_i\}, \sigma_G^2, Y, \Theta)$ can be easily calculated via Bayes' rule:

$$\begin{aligned} P(\{\alpha_i\}, \sigma_G^2|Y_G, X_G, \Theta) &\propto P(\{\alpha_i\}, \sigma_G^2|Y)P(X_G|\{\alpha_i\}, \sigma_G^2, Y_G)P(\Theta|\{\alpha_i\}, \sigma^2, Y, X) \\ &\propto \exp(-(\frac{U(\{\alpha_i\}, \sigma_G^2|Y)}{T_1} + l + \frac{KL}{T_2})). \end{aligned} \tag{3.5}$$

Similarly, we have

$$\begin{aligned} P(X_G|\{\alpha_i\}, \sigma_G^2, Y_G, \Theta) &\propto P(X_G|\{\alpha_i\}, \sigma_G^2, Y_G)P(\Theta|\{\alpha_i\}, \sigma^2, Y, X) \\ &\propto \exp(-(l + \frac{KL}{T_2})). \end{aligned} \tag{3.6}$$

Gaussian Emulator, a full Gibbs sampler scheme, circumvents the optimization steps, which can further boost the computational speed. On the other hand, sampling $\{\alpha_i\}$ and σ_G^2 from (3.5) combines the information from the cross-validation of its own Gaussian system and from the goodness-of-approximation to the dynamic system by its Gaussian system.

Recall that in our preceding sampling scheme for the partially observed case, we had to model the cross-correlation between observed components and missing components in our Gaussian system, adding more complexity to the original algorithms. However, we can still keep the independence assumption of $X_{G,i}$ for $1 \leq i \leq N$ in our Gaussian Emulator when we can apply the sampling scheme to the partially observed

case. We will elaborate this part in the following section. If we want to further speed up our sampling scheme, a shortcut of Gaussian Emulator could be:

- Sample $\{\alpha_i\}, \sigma_G^2$ from $\exp(-(\frac{U(\{\alpha_i\},\sigma_G^2|Y)}{T_1} + \frac{KL}{T_2}))$
- Sample X_G directly from $P_{\alpha_i}(x_{G,i}(\mathbf{t})|y_{G,i}(\mathbf{t}), \sigma_G^2) \sim N(\mu_i, \Sigma_i)$ for $1 \leq i \leq N$
- Substitute X_G, σ_G^2 in the Gaussian system to be X, σ^2 in the dynamic system
- Sample Θ from $P(\Theta|\{\alpha_i\}, \sigma^2, Y, X) \propto \exp(-\frac{KL}{T_2})$

This shortcut of Gaussian Emulator and the full version of Gaussian Emulator provide similar computational results for the fully observed case. However, for the partially observed case, the shortcut of Gaussian Emulator does not enjoy the same privilege as the full version does, which is preserving the independence assumption of $X_{G,i}$ for $1 \leq i \leq N$. A further modification, such as modeling the cross-correlation between observed components and missing components in Gaussian systems, has to be made for the shortcut.

3.2.1 Gaussian Emulator for the partially observed case

As we discussed in the previous section, Gaussian Emulator allows us to preserve the independence assumption of $X_{G,i}$ for $1 \leq i \leq N$ for the partially observed case, and we just need to modify two energy functions in our Gaussian Emulator for the fully observed case. Let J be the set of indices of the observed components in the dynamic systems. We define the energy functions, $U_M(\{\alpha_i\}, \sigma_G^2|Y)$ and l_M, for $\{\alpha_i\}, \sigma_G^2$ and

X_G respectively in the partially observed case as follows:

$$\begin{aligned} U_M(\{\alpha_i\}, \sigma_G^2|Y) &= \sum_{i\in J}\sum_{j=1}^{n}[(Y_{G,i}(t_j) - \hat{\mu}_i(t_j))^2 + \hat{\sigma}_i^2(t_j)] \\ l_M &= -\sum_{i\in J}\log P_{\alpha_i}(x_{G,i}(\mathbf{t})|y_{G,i}(\mathbf{t}), \sigma_G^2) - \sum_{i\in N\setminus J}\log\pi_{\alpha_i}(x_{G,i}(\mathbf{t})) \end{aligned} \tag{3.7}$$

Then a sampling scheme for the partially observed case could be:

- Sample $\{\alpha_i\}, \sigma_G^2$ from $P(\{\alpha_i\}, \sigma_G^2|Y_G, X_G, \Theta) \propto \exp(-(\frac{U_M}{T_1} + l_M + \frac{KL}{T_2}))$
- Sample X_G from $P(X_G|\{\alpha_i\}, \sigma_G^2, Y, \Theta) \propto \exp(-(l_M + \frac{KL}{T_2}))$
- Substitute X_G, σ_G^2 in the Gaussian system to be X, σ^2 in the dynamic system
- Sample Θ from $P(\Theta|X, Y, \sigma^2, \{\alpha_i\}) \propto \exp(-\frac{KL}{T_2})$

3.3 The Augmentation of Gaussian Emulator: Modeling the Mean of X_G

Notice that previously we assumed $\mu_{G,i}$ to be a zero vector. A further step could be that we put a functional form on $\mu_{G,i}$, for example $\mu_{G,i} = A_i + B_i\sin(C_it + D_i)$ and $C_i > 0$ and $D_i \in [0, 2\pi]$. By introducing such functional forms for $\mu_{G,i}$, our Gaussian Emulator would be better off capturing the dynamic behavior of the true solutions, notably X, especially in the partially observed case. In our functional form, A_i is a non-zero center amplitude and B_i, the amplitude, is the peak deviation of the function from zero. C_i, the *angular frequency*, is the rate of change of the function argument in units of radians per second. D_i, the *phase*, specifies (in radians) where in

its cycle the oscillation is at $t = 0$. We can further impose a hierarchical structure for C_i and D_i, $1 \leq i \leq N$, in which $C_i \sim \varphi_1(F, G)$ and $D_i \sim \varphi_2(H, J)$, where $\varphi_1(.)$ and $\varphi_2(.)$ are distribution forms. Since $C_i > 0$, $\varphi_1(.)$ could be a log-normal distribution, whereas for D_i, since $D_i \in [0, 2\pi]$, $D_i/2\pi$ can follow a Beta distribution, or a data transformation of $D_i/2\pi$, such as logit function or -log(-log(.)), could follow a normal distribution. Denoting $\mathcal{A} = (F, G, H, J)$ and $\mathcal{B} = \{A_i, B_i, C_i, D_i, 1 \leq i \leq N\}$, we propose an augmented version of Gaussian Emulator as follows:

The Augmented Gaussian Emulator for the Fully Observed Case

- Sample $\mathcal{A}$ from $P(\mathcal{A}|\mathcal{B}, \{\alpha_i\}, \sigma_G^2, Y, X, \Theta) = P(\mathcal{A}|\mathcal{B})$
- Sample $\mathcal{B}$ from $P(\mathcal{B}|\mathcal{A}, \{\alpha_i\}, \sigma_G^2, Y, X, \Theta) \propto P(\mathcal{B}|\mathcal{A})\exp(-(\frac{U}{T_1} + l + \frac{KL}{T_2}))$
- Sample $\{\alpha_i\}, \sigma_G^2$ from $P(\{\alpha_i\}, \sigma_G^2|Y, X, \Theta) \propto \exp(-(\frac{U}{T_1} + l + \frac{KL}{T_2}))$
- Sample X_G from $P(X_G|\{\alpha_i\}, \sigma_G^2, Y, \Theta) \propto \exp(-(l + \frac{KL}{T_2}))$ and substitute X_G, σ_G^2 to be X, σ^2
- Sample Θ from $P(\Theta|\{\alpha_i\}, \sigma^2, Y, X) \propto \exp(-\frac{KL}{T_2})$

The Augmented Gaussian Emulator for the Partially Observed Case

- Sample $\mathcal{A}$ from $P(\mathcal{A}|\mathcal{B}, \{\alpha_i\}, \sigma_G^2, Y, X, \Theta) = P(\mathcal{A}|\mathcal{B})$
- Sample $\mathcal{B}$ from $P(\mathcal{B}|\mathcal{A}, \{\alpha_i\}, \sigma_G^2, Y, X, \Theta) \propto P(\mathcal{B}|\mathcal{A})\exp(-(\frac{U_M}{T_1} + l_M + \frac{KL}{T_2}))$
- Sample $\{\alpha_i\}, \sigma_G^2$ from $P(\{\alpha_i\}, \sigma_G^2|Y, X, \Theta) \propto \exp(-(\frac{U_M}{T_1} + l_M + \frac{KL}{T_2}))$

- Sample X_G from $P(X_G|\{\alpha_i\}, \sigma_G^2, Y, \Theta) \propto \exp(-(l_M + \frac{KL}{T_2}))$ and substitute X_G, σ_G^2 to be X, σ^2
- Sample Θ from $P(\Theta|\{\alpha_i\}, \sigma^2, Y, X) \propto \exp(-\frac{KL}{T_2})$

3.4 Numerical Examples

3.4.1 Fitting the FitzHugh-Nagumo equations

We apply our Gaussian Emulator to the same simulated data in Chapter 2 for the FitzHugh-Nagumo equations. We examine our Gaussian Emulator on the fully observed case and the partially observed case. We first apply our Gaussian Emulator, assuming $\mu_{G,i} = 0$ for the fully observed cases and run for 50,000 iterations. We take $T_1 = T_2 = 1$ and $\gamma = 2$. Figure 3.1 shows that our Gaussian Emulator converges fast and the true values are within 95% confidence regions. Figure 3.2 shows that our Gaussian Emulator mimics the true solutions pretty well. The augmented Gaussian Emulator provides the similar simulated results, see Figure 3.3 and Figure 3.4. The comparison of histograms of Monte Carlo samples generated by the augmented Gaussian Emulator and numerical solvers in Figure 3.9 indicates that our Gaussian Emulator is superior to the numerical solver.

For the partially observed case, we again assume that data for V are unobserved. Although our Gaussian Emulator with $\mu_{G,i} = 0$ can estimate parameters fairly well in Figure 3.5, it does not well capture the amplitude of the true solutions V (see Figure 3.6). However, in addition to estimating parameters well in Figure 3.7, our

augmented Gaussian Emulator significantly improves the estimation of the trajectory of missing components, shown in Figure 3.8. The comparison with the numerical solver for the partially observed case again in Figure 3.10 again demonstrates that our Gaussian Emulator is superior to the numerical solver.

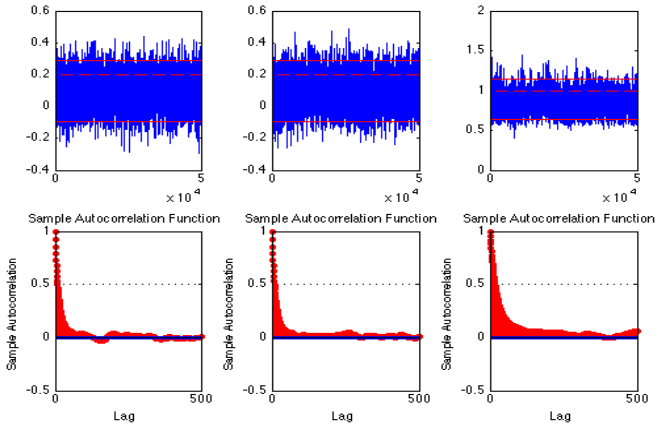

Figure 3.1: Gaussian Emulator for the FN model for the fully observed case. The 95% confidence regions are represented by two solid red lines and the true solutions are represented by dashed lines. **Left:** simulated results for parameter a. **Middle:** simulated results for parameter b. **Right:** simulated results for parameter c.

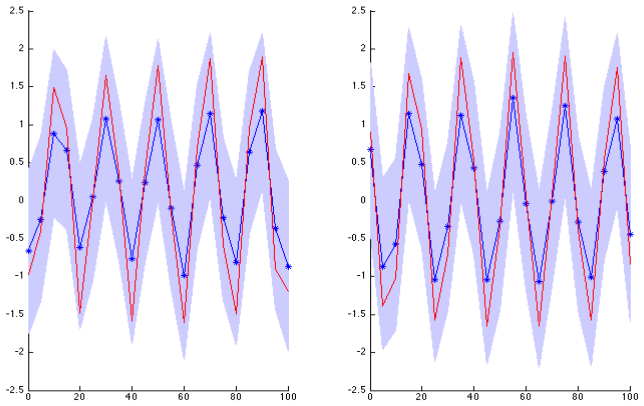

Figure 3.2: Gaussian Emulator for the FN model for the fully observed case. The grey area represents 95% confidence regions, the blue line represents the median of Monte Carlo samples, and the red line represents the numerical solutions. **Left:** simulated results for V. **Right:** simulated results for R.

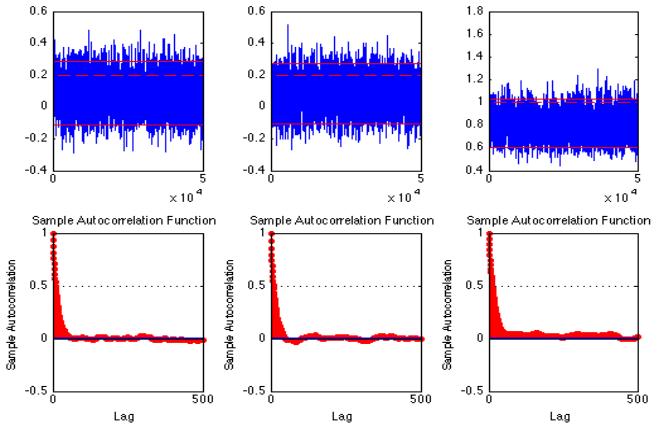

Figure 3.3: The augmented Gaussian Emulator for FN model for the fully observed case. The 95% confidence regions are represented by two solid red lines and the true solutions are represented by dashed lines. **Left:** simulated results for parameter a. **Middle:** simulated results for parameter b. **Right:** simulated results for parameter c.

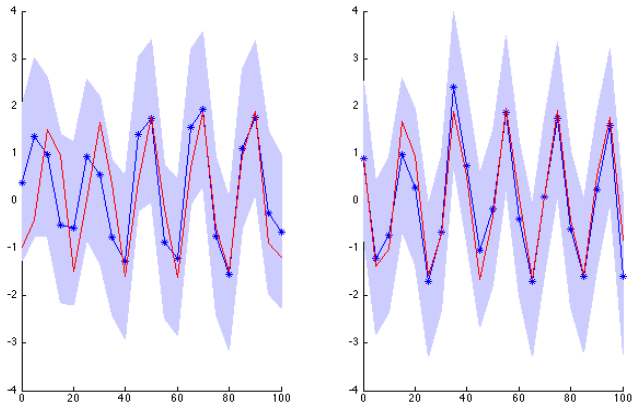

Figure 3.4: The augmented Gaussian Emulator for the FN model for the fully observed case. The grey area represents 95% confidence regions, the blue line represents the median of Monte Carlo samples, and the red line represents the numerical solutions. **Left:** simulated results for V. **Right:** simulated results for R.

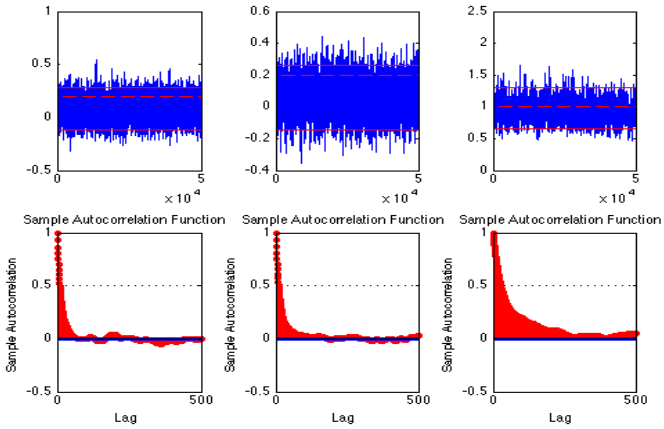

Figure 3.5: Gaussian Emulator for the FN model for the partially observed case. The 95% confidence regions are represented by two solid red lines and the true solutions are represented by dashed lines. **Left:** simulated results for parameter a. **Middle:** simulated results for parameter b. **Right:** simulated results for parameter c.

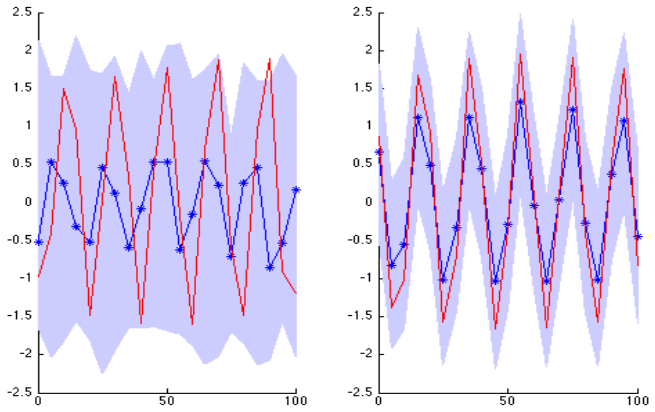

Figure 3.6: Gaussian Emulator for the FN model for the partially observed case. The grey area represents 95% confidence regions, the blue line represents the median of Monte Carlo samples, and the red line represents the numerical solutions. **Left:** simulated results for the missing component V. **Right:** simulated results for the observed component R.

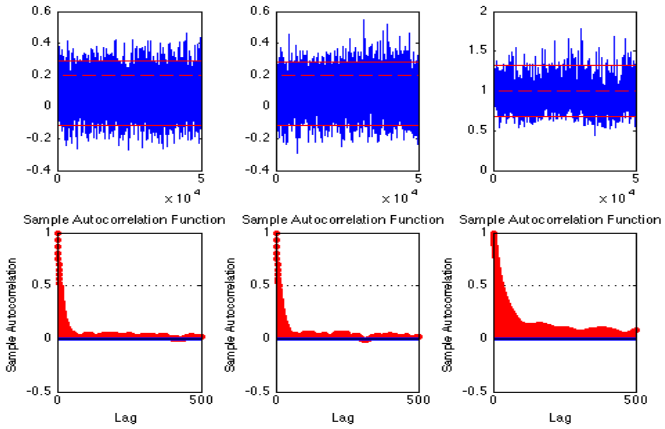

Figure 3.7: The augmented Gaussian Emulator for the FN model for the partially observed case. The 95% confidence regions are represented by two solid red lines and the true solutions are represented by dashed lines. **Left:** simulated results for parameter a. **Middle:** simulated results for parameter b. **Right:** simulated results for parameter c.

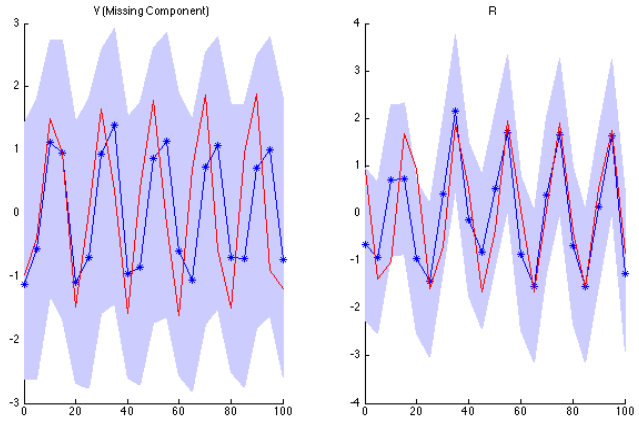

Figure 3.8: The augmented Gaussian Emulator for the FN model for the partially observed case. The grey area represents 95% confidence regions, the blue line represents the median of Monte Carlo samples, and the red line represents the numerical solutions. **Left:** simulated results for the missing component V. **Right:** simulated results for the observed component R.

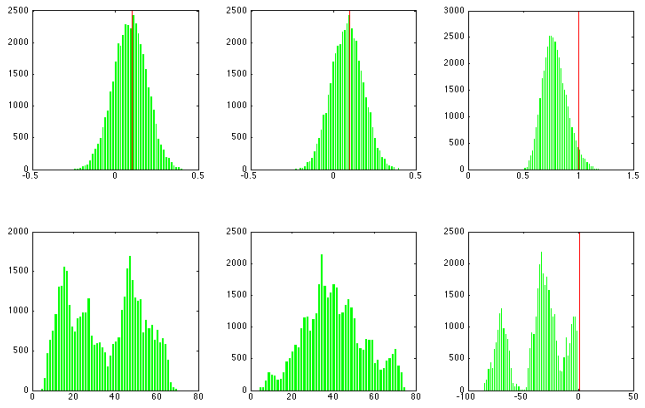

Figure 3.9: Comparison of the augmented Gaussian Emulator and the numerical solver for the FN model for the fully observed case. The red vertical line represents the numerical solutions. The first row is the histogram of the samples from Gaussian emulator and the second row is the histogram of the samples from numerical solver.

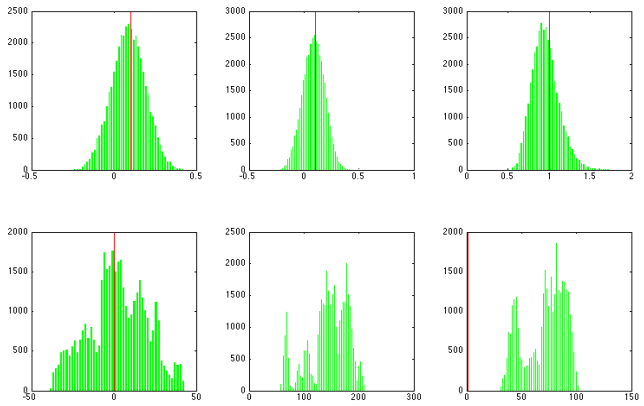

Figure 3.10: Comparison of the augmented Gaussian Emulator and the numerical solver for the FN model for the partially observed case. The red vertical line represents the numerical solutions. The first row is the histogram of the samples from Gaussian emulator and the second row is the histogram of the samples from numerical solver.

3.4.2 Fitting the Repressilator model

We apply our Gaussian Emulator on the same simulated data in Chapter 2 for the Repressilator model. We examine our Gaussian Emulator on the fully observed case and the partially observed case. We first apply our Gaussian Emulator, assuming $\mu_{G,i} = 0$ for the fully observed cases and run for 50,000 iterations. We take $T_1 = T_2 = 1$ and $\gamma = 10$. Figure 3.11 shows that our Gaussian Emulator converges fast and the true values are within 95% confidence regions. Figure 3.12 shows that our Gaussian Emulator mimics the true solutions pretty well. The augmented Gaussian Emulator provides the similar simulated results; see Figure 3.13 and Figure 3.14. The comparison of histograms of Monte Carlo samples generated by the augmented Gaussian Emulator and numerical solvers in Figure 3.19 indicates that our Gaussian emulator is superior to the numerical solver.

For the partially observed case, we again assume that data for protein components p_1, p_2 and p_3 are unobserved. In addition to estimating parameters well (see Figure 3.15), our Gaussian Emulator with $\mu_{G,i} = 0$ can estimate missing components fairly well (see Figure 3.16). However, with the additional ingredients from the augmented Gaussian Emulator, we can almost perfectly emulate the trajectory of missing components, shown in Figure 3.17 and Figure 3.18. The comparison with the numerical solver for the partially observed case again in Figure 3.20 again demonstrates that our Gaussian Emulator is superior to the numerical solver.

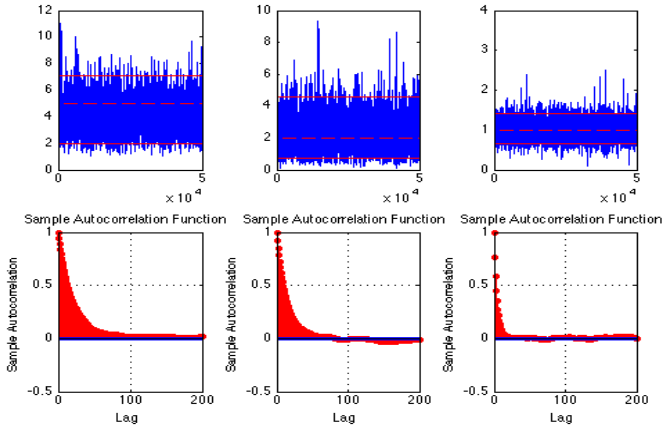

Figure 3.11: Gaussian Emulator for the Repressilator model for the fully observed case. The 95% confidence regions are represented by two solid red lines and the true solutions are represented by dashed lines. **Left:** simulated results for parameter a. **Middle:** simulated results for parameter b. **Right:** simulated results for parameter c.

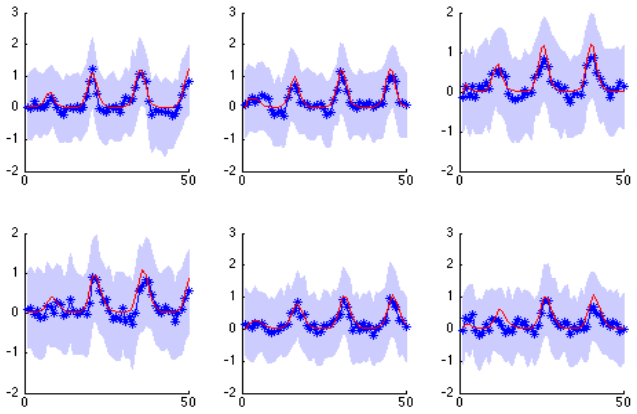

Figure 3.12: Gaussian Emulator for the Repressilator model for the fully observed case. The grey area represents 95% confidence regions, the blue line represents the median of Monte Carlo samples, and the red line represents the numerical solutions. The first row (from left to right): m_1, m_2, m_3 and the second row (from left to right): p_1, p_2, p_3.

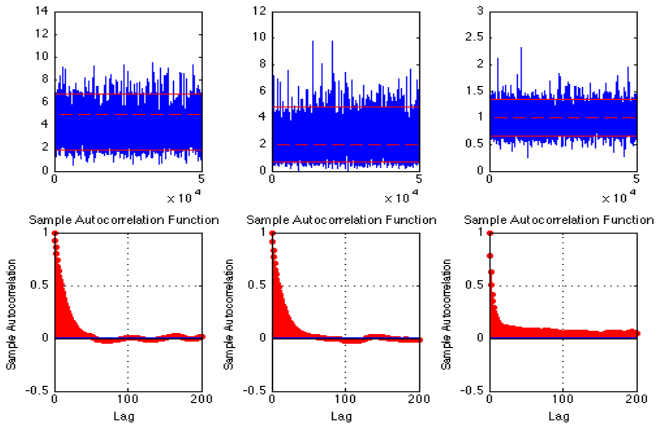

Figure 3.13: The augmented Gaussian Emulator for Repressilator model for the fully observed case. The 95% confidence regions are represented by two solid red lines and the true solutions are represented by dashed lines. **Left:** simulated results for parameter a. **Middle:** simulated results for parameter b. **Right:** simulated results for parameter c.

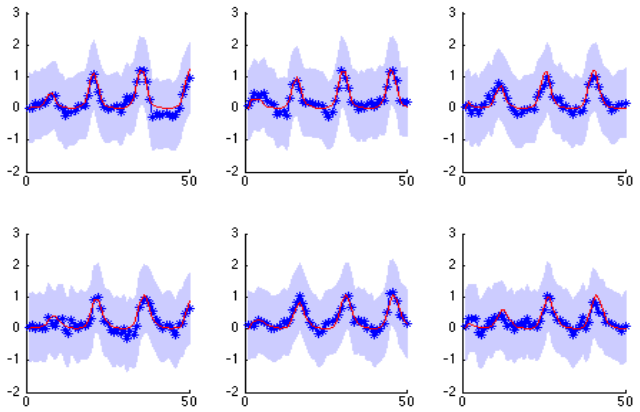

Figure 3.14: The augmented Gaussian Emulator for the Repressilator model for the fully observed case. The grey area represents 95% confidence regions, the blue line represents the median of Monte Carlo samples, and the red line represents the numerical solutions. The first row (from left to right): m_1, m_2, m_3 and the second row (from left to right): p_1, p_2, p_3.

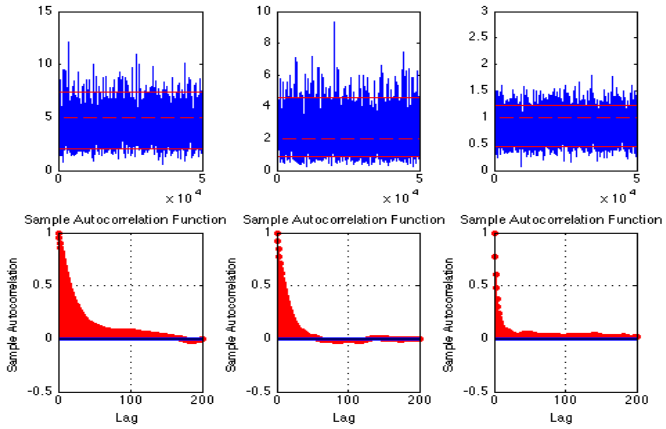

Figure 3.15: Gaussian Emulator for the Repressilator model for the partially observed case. The 95% confidence regions are represented by two solid red lines and the true solutions are represented by dashed lines. **Left:** simulated results for parameter a. **Middle:** simulated results for parameter b. **Right:** simulated results for parameter c.

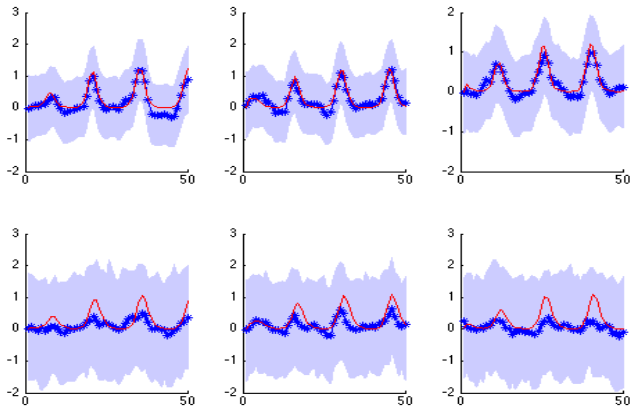

Figure 3.16: Gaussian Emulator for the Repressilator model for the partially observed case. The grey area represents 95% confidence regions, the blue line represents the median of Monte Carlo samples, and the red line represents the numerical solutions. The first row (from left to right): m_1, m_2, m_3 and the second row for missing components (from left to right): p_1, p_2, p_3.

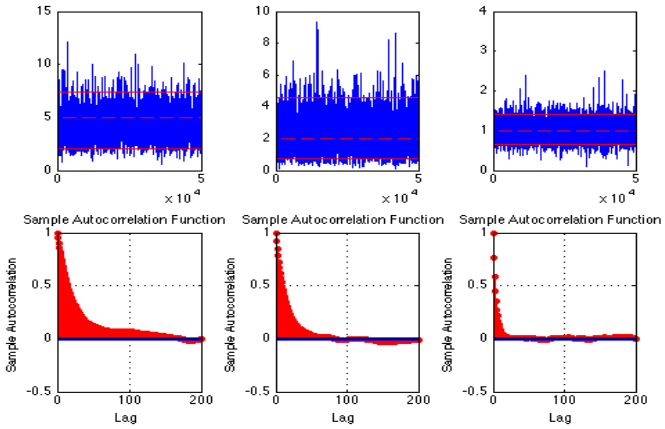

Figure 3.17: The augmented Gaussian Emulator for Repressilator model for the partially observed case. The 95% confidence regions are represented by two solid red lines and the true solutions are represented by dashed lines. **Left:** simulated results for parameter a. **Middle:** simulated results for parameter b. **Right:** simulated results for parameter c.

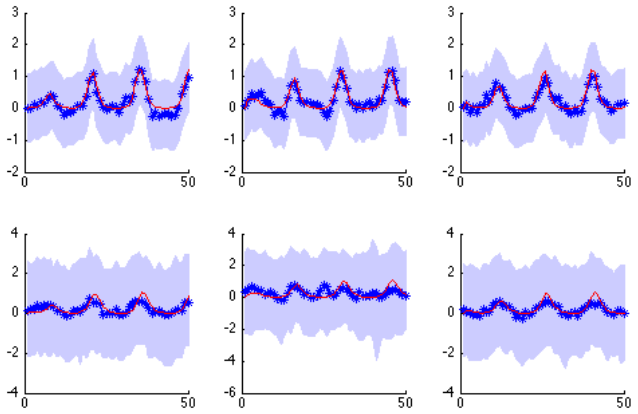

Figure 3.18: The augmented Gaussian Emulator for the Repressilator model for the partially observed case. The grey area represents 95% confidence regions, the blue line represents the median of Monte Carlo samples, and the red line represents the numerical solutions. The first row (from left to right): m_1, m_2, m_3 and the second row for missing components (from left to right): p_1, p_2, p_3.

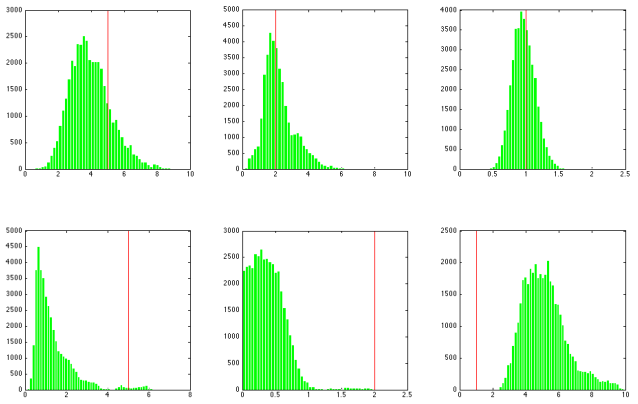

Figure 3.19: Comparison of the augmented Gaussian Emulator and the numerical solver for the Repressilator model for the fully observed case. The red vertical line represents the numerical solutions. The first row is the histogram of samples from Gaussian emulator and the second row is the histogram of the samples from numerical solver.

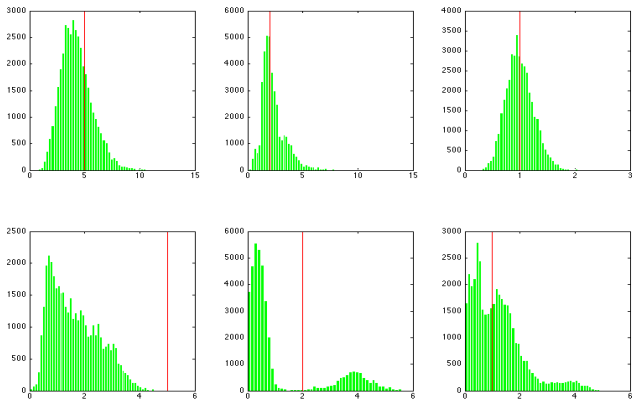

Figure 3.20: Comparison of the augmented Gaussian Emulator and the numerical solver for the Repressilator model for the partially observed case. The red vertical line represents the numerical solutions. The first row is the histogram of samples from Gaussian emulator and the second row is the histogram of the samples from numerical solver.

3.5 Delayed Differential Equation (DDE)

3.5.1 Introduction

The original repressilator model assumes that elongation, processing and export of primary gene transcripts are instantaneous processes. However, for some genes like Hes1 and p53, there is an average delay of around 10-20 minutes between the action of a transcription factor on the promotor of a gene and the appearance of the corresponding mature mRNA in the cytoplasm, see Lewin [2000]. Monk [2003] referred to this overall delay as the *transcriptional delay*. Similarly, synthesis of a typical protein from mRNA takes around 1-3 minutes and results in a translational delay. In principle, such delays can result in oscillatory mRNA and protein expression. However, experimental evidence of such delay-driven oscillations has been lacking. By

incorporating a time-delayed parameter into the original repressilator model, we show that the observed oscillatory expression and activity of Hes1 is most likely driven by transcriptional delays; see Figure 3.21.

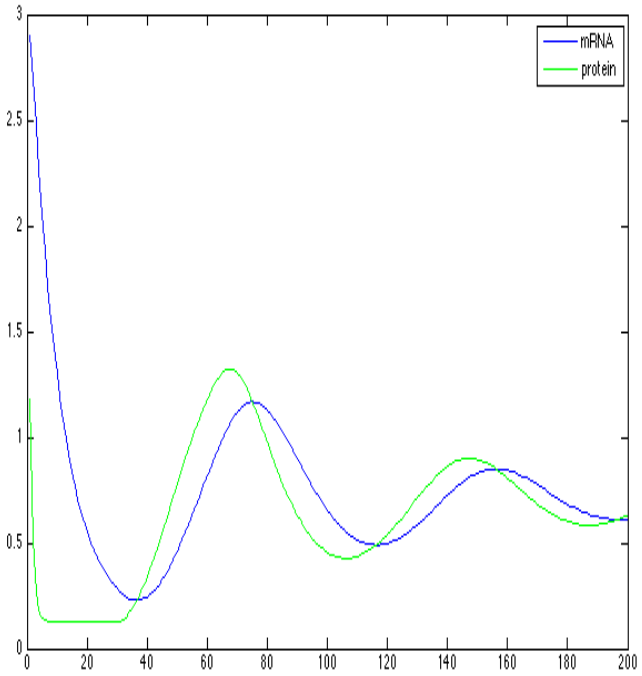

Figure 3.21: The oscillatory expression of mRNA and protein

Taking into account the transcriptional time delay, and denoting the concentration at time t of Hes1 mRNA by $M(t)$ and protein by $P(t)$, this system can be represented by DDEs as follows:

$$\begin{aligned} \frac{dM}{dt} &= \frac{a}{(1+P(t-\tau))^b} - M(t) \\ \frac{dP}{dt} &= c(M(t) - P(t)) \end{aligned}$$

Here the delay τ represents the sum of the transcriptional and translational time delays.

In addition to the above DDE model, other DDE models have been widely used in ecology (Gurney et al. [1980]), physiology (Mackey and Glass [1977]) and many other fields. A general form of DDE can be expressed as follows:

$$\dot{X}(t) = f(X(t), X(t-\tau)|\Theta) \tag{3.8}$$

where $X(t)$ is the dynamic process on $[t_1, t_n]$, τ is a constant delay parameter, and Θ is the parameter of interest. The DDE usually has no analytic solutions and hence can only be solved numerically. Notice that the DDE solutions will depend on not only the parameters Θ and τ, but also on the history of the dynamic process $H = \{x_i(t), t \in [t_1 - \tau, t_1], 1 \leq i \leq N\}$, which is an infinite-dimensional set.

There are a few papers on the parametric estimation of delayed differential equations. Fowler and Kember [1993] estimated the derivative $\dot{X}(t)$ using a finite difference $X(t) - X(t-\delta)$ to deal with the data assumed to be collected with little or no errors, where δ is a sufficiently small delay. They then embedded the dynamic process in a low-dimensional space $(X(t), X(t-\delta), X(t-\Delta))$ to identify the delay parameter, where Δ is selected as the location to have an abrupt change of the volume of this space. Bunner et al. [1996] studied a special DDE model

$$f(X(t), X(t-\tau)|\Theta) = -X(t) + g(X(t-\tau)).$$

Based on the fact that when $\dot{X}(t_i) = 0$ at the time point t_i, we have $X(t_i) = g(X(t_i -$

τ)). They then estimated the delay parameter by using the value that gave the smoothest path of $X(t_i)$ versus $X(t_i-\tau)$. Ellner et al. [1997] unified the above methods and extended them to the case where the data have measurement errors. They first applied nonparametric smoothing methods to estimate the derivatives $\dot{X}(t)$ from the noisy data, and then inferred $f(.)$ by using a generalized additive model. However, the inaccurate estimation for $\dot{X}(t)$ from the noisy data leads to the estimation for the DDE parameters with large errors. Horbelt et al. [2002] proposed a method for estimation of parameters of nonlinear delayed feedback systems. The idea is very similar to the nonlinear least squares method proposed by Biegler et al. [1986]. Guesses for initial values x_0 are chosen, as well as parameters of interest Θ including a delayed parameter τ. The DDE is solved numerically. The objective functional is then calculated as a sum of squared residues between the data and the model trajectory, weighted with inverse variances of the data:

$$\chi^2(x_0, \Theta) = \sum_{i=1}^{N} (y_i(t) - x_i(t))^2 / \sigma_i^2.$$

The required parameters are identified as those minimizing $\chi^2(x_0, \Theta)$. This is a very difficult optimization problem since the DDE numerical solutions depend on not only the parameter values and initial guess, but also on the history of the dynamic process H_τ. In this case, it becomes an infinite-dimensional optimization problem, more difficult than estimating parameters in ordinary differential equations. This method is further based on the numerical solution of a DDE, which is a computational hurdle.

In order to bypass the numerical solver, Wang and Cao [2012] extended the generalized smoothing method proposed by Ramsay et al. [2007] to estimating parameters

in the DDE. They used a set of flexible nonparametric functions to approximate the dynamic process. The coefficients c in these nonparametric functions are estimated by maximizing the penalized likelihood function $J(c|\Theta)$:

$$J(c|\Theta) = \sum_{i=1}^{N}\sum_{j=1}^{n} \log L(y_i|x(t_j)) - \lambda \sum_{i=1}^{N} \int_{t_1+\tau}^{t_n} \left\{\dot{x}_i(\mathbf{t}) - \mathbf{f_i}(\mathbf{X(t)}, \mathbf{X(t-\tau)}|\mathbf{\Theta})\right\}^{\mathbf{2}} \mathbf{dt}$$

where the smoothing parameter λ controls the trade-off between fit to the data and fidelity to the DDE model. Their simulation studies showed that their semiparametric method obtained more accurate estimates for the DDE parameters than some alternative approaches.

3.5.2 Numerical examples

In the this section, we will apply our Gaussian Emulator to the repressilator model with delayed parameter as we stated earlier. Given the measured mRNA and protein half-lives, a sustained oscillation with a period of 2 hours can be induced only if $b > 4$; see Monk [2003]. Therefore, in our case, we take $b = 6$ for illustrative purposes. The oscillatory expression of mRNA and protein, simulated by the DDE above when $a = 3, c = 0.1, \tau = 30$ is shown in Figure 3.21.

Data is generated from the model with $a = 3, c = 0.1, \tau = 30$ at 120 time points with Gaussian noise, $N(0, 1)$. We examine our Gaussian Emulator on the fully observed case. We apply our Gaussian Emulator, and run for 50,000 iterations. We take $T_1 = T_2 = 1$ and $\gamma = 2$. Figure 3.22 shows that our Gaussian Emulator converges fast and the true values are within 95% confidence regions. The histogram of Monte Carlo samples is in Figure 3.23. Figure 3.24 shows that our Gaussian Emulator mimics the

true solutions quite well.

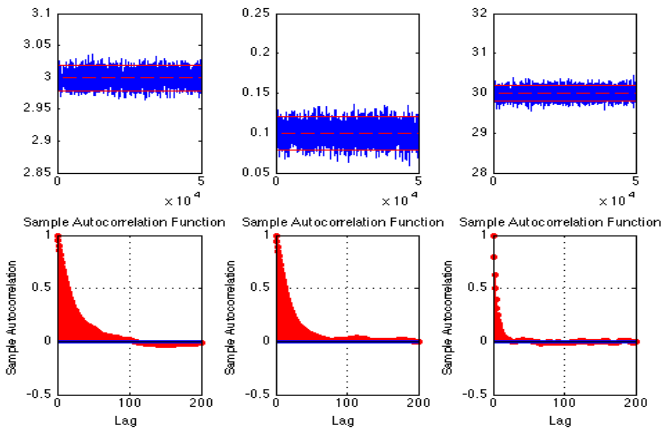

Figure 3.22: Gaussian Emulator for DDE. The 95% confidence regions are represented by two solid red lines and the true solutions are represented by dashed lines. **Left:** simulated results for parameter a. **Middle:** simulated results for parameter b. **Right:** simulated results for parameter τ.

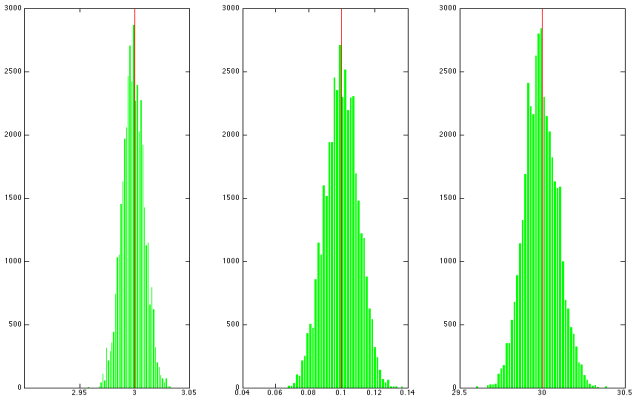

Figure 3.23: The histogram of Monte Carlo samples by Gaussian emulator for DDE. The red vertical line represents the true values. **Left:** simulated results for parameter a. **Middle:** simulated results for parameter b. **Right:** simulated results for parameter τ.

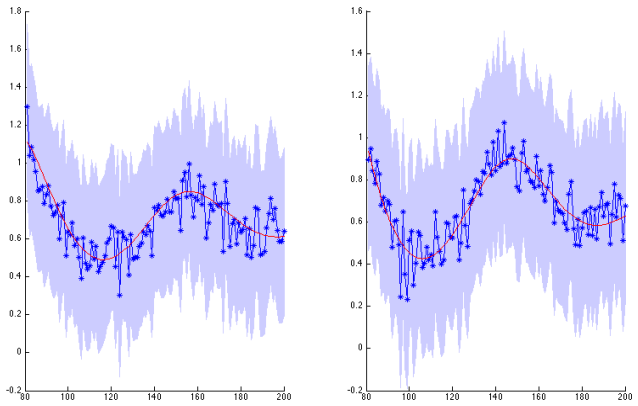

Figure 3.24: Gaussian Emulator for DDE. The grey area represents 95% confidence regions, the blue line represents the median of Monte Carlo samples, and the red line represents the numerical solutions. **Left:** simulated results for $M(t)$. **Right:** simulated results for $P(t)$.

3.6 Conclusion and Further Research

Differential equations have a long and illustrious history in mathematical modeling for physical and biological process. However, statistical inference such as parametric estimation of differential equations from complete or partially observed data has just emerged in the last decade. The Bayesian approach usually requires numerical solution of differential equations, which becomes a major bottleneck for inference of dynamic systems. We have addressed this problem by creating an artificial system driven by Gaussian process to approximate the dynamic system, bypassing the numerical solvers. We illustrated our method, named as Gaussian Emulator, on numerical examples in neuroscience and biology, where the collected data might be complete or partially observed. Our simulated results show that Gaussian Emulator can dramatically save the computational time and achieve a great estimation accuracy. Our

approach also benefited from not requiring an initial value of the dynamic system, which makes our method applicable in real scientific problems.

However, the theoretical justification on the convergence of Gaussian Emulator has yet to be done. We only provided some heuristic guidance on how to choose γ in Section 2.6.2. It is interesting to see some theoretical results on picking up the optimal γ. On the other hand, The normality assumption on the collected data might be restricted. It is possible to extend our method to experimental data that is not normally-distributed. In this case, a Gaussian process prior might not be directly applied. However, some transformation of Gaussian process could be an alternative candidate to address this issue. Our current approach deals only with constant parameters of dynamic systems. Another possible direction is to extend our method to estimate the time-varying parameters of dynamic systems.

Chapter 4

Statistical Learning of Market Microstructure of Hidden Liquidity: How Much Can Hidden Liquidity Improve Trading Prices?

Dim and dark,
Yet within it is an essence.
This essence is quite genuine
And within it is something that can be tested.
- Lao Tzu

4.1 Introduction

With the growth of new information processing and communications technologies, electronic limit order markets have become the dominant structure for trading in financial markets, where traders can submit limit orders to supply liquidity or market orders to demand liquidity. Orders posted in those order-driven markets must show specifying sign (ask or bid), size and limit price. The information contained in displayed limit orders indicates trading intentions and may induce adverse selection effects [see e.g. Hautsch and Huang, 2012a]. In order to avoid adverse price effects, big buy side traders typically use sophisticate strategies, e.g. "slicing and dicing" or multiple brokers to reduce the information leakage. In the current decade, many exchanges and ECN re-introduce undisclosed order facilities to traders to achieve this purpose by limiting exposure of their order sizes. These facilities have driven equity markets toward more opaque market structures and away from fully transparency. The question of how much transparency should be provided in markets has become a important topic in recent market microstructure research.

Depending on the undisclosed order type, traders can partially reveal size (iceberg or reserve orders) or completely hide size (non-display or hidden orders). Unlike limit orders, reserve orders may display a fraction of their orders to the market. Hidden orders are even more extreme to reduce transparency. They are totally invisible and do not reveal even the posted limit price. They can be pegged to the best bid (ask) or spread midpoint, which can be viewed as hidden liquidity supply in the limit order book (LOB). In other words, displayed limit orders bear exposure risk but they have priority to be executed compared to hidden orders. Therefore, the submission of

hidden orders faces a trade-off between exposure risk and execution risk.

Recent empirical evidence shows a growing usage of undisclosed order facilities and the existence of huge volumes of hidden liquidity in markets. For instance, Bessembinder et al. [2009] report that more than 44% of Euronext Paris order volume is hidden. Frey and Sandås [2009] show that iceberg orders account for 9% of non-marketable orders in German Xetra Trading. Hautsch and Huang [2012b] find that hidden liquidity contributes to more than 17% of NASDAQ trading volume. We find that the hidden liquidity accounts for 20% to 30% of depth at the first five best quotes in NASDAQ's order book.

This paper sheds some light on the hidden liquidity in an opaque limit order market. It is closely related to the study on undisclosed order strategies, e.g. Bessembinder et al. [2009] and Hautsch and Huang [2012b], but from a totally new perspective. We use the price improvement, which is the difference between *ex ante* expected transaction price and *ex post* realized price of a market order with an artificial size, as the measure for hidden liquidity. This measure naturally takes both hidden volumes and the corresponding prices into account. By varying the artificial order size, we are able to study reserve and non-display orders separately.

To our best knowledge, this paper is *first* empirical study of the dynamics of hidden orders under different market conditions. We compute the price improvement by using NASDAQ ModelView data that records the aggregated displayed and hidden volume available at each price level at one-minute intervals during pre-market and normal trading day. The price improvement of a *small* artificial market order serves as a simple benchmark for hidden liquidity within the spread, in particular, the

aggressiveness of non-display orders within the spread, whereas the price improvement of a *large* order serves as a benchmark for hidden liquidity across the whole LOB.

In order to study the effects of market conditions on hidden liquidity, we retrieve order flow information from NASDAQ Totalview data. We then propose a zero-inflated gamma model on the price improvement of a *small* order in the presence of hidden orders given the state of the market. The evidence shows that it is adequate to identify hidden volumes within spread under different market conditions. However, it is insufficient to explain the price improvement of a *large* order due to the noisy levels of undisclosed (reserve and non-display) orders outside the spread. We further propose a nonparametric model with additional economic motivation to identify hidden liquidity across the whole limit order book.

Statistical inference on those models allows us to test some theoretical results on the submission strategy of undisclosed orders, as well as the associated economic theory on the relation between market conditions and traders' incentive to use undisclosed orders. Our findings, based on several NASDAQ stocks, show that the distribution of hidden orders is significantly driven by market conditions, which is reflected by the (visible) bid-ask spread, (visible) depth, recent price movement, (visible) executed hidden volumes and displayed limit orders updated by low-frequency traders, and thus is reliably predictable under different market conditions.

The remainder of this essay is organized as follows: Section 4.2 reviews some theoretical literatures and formulates some economic hypotheses. In Section 4.3, we construct market conditions and describe the data we use. In Section 4.4, we propose statistical models for the price improvement of a *small* order as well as a *large* order,

and report the empirical findings. Section 4.5 concludes the paper.

4.2 Review of Literature on Order Display

4.2.1 Economic reasoning

Harris [1996] and Harris [1997] discuss the difficult trading dilemma faced by a large buy-side trader. On one hand, she must show at least some of her trading interest to the other market participants. By showing her interest, e.g., using instruments like visible limit orders or market orders, she anticipates that traders with the opposite trading interest are attracted so that the execution time of her position would consequently be shortened and the transaction costs would be reduced. On the other hand, other traders may react her in unfavorable ways. For instance, the "defensive" trader who interprets the signal as inside information would refrain from trading by refusing to submit market orders or canceling existing aggressive limit orders. The parasitic trader exploits the option value of the big order by using front-running strategies, e.g., quote-matching. Consequently, the execution time and transaction costs may significantly increase.

Based on such economic reasoning, several theories on the usage of undisclosed orders have been developed. Esser and Mönch [2007] propose a static framework in which the trader optimizes the peak size and limit price of reserve orders by continuously monitoring and balancing the exposure risk with the execution risk. Moinas [2010] presents a theoretical model where informed traders, as well as large liquidity traders, use reserve orders to mitigate the information leakage. Cebiroglu and Horst

[2011] propose a model where the traders decide on the peak size of the reserve order based on the market impact of exposure defined as the effect of LOB depth imbalance on the expected execution price. Buti and Rindi [2011] present a dynamic framework where the trader decides her optimal strategy by simultaneously choosing the trading direction, the aggressiveness, the size and the peak proportion of her order. To our best knowledge, it is the only theoretical model that explicitly takes the hidden order into traders' trading options.

4.2.2 Empirical evidence

The empirical literature on reserve orders has grown remarkably in the last decade, partially due to its proliferation in LOB markets and increased availability of data. Bessembinder et al. [2009] study Euronext Paris, documenting that reserve orders are associated with lower implementation short fall costs but longer time-to-fill. De Winne and D'Hondt [2007] examine the same platform, finding that the detection of the hidden depth increases the order aggressiveness on the opposite side. Both studies show that the decision on using reserve orders is strongly related to the prevailing market conditions, such as bid-ask spread, depths in LOB and volatility.

Aitken et al. [2001] study the Australian Stock Exchanages (ASX), finding that the reserve order does not have a different price impact from the visible limit order and that the use of reserve orders increases with volatility and the average order value, while decreases in tick size and trading activity.[1] Frey and Sandås [2009] study the Deutsche Börse's trading platform Xetra, reporting that the price impact of the

[1]Although the quality of the reserve order is hidden in the ASX, the trading screen displays a "U" in the quantity field when a trader submits a reserve order.

reserve order depends on the executed fraction of its size and the profitability of it is higher if has not been detected. Pardo Tornero and Pascual [2007] study the Spanish Stock Exchange, reporting that there is no significant price impacts associated with the execution of hidden parts of reserve orders. This evidence supports the hypothesis that a liquidity trader uses reserve orders to compete the liquidity provision and to prevent pick-off risk.

Tuttle [2006] shows that the overall inside depth increases significantly after NASDAQ introduced the undisclosed orders and the hidden size is predictable for future market price movements while the visible size conveys little information. Anand and Weaver [2004] examine the abolition in 1996 and re-introduction in 2002 of reserve orders in the Toronto Stock Exchange, finding that the spread and visible depth remained unchanged after either event. However, the total depth at the inside, including both visible and hidden volume, significantly increased after the re-introduction. Both studies show that the market quality is improved after introducing reserve orders and informed traders use them to reduce price impact.

Moreover, Fleming and Mizrach [2009] examine BrockerTec, the leading interdealer ECN of the U.S. Treasury, documenting that the use of reserve orders varies considerably and the quantity of hidden depth increases with price volatility.

4.3 Market Structure and Data

4.3.1 Institutional background

As one of the largest electronic LOB markets in the world, the NASDAQ SingleBook platform provides a unified procedure for passing limit orders from ECNs (Brut and INET) and the traditional dealer-quote system. In particular, it treats the market maker's quote as a pair of limit orders on both sides (ask and bid sides) and aggregates them into a centralized order book as other orders. During continuous trading between 9:30 and 16:00 Eastern Time, the system matches incoming orders against the best (in term of price) prevailing orders (possible undisclosed) in LOB on the opposite side on the LOB. If there is insufficient volume to fully execute the incoming order, the remaining part will be consolidated into the LOB. Besides limit orders and market orders, NASDAQ provides both reserve orders and hidden orders.[2]

To encourage traders to disclose their orders, NASDAQ uses secondary order precedence rule to reward traders for disclosing their orders. As a consequence, the hidden parts of undisclosed orders lose their time priority to visible limit orders or peaks of reserve orders at the same price.

Moreover, the NASDAQ Stock Market trading rule [NASDAQ, 2008] requires a market maker to display at least one round lot size for the security at her quote

[2]NASDAQ also provides a so-called "discretionary order" which has a displayed price and size, as well as a non-displayed discretionary price range. When the discretionary price range is touched by the opposite order, the discretionary order converts to an IOC (Immediate or Cancel) market order. This order type is clearly related to trading intention hiding. However, we do not consider discretionary orders as undisclosed orders because 1) they take the liquidity from LOB rather than provide it; 2)they are not recorded as hidden liquidity in ModelView; 3) it is very difficult to identify them in TotalView-ITCH data by the method proposed by Hautsch and Huang [2012b] due to HFT algorithms generating an enormous number of IOC orders.

prices if she is not disclosing them as limit orders. In this case, the market maker's quotation is like a pair of reserve orders.

4.3.2 A measure of aggressiveness of hidden liquidity

The NASDAQ ModelView data provides insight into the markets full liquidity, including both reserve and hidden interests. In particular, it releases the historical minute-by-minute (entire) LOB with hidden liquidity explicitly recorded at each price level on a T+10 basis. It is difficult to quantify the distribution of the whole hidden orders across different price levels. Consequently, we shall develop a measure that can summarize hidden liquidity across LOB with sound economic reasoning. Quoted spread and depth are standard liquidity measures applied in dealer markets, but they are problematic in the sense that they do not reflect the liquidity supply beyond the inside quotes, and they do not incorporate the demand for liquidity. Also, they are *ex post* measures of liquidity, which might not be interesting for practitioners.

Since supply and demand are expressed most visibly in a limit-order market, we intend to define a measure of liquidity from this perspective. In a limit-order market setting, Coppejans et al. [2000] and Irvine and Kandel [2000] propose a size-related, *ex ante* liquidity measure that aggregates all limit orders on the book. We basically follow their ideas.

Let A_i denote the ith best visible ask price, and S_i denote the corresponding depth at price A_i. If we do not have hidden orders, then if an investor want to buy q shares

of the stock, he or she has to pay

$$P(q)_{ask} = \frac{1}{q}\left\{\sum_{j=1}^{k-1} S_j A_j + \left(q - \sum_{j=1}^{k-1} S_j\right) A_k\right\} \tag{4.1}$$

where k denotes the index of the last buy limit order. However, if we take hidden orders into account, let A'_i denote the ith best ask price, and S'_i denote the corresponding depth at price A'_i. Then, with the same amount q the investor wants to purchase, he or she has to pay

$$P'(q)_{ask} = \frac{1}{q}\left\{\sum_{j=1}^{k'-1} S'_j A'_j + \left(q - \sum_{j=1}^{k'-1} S'_j\right) A'_k\right\} \tag{4.2}$$

where k' denotes the index of the last buy limit order in the presence of hidden orders. Due to the possibility of hidden orders within the spread, we have $P'(q)_{ask} \leq P(q)_{ask}$. We can have similar definitions for $P'(q)_{bid}$ and $P(q)_{bid}$. Hence, we see a price improvement of the ask/bid side, defined as $PI(q)_{ask/bid} = |\log(P'(q)_{ask/bid}) - \log(P(q)_{ask/bid})|$, as a result of hidden orders in the limit order book. The price improvement summarizes the distribution of hidden order volumes in the order book, which can be measured for hidden liquidity. The possibility of ample hidden orders in the order book would result in a big price improvement, whereas few or no hidden orders would make the price improvement negligible. On the other hand, the price improvement of a *small* order would serve a reasonable proxy for hidden order volumes within the spread; in particular, it measures the aggressiveness of hidden orders within the spread. For example, a big price improvement of a *small* order means that a high volume of hidden orders exists within the spread, and the location of those hidden

orders is close to the opposite side. The price improvement of a *large* order would serve as a proxy for hidden order volumes within and beyond the spread. Moreover, the price improvement can be utilized by investors for executions or designing arbitrage trading strategies. For example, a big price improvement due to the presence of hidden orders would give investors incentive to execute a large order.

To determine a *small* order versus a *large* order, we look at average trading sizes in the investigation period (our investigation period is October 15, 2010 - November 30, 2010). We define a *small* trade with the number of shares q equal to 10 % of the average trading sizes, which is favored by retail investors, whereas a *large* trade with q equal to 150 % of the average trading sizes, favored by institutional investors.

Take AMZN (ticker for Amazon.com, Inc.) for example: we look at the histogram of the price improvement for a small trade with $q = 263$ and a large trade with $q = 3952$ respectively in Figure 4.1, which exhibits a mixture of mass zero and the gamma distribution in both cases.

Therefore, we propose modeling the price improvement using a zero-inflated gamma model, relating market condition variables as constructed in the following section, for a *small* trade. However, the hidden order data across the whole LOB is noisier than those within the spread, making a model of the price improvement of a *large* trade more difficult. A generalization of zero-inflated gamma model is proposed in a later section.

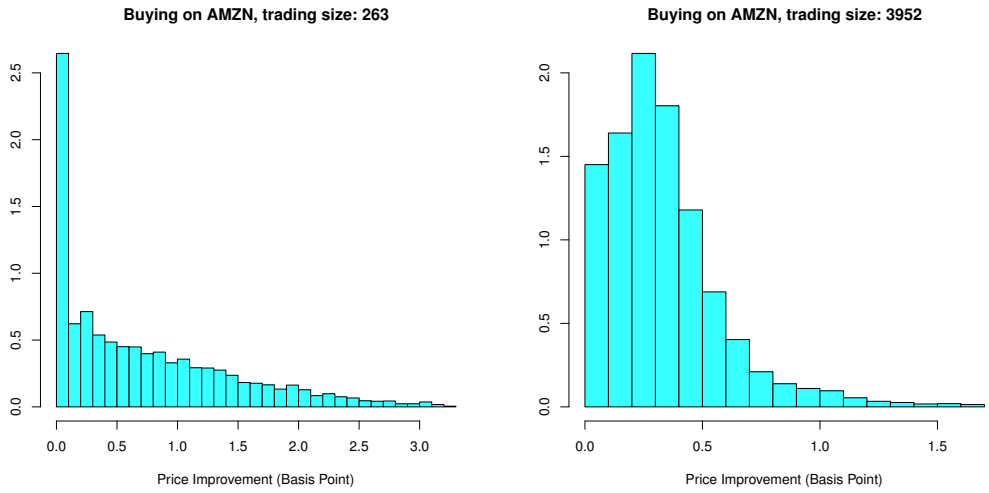

Figure 4.1: **Left:** Price improvement on a small buy trade with size 263 shares (against the ask side on LOB). **Right:** Price improvement on a large buy trade with size 3952 shares (against the ask side on LOB). Trading on AMZN, NASDAQ, October-November 2010.

4.3.3 Capturing visible market conditions

We retrieve the historical market conditions from TotalView-ITCH data. The NASDAQ TotalView^{SM}, surpassing NASDAQ Level 2, is nowadays the standard NASDAQ data feed for displaying the real time full order book depth for market participants. The historical data files record rich information on order activities, including limit order submissions, cancellations, executions and hidden order executions, as well as a unique identification number for every limit order and peak of reserve orders. First, we reconstruct the historical LOB by using the algorithm proposed by Huang and Polak [2011]. Their algorithm continuously updates the LOB according to the incoming message and represents the exact state of LOB that is historically shown to the TotalView subscribers in real time. Second, we identify the state of limit orders (cancelled or filled) and compute their life time by tracking them through their

order IDs.[3] Finally, we aggregate the sequence of executions on buy (sell) limit or hidden orders, which occur in half second to next, into a sell (buy) market order. If a limit order is recorded immediately after the sequence in half second, it would also be aggregated into the sequence considered a marketable limit order. Finally, to avoid erratic effects during the market opening and closing, our sample period covers only the periods between 9:45 and 15:45, i.e. 15 minutes after the opening and before the closing.

To relate the usage of undisclosed orders to *prevailing* market conditions, we construct different variables representing various states of the market. Here we outline the exact definitions of constructed variables used for hidden order submission on the buy side:

- $SPR \equiv$ log(best ask/best bid)
- $DPA \equiv$ log(depth at best ask)
- $DPB \equiv$ log(depth at best bid)
- $DPI \equiv DPB - DPA$
- $RET \equiv$ log return over the prevailing 1 minute
- $VOL \equiv$ market price range (maximum − minimum) over the prevailing 1 minute
- $HVA \equiv$ log(1 + volume of executed hidden ask depth during the prevailing 1 minute)

[3]The limit order book reconstruction and limit order tracking is performed by the software "LOBSTER" which can be freely accessed at `http://lobster.wiwi.hu-berlin.de`.

- $HVB \equiv \log(1 +$ volume of executed hidden bid depth during the prevailing 1 minute)
- $HVA_5 \equiv \log(1 +$ volume of executed hidden ask depth during the prevailing 5 minutes)
- $HVB_5 \equiv \log(1 +$ volume of executed hidden bid depth during the prevailing 5 minutes)
- $HRA \equiv HVA - HVA_5$
- $HRB \equiv HVB - HVB_5$
- $ALA \equiv \log(1 +$ number of aggressive sell limit orders that are not canceled during the prevailing 1 minute)
- $ALB \equiv \log(1 +$ number of aggressive buy limit orders that are not canceled during the prevailing 1 minute)
- $HFA \equiv \log(1 +$ number of fleeting sell limit orders during the prevailing 1 minute)
- $HFB \equiv \log(1 +$ number of fleeting buy limit orders during the prevailing 1 minute)

The prevailing LOB state is represented by the visible bid-ask spread (SPR), the visible depth on the best level on the bid side (DPB) and the visible depth on the best level on the ask side (DPA). Here we use visible depth imbalance (DPI) to summarize the relative level of the visible depth on the two sides. To capture the

impact of prevailing trade signals, we include the prevailing one-minute mid-quote return (RET) capturing short-term price movements and price volatility (VOL) in term of the (max/min) range of trade prices during the last one minute. Information on hidden depth is incorporated by the short-run executed hidden depth on the ask side and the bid side (HVA, HVB), representing how successfully traders have detected pending hidden depth. Moreover, to assess the relative intensity of temporary hidden order executions, we compute the executed hidden depth during the last minute relative to that executed during the last five minutes (HRA, HRB). Also, HFT activities are captured by two variables, HFA and HFB, which are the number of fleeting orders on the ask side and the bid side respectively. A "fleeting order" is a limit order that is canceled within one second after the submission and thus is posted to "test" for the existence of hidden volume. To differentiate between fleeting orders and "normal" limit orders, we also include the number of aggressive limit orders that have not been canceled (ALA, ALB) and thus represent the frequency of quote updating by low frequency traders.

4.4 Statistical Models for Hidden Liquidity

4.4.1 A Zero-inflated model for price improvement in a small trade

Let Y_i be the price improvement at time i and X_i represent market conditions up to time i. A zero-inflated gamma model is proposed as follows:

$$Y_i|X_i \sim \begin{cases} 0 & \text{with probability } 1-p \\ f(y_i|\theta_i) & \text{with probability } p \end{cases}$$

where $f(y_i|\theta_i)$ is the probability density function that belongs to some one-parameter exponential family distribution with θ_i as the canonical parameter to be linked to the covariate X_i (see below). The exponential family density we take is of Gamma form:

$$f(y_i|\theta_i) \propto y_i^{\theta_i - 1} e^{-y_i}$$

For simplicity, we assume that p is constant first. Later on, we will model p as a function of market conditions when order size becomes large. We link $\mu_i = E(Y_i)$, which is the expectation of Y_i evaluated under f, to X_i via a monotonic link function g_1, i.e., $g_1(\mu_i) = \alpha_0 + \beta^T X_i$. α_0, β and p can be estimated via a standard maximizing likelihood procedure where the log-likelihood is $l(\alpha_0, \beta, p) = \sum_{i=1}^{n} [I(y_i = 0)\log(1-p) + I(y_i \neq 0)(\log p - \log f(y_i|\theta_i))]$.

Model diagnostics may be inferred by examining the Pearson residuals, which are obtaining by rescaling the raw residuals $\hat{\epsilon}_i = y_i - \hat{\mu}_i$ by their estimated standard

deviation:

$$\hat{\epsilon}_i^p = \frac{y_i - \hat{\mu}_i}{\sqrt{V(\hat{\mu}_i)}}$$

If the model fits the data well, then we expect the Q-Q normal plot of the residuals to be close to a line with slope 1. We apply this zero-inflated gamma model to the price improvements on the ask and bid sides for three tickers, namely AMZN, BIDU, GOOG. The estimates are summarized in Table 1, and the Q-Q normal plots in Figure B.1 show that this model fits the data well.

Table 4.1: Summary Statistics for the zero-inflated Gamma model for a small trading: AMZN, BIDU, GOOG. Significant estimates (5% level) for β are highlighted in boldface.

	AMZN		BIDU		GOOG	
	Ask	Bid	Ask	Bid	Ask	Bid
SPR	**0.26**	**0.26**	**0.21**	**0.21**	**0.26**	**0.26**
DPI	**-0.06**	**0.13**	**-0.10**	**0.12**	**-0.05**	**0.14**
RET	**0.15**	**-0.15**	**0.13**	**-0.14**	**0.07**	**-0.07**
VOL	**-0.54**	**-0.47**	**-0.18**	0.01	0.02	0.01
HVA	**0.25**		**0.42**		**0.35**	
HVB		**0.29**		**0.37**		**0.51**
HRA	**-0.23**		**-0.30**		**-0.21**	
HRB		**-0.19**		**-0.18**		**-0.43**
ALA	**-0.04**	**0.06**	**-0.03**	**-0.10**	**-0.05**	**-0.03**
ALB	**0.08**	**-0.09**	**-0.09**	**-0.03**	**0.02**	**-0.02**
HFA	0.02	-0.02	-0.05	-0.03	-0.01	-0.01
HFB	0.01	-0.02	0.01	**-0.06**	-0.01	-0.01

4.4.2 Empirical evidence for a small trade

Price Improvement and Spread Sizes

We find that the size of visible bid-ask spread (SPR) has a significant positive impact on price improvement for a small trading, both on ask and bid sides. It implies that a widening of the spread leads to a stronger accumulation of hidden volume inside the spread. This confirms with Cebiroglu and Horst [2011] who argue that the spread size is the dominant factor for hidden volumes inside it.

Price Improvement and Visible Liquidity Provision

Our finding is that when the visible depth at the best bid is relatively higher than the one at the best ask (DPI), i.e. a positive DPI, the price improvement on the ask side decreases while the price improvement on the bid side increases. It is consistent with the theoretical prediction by buti-rindi-2011 that hidden order traders submit their order more aggressively to compete for the provision of liquidity. Consequently, we expect a bigger price improvement on the side with higher visible depth.

Price Improvement after Price Movements and Trading Signals

We observe that the trading price movement is positively correlated to the price improvement. It implies that the hidden liquidity on the ask (bid) side is distributed closer to the spread when trading prices are moving up (down). When prices are moving up (down), the execution probability of ask (bid) hidden orders increases due to the momentum effect. This finding is in line with Buti and Rindi [2011] who argue that hidden liquidity providers submit more hidden volumes when the execution probability is high. Note that this finding is *not* necessary conflict with the argument by Hautsch and Huang [2012b] that the hidden orders become less aggressive when

the trading price moving in their favor direction. Instead, we believe that it is more likely that the seller increases the aggressiveness of hidden order but reduce the hidden order size, see e.g., the pronounced "U" shape distribution of hidden liquidity found by Cebiroglu and Horst [2011].

However, the prevailing return volatility VOL does not always have a significant impact on price improvement, which is also confirmed by Hautsch and Huang [2012b].

Price Improvement and Hidden Liquidity Provision

Our estimates (HVA or HVB) show clear evidence for the competition between hidden liquidity providers, as the price improvement increases on its own side when the execution of hidden volumes on its own side increases. This means that liquidity suppliers are encouraged to provide further hidden volume if they realize liquidity demand from the opposite side and competition on their own side.

However, according to Buti and Rindi [2011], these effects persist as long as adverse selection risk does not become too high. Indeed, we find the negative effects when we control the relative intensity of executed hidden volumes (HRA or HRB). Price improvement tends to decline if hidden depth demand become extraordinarily high, as price pressure from the opposite side becomes too strong and makes adverse selection risk too high.

Price Improvement and HFT

Although the frequency of quote updating by low frequency traders has impact on price improvement, HFT activities (approximated by the intensity of fleeting orders) do not have significant empirical evidence, which contradicts the intuition of many people. We believe that the statistical insignificance for HFT activities is likely due

to the possible failure for"pinging" a huge amount of small market orders to buy and sell initiated by algorithmic traders during the investigation period.

4.4.3 Nonparametric zero-inflated model for a large trade

While a simple zero-inflated gamma model forecasts the price improvement for q small trade quite well, it fits the price improvement for a large trade poorly, as it is shown by the Q-Q plot in Figure B.2, which suggests a further improvement for this zero-inflated gamma model. Now we generalize our original zero-inflated model to a certain extent such that our model is flexible and can capture the dynamics of hidden liquidity across the whole limit order book under market conditions for which we are concerned. The generalization comes in two parts: the first part is that we allow the non-zero inflation probability, denoted by p_i, to be time varying and be linked to the covariates via a link function g_2 (e.g. logit or probit functions):

$$g_2(p_i) = \beta_0 + \sum_{i=1}^{k} h_i(X_i) \tag{4.3}$$

and the second part is that we allow greater flexibility in the mean response μ such that

$$g_1(\mu) = \gamma_0 + \sum_{i=1}^{k} s_i(X_i) \tag{4.4}$$

where $h_i, s_i, i = 1, \ldots, k$ are two sets of nonparametric smooth functions. They can be estimated non-parametrically through cubic regression splines, which can be further extended to high-dimensional smoothing that can accommodate the interaction between some covariates.

Equations (4.3) and (4.4) formulate an unconstrained nonparametric zero-inflated model, which assumes that the market conditions effect on the probability of having a non-zero price improvement may have different data generating mechanisms from the magnitude of the non-zero price improvement. However, an interesting question arises that some market conditions might influence the two processes simultaneously. In this case, the constrained zero-inflated models (Liu and Chan [2011]) can be used to test the above hypothesis, which assumes a monotonic relationship between g_1 and g_2. In particular, we consider the case that g_2 is a linear function of g_1:

$$g_2 = \alpha + \delta g_1 \tag{4.5}$$

where α and δ are two unknown coefficients.

The reason that we consider the constrained zero-inflated model is not only that this parsimonious model would promote estimation efficiency, but also it connects to some latent threshold model (Liu et al. [2012]) which has an economic interpretation. Assume that Y' is a latent response variable following the $Gamma(\alpha, \beta)$ distribution. The observed response Y is zero if the latent mean response μ is less than a random threshold T, and is equal to Y' if μ exceeds the threshold. This threshold T is determined by market conditions, in which we can view T as a function of market conditions X. Now the non-zero-inflated probability $p = P(Y = Y') = P(T \leq \mu) = F_T(\mu)$, where F_T is the cumulative distribution function of T. This implies that the link function g, which is defined as $g(p) = \mu$, is the inverse of F_T, which is unknown. Nevertheless, Li and Duan [1989] showed that, even under a misspecified link function, any maximum likelihood estimator is consistent up to a multiplicative

scalar, i.e., $\hat{h_i} = \delta s_i, i = 1, \ldots, k$, for some scalar δ.

Model Estimation and Inference

The proposed nonparametric zero-inflated model can be estimated by the penalized likelihood approach, which is to maximize the following penalized log-likelihood:

$$l(\alpha, \delta, \beta) - \sum_{i=1}^{k} \lambda_i^2 J(h_i) - \sum_{i=1}^{k} \varphi_i^2 J(s_i) \tag{4.6}$$

where $l = \sum_{i=1}^{n} [I(y_i = 0) \log(1 - p_i) + I(y_i \neq 0)(\log p_i - \log f(y_i|x_i)]$ is the log-likelihood function for all the observations, $J(h)$ defines a roughness penalty functional of h, and $\lambda_i, \varphi_i, i = 1, \ldots, k$, are the smoothing parameters corresponding to each penalty term, which control the trade-off between the smoothness of the function estimates and goodness-of-fit of the model. In our case, we adopt the roughness penalty $J(h) = \int \{h^{(2)}(x)\}^2 dx$, where $h^{(2)}(x)$ denotes the second derivative of a univariate function $h(x)$. The spline estimate can be represented as a linear combination of some basis functions: $h(x) = \theta_0 + \theta_1 x + \sum_{j=1}^{K-1} \theta_{j+1}(x - x_j^*)_+^3$, where x_j^* $(j = 1, \ldots, K - 1)$ are fixed knots placed evenly over the corresponding observed covariate values. Denote $\boldsymbol{\theta} = (\theta_0, \ldots, \theta_K)$, the roughness penalty can be written as a quadratic form of $\boldsymbol{\theta}$ such that $J(h) = \boldsymbol{\theta}' \boldsymbol{S} \boldsymbol{\theta}$, where $\mathbf{S}$ is the penalty matrix. The smoothing parameters can be chosen by generalized cross-validation or similar procedures.

Model Selection

The model selection procedure (see Appendix B) suggests that the constrained zero-inflated model has higher marginal likelihood (see the calculation in (B.1)) than the unconstrained zero-inflated model (see the calculation in (B.2)) for all three tick-

ers. Then we perform the variable selection procedure, that is picking up the **most parsimonious** constrained zero-inflated nonparametric model, which has the *highest* marginal likelihood among the possible candidates. The most parsimonious constrained zero-inflated nonparametric model for the price improvement for a large trade is a mixture distribution that equals zero with probability $1 - p_i$ but otherwise is log-normal with mean μ given by

$$\mu = c + s(RET) + s(SPR) + s(HVA) + s(HRA) + s(ALA, ALB) + s(DPA, DPB)$$

for a buy order, and

$$\mu = c + s(RET) + s(SPR) + s(HVA) + s(HRA) + s(ALA, ALB) + s(DPA, DPB)$$

for a sell order. And

$$\text{logit}(p_i) = \alpha + \delta\mu,$$

where c, α, δ are parameters, and s are assumed to be distinct smooth functions. The normal QQ plots of residuals in Figure B.3 suggest that this constrained nonparametric zero-inflated model fits the data well.

4.4.4 Empirical finding for a large trade

This nonparametric zero-inflated model shows some persistent patterns across the selected stocks, see Figure B.4- B.9. For reasons of clarity, we shall discuss these patterns in detail in the case of buying 3206 shares GOOG, which is fifteen times the

average trading sizes in the investigation period.

Our findings on the effects of market condition on the price improvement of a large trade are largely consistent with those for a small trade; see Section 4.2. However, it demonstrates some more interesting patterns, in particular, the *interactive* effect of some market variables (ALA and ALB, DPA and DPB) on the ask side and the bid side. We find that the best price improvement happens when the number of aggressive buy limit orders is big but not too big. This finding is consistent with Buti and Rindi [2011] who argue that traders implement more aggressive hidden order strategies when the opposite traders (in this case, they submit the aggressive buy limit orders) are more active. On the other hand, when there are many aggressive buy limit orders, the ask dark liquidity has likely been exhausted. Consequently, the price improvement declines. On the other hand, we see that the isotropic curve for the number of the aggressive sell limit orders, given a certain amount of the aggressive buy limit orders, is quite parallel to the vertical axis in Figure 7, which implies that though the aggressive sell limit orders have a negative effect on the price improvement, that effect is rather insignificant.

The model shows that the price improvement for a large trade depends on displayed liquidity in a complex way. The highest price improvement is in the region around the point corresponding to the average of displayed depth on both sides. Surprisingly, when displayed ask depth is too big, we expect a smaller pricing improvement in buying a huge amount of shares. The underlying reason is that, although large displayed ask depth may imply more aggressive sell hidden orders, the hidden order size might be relatively small compared to (1) the big displayed ask depth and

(2) the huge size of a big trading. Note that sizable displayed depth also indicates a better expected price against visible liquidity. Overall, the price improvement by small aggressive hidden orders becomes relatively insignificant.

4.5 Conclusion

Trading under limited pre-trade transparency has become increasingly popular in financial markets, due to the emergence of iceberg (partially hidden) or hidden (completely hidden) orders. A growing body of empirical evidence studies the iceberg orders as well as some theoretical work on how hidden orders can be used to control exposure cost. This paper sheds light on the use of hidden orders to improve trading prices, and first provides empirical evidence on how different market conditions can affect price improvement due to hidden liquidity.

We retrieve information on market conditions from NASDAQ TotalView message data, and construct price improvement due to hidden orders from NASDAQ ModelView data. We propose two novel statistical models for predicting price improvement based on visible market conditions, from a small retail investor's and a large institutional investor's perspective. Our finding shows that price improvement is significantly correlated with market conditions and thus is predictable in terms of the state of prevailing visible LOB and order flow. Our empirical evidence is consistent with some of theoretical predictions, and show the following pronounced effects: First, price improvement is positively correlated with observable spreads, and follows recent price movement and trading signals. Second, price improvement becomes more pronounced when traders competes with displayed liquidity and hidden liquidity on

the own side of the market. Third, price improvement decreases when the adverse selection risk is too high. Fourth, the interaction of aggressive orders on both sides, as well as displayed liquidity on both sides, has complex and dynamic effects on price improvement for a large trade.

Our finding can be utilized to further develop theoretical models for hidden order submission strategies. The proposed statistical models can be extended in various directions to better understand hidden liquidity in order-driven electronic markets.

Appendix A

Supplementary Material for Chapter 2

A.1 The derivation of (2.2), (2.3) and (2.4)

Since $y_i(\mathbf{t})|x_i(\mathbf{t}), \sigma^2 \sim N(x_i(\mathbf{t}), \sigma^2 I_n)$, it follows that $y_i(\mathbf{t})|\alpha_i, \sigma^2$ is also normally distributed with mean

$$E(y_i(\mathbf{t})|\alpha_i, \sigma^2) = E(E(y_i(\mathbf{t})|x_i(\mathbf{t}), \alpha_i, \sigma^2)|\alpha_i, \sigma^2) = E(x_i(\mathbf{t})|\alpha_i, \sigma^2) = \mu_{G,i}$$

and covariance

$$\begin{aligned} Cov(y_i(\mathbf{t})|\alpha_i, \sigma^2) &= E(Cov(y_i(\mathbf{t})|x_i(\mathbf{t}), \alpha_i, \sigma^2)|\alpha_i, \sigma^2) + Cov(E(y_i(\mathbf{t})|x_i(\mathbf{t}), \alpha_i, \sigma^2)|\alpha_i, \sigma^2) \\ &= E(\sigma^2 I_n|\alpha_i, \sigma^2) + Cov(x_i(\mathbf{t})|\alpha_i, \sigma^2) \\ &= \sigma^2 I_n + C_{\alpha_i} \end{aligned}$$

Therefore, the joint distribution of $(y_i(\mathbf{t}), x_i(\mathbf{t}))$ given α_i, σ^2 is a bivariate normal distribution with mean $(\mu_{G,i}, \mu_{G,i})^T$ and covariance $\begin{pmatrix} \sigma^2 I_n + C_{\alpha_i} & C_{\alpha_i} \\ C_{\alpha_i} & C_{\alpha_i} \end{pmatrix}$

Hence, by the property of conditional distribution for multivariate normal distributions, we have

$$p(x_i(\mathbf{t})|y_i(\mathbf{t}), \sigma^2, \alpha_i) = N(\mu_i, \Sigma_i)$$

where $\mu_i = \mu_{G,i} + C_{\alpha_i}(C_{\alpha_i} + \sigma^2 I_n)^{-1}(y_i(\mathbf{t}) - \mu_{G,i})$ and

$$\begin{aligned} \Sigma_i &= C_{\alpha_i} - C_{\alpha_i}(C_{\alpha_i} + \sigma^2 I_n)^{-1} C_{\alpha_i} \\ &= C_{\alpha_i}(C_{\alpha_i} + \sigma^2 I_n)^{-1}(C_{\alpha_i} + \sigma^2 I_n - C_{\alpha_i}) \\ &= \sigma^2 C_{\alpha_i}(C_{\alpha_i} + \sigma^2 I_n)^{-1} \end{aligned}$$

which is (2.2).

To obtain (2.3), we define C'_{α_i} =Cov$(\dot{x}_i(\mathbf{t}), x_i(\mathbf{t}))$, C''_{α_i} =Cov$(\dot{x}_i(\mathbf{t}), \dot{x}_i(\mathbf{t}))$ and note that $\dot{x}_i(\mathbf{t}))|\alpha_i \sim N(\dot{\mu}_{G,i}, C''_{\alpha_i})$.

Then the joint distribution of $(y_i(\mathbf{t}), \dot{x}_i(\mathbf{t}))$ given α_i, σ^2 is a bivariate normal distribution with mean $(\mu_{G,i}, \dot{\mu}_{G,i})$ and covariance $\begin{pmatrix} \sigma^2 I_n + C_{\alpha_i} & C'_{\alpha_i} \\ C'_{\alpha_i} & C''_{\alpha_i} \end{pmatrix}$

Hence, by the property of conditional distribution for multivariate normal distributions, we have

$$p(\dot{x}_i(\mathbf{t})|y_i(\mathbf{t}), \alpha_i, \sigma^2) = N(a_i, b_i)$$

where $a_i = \dot{\mu}_{G,i}) + C'_{\alpha_i}(C_{\alpha_i} + \sigma^2 I_n)^{-1}(x_i(\mathbf{t}) - \mu_{G,i})$ and $b_n = C''_{\alpha_i} - C'_{\alpha_i}(C_{\alpha_i} + \sigma^2 I_n)C'_{\alpha_i}$.

Similarly, the joint distribution of $(y_i(\mathbf{t}), x_i(\mathbf{t}), \dot{x}_i(\mathbf{t}))$ given α_i, σ^2 is a trivariate nor-

mal distribution with mean $(\mu_{G,i}, \mu_{G,i}, \dot{\mu}_{G,i})$ and covariance matrix $\begin{pmatrix} \sigma^2 I_n + C_{\alpha_i} & C_{\alpha_i} & C'_{\alpha_i} \\ C_{\alpha_i} & C_{\alpha_i} & C'_{\alpha_i} \\ C'_{\alpha_i} & C'_{\alpha_i} & C''_{\alpha_i} \end{pmatrix}$ .By the property of conditional distribution for multivariate normal distributions, we obtain (2.6).

A.2 The form of C'_{α_i} and C''_{α_i}

For the notation ease, let l denote α_i. Since $\mathrm{Cov}(\dot{x}_i(\mathbf{t}), x_i(\mathbf{t})) = \frac{d}{d\mathbf{t}}\mathrm{Cov}(x_i(\mathbf{t}), x_i(\mathbf{t}))$ and $\mathrm{Cov}(\dot{x}_i(t_1), \dot{x}_i(t_2)) = \frac{\partial^2}{\partial t_1 \partial t_2}\mathrm{Cov}(x_i(\mathbf{t}), x_i(\mathbf{t}))$, if C_l is a squared covariance function, then the (i,j)th element of C'_l and C''_l is

$$\begin{aligned} C'_l(i,j) &= -\sigma_C^2 r e^{-\frac{r^2}{2l}} \\ C''_l(i,j) &= \sigma_C^2(1-r^2)e^{-\frac{r^2}{2l}} \end{aligned}$$

where $r = |t_i - t_j|$.

If C_l is a Matern class with $v = 5/2$, then the (i,j)th element of C'_l and C''_l is

$$\begin{aligned} C'_l(i,j) &= -\sigma_C^2 e^{-\frac{\sqrt{5}r}{l}}(\frac{5r}{3l^2} + \sigma_C^2\frac{5\sqrt{5}r^2}{3l^3}) \\ C''_l(i,j) &= -\sigma_C^2\frac{\sqrt{5}}{l}e^{-\frac{\sqrt{5}r}{l}}(\frac{5r}{3l^2} + \sigma_C^2\frac{5\sqrt{5}r^2}{3l^3}) + \sigma_C^2 e^{-\frac{\sqrt{5}r}{l}}(\frac{5}{3l^2} + \sigma_C^2\frac{10\sqrt{5}r}{3l^3}) \end{aligned}$$

A.3 K-L divergence of (2.6) and (2.7)

Given two d-dimensional Gaussian random variables $f \sim N(\mu_f, \Sigma_f)$ and $g \sim N(\mu_g, \Sigma_g)$, the K-L divergence of f and g has a closed-formed expression:

$$KL(f||g) = \frac{1}{2}\Big(\ln \frac{|\Sigma_g|}{|\Sigma_f|} + Tr[\Sigma_g^{-1}\Sigma_f] - d + (\mu_f - \mu_g)^T \Sigma_g^{-1}(\mu_f - \mu_g)\Big) \tag{A.1}$$

To calculate K-L divergence of (2.6) and (2.7), we simply let $\mu_f = \tilde{\mu}_i, \Sigma_f = \tilde{\Sigma}_i + \frac{1}{2}\gamma I_n$ and $\mu_g = f_i, \Sigma_g = \frac{1}{2}\gamma I_n$, and substitute into (A.1).

Appendix B

Supplementary Material for Chapter 4

B.1 Model Selection on Zero-inflated Models

Although the constrained zero-inflated model results in a parsimonious form, and has some economic and financial interpretation, the theoretical ground is yet to be clear. Therefore, a model selection procedure among multiple competing models should be performed. In statistical analysis, one of the widely used model selection criteria is the Bayesian information criterion (BIC, Schwarz [1978]), which selects the model with maximum posterior probability. Under the Bayesian framework, the posterior probability of model M_i is:

$$P(M_i|D) = \frac{P(D|M_i)P(M_i)}{P(D)}$$

where $P(M_i)$ is the prior probability of model M_i, D denotes the data, and $P(D)$ is the normalizing constant. $P(D|M_i)$ is the marginal likelihood of the model M_i, and it can be written as

$$P(D|M_i) = \int P(D|\theta, M_i)P(\theta|M_i)d\theta$$

where $P(D|\theta, M_i)$ is the likelihood of the parameter θ under the model M_i, and $P(\theta|M_i)$ is the prior probability of θ under M_i. Assuming we put a flat prior on $P(M_i)$, the posterior probability $P(M_i|D)$ is proportional to the marginal likelihood $P(D|M_i)$. Reminiscent to the BIC, we use the marginal likelihood as a model selection criterion for constrained and non-constrained zero-inflated models, where the model with larger marginal likelihoods would be preferred. However, there is no closed-form solution for constrained and non-constrained zero-inflated models. The Laplace method is used to approximate the marginal likelihoods.

Liu and Chan [2011] gave the following approximate formula of the logarithmic marginal likelihood for the constrained zero-inflated model:

$$logE_c \approx l(\hat{\theta}) - \frac{K+2}{2}\log n - \frac{1}{2}\log|H| + \frac{K+2-B}{2}\log 2\pi + \frac{1}{2}\sum_{j=1}^{m} log|\lambda_j^2 \boldsymbol{S}_{j+}| \quad \text{(B.1)}$$

where $\hat{\theta}$ is the maximum penalized likelihood estimator, $K = \dim(\beta)$, $\boldsymbol{S}_{j+}$ is a diagonal matrix of dimension b_j with all the strictly positive eigenvalues of the penalty matrix $\boldsymbol{S}_j$ arranged in descending order on the leading diagonal, $B = \sum_{j=1}^{m} b_j$, and H is the negative Hessian matrix of l/n evaluated at $\hat{\theta}$.

For the unconstrained zero-inflated model, Liu and Chan [2011] provided the

following approximation, $logE \approx l(\hat{\beta}, \hat{\gamma}) - l$:

$$l = \frac{K + \tilde{K}}{2} \log n - \frac{1}{2} \log |H| + \frac{K + \tilde{K} - B - \tilde{B}}{2} \log 2\pi + \frac{1}{2} \sum_{j=1}^{m} log|\lambda_j^2 \boldsymbol{S}_{j+}| + \frac{1}{2} \sum_{j=1}^{m'} log|\varphi_j^2 \boldsymbol{S'}_{j+}| \tag{B.2}$$

where $\tilde{K} = \dim(\gamma), \boldsymbol{S'}_{j+}$ is a diagonal matrix of dimension b'_j with all the strictly positive eigenvalues of the penalty matrix $\boldsymbol{S'}_j$ arranged in descending order on the leading diagonal and $B' = \sum_{j=1}^{m'} b'_j$.

B.2 Graphical illustration of goodness-of-fit for constrained zero-Inflated nonparametric models on three tickers

Figure B.1, B.2 and B.3 illustrate the QQ-plot for the diagnosis of our proposed statistical model on three tickers to explain the price improvement of hidden liquidity under different market condition.

Figure B.4 -B.9 illustrate the goodness-of-fit for constrained zero-Inflated nonparametric models on three tickers. The notations in those figures: r is RET, bidAskSpread is SPR, hiddenExeAsk(Bid)_1 is $HVA(HVB)$, hiddenExeAsk(Bid)_r is $HVA_5(HVB_5)$, aggBuy(aggSell) is $ALB(ALA)$ and dispAsk(dispBid) is $DPA(DPB)$.

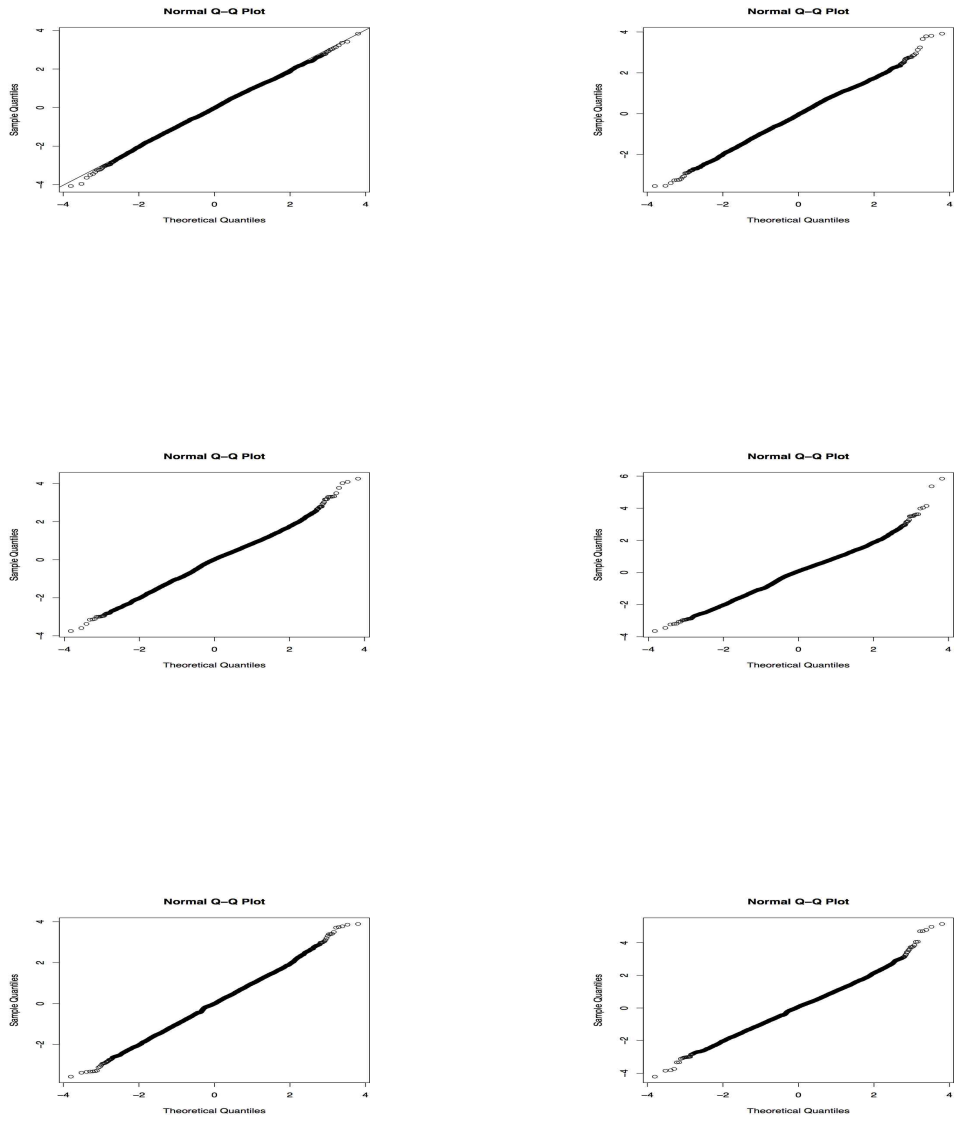

Figure B.1: Normal QQ plots for a small trading against the ask side (left) and the bid side (right) using zero-inflated Gamma models. **First row:** AMZN with $q = 263$. **Second row:** GOOG with $q = 213$. **Third row:** BIDU with $q = 316$.

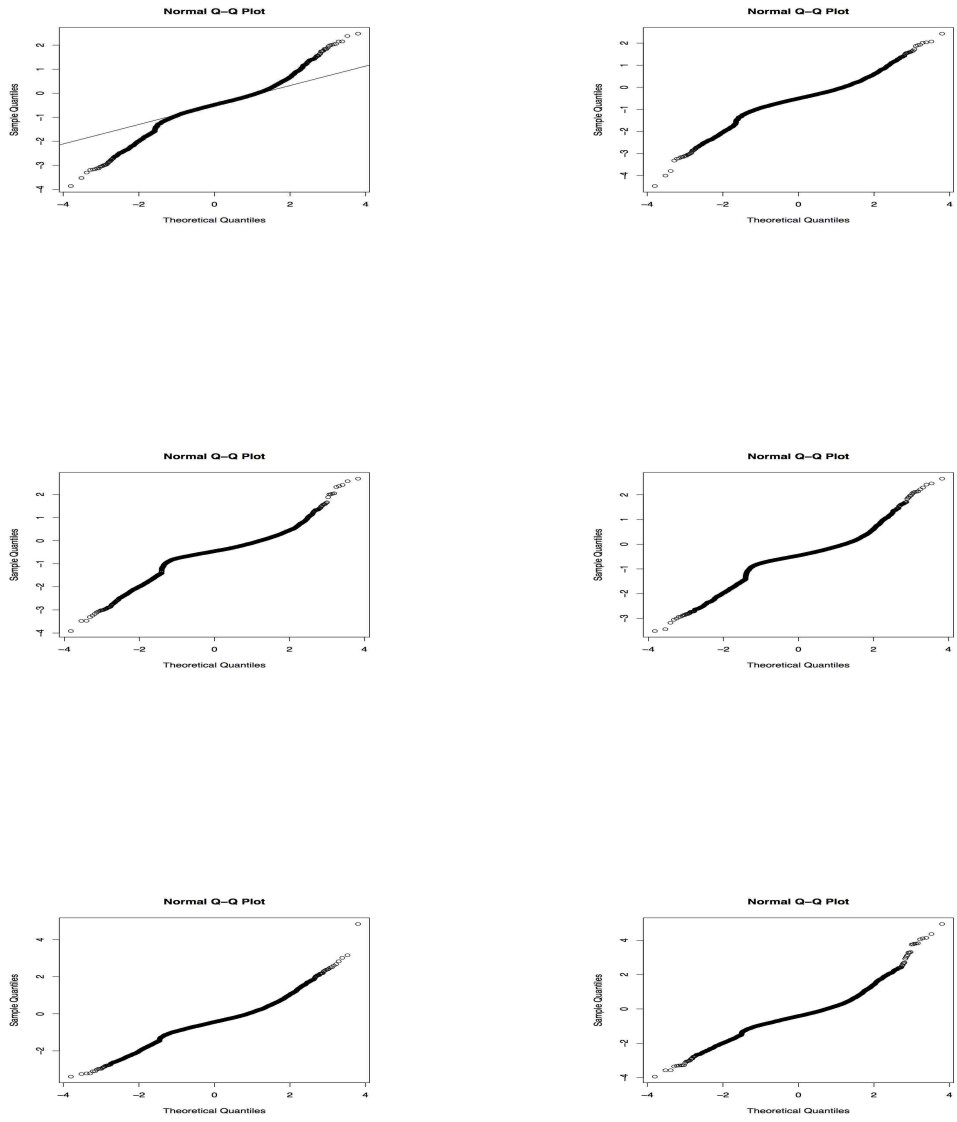

Figure B.2: Normal QQ plots for a large trading against the ask side (left) and the bid side (right) using zero-inflated Gamma models. **First row:** AMZN with $q = 3952$. **Second row:** GOOG with $q = 3206$. **Third row:** BIDU with $q = 4743$.

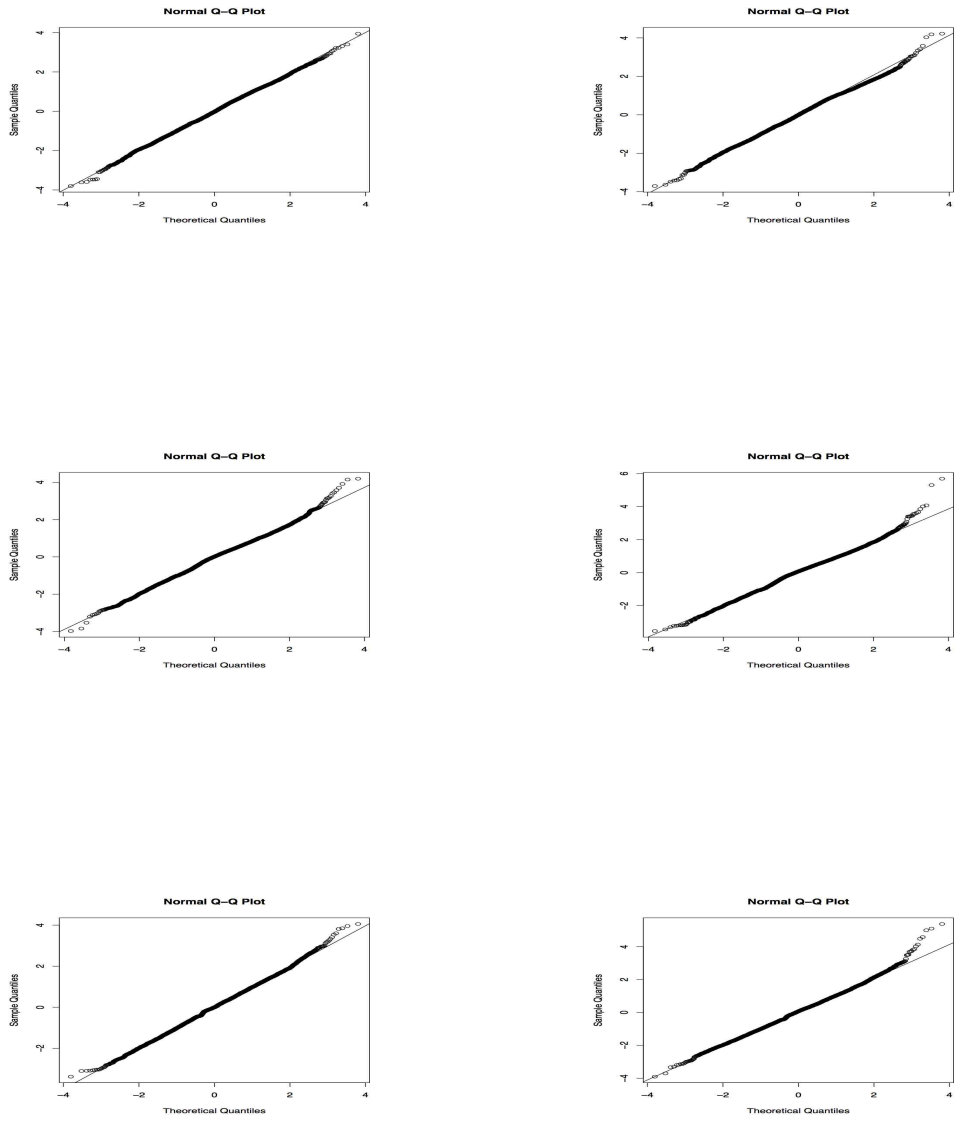

Figure B.3: Normal QQ plots for a large trading against the ask side (left) and the bid side (right) using constrained nonparametric models. **First row:** AMZN with $q = 3952$. **Second row:** GOOG with $q = 3206$. **Third row:** BIDU with $q = 4743$.

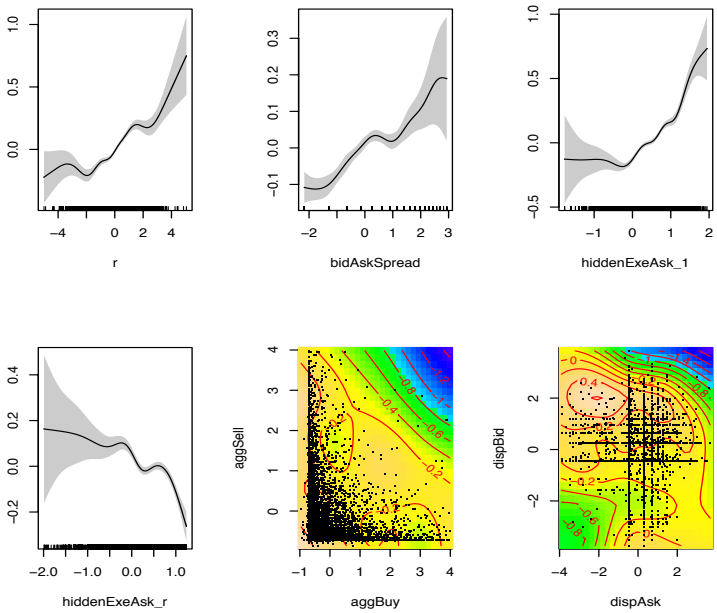

Figure B.4: The goodness-of-fit for constrained zero-Inflated nonparametric models for buying AMZN on NASDAQ with trade size 3952.

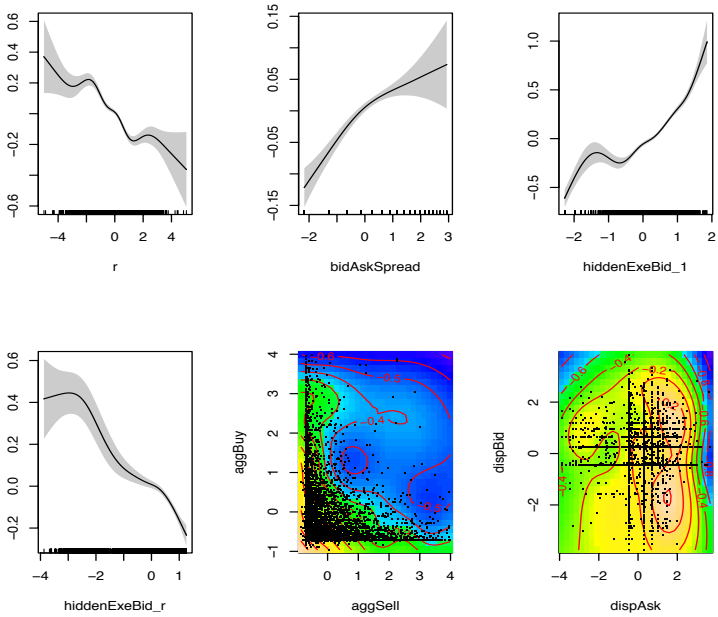

Figure B.5: The goodness-of-fit for constrained zero-Inflated nonparametric models for selling AMZN on NASDAQ with trade size 3952.

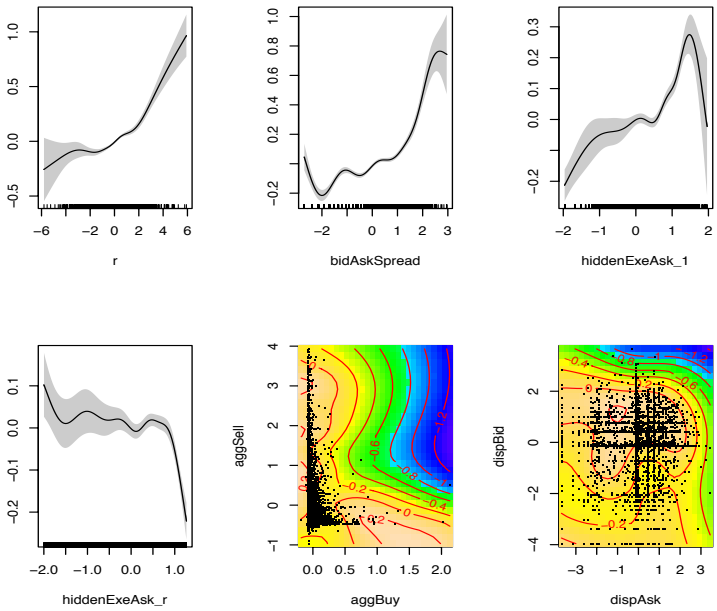

Figure B.6: The goodness-of-fit for constrained zero-Inflated nonparametric models for buying GOOG on NASDAQ with trade size 3206.

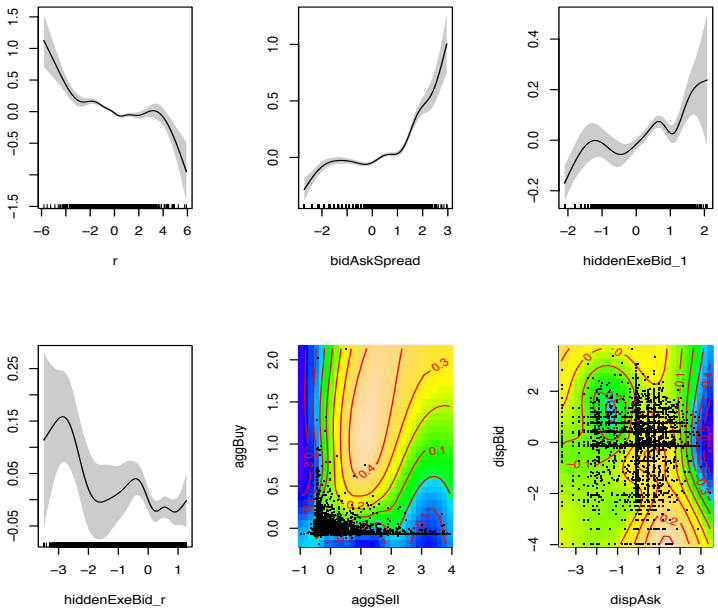

Figure B.7: The goodness-of-fit for constrained zero-Inflated nonparametric models for selling GOOG on NASDAQ with trade size 3206.

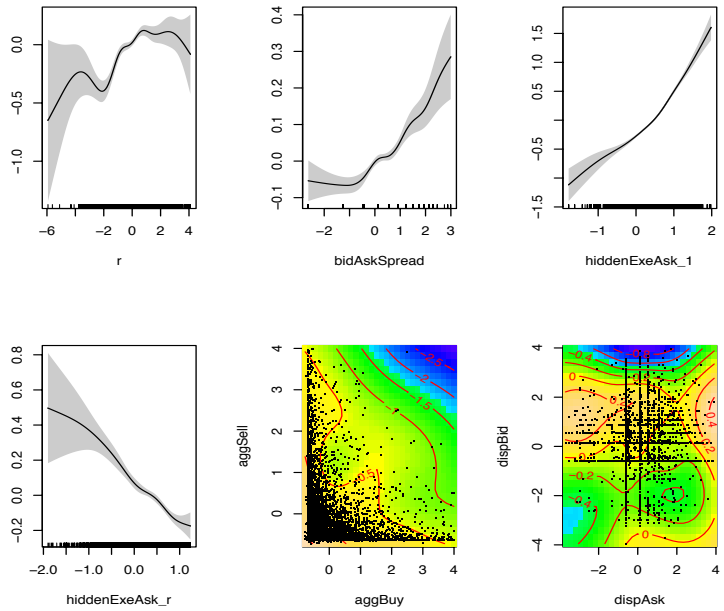

Figure B.8: The goodness-of-fit for constrained zero-Inflated nonparametric models for buying BIDU on NASDAQ with trade size 4743.

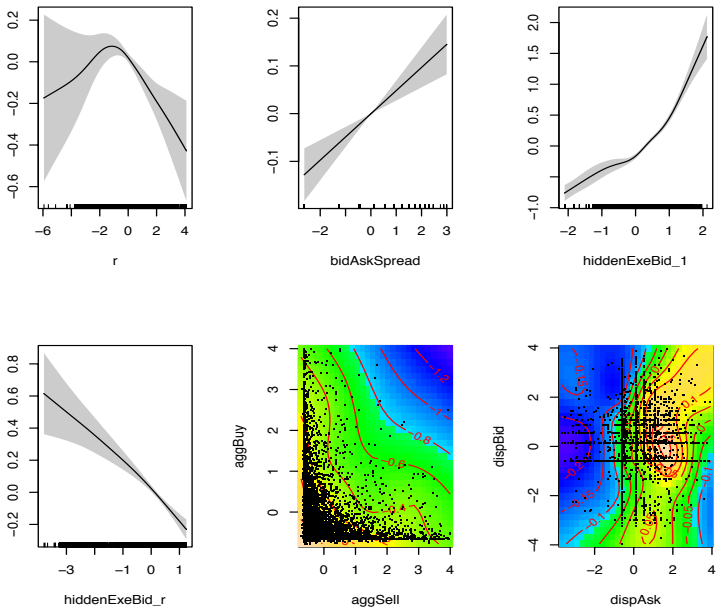

Figure B.9: The goodness-of-fit for constrained zero-Inflated nonparametric models for selling BIDU on NASDAQ with trade size 4743.

Bibliography

M. Aitken, H. Berkman, and D. Mak. The use of undisclosed limit orders on the Australian Stock Exchange. *Journal of Banking and Finance*, 25:1589 – 1603, 2001.

A. Anand and D. Weaver. Can order exposure be mandated? *Journal of Financial Markets*, 7:405 – 426, 2004.

H. Bessembinder, M. Panayides, and K. Venkataraman. Hidden liquidity: an analysis of order exposure strategies in electronic stock markets. *Journal of Financial Economics*, 94:361 – 383, 2009.

L. Biegler, J. Damiano, and G. Blau. Nonlinear parameter estimation: A case study comparison. *AIChE J.*, 32:29–45, 1986.

C. Bishop. *Pattern Recognition And Machine Learning*. Springer, 2006.

M. Bunner, M. Popp, Th. Meyer, A. Kittel, U. Ran, and J. Parisi. Recovery of scalar time-delay systems from time series. *Physics Letter. A*, 211:345–349, 1996.

S. Buti and B. Rindi. Undisclosed orders and optimal submission strategies in a dynamic limit order market. EFA 2008 Athens Meeting Paper, AFA 2009 San Francisco Meeting Paper, 2011.

G. Cebiroglu and U. Horst. Optimal display of iceberg orders. SFB 649 Discussion Paper 2011-057, Humboldt Universität zu Berlin, 2011.

J. Chen and H. Wu. Efficient local estimation for time-varying coefficients in deterministic dynamic models with applications to hiv-1 dynamics. *Journal of the American Statistical Association*, 103:369–384, 2008.

M. Coppejans, I. Domowitz., and A. Madhavan. Liquidity in an automated auction. *SSRN eLibrary*, 2000.

C. Cosentino and D. Bates. *Feedback Control in Systems Biology*. CRC Press, 2011.

R. De Winne and C. D'Hondt. Hide-and-seek in the market: placing and detecting hidden orders. *Review of Finance*, 11(4):663 – 692, 2007.

A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm (with discussion). *Journal of the Royal Statistical Society. Series B (Methodological)*, 39:1–38, 1977.

S. Ellner, B. Kendall, S. Wood, E. McCauley, and C. Briggs. Inferring mechanism from time-series data: Delay-differential equation. *Physica D*, 110:182–194, 1997.

N. Elowitz and S. Leibler. A synthetic oscillatory network of transcriptional regulators. *Nature*, 403:335–338, 2000.

A. Esser and B. Mönch. The navigation of an iceberg: The optimal use of hidden orders. *Finance Research Letters*, 4:68 – 81, 2007.

M. Fielding and D. NottShie-Yui Liong. Efficient mcmc schemes for computationally expensive posterior distributions. *Technometrics*, 53:1628, 2011.

R. FitzHugh. Impulses and physiological states in models of nerve membrane. *Biophys. J.*, 1:445–466, 1961.

M. Fleming and B. Mizrach. The microstructure of a U.S. treasury ecn: The BrokerTec platform. Staff Report 381, Federal Reserve Bank of New York, 2009.

A. Fowler and G. Kember. Delay recognition in chaotic time series. *Physics Letter. A*, 175:402–408, 1993.

S. Frey and P. Sandås. The impact of iceberg orders in limit order books. CFR Working Paper 09-06, University of Cologne, Centre for Financial Research (CFR), 2009.

A. Gelman, F. Bois, and J. Jiang. Physiological pharmacokinetic analysis using population modeling and informative prior distributions. *Journal of the American Statistical Association*, 91:1400–1412, 1996.

W. Gurney, S. Blythe, and R. Nisbet. Nicholson's blowflies revisited. *Nature*, 287: 17–21, 1980.

L. Harris. Does a large minimum price variation encourage order exposure. Working paper, Marshall School of Business, University of Southern California, 1996.

L. Harris. Order exposure and parasitic traders. Working paper, Marshall School of Business, University of Southern California, 1997.

N. Hautsch and R. Huang. The market impact of a limit order. *Journal of Economic Dynamics & Control*, 36:501 – 522, 2012a.

N. Hautsch and R. Huang. On the dark side of the market: Identifying and analyzing hidden order placement. Discussion Paper 2012-14, CRC 649, Humboldt Universität zu Berlin, Germany, Dec 2012b.

W. Horbelt, J. Timmer, and H. U. Voss. Parameter estimation in nonlinear delayed feedback systems from noisy data. *Physics Letter. A*, 299:513–521, 2002.

R. Huang and T. Polak. LOBSTER: The limit order book reconstructor. Technical report, School of Business and Economics, Humboldt Universität zu Berlin, 2011. `http://lobster.wiwi.hu-berlin.de/Lobster/LobsterReport.pdf`.

Y. Huang, D. Liu, and H. Wu. Hierarchical bayesian methods for estimation of parameters in a longitudinal hiv dynamic system. *Biometrics*, 62:413–423, 2006.

E. Ionides, C. Breto, and A. King. Inference for nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 103:18438–18443, 2006.

G. Irvine, P.and Benston and E. Kandel. Liquidity beyond the inside spread: Measuring and using information in the limit order book. *SSRN eLibrary*, 2000.

S. Kou, Q. Zhou, and W. Wong. Equi-energy sampler with applications in statistical inference and statistical mechanics (with discussion). *Annals of Statistics*, 34:1581–1652, 2006.

S. Kou, B.Olding, M. Lysy, and J. Liu. A multiresolution method for parameter estimation of diffusion processes. *Journal of American Statistical Association*, 107: 1558–1574, 2012.

B. Lewin. *Genes VII*. Oxford: Oxford University Press, 2000.

K.-C. Li and N. Duan. Regression analysis under link violation. *The Annals of Statistics*, 17:1009 – 1052, 1989.

H. Liu and K.S. Chan. Generalized additive models for zero-inflated data with partial constraints. *Scandinavian Journal of Statistics*, 38:650 – 665, 2011.

H. Liu, S. Ma, R. Kronmal, and K.S. Chan. Semiparametric zero-inflated modeling in multi-ethnic study of atherosclerosis (mesa). *Annals of Applied Statistics*, 6(3): 1236–1255, 2012.

M. Mackey and L. Glass. Oscillations and chaos in physiological control systems. *Science*, 197:287–289, 1977.

S. Moinas. Hidden limit orders and liquidity in order driven markets. working paper 10-147, Toulouse School of Economics, 2010.

N. Monk. Oscillatory expression of hes1, p53, and nf-κb driven by transcriptional time delays. *Current Biology*, 13:1409–1413, 2003.

J. Nagumo, S. Arimoto, and S. Yoshizawa. Impulses and physiological states in models of nerve membrane. *Proc.Inst.Radio Engrs*, 50:2061–2070, 1962.

NASDAQ. NASDAQ stock market rules. http://nasdaq.cchwallstreet.com/, 2008.

A. Pardo Tornero and R. Pascual. On the hidden side of liqudity. Available at SSRN: http://ssrn.com/abstract=459000 or doi:10.2139/ssrn.459000, 2007.

J. O. Ramsay, G. Hooker, D. Campbell, and J. Cao. Parameter estimation for differential equations: a generalized smoothing approach (with discussion). *Journal of the Royal Statistical Society, Series B*, 69:741–796, 2007.

C. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. MIT: MIT Press, 2004.

G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.

M. Stein. *Interpolation of Spatial Data*. Springer-Verlag, New York, 1999.

L. Tuttle. Hidden orders, trading costs, and information. Unpublished working paper, University of Kansas, Lawrence, KS., 2006.

L. Wang and J. Cao. Estimating parameters in delay differential equation models. *Journal of Agricultural, Biological, and Environmental Statistics*, 17:68–83, 2012.

H. R. Wilson. *Spikes, Decisions and Actions: the Dynamical Foundations of Neuroscience*. Oxford: Oxford University Press, 1999.