TWO STUDIES IN NONLINEAR BIOLOGICAL SYSTEM MODELING AND IDENTIFICATION

By

Jinyao Yan

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Electrical Engineering – Doctor of Philosophy

2018

ABSTRACT

TWO STUDIES IN NONLINEAR BIOLOGICAL SYSTEM MODELING AND IDENTIFICATION

By

Jinyao Yan

Biological systems are often complex, nonlinear and time-varying. The modeling of biological systems, therefore, presents significant challenges that are not overcome by the classical linear methods. In recent decades, intensive research has begun to produce methods for analyzing and modeling isolated classes of nonlinear systems. However, this vast class of models still presents many challenges, especially in complex biological systems. In this research, two novel methods are introduced for analyzing time series resulting from nonlinear systems.

In the first approach, we study a class of dynamical systems that are nonlinear, discrete and with a latent state-space. We solve the probabilistic inference problem in these latent models using a variational autoencoder (VAE). Compared to continuous latent random variables, the inference of discrete latent variables is more difficult to solve. However, stochastic variational inference provides us with a general framework that tackles the inference problem for this class of model. We focused on an important neuroscience application – inferring pre- and post-synaptic activities from dendritic calcium imaging data. For it, we developed families of generative models, a deep convolutional neural network recognition model, and methods of inference using stochastic gradient ascent VAE. We benchmarked our model with both synthetic data, which resembles real data, and real experimental data. The framework can flexibly support rapid model prototyping. Both the generative model and recognition model can be changed without perturbing the inference. This is especially beneficial for testing different biological hypotheses.

As a second approach, we treat a subclass of nonlinear autoregressive models: linear-time-invariant-in-parameters models. This class of models is useful and easy to work with. We propose an identification algorithm that simultaneously selects the model and does parameter estimation. The algorithm integrates two strategies: set-based parameter identification, and evolutionary algorithms

that optimize fitness measures derived from these solutions. The algorithm can identify nonlinear models in novel noise scenarios. We show the performance of the algorithm in various simulated systems and practical datasets. We demonstrate its application to identify causal connectivity in a graph. This problem is often posed in recovering functional connectivity in the brain.

The main contribution of this thesis is that we provide two framework for identifying nonlinear, biological systems from time series data. These two classes of nonlinear models and their applications are significant as each class is broad enough for modeling many complicated biological systems. We develop general, fast algorithms for learning these systems from data for these two model classes.

In memory of my aunt, Yaling Yan, who has inspired me for the pursuit of science.

ACKNOWLEDGEMENTS

I am grateful to my advisers: Prof. John Deller, who has patiently guided me through my graduate study; Prof. Erik Goodman, for the passionate discussion and immediate help; Dr. Srini Turaga, who has given me countless insightful suggestions and feedback.

I would also like to thank Aaron Kerlin for providing me with the calcium imaging data, and spending lots of time debugging and testing my code; Laurence Aitchison for coaching me on the model development; Artur Speiser for sharing the code and expertise with me; and everyone who we shared inspiring conversation during my study in the two laboratories at Michigan State University and Janelia Research Campus.

Special thanks to my thesis committee members, Anil Jain and Selin Aviyente, for your generous feedback on my thesis.

To my two best friends who have shared my joys and sorrows, and being my life-coaches: Zejia Zheng and Maxium Manakov.

Last but not least, I'd like to thank my families – grandparents, parents, and sister (especially my grandmom, my mom, and my twin sister) for years of support and belief in me.

PREFACE

This thesis contains two disjoint research I have done during my graduate study. The first research (Part 2) was guided by my advisers at Michigan State University, Prof. John Deller and Prof. Erik Goodman, during my first three years of study. At the end of my third graduate year, I had the honor of receiving a prestigious fellowship from Janelia Research Campus. I was invited as a Graduate Research Fellow to work with Dr. Srini Turaga. Together, we have worked on a different research topic (Part 1). Though employing different approaches, both the research is designated for analyzing time series data of nonlinear, biological systems.

TABLE OF CONTENTS

LIST OF TABLES xi

LIST OF FIGURES xii

LIST OF ALGORITHMS xvii

PART 1 Variational Auto-Encoder for Discrete State-Space Dynamical Systems 1

CHAPTER 1 INTRODUCTION 2
- 1.1 Directed Graphical Model 2
- 1.2 Variational AutoEncoder 3
  - 1.2.1 Variational Inference and ELBO 4
  - 1.2.2 Stochastic Variational Inference 5
    - 1.2.2.1 Score Function Gradients 5
    - 1.2.2.2 Reparameterization Gradients 7
  - 1.2.3 Amortized Inference and Deep Neural Network 8
- 1.3 Overview of the Research 9

CHAPTER 2 INFERRING NEURONAL PRE- AND POST-SYNAPTIC ACTIVITY . . . 10
- 2.1 Dendritic Calcium Imaging 10
- 2.2 Generative Model 12
  - 2.2.1 Spike Interactions 13
  - 2.2.2 Calcium Indicator Models 15
    - 2.2.2.1 Linear Model 15
    - 2.2.2.2 Static Nonlinear Model 16
    - 2.2.2.3 Dynamical Nonlinear Model 16
    - 2.2.2.4 Implementation 17
- 2.3 Recognition Model 19
  - 2.3.1 Soma Network 20
  - 2.3.2 Spine Network 20
- 2.4 Inference Algorithm 22

CHAPTER 3 EXPERIMENTS AND RESULTS 26
- 3.1 Low Firing Rate Cell with Moderate Noise 28
- 3.2 High Firing Rate Cell with Moderate Noise 36
- 3.3 High Firing Rate Cell with Large Noise 42
- 3.4 Real Data 48

CHAPTER 4 CONCLUSIONS FOR PART I 51
- 4.1 Contributions 52
- 4.2 Limitations of the Approach 53
- 4.3 Future Work 54

PART 2 Nonlinear System Identification Using a Set-Theoretic Evolutionary Approach 55

CHAPTER 5 INTRODUCTION 56
5.1 Methods for parameter estimation 58
5.2 Review of OBE algorithms 60
5.3 Motivation for a new algorithm 62

CHAPTER 6 NONLINEAR EVOLUTIONARY SYSTEM IDENTIFICATION 64
6.1 Introduction and Overview 64
6.2 Identification framework 66
6.3 Set-theoretic parameter estimation 68
6.3.1 Formulation of OBE algorithms 68
6.3.2 Technical adjustments for evolving regressor signals 74
6.4 Evolutionary model selection 76
6.4.1 "Cell biology" of the LTIiP model 76
6.4.2 Survival fitness measures 78
6.4.3 Evolutionary operations 80
6.5 Multi-objective optimization 81
6.5.1 Optimization formulation 81
6.5.2 Multi-objective genetic algorithm 82

CHAPTER 7 EXPERIMENTS, APPLICATIONS AND DISCUSSION 85
7.1 Introduction 85
7.2 Results on simulated sequences: Single-objective optimization 85
7.2.1 IID noise sequence 85
7.2.1.1 System 1 85
7.2.1.2 System 2 88
7.2.2 Colored noise sequence 89
7.2.2.1 System 3 89
7.2.3 General noise sequence 90
7.2.3.1 System 4 90
7.2.4 Comparison study 92
7.2.4.1 System 5 92
7.3 Simulation results: Multi-objective optimization 94
7.3.1 White noise disturbances 94
7.3.1.1 Empirical combination of objective functions 95
7.3.1.2 Multi-objective optimization 96
7.3.1.3 Selecting the final model – Regressor significance 96
7.3.1.4 Comparison with classical method 99
7.3.2 Colored noise disturbances 101
7.4 Practical datasets 104
7.5 Application to causality analysis 108
7.5.1 Methods 109
7.5.1.1 Bivariate time series 109

7.5.1.2 Multivariate time series . . . 110
7.5.2 Simulation results . . . 112
7.5.2.1 Colored noise disturbances . . . 112
7.5.2.2 Nonlinear network . . . 114

CHAPTER 8 CONCLUSION FOR PART II . . . 116
8.1 Contributions . . . 117
8.2 Limitations of the Approach . . . 118
8.3 Future Work . . . 118

APPENDICES . . . 119
APPENDIX A INTEGRATING FACTOR TRICK . . . 120
APPENDIX B INFERENCE ALGORITHMS . . . 121

BIBLIOGRAPHY . . . 125

LIST OF TABLES

Table 6.1: Specification of popular OBE algorithms. The notation of λ_* is used to denote the existence of an optimal weight that minimizes the optimization criterion [Deller Jr. et al., 1994] 74

Table 6.2: Adaptation of system modeling to a genetic algorithm 78

LIST OF FIGURES

Figure 1.1: Example of Directed Graphical Model. 3

Figure 2.1: A Cell Cartoon. 11

Figure 2.2: Variational AutoEncoder. 12

Figure 2.3: Generative Model. The dendritic model consists of multiple sites, including time series at both soma and spines. The processes of spikes giving rise to fluorescence is given by dynamical system Eqn. 2.5 and Eqn. 2.11. s is the spike train, c represents calcium activity, and y represents the dye activity, and f is our noisy measurement. 19

Figure 2.4: Recognition Model. The recognition model is a hierarchical model that has two deep convolutional neural networks. Left: soma network, which takes in the fluorescent traces from all sites and predicts the spike trains for the soma. Right: spine network, which has not only traces, but also the soma probability and samples, and estimated generative parameters as input. The structure provides us an efficient way to sample from the distribution and evaluate the probabilities of samples. 22

Figure 2.5: Illustrations of Our Inference Procedure. The recognition model takes the fluorescent observations (data) as input, and predict spike probabilities. We then draw samples from the distribution and pass the samples through the generative model. The objective can be seen as a variant of reconstruction error between the output of generative model and observations with some regularizer in the spike probability space. The parameters of both the generative model and the recognition model are learned by gradient descent. 25

Figure 3.1: Low firing rate, good SNR cell: Inference and reconstruction using different algorithms. The cell has a firing rate of about 0.5Hz. The SNR at the cell body is 18.41, whereas the spines have a range of SNR from 18.4 to 1. Shown here are the results of the inferred spikes for a rather clean spine. Our model has shown superior performance for removing the backpropagated action potential from spines. 29

Figure 3.2: Low firing rate, good SNR cell: Inference and reconstruction using different algorithms. The inference is robust to large noise since we have a shared structure in the spine network. Also, the marginal distribution reflects the quality of the data, and the uncertainty in the estimation. 30

Figure 3.3: Low firing rate, good SNR cell: Correlation coefficient and root mean squared error sitewise with ground truth. 31

Figure 3.4: Low firing rate, good SNR cell: Correlation Matrix: Ground Truth, VIMCO, ELBO 32

Figure 3.5: Low firing rate, good SNR cell: Correlation Matrix: GUMBEL, DRR 33

Figure 3.6: Low firing rate, good SNR cell: Soma-Spine (Input-Output) Correlation and Spine-Spine (Input-Input) Correlation. The two parameters associated for each method are the slope and intercept of the line fitted to the scatter points. . . 34

Figure 3.7: Low firing rate, good SNR cell: Generative Parameters. 35

Figure 3.8: High firing rate, good SNR: Inference and reconstruction using different algorithms on synthetic data. 36

Figure 3.9: High firing rate, moderate noise: Inference and reconstruction using different algorithms on synthetic data. 37

Figure 3.10: High firing rate, moderate noise: : Correlation coefficient and Root mean squared error 38

Figure 3.11: High firing rate, moderate noise: Correlation Matrix: Ground Truth, VIMCO, ELBO 39

Figure 3.12: High firing rate, moderate noise: Correlation Matrix: GUMBEL, DRR 40

Figure 3.13: High firing rate, moderate noise: Soma-Spine (Input-Output) Correlation and Spine-Spine (Input-Input) Correlation. 40

Figure 3.14: High firing rate, moderate noise: Generative Parameters. 41

Figure 3.15: High firing rate, large noise: Inference and reconstruction using different algorithms on synthetic data. 42

Figure 3.16: High firing rate, large noise: Inference and reconstruction using different algorithms on synthetic data. 43

Figure 3.17: High firing rate, large noise: : Correlation coefficient and Root mean squared error 44

Figure 3.18: High firing rate, large noise: Correlation Matrix: Ground Truth, VIMCO, ELBO 45

Figure 3.19: High firing rate, large noise: Correlation Matrix: GUMBEL, DRR 46

Figure 3.20: High firing rate, large noise: Soma-Spine Correlation and Spine-Spine Correlation. 46

Figure 3.21: High firing rate, large noise: Generative Parameters. 47

Figure 3.22: Real Data 1: The first two rows are the trace, reconstruction and inferred spikes for the soma. The second and third two rows are the trace, reconstruction and inferred spikes for two different spines. This plot shows that our algorithm successfully removed back-propagated action potentials. 49

Figure 3.23: Real Data 2: The first two rows are the trace, reconstruction and inferred spikes for soma. The second and third two rows are the trace, reconstruction and inferred spikes for two different spines. The two spines receive many independent events. As we can see, our algorithm identifies independent inputs. 50

Figure 6.1: Hyperstrips ω_n and their intersection Ω_t as described in (6.12) for a system of order 2 ($m = 2$) at time $t = 3$. 70

Figure 6.2: The ellipsoid superset $\overline{\Omega}_n \supseteq \Omega_n$ corresponding to the system of Fig. 6.1. 72

Figure 7.1: Simulated System 1: input and output data. 87

Figure 7.2: System 1: Residual and linear correlation Test: the horizontal line on the second figure is the 95% confidence interval. 87

Figure 7.3: System 1: Identification results: True data (continuous curve) and estimated data (dash-dot curve). 88

Figure 7.4: System 3 with colored noise: Residual and linear correlation test: the horizontal line on the second figure is the 95% confidence interval. 90

Figure 7.5: System 3 with colored noise: Identification results: True data (continuous curve) and estimated data (dash-dot curve). 91

Figure 7.6: System 4 with general noise: Residual and linear correlation test: the horizontal line on the second figure is the 95% confidence interval. 92

Figure 7.7: System 1 input and output signals. The horizontal axis is the sample index, and the vertical axis is the amplitude. 95

Figure 7.8: Landscape of the objective space (horizontal axis is the number of parameters, vertical axis is the residual squared error): the '$\times$' points are uniform randomly generated binary strings, and the '$\bullet$' point is the ideal point. 97

Figure 7.9: Pareto efficient points (horizontal axis is the number of parameters, vertical axis is the residual squared error). The point $(0.0875, 5)$ is not plotted for visualization purpose. 97

Figure 7.10: The significant value of each term (horizontal axis is the index of regressors, vertical axis is the significant value). 98

Figure 7.11: System (7.3.1) residual and linear correlation tests. Horizontal lines inn the lower figure show the 95% confidence interval. 99

Figure 7.12: System Identification results: True data (continuous curve with '$\cdot$') and estimated data (dashed curve with '$\times$'). 100

Figure 7.13: System (7.3.1) – Comparison of models. The horizontal axis is the index of regressors, and the vertical axis is the true or estimated parameter values: True parameter values are indicated by $\circ$, estimated system model using evolutionary algorithm $\triangle$, and estimated model using OMP $\times$. 101

Figure 7.14: The significance value of each term (horizontal axis is the index of regressors, vertical axis is the significance value). 102

Figure 7.15: System (7.15) residual and linear correlation tests. Horizontal lines in the lower figure show the 95% confidence interval. 103

Figure 7.16: System (7.3.2) – Comparison of models. The horizontal axis is the index of regressors, and the vertical axis is the true or estimated parameter values: True parameter values are indicated by $\circ$, estimated system model using evolutionary algorithm $\triangle$, and estimated model using OMP $\times$. 104

Figure 7.17: Two-tank system input and output data. 105

Figure 7.18: Two-tank system identification. True data (continuous curve) and estimated data (dash-dot). 107

Figure 7.19: The connection regimes that are ambiguous under pair-wise GCA, but can be distinguished by conditional GCA. A: the connection from process X to process Y is directed; right: the connection from process X to Y is mediated through Z. B: process X drives the other two processes Y and Z with differential time delays, while there is no information flow from Y to Z. Right: Y and Z have a common source, but are not directly related. 111

Figure 7.20: Schematic illustration of the system: a simulated five-node structurally connected with different time delays. 112

Figure 7.21: Causal connections identified by conventional OLS for System (7.24). A: Identified system connection using OLS - white noise scenario. The GCA using the OLS algorithm identified the correct connections. Green (red) indicates uni (bi-) directional causality. Width of connector indicates strength. B: Identified system connection using OLS - colored noise scenario. The GCA using OLS algorithm identified incorrect connections. 113

Figure 7.22: Identified system connection using proposed algorithm - colored noise scenario. The proposed algorithm identifies the proper connections even in colored noise. 114

Figure 7.23: Causal connections identified for System 7.34. The proposed algorithm identifies the nonlinear connections. A: Identified system connection using OLS. The GCA using the OLS algorithm incompletely identified the connections, missing nonlinear connections. B: Identified system connection using proposed algorithm. The proposed algorithm identified all connections. 115

LIST OF ALGORITHMS

Algorithm 1: OBE-ERS 76

Algorithm 2: Evolutionary model selection 81

PART 1

Variational Auto-Encoder for Discrete State-Space Dynamical Systems

CHAPTER 1

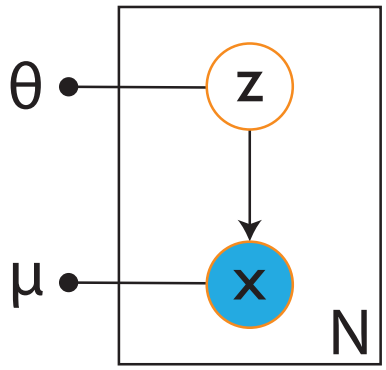
INTRODUCTION

Though providing service to each other, the fields of Statistics and Computer Science have followed separate development paths for the past few decades, with core concerns of the two fields being quite different [Jordan, 1998, Wainwright et al., 2008]. Statisticians have been focusing on developing and applying probability theory to study interactions of random variables in the data, while computer scientists mainly focus on solving computational problems with fast and efficient algorithms. However, in recent years, the interests of these two fields have witnessed increasing overlap. With the arrival of big data, statistical applications require more and more large-scale and complex models, such as applications in the fields of biomedical signal processing, neuroscience, genetic information, image and speech processing *etc.*, which calls for more powerful and efficient inference algorithms. At the same time, computer scientists have growing interests in modeling and analyzing real data, and quantifying uncertainties in both their data and results. One area that is most evident regarding this trend is the probabilistic graphic model, and such models have attracted more attention recently in the machine learning and pattern recognition community. The distinctive feature of a graphic model is that it provides a natural and systematic formalism for probabilistic models, but it also parts with control over the computational complexity. As a result, graphic models provide a general methodology for approaching large-scale models involving thousands of random variables interacting in complex ways.

1.1 Directed Graphical Model

A graphic model is a diagrammatic representation of a family of probability distributions in which the nodes of a graph are identified with random variables, and the links between nodes represent probabilistic relationships between these variables [Bishop, 2006]. The graph can be categorized as a directed (acyclic) graph or an undirected graph. The directed graph offers more than just an appealing visualization of the joint distributions. It also provides a convenient factorization

Figure 1.1: Example of Directed Graphical Model.



based on conditional independence. A node A is called a parent node of B, if there is a directed link connecting from node A to node B. Given the graph, the joint distribution is represented as products of conditional probabilities of all nodes given their parents' nodes [Bishop, 2006]. Directed graph models have found applications in many areas, such as clustering, time series analysis, stochastic dynamical systems, *etc.* They are especially familiar as representing causal structure and hierarchical Bayesian models. To distinguish data and parameters, we use an unshaded circle to denote the hidden (not observed) random variables, a shaded circle for observed random variables and a dot for the parameters. Succinctly, *plate*, a square outskirts nodes are used to designate replication of graphs. Fig. 1.1 is an example of a directed graphical model.

1.2 Variational AutoEncoder

The inference problem in graphical models refers to the problem of inferring the posterior distribution of one or more subsets of hidden nodes given observed nodes. There are two big schools of methods used to solve a probabilistic inference problem: exact inference *vs.* approximate inference. Exact Inference algorithms such as the sum-product algorithm compute marginal probabilities by exploiting the conditional independence encoded in the graph [Jordan, 1998, Jordan et al., 1999]. Classical graphical models, such as Kalman filters and Hidden Markov models, can be solved efficiently using exact inference. However, it is often infeasible to use exact inference in many problems of practical interest, for instance, nonlinear time series models or Bayesian neural networks, *etc*. The posterior distribution could have a very complicated form where expectations are

not analytically tractable, or the dimensionality of the latent space is too large to work with. This is especially the case when we have large-scale models and we want to model complex relationships. On the other hand, approximate inference, including sampling methods and variational inference, provide a general methodology for probabilistic inference. In particular, variational inference, especially recent development in black-box variational autoencoders (VAE), has progressed toward solving more complicated, larger-scale problems.

1.2.1 Variational Inference and ELBO

In a nutshell, variational inference solves an intractable probabilistic inference problem by transforming it into an optimization problem. Suppose $\mathbf{x} = \{x_1, x_2, \ldots, x_N\}$ consists of independent and identically distributed (i.i.d) samples of random variable $\mathbf{x}$. We model the data as generated from a random process conditioned on hidden variables $\mathbf{z}$. The variational inference method uses a parametric distribution $q_\phi(\mathbf{z}|\mathbf{x})$ to approximate the posterior distribution $p(\mathbf{z}|\mathbf{x})$. The 'distance' between the two distributions is measured by the Kullback–Leibler divergence,

$$\begin{aligned} \mathrm{KL(q||p)} &= -\mathbb{E}_{q_\phi(\mathbf{z}|x)}[\log p(\mathbf{z}|\mathbf{x}) - \log q_\phi(\mathbf{z}|\mathbf{x})] \\ &= -\mathbb{E}_{q_\phi(\mathbf{z}|x)}[\log p(\mathbf{z},\mathbf{x}) - \log q_\phi(\mathbf{z}|\mathbf{x})] + \log p_\theta(\mathbf{x}) \end{aligned} \tag{1.1}$$

Usually $p(\mathbf{z},\mathbf{x})$ describes a generative process of the data, so is also called a generative model. It usually has some parameters associated with it; let's denote them as θ. Since θ parameterizes the forward model, we will call them generative model parameters. We will refer to ϕ as variational parameters.

Let

$$L(\mathbf{x};\theta,\phi) = \mathbb{E}_{q_\phi(\mathbf{z}|x)}[\log p_\theta(\mathbf{z},\mathbf{x}) - \log q_\phi(\mathbf{z}|\mathbf{x})] \tag{1.2}$$

Rearranging the terms of Eqn. 1.1, we can write the data likelihood $p_\theta(\mathbf{x})$ as,

$$\log p(\mathbf{x}) = \mathrm{KL(q||p)} + L(\mathbf{x};\theta,\phi) \tag{1.3}$$

Since by definition, $\mathrm{KL(q||p)} \geq 0$, $L(\mathbf{x};\theta,\phi)$ is a lower bound on the data likelihood,

$$\log p(\mathbf{x}) \geq L(\mathbf{x};\theta,\phi) \tag{1.4}$$

$L(\mathbf{x}; \theta, \phi)$ is called the evidence lower bound (ELBO). The variational inference approach solves the inference by maximizing the objective ELBO, which is equivalent to minimizing the Kullback–Leibler divergence between the approximate distribution and the true posterior distribution. To interpret ELBO, we can see that there are two parts of this objective; the first term encourages $q(z|x)$ to concentrate its mass on the Maximum A Posteriori (MAP) estimate, while the second term $\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\{-\log q_\phi(\mathbf{z}|\mathbf{x})\}$, which equals the entropy of the approximate distribution, encourages the distribution to be diffuse.

1.2.2 Stochastic Variational Inference

Traditionally, VI can be solved quickly for conditionally conjugate exponential family models, where the distribution of each latent variable given its Markov blanket [Bishop, 2006] falls in the same family as the prior distribution [Jordan et al., 2001]. Closed-form coordinate-ascent updates can be derived whenever such requirements are satisfied. However, whenever the variational family falls outside this small distribution family, a problem arises. To solve VI for general cases, and to avoid model-specific derivation, [Ranganath et al., 2014] introduced black box variational inference, which uses stochastic optimization of the variational objective, where gradients are carefully estimated. With this recent development, the modern VI can scale up to handle massive data, can have more flexible and expressive families of approximation, and is easier to derive and apply to more difficult models and problems.

The key to the success of the black box variational inference is estimating the gradients of the parameters. The original paper [Ranganath et al., 2014] achieved a noisy, but unbiased Monte Carlo estimate of the gradients, which is also called the score function gradients.

1.2.2.1 Score Function Gradients

Define,

$$g(\mathbf{z}, \xi) = \log p_\theta(\mathbf{x}, \mathbf{z}) - \log q_\phi(\mathbf{z}) \tag{1.5}$$

where $\xi = \{\theta, \phi\}$. The gradients of the ELBO (Eqn. 1.2),

$$\nabla L(\mathbf{x}; \xi) = \nabla_\xi \int q_\xi(z) g(\mathbf{z}, \xi) dz \tag{1.6}$$

$$= \int \Big[\nabla_\xi q_\xi(z) g(\mathbf{z}, \xi) + q_\xi(z) \nabla_\xi g(\mathbf{z}, \xi) \Big] dz \tag{1.7}$$

Using the **score function trick**,

$$\nabla_\xi \log q = \frac{\nabla_\xi q}{q} \tag{1.8}$$

We have,

$$\nabla_\xi L = \int \Big[q_\xi(\mathbf{z}) \nabla_\xi \log q_\xi(\mathbf{z}) g(\mathbf{z}, \phi) + q_\xi(\mathbf{z}) \nabla_\phi g(\mathbf{z}, \phi, \theta) \Big] dz \tag{1.9}$$

$$= \mathbb{E}_{q_\xi}[q_\xi(z) g(\mathbf{z}, \xi) + q_\xi(z) \nabla_\xi g(\mathbf{z}, \xi)] \tag{1.10}$$

Note that,

$$\mathbb{E}_{q_\xi}[\nabla_\xi g(\mathbf{z}, \xi)] = \mathbb{E}_q[\nabla_\xi q(\mathbf{z}, \xi)] = 0 \tag{1.11}$$

The gradients for the variational parameters ϕ,

$$\nabla_\phi L = \mathbb{E}_q[\nabla_\phi \log q(\mathbf{z}; \mathbf{x})(\log p(x, z) - \log q(z))] \tag{1.12}$$

Specifically, the gradient for the generative model parameter is,

$$\nabla_\theta L = \int q_\phi(z) \nabla_\theta g(\mathbf{z}, \phi, \theta) dz \tag{1.13}$$

$$= \mathbb{E}_{q_\phi}[\nabla_\theta g(z, \phi, \theta)] \tag{1.14}$$

Simplifying,

$$\nabla_\theta L = \mathbb{E}_q[\nabla_\theta \log p_\theta(\mathbf{x}, \mathbf{z})] \tag{1.15}$$

Both the gradient for generative parameters $\nabla_\theta L$, and the gradient for variational parameters $\nabla_\phi L$ can then be estimated by Monte Carlo methods.

$$\nabla_\theta L \approx \frac{1}{n} \sum_{i=1}^{n} \nabla_\theta \log p_\theta(\mathbf{x}, \mathbf{z}^i) \tag{1.16}$$

$$\nabla_\phi L \approx \frac{1}{n} \sum_{i=1}^{n} (\log p_\theta(\mathbf{x}, \mathbf{z}^i - \mathrm{q}_\phi(\mathbf{z}^i | \mathbf{x})) \nabla_\phi \log q_\phi(\mathbf{z}^i | \mathbf{x})) \tag{1.17}$$

This is an unbiased estimator of the gradient [Ranganath et al., 2014]. However, the variance of this estimator is too high to be used in practice.

To address the problem, various variance reduction techniques have been proposed [Mnih and Rezende, 2016, Mnih and Gregor, 2014]. The essence of variance reduction techniques is to replace the function whose value is approximated by Monte Carlo by another function that has the same expectation but less variance. In particular, VIMCO replaces ELBO with a tighter multi-sample objective (Eqn. 1.18), and uses a control variate technique to produce a much lower variance per-sample learning signal [Mnih and Rezende, 2016, Burda et al., 2015].

$$L^K = \mathbb{E}_{\mathbf{z}_1,\ldots,\mathbf{z}_K \sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{1}{K}\sum_{k=1}^{K}\frac{p_\theta(\mathbf{z}^k,\mathbf{x})}{q_\phi(\mathbf{z}^k|\mathbf{x})}\right] \tag{1.18}$$

VIMCO has been proven to have better performance with relatively easier implementation, and it can be used for both discrete and continuous latent variables [Mnih and Rezende, 2016]. The latents in our model are binary random variables, which makes VIMCO a candidate algorithm for fitting the model.

1.2.2.2 Reparameterization Gradients

The reparamterization gradient [Kingma and Welling, 2013], which is also called the path-wise gradient, adds an additional assumption to achieve a better gradient estimator. Assuming there exists a known function t

$$\mathbf{z} = t(\epsilon, \phi), \quad \epsilon \sim u(\epsilon) \quad \text{implies} \quad \mathbf{z} \sim q_\phi(\mathbf{z}) \tag{1.19}$$

which means there is a continuous transformation from random variable ϵ to $\mathbf{z}$. Then, we have,

$$\nabla_\xi L = \mathbb{E}_{q_\xi(z)}[\nabla_\xi \log q_\xi(z) g(z,\xi) + \nabla_\xi g(z,\xi)] \tag{1.20}$$

$$= \mathbb{E}_{u(\epsilon)}[\nabla_\xi \log u(\epsilon) g(t(\epsilon,\xi),\xi) + \nabla_\xi g(t(\epsilon,\xi),\xi)] \tag{1.21}$$

$$= \mathbb{E}_{u(\epsilon)}[\nabla_\xi g(t(\epsilon,\xi),\xi)] \tag{1.22}$$

Rewrite $g(z, \xi)$,

$$\nabla_\phi L = \mathbb{E}_{u(\epsilon)}\Big[\nabla_{\mathbf{z}}[\log p_\theta(\mathbf{x}, \mathbf{z}) - \log q_\phi(\mathbf{z})]\nabla_\phi t(\epsilon, \phi) - \nabla_\phi \log q(\mathbf{z}, \phi)\Big] \tag{1.23}$$

$$= \mathbb{E}_{u(\epsilon)}\Big[\nabla_{\mathbf{z}}[\log p_\theta(\mathbf{x}, \mathbf{z}) - \log q_\phi(\mathbf{z})]\nabla_\phi t(\epsilon, \phi)\Big] \tag{1.24}$$

$$\nabla_\theta L = \mathbb{E}_{u(\epsilon)}[\nabla_\theta \log p_\theta(\mathbf{x}, \mathbf{z})] \tag{1.25}$$

Now we can sample ϵ, and estimate the gradients using Monte Carlo methods. In practical applications, it is seen that this reparameterization-based estimate of the gradient exhibits much less variance than those of competing estimators [Kingma and Welling, 2013].

However, the reparameterization trick only works for continuous random variables, since there is no differential function that maps a continuous set onto a discrete set. To mitigate the problem, [Maddison et al., 2016] proposed the GUMBEL-softmax trick. The idea of Gumbel-Softmax is to use a continuous distribution to approximate categorical samples, such that their parameter gradients can be easily computed via the reparameterization trick [Jang et al., 2016]. Thus it can be used with discrete latent variables. The GUMBEL-softmax trick results in a biased estimator; however, the variance of the estimator is usually smaller. We will use this method, and a variant of the estimator that optimizes the multi-sample objective Eqn. 1.18, to solve our binary, discrete inference problem.

1.2.3 Amortized Inference and Deep Neural Network

To speed up the inference and extract more information from data, amortized inference, also called variational autoencoder (VAE), uses an inference network (also called recognition model) to predict the parameters in the approximate posterior distribution [Gershman and Goodman, 2014], i.e.,

$$q_\phi(\mathbf{z}) = q(\mathbf{z}|\mathbf{x}; f_\phi(x)) \tag{1.26}$$

The downside of using the inference network is that it only admits a smaller function class of approximation, which depends on the flexibility of f. To make f expressive, usually the inference network is embodied as a neural network. In practice, we found that using an inference network

not only not only speeds up the training several fold, but also makes the predictions much more reliable, since the parameters are learned globally.

1.3 Overview of the Research

Focused around an important neuroscience application – inferring pre- and post-synaptic activities from calcium imaging, we have developed a framework for analyzing time series data using a stochastic, discrete state-space model. The framework uses an inference strategy called the stochastic variation autoencoder. In Chapter 2, we describe in detail the problem formulation and the compartments of the model: generative model, recognition model, and the inference algorithm. In Chapter 3, we demonstrate the performance of our method on both synthetic data and real data. Chapter 4 contains further discussion and conclusions.

CHAPTER 2

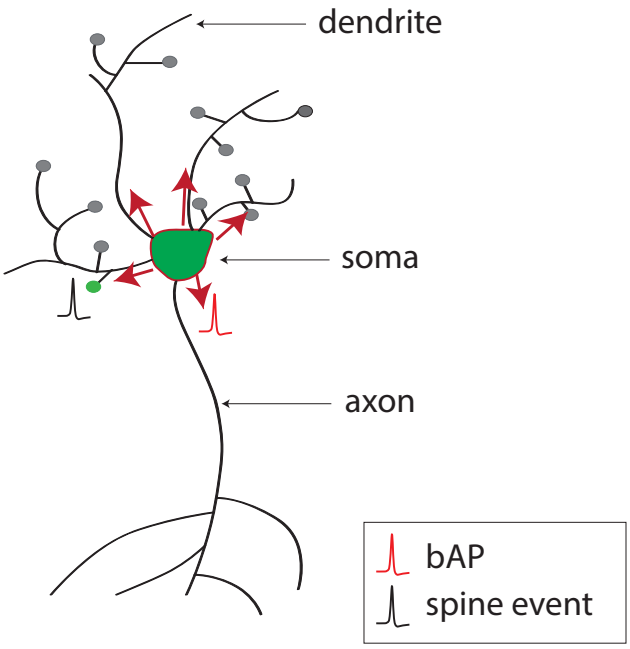
INFERRING NEURONAL PRE- AND POST-SYNAPTIC ACTIVITY

In-vivo calcium imaging can be used to probe the functional organization of the synaptic inputs to a neuron. Excitatory inputs connect to the dendritic arbor of a neuron via small protrusions called spines. Activation of an input can produce a calcium transient that is largely isolated to the spine. Output information from the neuron is carried forward by action potentials (spikes) generated at the cell body, which also back-propagate and contribute strongly to calcium influx at individual spines. Thus, calcium signals in spines reflect a mixture of input and output signals. We propose a statistical model to separate these sources and infer both pre- and post-synaptic action potential activity. This model is a simplified nonlinear approximation of the biophysical processes by which synaptic input and the bAP contribute to the fluorescent measurements at different sites. We use the framework of variational autoencoders (VAE), a recent advance in machine learning, to demix the signals. The VAE is composed of a generative model, which describes the forward process of calcium generation from spike events, and a recognition model. We jointly optimize the parameters of the generative model, a forward model and the recognition network – a deep neural network (DNN) – as part of the VAE to efficiently infer an approximate posterior distribution over spike trains from fluorescence traces [Speiser et al., 2017]. Our training procedure jointly learns parameters of the generative model and the recognition network. These methods are a crucial step towards understanding the transformation of dendritic inputs to somatic spike output in vivo [Yan et al., 2018].

2.1 Dendritic Calcium Imaging

As the fundamental computational unit in the brain, neurons receive, process, and transmit information that is critical for brain function. A neuron receives thousands of input signals via a tree-like branch structure called a dendrite, integrates this information, and then initiates regenerative signals at the cell body that carry information downstream through a wire-like structure called the

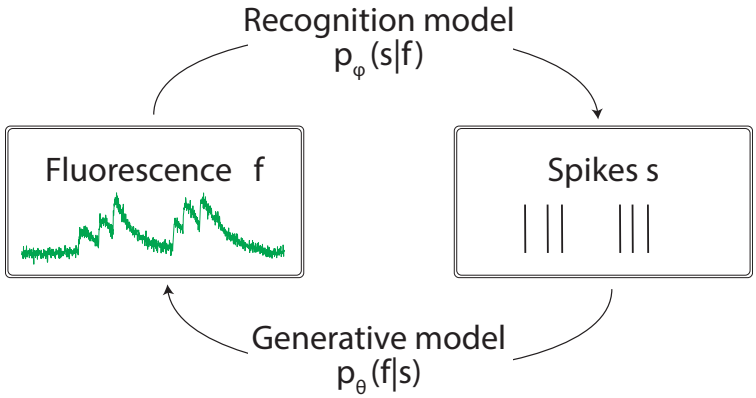
Figure 2.1: A Cell Cartoon.



axon (Fig. 2.1). Axons travel tortuous paths through the brain, intermingled with other thousands of miles worth of other axons that if connected end-to-end would approximately reach the moon from earth. Extracting complete connections between neurons directly from this nanometer-scale spaghetti of anatomical wiring is not yet feasible for the mammalian brain. However, by imaging the input and output of individual neurons while the brain is processing information, we could begin to understand the functional logic of connections between neurons.

In-vivo imaging of the activity of individual neurons and their inputs has been accomplished using fluorescent calcium sensors [Chen et al., 2013]. Initiation of an output signal (spike) in the soma leads to a large influx of calcium. Excitatory input signals arrive at specialized protrusions on the dendrite called dendritic spines. Activation of an input can lead to calcium influx that is restricted to a spine, however output spikes also back-propagate within a few milliseconds to spines which also trigger calcium influx. Given that calcium sensors are significantly slower (>100 milliseconds) than this underlying electrical signaling, the input and output signals become intermixed. Thus, even though in vivo calcium imaging of spines has been around for nearly two decades, little can be said about the transformation of inputs to output.

Figure 2.2: Variational AutoEncoder.



Our goal is to infer spike events from calcium imaging data for both soma and spine. In the meantime, we want to demix the synaptic input events from backpropagated cell output events. Since we don't have the ground truth spike activities, we use a generative-model-based method for unsupervised training of a network for predicting spike probabilities. The problem is formulated as a probabilistic graphical model (Fig. 2.2). The random variables **s** are the latent spike trains, and **f** is our fluorescent measurements. We want to solve the inference problem – the posterior distribution of spike trains given data $p(\mathbf{s}|\mathbf{f})$. Because the models are highly nonlinear, and complicated, exact inference is intractable. Thus we use a variational autoencoder (VAE) to solve the problem. The VAE consists of two parts, the generative model (forward model), which describes the data generative process, *i.e.*, how spikes produce fluorescence observations, and the recognition model (backward model), which predicts the spike trains given fluorescence data. In the following section, we will explain the parts that constitute the VAE, *i.e.*, generative model and recognition model, as well as the inference algorithms.

2.2 Generative Model

"Generative model" is machine learning jargon which refers to the joint distribution of input and output data. We denote our generative model as $p_\theta(\mathbf{s}, \mathbf{f})$. We use **s** to represent the spike train, and **f** for the fluorescent measurements. Both **s**, **f** are random variables, where **s** is the latent variables, and **f** is the observation variables. θ are the generative model parameters. Using the product rule

of probability, we have $p(\mathbf{s}, \mathbf{f}) = p(\mathbf{f}|\mathbf{s})p(\mathbf{s})$. We will refer to $p(\mathbf{s})$ as the prior distribution of the spikes. As the name suggests, the prior distribution describes *a prior* knowledge about the spike trains. Here we will use a Bernoulli process with a constant firing rate for the prior distribution.

$$p(\mathbf{s}) \sim \text{Bernoulli}(\mu = c) \tag{2.1}$$

where c is the fixed firing rate of the cell that is estimated from data prior to model fitting.

The conditional distribution $p(\mathbf{f}|\mathbf{s})$ describes how, given the latent variables, the spike trains interact and give rise to the florescent measurement at different locations in the cells. In other words, the conditional distribution describes the interactions of the input spikes, the calcium (Ca^{2+}) activity, as well as the physiological process of the fluorescence changes of a synthetic or genetically encoded Ca^{2+} indicator. Spikes generate intracellular Ca^{2+} increases that have a fast rise and slow decay. Though the signature looks similar, the exact rise times and decay times depend on the location in the cell at which the measurement is taken. For instance, the calcium dynamics are usually $2 \sim 3$ times faster at dendrites than at the cell body. Thus we will estimate a set of calcium parameters at different sites. Since the indicator is the same at different sites (GCaMP6f in our case), we will use shared indicator parameters.

In the following section, we will elaborate on the generative models (with increasing biological details) that we have explored. The first model is a linear model, as many methods for single cell spike inference require. However, our method doesn't require the model to be linear, which allows for fitting more complicated biophysical calcium models. In the second model, we will use a static nonlinear model, *i.e.* a Hill equation, which describes the cooperative nonlinearity and saturation of Ca^{2+} indicators. The third model uses a dynamical nonlinear indicator model that also has a rise time.

2.2.1 Spike Interactions

The computations at dendrites are diverse and complicated [Byrne et al., 2014]. One can build models with different levels of detail for modeling the interactions between synaptic input events

and the neuronal action potentials. According to our data collection setup and procedure, we have made some minor assumptions: Since our sampling rate is about 15Hz and our scan view spans usually centimeters of the cell (the action-potential speed ranges from 1 meter per second to 100 meters per second), we assume that the backpropogated action potential is seen everywhere on the dendrites without delay. In other words, we do not have the resolution needed to see the propagation of the action potential on dendrites. We will assume the electrical signals of the cell's backpropagated action potential and synaptic inputs combine in a linear additive way with different amplitudes. Though simple, it already enables us to tackle the most important aspect of our problem – demixing the input and output signals of the cell's computation. Once demixed, the spikes can be used for calculating the tuning of the cell, the input and output statistics, and the computation carried on by the cell.

We will represent the temporal sequence of spikes as follows,

$$\mathbf{s} = \{\mathbf{s}_t^s, \mathbf{s}_t^{d_j}\} \tag{2.2}$$

where $\mathbf{s}_t^s$ is the spike train measured at the soma, $\mathbf{s}_t^{d_j}$ is the spike train measured at isolated synaptic heads (spines) on the dendritic branches (the spines are ordered by their distance along the branches from the cell body), $j \in 1, 2, ...m$, is the index for different sites, m is the total number of distinctly separated spines.

$$\mathbf{s}^s(t) = \sum_i \delta_{t_i}^s(t) \tag{2.3}$$

$$\mathbf{s}^{d_j}(t) = \sum_i (\delta_{t_i}^{d_j}(t) + \rho^{d_j}\delta_{t_i}^c(t)) \tag{2.4}$$

where $\delta_{t_i}(t)$ is the delta function, $\delta_{t_i} = 1$ when $t = t_i$, $\delta_{t_i} = 0$ elsewhere). 1 indicates that the neuron has produced an action potential. ρ is the ratio between the backpropagated action potential and the synaptic events. As we can see, the spikes at the spines contain both synaptic events and a backpropagated action potential.

2.2.2 Calcium Indicator Models

Several Ca^{2+} indicator models (functions mapping from spike activities to fluorescence) have been developed for the purpose of spike inference [Pnevmatikakis et al., 2014, Deneux et al., 2016, Friedrich and Paninski, 2016, Aitchison et al., 2017, Speiser et al., 2017]. Our method has advantages compared to other approaches in the following respects. First, in most popular spike inference algorithms, the fluorescent process is modeled to be linear [Pnevmatikakis et al., 2014, Friedrich and Paninski, 2016], whereas the underlying process is well known to exhibit nonlinear phenomena – for instance, saturation and cooperativity. In contrast, our framework can model complicated generative models. Due to the black-box nature of our fitting procedure (*i.e.*, no need for model-specific derivations), we can quickly prototype and fit different models with increasing biophysical details. Second, instead of providing only point estimates for the spike trains, our method estimates a distribution over spike trains, which is important for the subsequent analysis. Third, we have carefully developed fast implementations of our benchmarked models, which is crucial for fitting such a large dataset as the dendritic imaging data. Since we know the measurement timing of pixels of the image, we can use that information to infer the spike timing of backpropagated action potentials in super-resolution. This gives us a chance to create a more accurate model, but at the same time it requires finer time binning, thus more computation. Besides, during one session, there could be hundreds of spines being measured. Thus, a fast, scalable algorithm is important for our application.

2.2.2.1 Linear Model

The first model is a linear model, where we model the calcium process as an exponential decay, and the fluorescent measurement is a scaled readout of the calcium process.

$$\frac{dc}{dt} = -\frac{1}{\tau}c + s(t) \tag{2.5}$$

$$f(t) = Ac(t) + b \tag{2.6}$$

where τ is the decay constant, A is the amplitude of the fluorescence for one spike, and b is the baseline. Thus the indicator generative parameters are $\theta = \{\tau, A, b\}$. Note that in fitting the dendritic imaging data, we will fit a set parameters for each measurement site, since the dynamics are different from site to site, $\theta = \{\theta^s, \theta^{d_j}, \rho^{d_j}\}$. $s(t)$ represents one spike train. For simplicity of notation, we will drop the superscript. For the remainder of the chapter, the reader should assume a unique set of parameters for soma and each spine unless specifically noted otherwise.

As we can see, the fluorescence is modeled as a direct, scaled readout of the calcium activity. Notice that we can extend Eqn. 2.5 to include more exponential components fairly straightforwardly. Moreover, in practice, we found that one decay constant explains data quite well; thus we will only report results using Eqn. 2.5.

2.2.2.2 Static Nonlinear Model

In the second model, we use the Hill equation [Hill, 1910] to model the nonlinear effects – cooperativity and saturation of Ca^{2+} indicators,

$$\frac{dc}{dt} = -\frac{1}{\tau}c(t) + s(t) \tag{2.7}$$

$$y = \frac{c^h}{1 + \gamma c^h} \tag{2.8}$$

$$f(t) = Ay(t) + b \tag{2.9}$$

where γ is the saturation level (y saturates at $\frac{1}{\gamma}$), and h is the Hill coefficient. Eqn. 2.7 describes the sigmoidal binding property of the calcium indicators.Thus our generative parameters are $\{\gamma^s, \gamma^{d_j}, h^s, h^{d_j}, \tau^s, \tau^{d_j}, A^s, A^{d_j}, b^s, b^{d_j}, \rho^{d^j}\}$.

2.2.2.3 Dynamical Nonlinear Model

In the third model, we use a more precise dynamical model to describe the activity of the dye, to which, Eqn. 2.7 is the static solution. This model enables us to model the rise time of the dye, thus

is more accurate for the transient fluorescent activities.

$$\frac{dc}{dt} = -\frac{1}{\tau}c(t) + s(t) \tag{2.10}$$

$$\frac{dy}{dt} = \frac{1}{\tau_{\text{on}}}(1 + \gamma c^h)(\frac{c^h}{1 + \gamma c^h} - y) \tag{2.11}$$

$$f(t) = Ay(t) + b \tag{2.12}$$

As compared to the second model, there is another shared dye parameter τ_{on} added to the generative parameter set. For the biophysical inspiration of the model, please refer to [Deneux et al., 2016]. Fig. 2.3 shows the diagram of the generative model.

We assume the measurement noise is an additive Gaussian noise with unknown variance,

$$p(\mathbf{f}|\mathbf{s}) \sim \text{Normal}(f(t), \sigma) \tag{2.13}$$

where $\sigma = \{\sigma_s, \sigma_{d_j}\}$, and will be estimated from data during training.

2.2.2.4 Implementation

Given the spike trains, a naive way to implement these models would be to discretize the model and numerically simulate the corresponding difference equations. However, since speed is important in our application, we have developed a fast convolution way for calculating the output of the model (convolution is much faster on a GPU than the recurrent simulation, since the GPU is optimized for doing such convolution). The method utilizes the integrating factor trick for solving the differential equation first, and then cleverly recognizes the convolution structure inside the integration.

Using the integrating factor trick [Morse and Feshbach, 1946], we can derive an analytical solution for Eqn. 2.5 and Eqn. 2.11. To see the detailed derivation, please see AppendixA. For Eqn. 2.5, we have,

$$c(t) = \int_0^t s(x)e^{\frac{1}{\tau}(x-t)}dx + c_0 e^{-\frac{1}{\tau}t} \tag{2.14}$$

where c_0 is the initial state of the calcium. The integration part is the convolution of s_t with $e^{-\frac{1}{\tau}t}$, which can be computed rapidly on a GPU.

Similarly, we can solve Eqn. 2.11 using the same procedure. Denote $P(t) = \frac{1}{\tau_{\text{on}}}(1 + \gamma c^h(t))y$, $Q(t) = \frac{1}{\tau_{\text{on}}}c^h(t)$, then we have

$$y(t) = e^{-\int_0^t P(v)dv} \int_0^t Q(x)e^{\int_0^x P(v)dv}dx + y_0 e^{-\int_0^t P(v)dv} \tag{2.15}$$

As the exponentiated integrals may become very large or very small, for numerical stability, we subtract a constant, k, which in principle is arbitrary, but in practice is set to the mean of P(t).

$$y(t) = e^{-\int_0^t (P(v)-k)dv} \int_0^t Q(x)e^{\int_0^x P(v)dv}e^{-(t-x)k}dx + y_0 e^{-\int_0^t (P(v)-k)dv}e^{-kt} \tag{2.16}$$

where y_0 is the initial state of the dye. To calculate the integration, we perform the convolution of $Q(x)e^{\int_0^x P(v)dv}$ with e^{-tk}. This implementation is much faster on a GPU than the naive recurrent implementation.

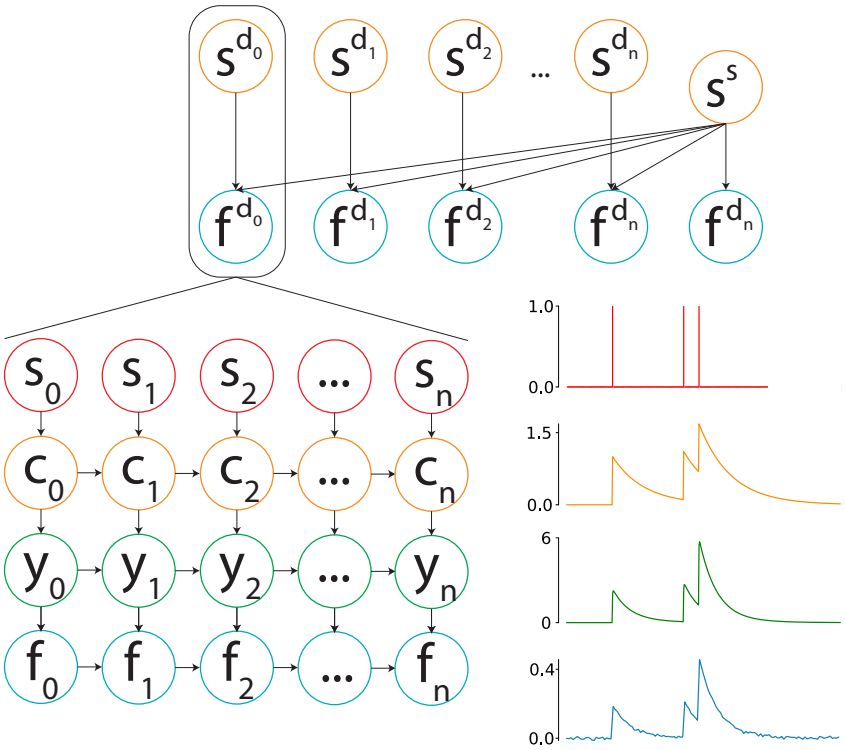
We estimate the initial state c_0 and y_0. During training, at each stochastic gradient step, we randomly draw a snippet of the data. We keep a record of the calcium state matrix (number of time points by number of sites) and the dye state matrix. We update the matrix by averaging the previous estimate with the new estimate.

$$c_0[i,j] = \frac{1}{2}(c_0^{\text{old}}[i,j] + c_0^{\text{new}}[i,j]) \tag{2.17}$$

$$y_0[i,j] = \frac{1}{2}(y_0^{\text{old}}[i,j] + y_0^{\text{new}}[i,j]) \tag{2.18}$$

During testing, we draw samples continuously, thus the initial value is stored and used for later samples.

Figure 2.3: Generative Model. The dendritic model consists of multiple sites, including time series at both soma and spines. The processes of spikes giving rise to fluorescence is given by dynamical system Eqn. 2.5 and Eqn. 2.11. s is the spike train, c represents calcium activity, and y represents the dye activity, and f is our noisy measurement.



2.3 Recognition Model

The goal of the recognition model $q(z|x)$ is to provide an efficient and flexible approximation to the true posterior distribution. Thus it is usually chosen by trading off between expressiveness and convenience. Specifically, in our dendritic modeling problem, the recognition model describes the distribution of spike trains given the fluorescence measurements. As the correlations between soma and spines are what we are after, it therefore is important to model it explicitly. We factorize the posterior distribution $q(z|x)$ as a hierarchical model,

$$q(z|x) = q(\mathbf{s}_c|\mathbf{f})\Pi_j q(\mathbf{s}_{d_j}|\mathbf{s}_c, \mathbf{f}) \tag{2.19}$$

$$q(\mathbf{s}_c|\mathbf{f}) \sim \text{Bernoulli}(\mu_s = g_s(f)) \tag{2.20}$$

$$q(\mathbf{s}_{d_j}|\mathbf{s}_c, \mathbf{f}) \sim \text{Bernoulli}(\mu_d = g_d(f, \dot{s}, \mu_s)) \tag{2.21}$$

where μ_s, μ_d is the mean of the Bernoulli distribution, and $\dot{s}$ is the sample from the soma distribution. g_s and g_d is the soma network and spine network parameterized as a convolutional neural network, $q_\phi(z|x)$.

2.3.1 Soma Network

Since not only the measurement at the soma has information about cell firing, but also the measurements at spines contain backpropogated action potentials which are scaled signals from the soma firing, we use all the fluorescence from the various sites as input to the soma network. The network g_s is a deep neural network which is convolutional in time. For each layer, we create feature maps from the input using local filters of the fluorescence trace centered at the prediction time,

$$T_j(t) = \sigma(w_i * f[t - \omega_{i,j}, t + \omega_{i,j}]), i \in [1, 2, \ldots, n_j] \tag{2.22}$$

where $*$ denotes convolution, $2\omega_i$ is the length of the filter, and n_j is the total number of filters per layer, and $j \in [1, 2, \ldots, m]$, m is the total number of layers, and σ is a nonlinear function.

In order for the network to be invariant to the number and the orders of the spines, in the first layer, we process the soma measurement and the spine measurements separately. The spine inputs are filtered by shared filters, and then averaged, *i. e.*,

$$T_1(t) = \sum_i \sigma(w_i * f[t - \omega_{i,1}, t + \omega_{i,1}]) \tag{2.23}$$

The averaged feature map is then combined with soma input feature maps as different channels as the input to the second layer. The rest of the layers are straightforward convolution layers as in Eqn. eqn: convlayer. We use five hidden layers and 20 filters per layer with selu units [Klambauer et al., 2017]. We use a sigmoid nonlinearity to compute the Bernoulli spike probabilities $q(\mathbf{s}_c|\mathbf{f})$ at the output layer.

2.3.2 Spine Network

The spine network has similar architecture to the soma network. It consists of several hidden convolution layers. Note that since we believe the biophysics of fluorescence is the same among

spines, in order to harvest the statistics power, we use shared filters among spines. The input to the spine network includes fluorescence measurements, soma spike probabilities, and soma samples. The input of soma samples makes spine spike probability conditional. In addition, to differentiate individual properties of each spine, inspired by Wavenet structure [Oord et al., 2016], we also made the spine network conditional on the sitewise generative parameters which are being fitted at the same time.

$$T_j(t) = \sigma(w_i * f[t - \omega_{i,j}, t + \omega_{i,j}] + \nu_i\theta) \tag{2.24}$$

where θ is the generative parameters. This structure facilitates the learning of a family of functions of a shared, global biophysical process of fluorescence to spikes, at the same time that it differentiates the mapping based on the dynamics estimated from the data. For instance, for spines that have a slightly different decay constant, the network can recognize the difference and adjust the weight configuration; or adjust the baseline probability estimate to reflect the uncertainty of the estimates. Since we are fitting both generative parameters θ and the recognition parameters ϕ together, it is important to emphasize that we use estimated generative parameters as conditional input to the spine network. Moreover, when used as a conditional input, the generative parameters are treated as constant. For implementation, we use `tf.stop_gradient` to stop the gradient calculation chain.

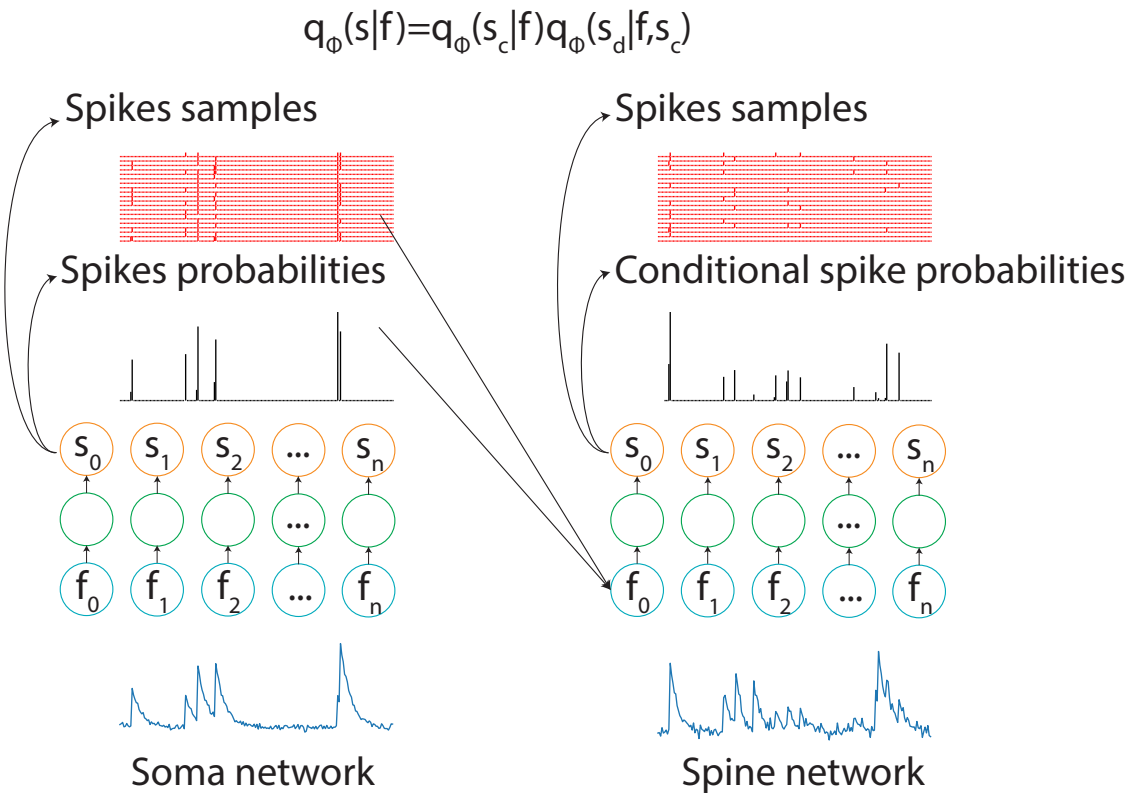
The recognition model, which consist of two main networks, is illustrated in Fig. 2.4. The model gives us an efficient, yet quite flexible distribution to sample, and allows us to evaluate the log probabilities of the spike samples. During sampling, we sample soma spikes in parallel across time first, and then sample spine spikes conditioned on the soma samples. Note that we are not modeling the dependence across time. In other words, we assume,

$$q(\mathbf{s}_c|\mathbf{f}) = \Pi_t q(\mathbf{s}_c(t)|\mathbf{f}) \tag{2.25}$$

$$q(\mathbf{s}_{d_j}|\mathbf{s}_c, \mathbf{f}) = \Pi_t \Pi_j q(\mathbf{s}_{d_j}(t)|\mathbf{s}_c, \mathbf{f}) \tag{2.26}$$

This assumption is a typical one in variational inference as a result of the trade-off between model expressiveness and inference efficiency. We chose this factorization to model the structure we

Figure 2.4: Recognition Model. The recognition model is a hierarchical model that has two deep convolutional neural networks. Left: soma network, which takes in the fluorescent traces from all sites and predicts the spike trains for the soma. Right: spine network, which has not only traces, but also the soma probability and samples, and estimated generative parameters as input. The structure provides us an efficient way to sample from the distribution and evaluate the probabilities of samples.



are most interested in *i.e.*, soma and spine correlations in the data. Because the networks are feedfoward, it is very fast to train them.

2.4 Inference Algorithm

Our goal is to estimate latent spike trains given only fluorescence observations. We use an unsupervised training procedure, which jointly optimizes parameters of the generative model θ, and the recognition network parameters ϕ with respect to a lower bound of the log likelihood of observed data $\log p(\mathbf{f})$. Because our latent variables are discrete, special care needs to be taken for calculating the gradients.

We simultaneously learn the parameters θ and ϕ by jointly maximizing the ELBO 1.2,

$$L = \mathbb{E}_{\mathbf{s}\sim q_\phi(\mathbf{s}|\mathbf{f})}\left[\log \frac{p_\theta(\mathbf{s},\mathbf{f})}{q_\phi(\mathbf{s}|\mathbf{f})}\right] \tag{2.27}$$

We use a minibatch of K samples to estimate the lower bound,

$$\hat{L} = \frac{1}{K}\sum_{i=1}^{K}\left[\log \frac{p_\theta(\mathbf{s}^k,\mathbf{f})}{q_\phi(\mathbf{s}^k|\mathbf{f})}\right], \quad \mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_K \sim q_\phi(\mathbf{s}|\mathbf{f}) \tag{2.28}$$

where $\mathbf{s}^k$ are samples of spike trains from the recognition model.

We also use another multi-sample importance-weighting lower bound on the log likelihood,

$$L = \mathbb{E}_{\mathbf{s}_1,...,\mathbf{s}_K\sim q_\phi(\mathbf{s}|\mathbf{f})}\left[\log \frac{1}{K}\sum_{k=1}^{K}\frac{p_\theta(\mathbf{s}^k,\mathbf{f})}{q_\phi(\mathbf{s}^k|\mathbf{f})}\right] \tag{2.29}$$

During training, we draw one set of K samples from the recognition model $q_\phi(\mathbf{s}|\mathbf{f})$ for one batch of data, which results in a stochastic estimate of the importance-weighted bound,

$$\hat{L} = \log \frac{1}{K}\sum_{k=1}^{K}\frac{p_\theta(\mathbf{s}^k,\mathbf{f})}{q_\phi(\mathbf{s}^k|\mathbf{f})} \tag{2.30}$$

Note the difference between Eqn. 2.28 and Eqn. 2.30. Both lower bounds involve drawing K samples from the recognition model, but they are derived from different lower bounds. When $K = 1$, the importance-weighted bound Eqn. 2.29 reduces to ELBO Eqn. 2.27. Increasing K yields a tighter lower bound than the ELBO on the marginal log likelihood, at the cost of more training time. The importance-weighted lower bound has been reported to provide better fitting of the generative parameters in previous papers [Burda et al., 2015].

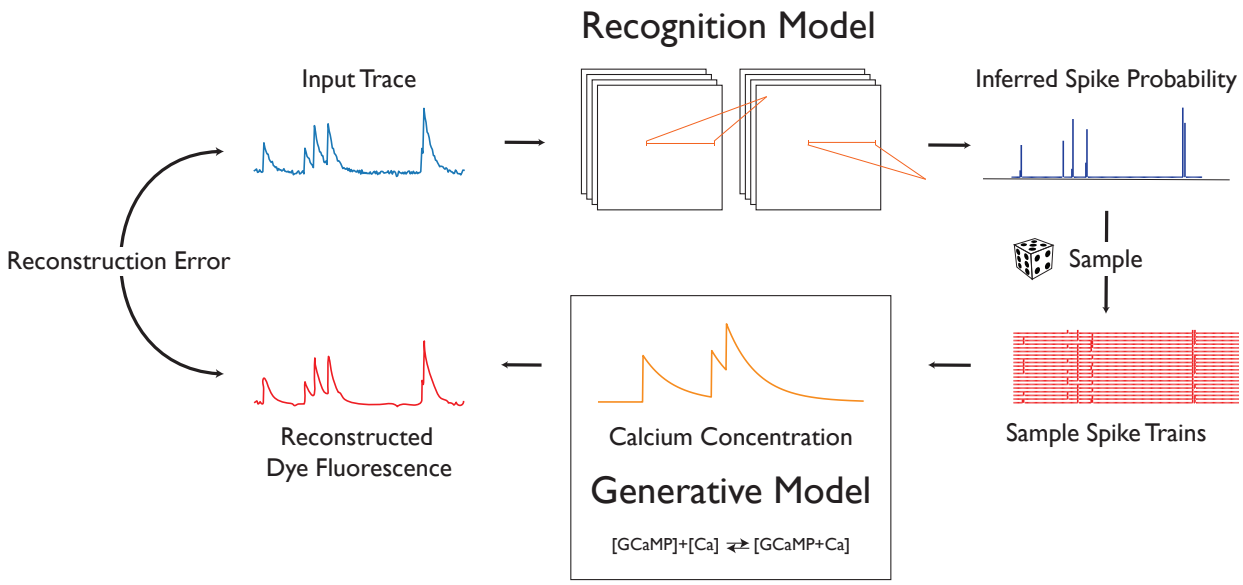
We use stochastic gradient ascent to train the parameters θ, ϕ. We estimate the gradients $\nabla_{\theta,\phi}L$ based on samples. As explained in Chapter 1, obtaining a good estimate of gradients with respect to recognition model parameters is challenging, especially for discrete latent variables. We used two methods: the Gumbel-Softmax Trick for using path-wise gradient estimation [Maddison et al., 2016], and VIMCO [Mnih and Rezende, 2016] – an effective control variate score function approach. Both approaches produce low-variance estimates of the gradients. However, the Gumbel-Softmax approach produce a biased estimate. To compare the performance of different estimators for discrete latents, we have tested thoroughly three inference methods, and we will use the following acronyms here and in the following chapters to refer to these three different algorithms,

- ELBO: reparameterization trick with GUMBEL-Softmax relaxation for gradient estimate with minibatch ELBO objective (Eqn. 2.28).
- GUMBEL: reparameterization trick with GUMBEL-Softmax relaxation for gradient estimate with importance-weighted objective (Eqn.2.30).
- VIMCO: score function with control variates for gradient estimate with importance-weighted objective (Eqn.2.30).

The detailed derivations of all the algorithms are listed in Appendix B. A diagram of the overall framework is shown in Fig. 2.4

Generally, one advantage of our model and inference framework is that it allows quick modular development. The generative model and recognition model can be easily changed, which makes it especially beneficial for prototyping new models. For instance, we can develop our generative model (as long as it is differential, and we can evaluate the log probability of the data) by including more and more biophysical details without model specific derivation. The advantage of black-box inference makes it convenient for customized development and updates. Secondly, we have specifically designed our model to run fast on a GPU. Our software package is developed using an open source software library – python and tensorflow, which allows deployment of computation on CPUs and GPUs.

Figure 2.5: Illustrations of Our Inference Procedure. The recognition model takes the fluorescent observations (data) as input, and predict spike probabilities. We then draw samples from the distribution and pass the samples through the generative model. The objective can be seen as a variant of reconstruction error between the output of generative model and observations with some regularizer in the spike probability space. The parameters of both the generative model and the recognition model are learned by gradient descent.



CHAPTER 3

EXPERIMENTS AND RESULTS

We evaluated our method on simulated and experimental data. We generated spikes using methods in paper [Macke et al., 2009]. The method provides a quick way for generating correlated spike trains with a range of correlation values. The correlations between different sites in our simulated data are in the range of $[0, 0.5]$. The time bin between spikes is about 60ms, which corresponds to the real data. Then we used our developed generative model – the static nonlinear model 2.2.2.2 to create synthetic calcium fluorescence traces. In our simulation, one spike will trigger a fluorescence influx of around 0.1 $\Delta f/f$. The baseline is constant, and fitted according to data. The ratio between the backpropagated action potential and individual events ρ is in the range of $[1.5, 2.5]$. We simulated cells of different firing rates and different signal to noise ratios. Each cell has 15 spines with different dynamics (much faster decay) than the cell body. The dynamics compared between spines, although similar, are slightly different from each other. The traces are of length $T = 10^5$, comparable to the real data length. We benchmarked the performance of our model with three different inference algorithms, *i.e.*, ELBO, GUMBEL, and VIMCO. We also compared our method with a naive one that use off-the-shelf techniques. During each training, we conducted 8 experiments with different initializations and selected the best model based on objective value.

Finally, we showed the performance of our algorithm on real data. The data was collected using a two-photon microscope. Cells at the frontal cortex of mice were imaged using the genetically encoded calcium-indicators GCaMP6f.

For the simulated data, since we have ground truth spikes, we report results using the correlation and root mean square error between true and predicted spike-rates, at the sampling discretization of 15Hz. Moreover, we test our model's performance on reconstructing the correlation matrix between soma and spines. Finally, we compared the estimated generative parameters to the ground truth model parameters. By reporting these measures, we not only show that the recognition model can predict the spike probabilities accurately and robustly, but also that the generative model can

discover the correct model settings.

Training: We use ADAM [26], an adaptive gradient update scheme, to perform stochastic gradient ascent. The training data is cut into short chunks of one hundred fifty time steps and arranged in batches containing samples from a single cell. The generative model parameters are roughly estimated by eye, and randomly initialized near the estimates. The florescence noise is initialized at a large value to smooth the optimization landscape at the beginning of training, which helps the model avoid bad local optima. The recognition model is randomly initialized with a low spike-firing rate. The model is trained up to 150 epochs, with each epoch containing 50 iterations. Early stopping is turned on after 50 epochs of training, based on the convergence of the objective. Since the goal is to predict spikes for the dataset we used for training, and the function of the recognition network is more about speeding up the training than generalization, we do not split the data into separate training and testing sets. All the data from one cell is used for training, and during testing, we predict spikes for all the time points. To avoid local optima caused by having the spine network fitting to both backpropagated action potentials and individual events, we start by training the soma network first, and then start training the spine network after 5 epochs. We use norm-clipping to scale all the gradients: the norm of all gradients is calculated, and if it exceeds a fixed threshold, the gradients are rescaled. We found norm-clipping to be beneficial for achieving high performance for our model. The threshold value is set to 1.0, which yielded the best results empirically.

Testing: After training, we apply the model to the same data. We use the same window size that breaks the data into chunks, and infer the spike train recursively.

The training and testing of the model is very fast, since we have chosen our model and algorithns to be fast, and also most of the calculation happens on a GPU. The average training time per iteration is about 1.2s. The total fitting time for a cell with 1 soma and 14 spines, each 100,000 measurements long, is under one hour. The testing is very fast, taking only minutes [1].

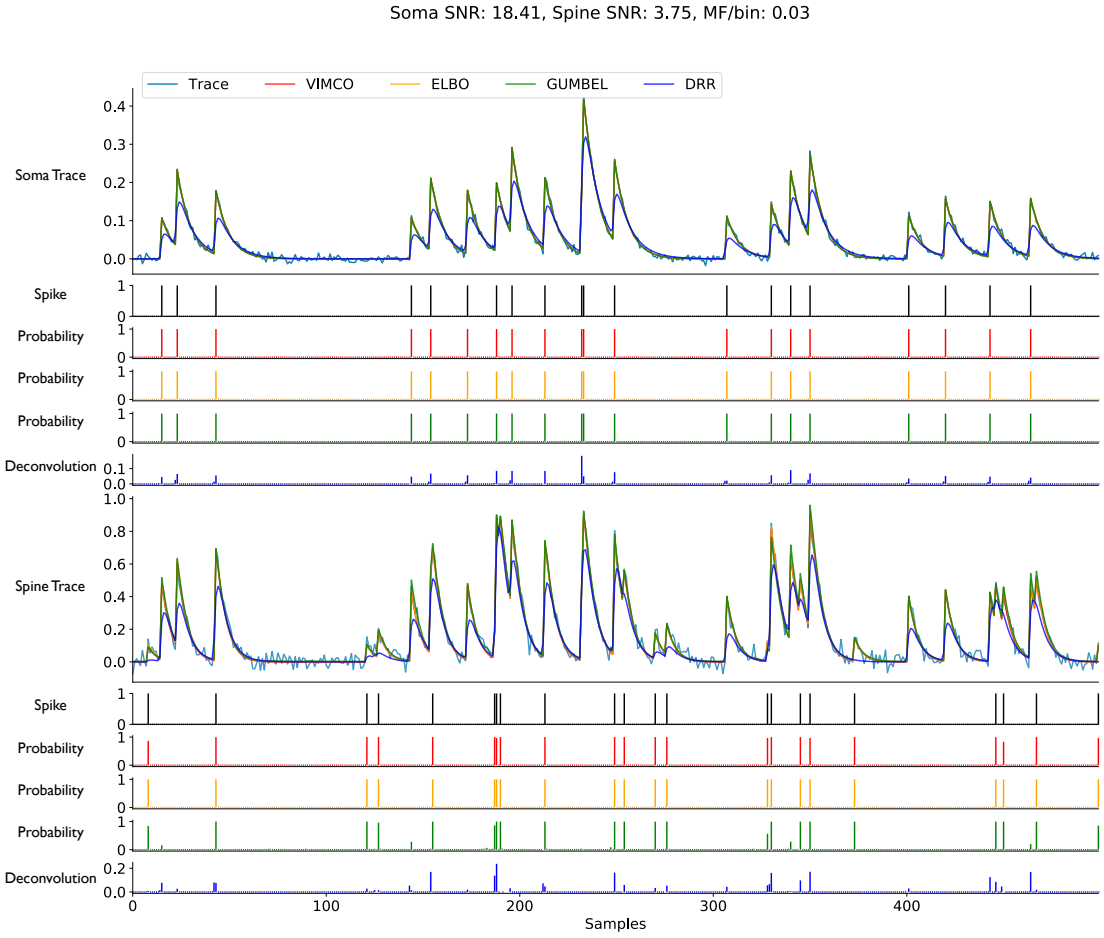
[1]The comparison of the speed of our algorithm versus other methods is benchmarked in our paper on a smaller dataset [Speiser et al., 2017].

3.1 Low Firing Rate Cell with Moderate Noise

For this experiment, we simulated a cell with a firing rate of about 0.5Hz, thus a mean spike probability of $0.5Hz/15 \simeq 0.03$ spikes per bin. The measurement at the soma has a high signal-to-noise ratio of about 18.41. The spines have a range of signal-to-noise ratios ranging from 18 to 1. We fit the data using our static nonlinear model with three different inference algorithms: ELBO, GUMBEL, and VIMCO. Also, inspired by the previous paper about analyzing dendritic imaging data, as a baseline model, we designed another method using off-the-shelf tools popular among neuroscientists. The method used Foopsi [Pnevmatikakis et al., 2014] to deconvolve each calcium trace first. Constrained-Foopsi is a fast, non-negative deconvolution spike-inference algorithm that uses autoregressive linear models for calcium generation. It creates continuous-valued point estimates of spikes from calcium traces. Second, in order to remove the backpropagated action potential in the spine traces, for each spine, we used linear robust regression to estimate the ratio parameter ρ_{d_j}, $j \in [1, 2, \ldots, 15]$, and subtracted the $\rho_{d_j} S_c$ from each spine.

Fig. 3.1 and Fig. 3.2 show the results of the inferred spikes using different methods. We plot the trace with reconstructions from different methods. Also, we show the spike inference as compared to the ground truth spikes which the model is agnostic to. As we can see from the figures, our model (VIMCO, GUMBEL, and ELBO) has demonstrated superior performance compared to the baseline model. First, our model produced more accurate results, both in terms of reconstruction and spike inference. Especially, it removes the backpropagated action potential from spines much better than the baseline method, which is very important in estimating the correct correlation structure later. Secondly, instead of predicting a point estimate at each time bin, our model predicts a distribution which reflects the uncertainty about the inference. The spikes predicted for the soma are very certain because first, we have a lot of information about the somatic spikes (both in the soma trace and spine traces); second, the signal-to-noise ratio is high for extracting this information. The spike distribution predicted for the spines reflects the quality of the data. As we can see, the spike distribution for noisy spines has more variance than that for the clean spines, which reflects the uncertainty in its inference. This information is especially beneficial for the downstream analysis

Figure 3.1: Low firing rate, good SNR cell: Inference and reconstruction using different algorithms. The cell has a firing rate of about 0.5Hz. The SNR at the cell body is 18.41, whereas the spines have a range of SNR from 18.4 to 1. Shown here are the results of the inferred spikes for a rather clean spine. Our model has shown superior performance for removing the backpropagated action potential from spines.

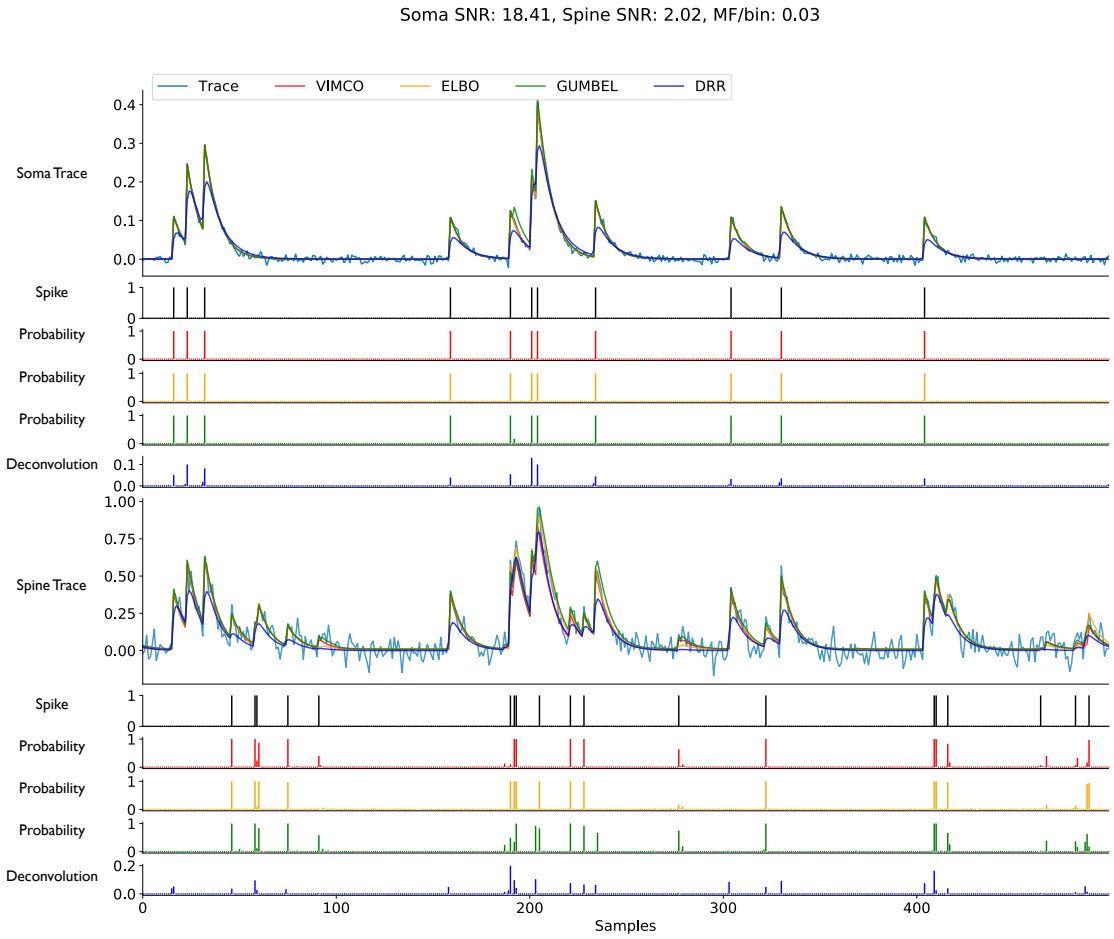


for neuroscientists.

Comparing different training methods for our model, we can see that the performance is comparable, and slightly better performance is achieved by VIMCO and ELBO.

In Fig. 3.3, we evaluate the correlation and root mean square error (RMSE) between the inferred spikes and the ground truth spikes. The index zero represents the soma, and the spines are numbered from 1 to 14, ordered by the level of noise. The correlation shows on average how well we are identifying when there is a spike; thus the higher, the better, bounded by 1. The RMSE shows

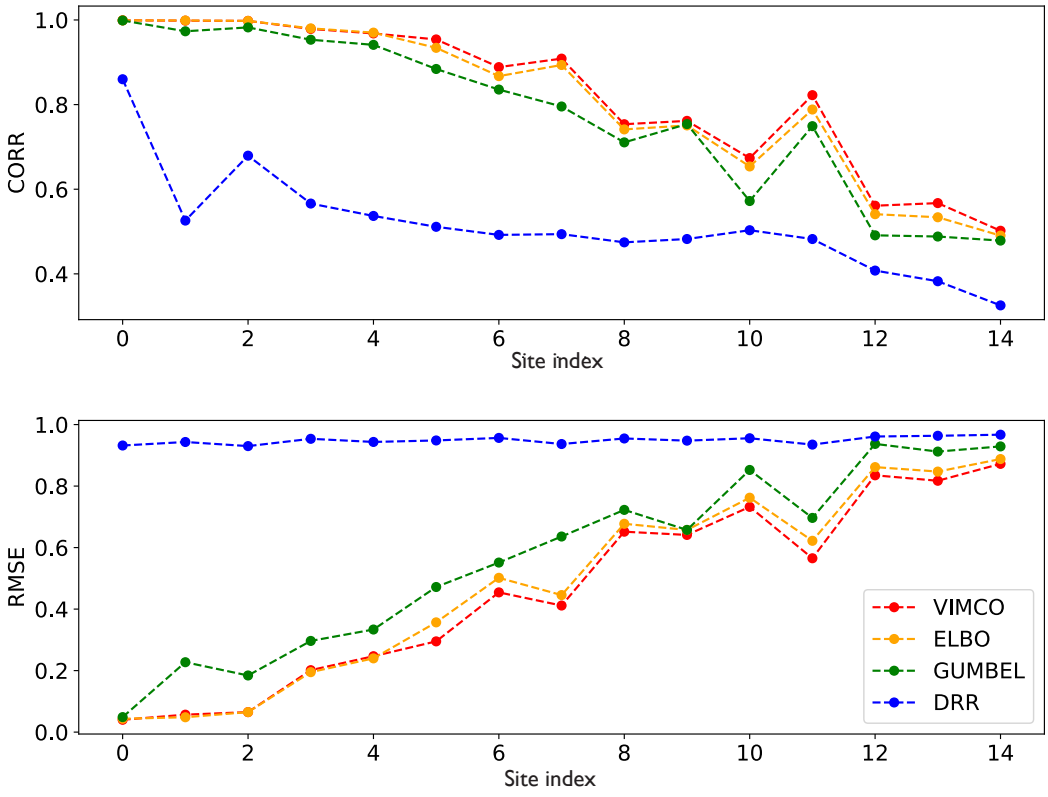
Figure 3.2: Low firing rate, good SNR cell: Inference and reconstruction using different algorithms. The inference is robust to large noise since we have a shared structure in the spine network. Also, the marginal distribution reflects the quality of the data, and the uncertainty in the estimation.



how well we are matching the firing rate of the cell, and it is the lower, the better, bounded by 0. Our methods show excellent results, and as the problem becomes harder (has more noise), the performance degrades. This figure shows the model's performance as a function of noise.

In Fig. 3.4, and Fig. 3.5, we show the correlated matrix recovered by different methods as compared to the ground truth. The sites are arranged the same way as in Fig. 3.3. The first row of the matrix shows the correlation between soma and spine, *i.e.*, output and input of a cell, and the the other shows the correlation between the spines, *i.e.*, input and input correlations. For better

Figure 3.3: Low firing rate, good SNR cell: Correlation coefficient and root mean squared error sitewise with ground truth.



visualization, the diagonal of the matrix is set to zero. For different methods, we also plot an error matrix between inference and ground truth. The left upper corner entries are for the clean traces, and the noise increases going towards the bottom right. Our model trained by VIMCO and ELBO demonstrated superior performance.

To better quantify this result, we plot the input-output correlation, and output-output correlation in a scatter plot as in Fig. 3.6. Different colors show different methods. We fitted the points with a line with slope and intercept for each method, and they are listed in the figure.

Another advantage of our model is that after training, not only have we inferred spike distribution, we also have estimated the generative model. This can be used by the biologist to quickly validate the model and interpret the data. In Fig. 3.7, we showed the estimated generative parameters with respect to the ground truth generative parameters. Note that the estimation came very close to the real values.

Figure 3.4: Low firing rate, good SNR cell: Correlation Matrix: Ground Truth, VIMCO, ELBO

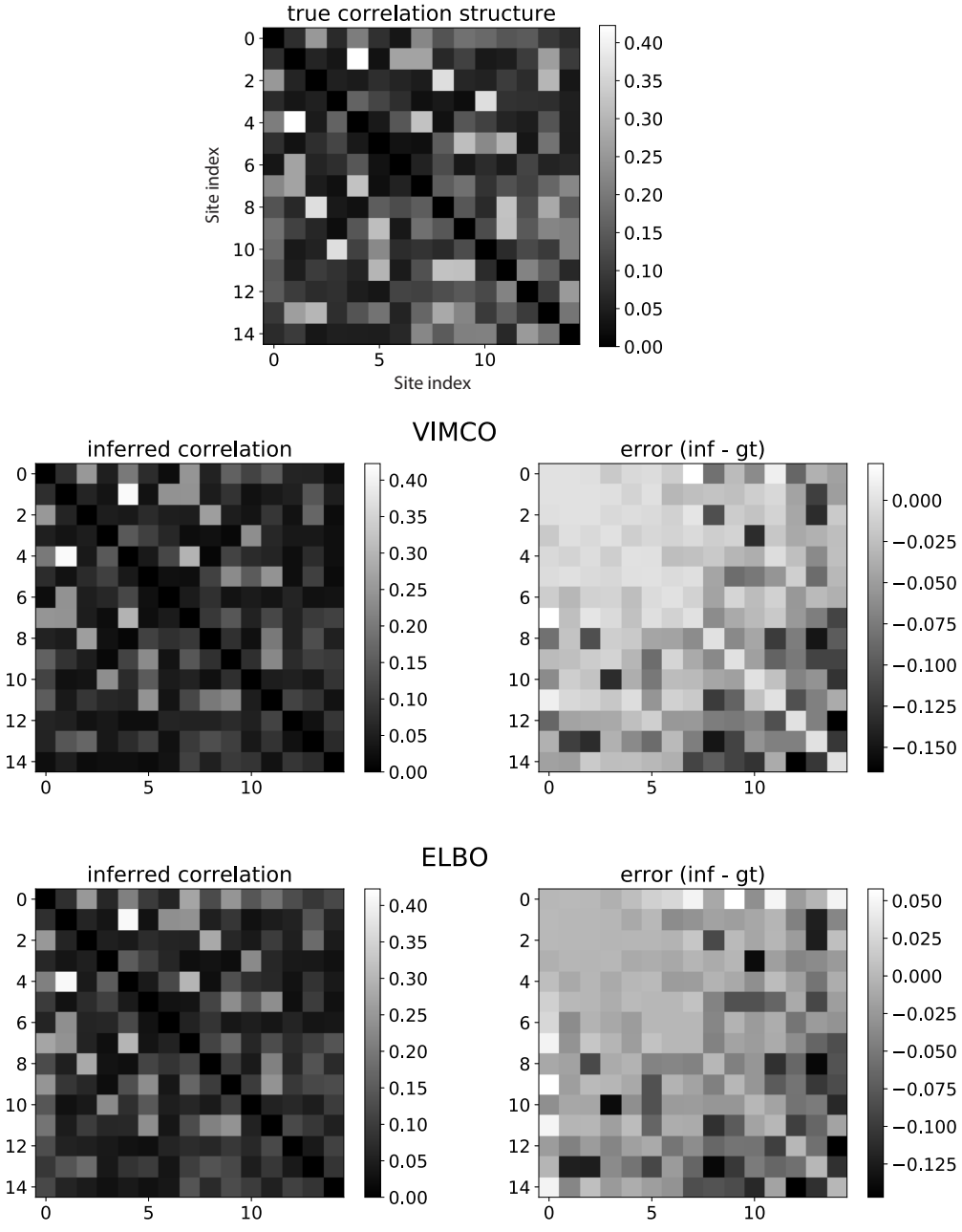


Figure 3.5: Low firing rate, good SNR cell: Correlation Matrix: GUMBEL, DRR

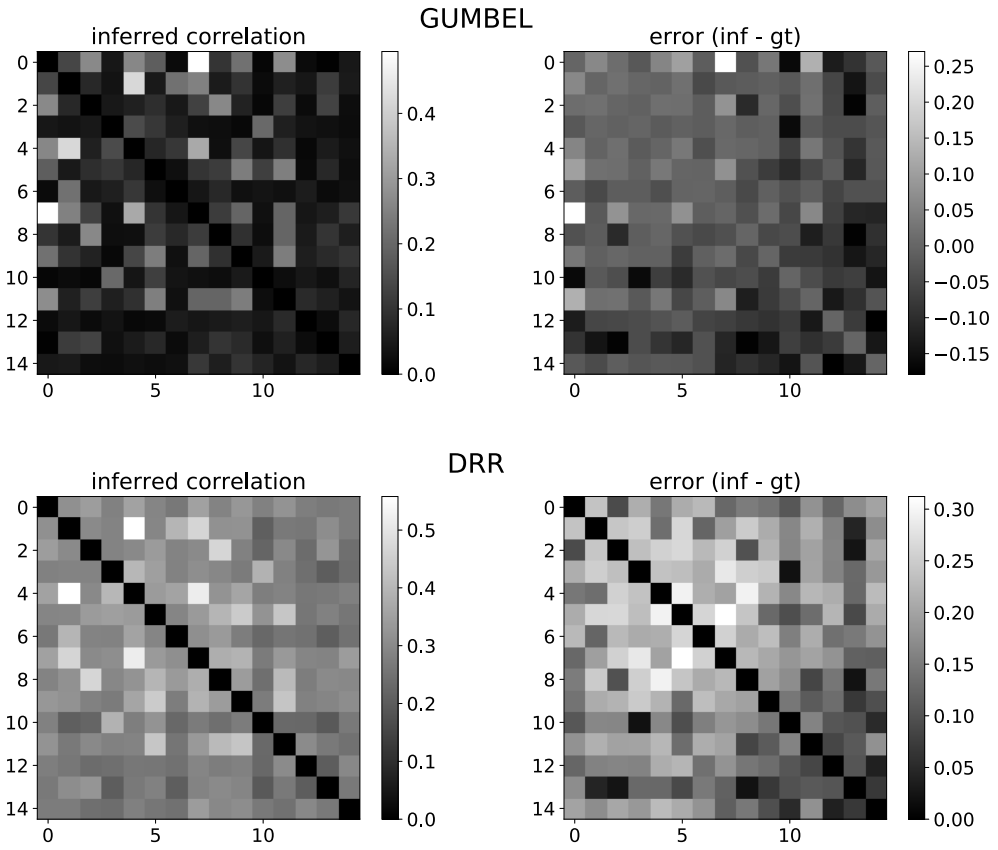


Figure 3.6: Low firing rate, good SNR cell: Soma-Spine (Input-Output) Correlation and Spine-Spine (Input-Input) Correlation. The two parameters associated for each method are the slope and intercept of the line fitted to the scatter points.

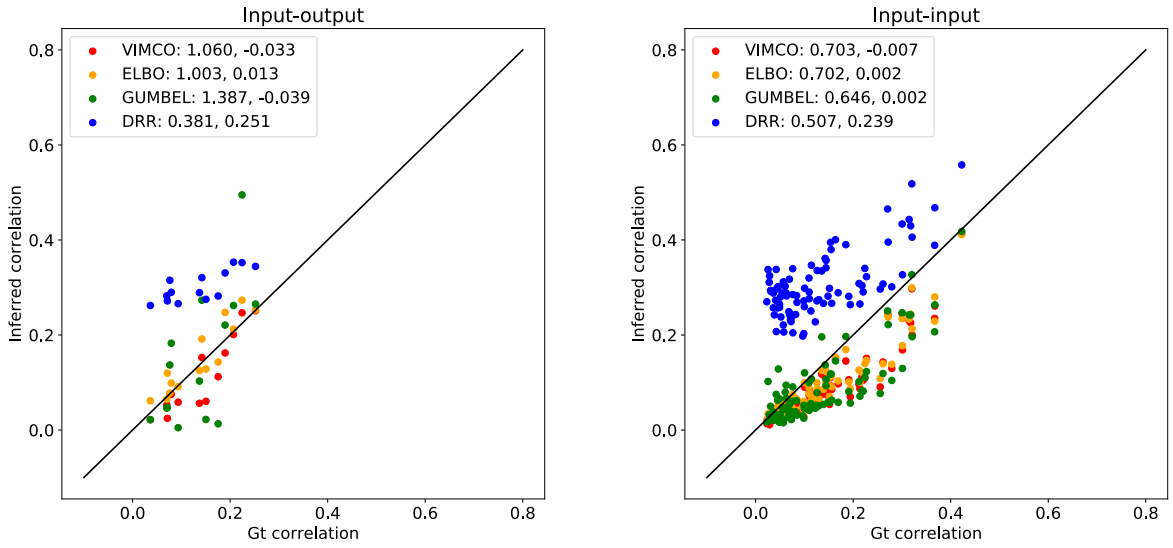
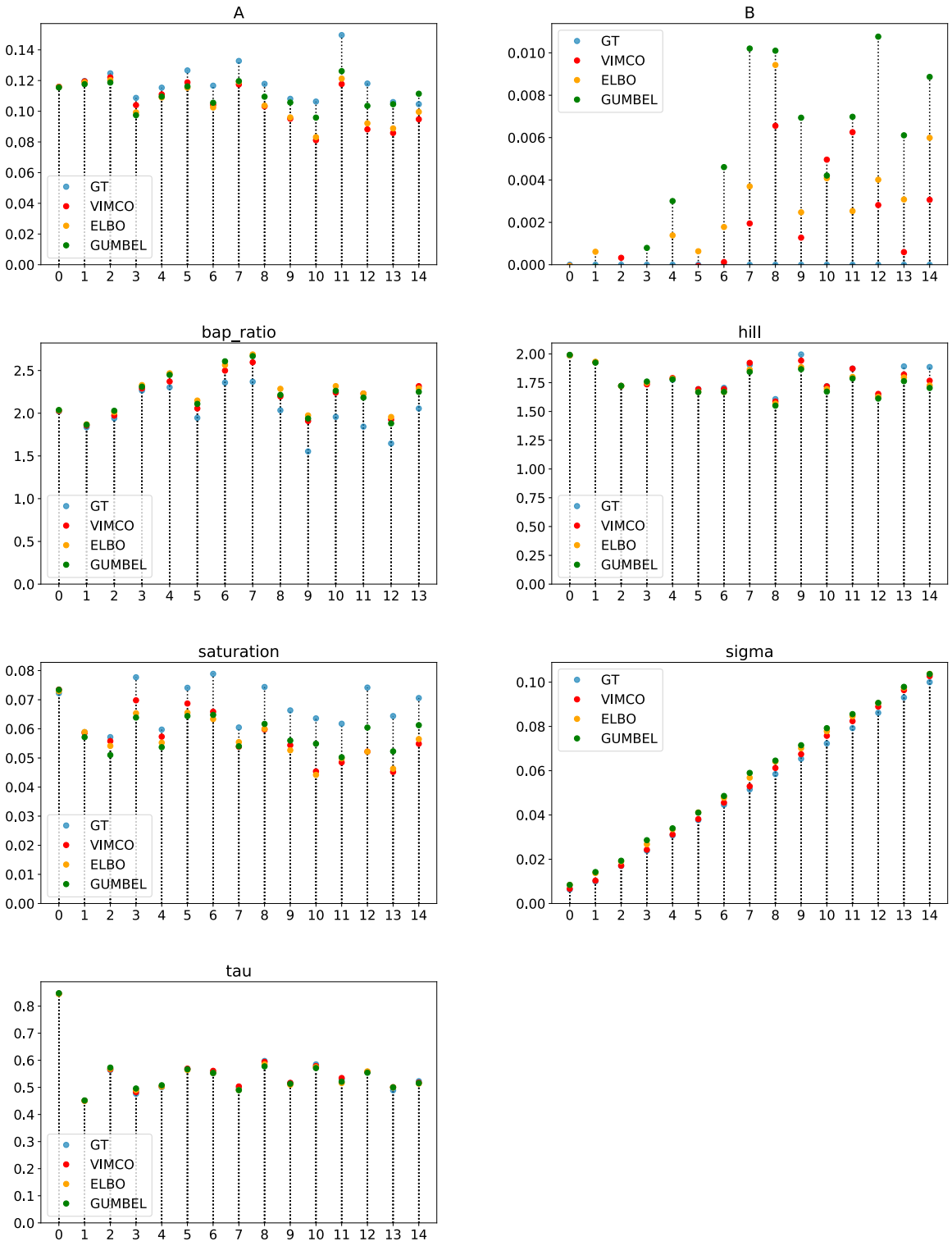


Figure 3.7: Low firing rate, good SNR cell: Generative Parameters.



3.2 High Firing Rate Cell with Moderate Noise

In the second experiment, we simulated a cell with a high firing rate, about 1Hz. The data is generated using the same settings as in the first experiment. Fig. 3.8 to Fig. 3.14 show the fitting results. There are more nonlinear effects when the firing rate of the cell is high. Our method has demonstrated very good results even on high firing rate cells.

Figure 3.8: High firing rate, good SNR: Inference and reconstruction using different algorithms on synthetic data.

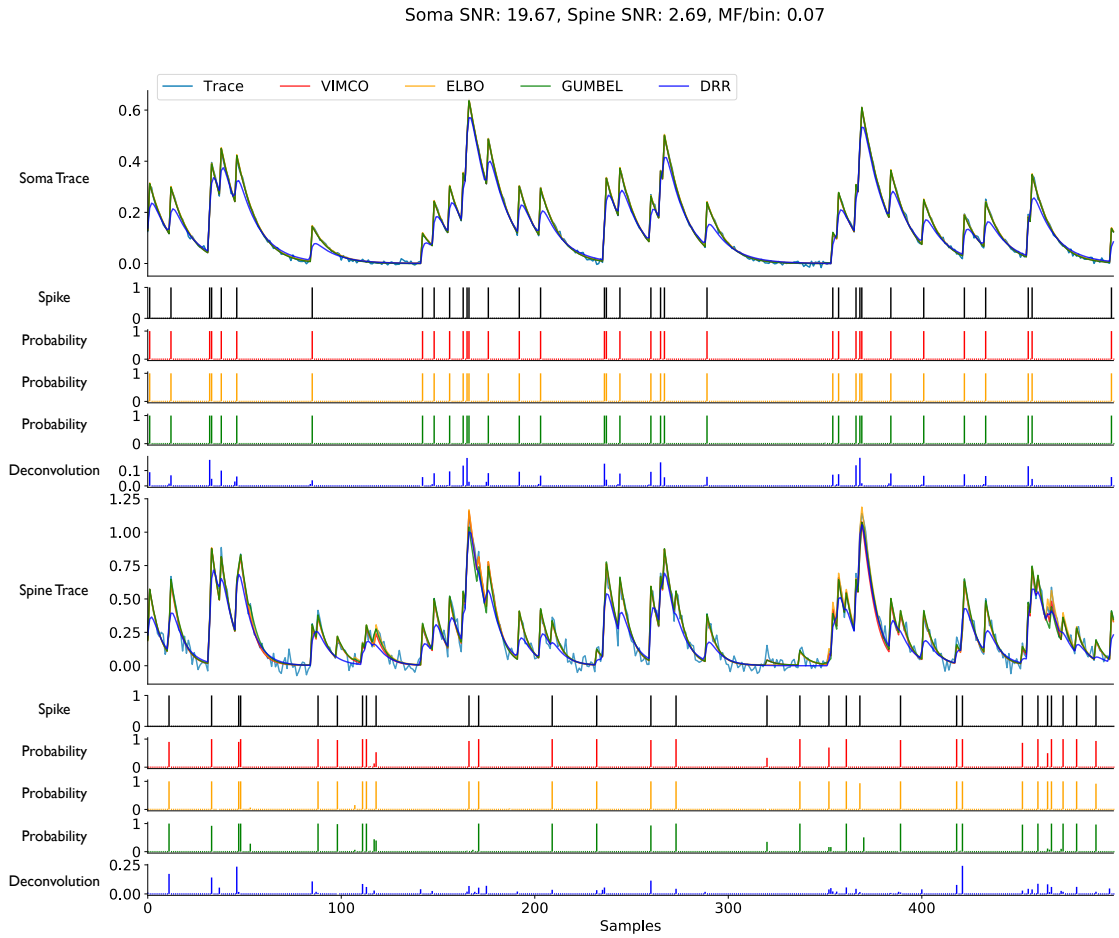


Figure 3.9: High firing rate, moderate noise: Inference and reconstruction using different algorithms on synthetic data.

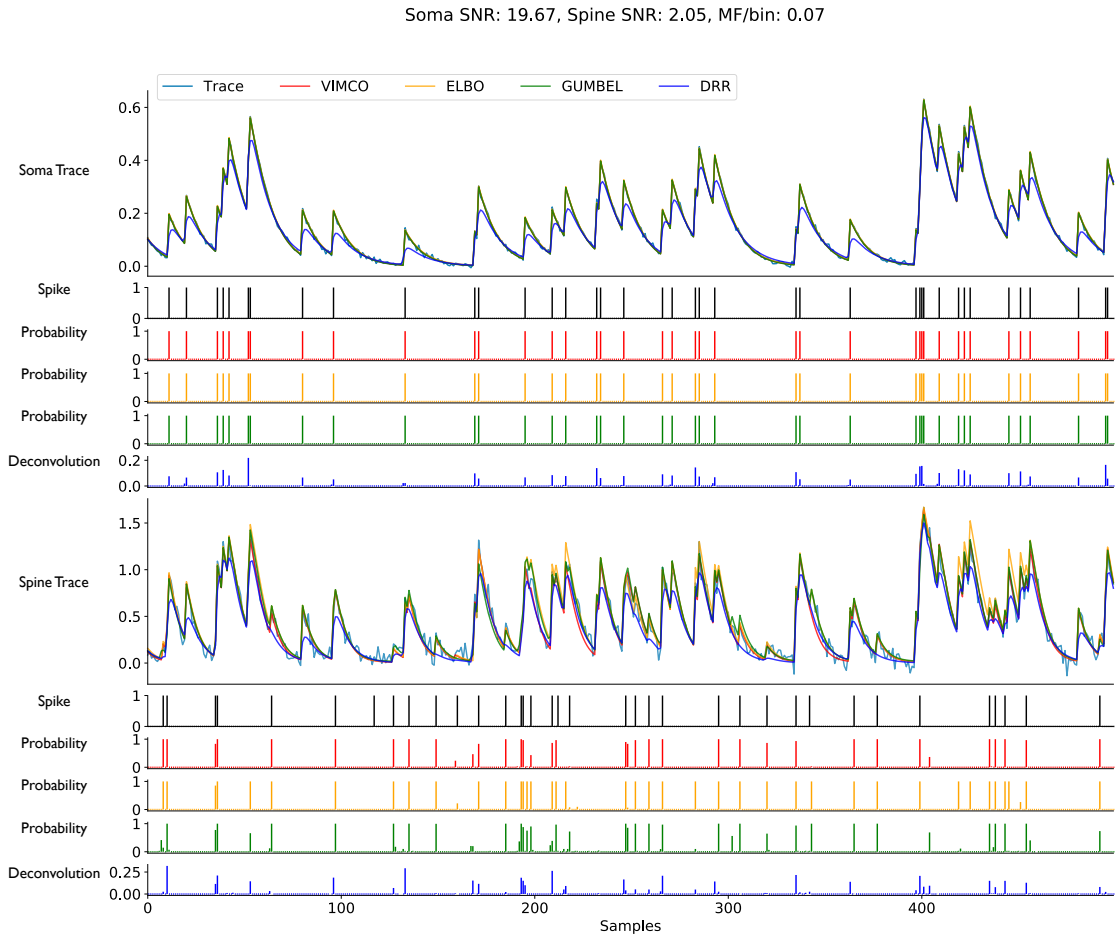


Figure 3.10: High firing rate, moderate noise: : Correlation coefficient and Root mean squared error

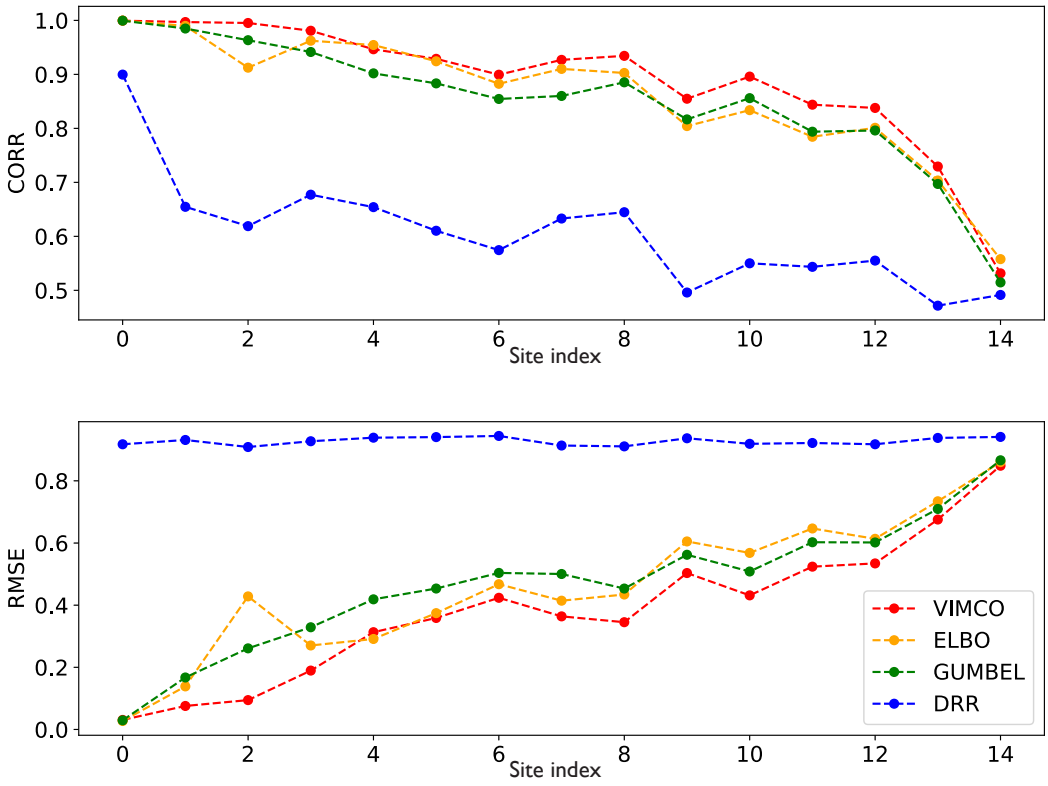


Figure 3.11: High firing rate, moderate noise: Correlation Matrix: Ground Truth, VIMCO, ELBO

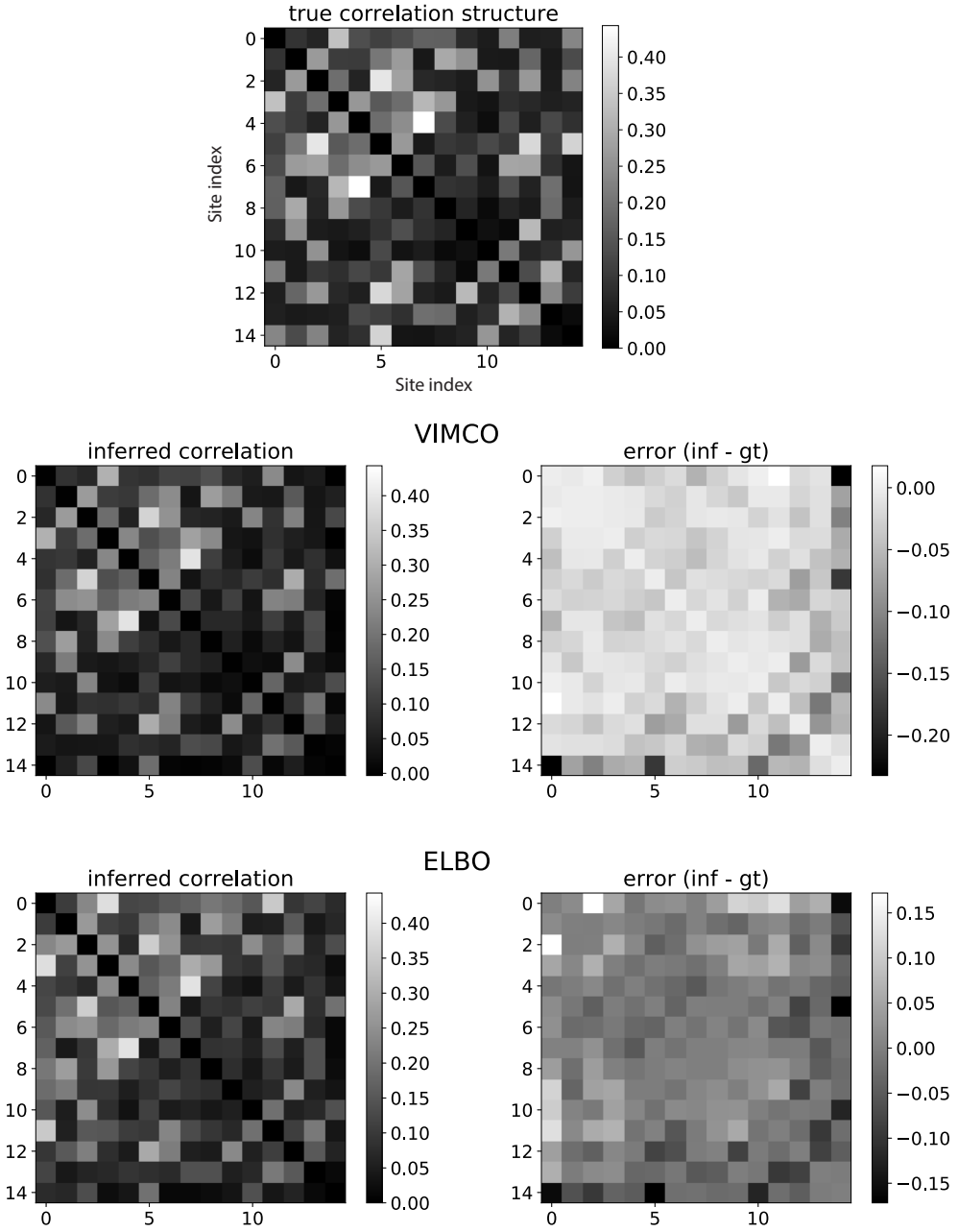


Figure 3.12: High firing rate, moderate noise: Correlation Matrix: GUMBEL, DRR

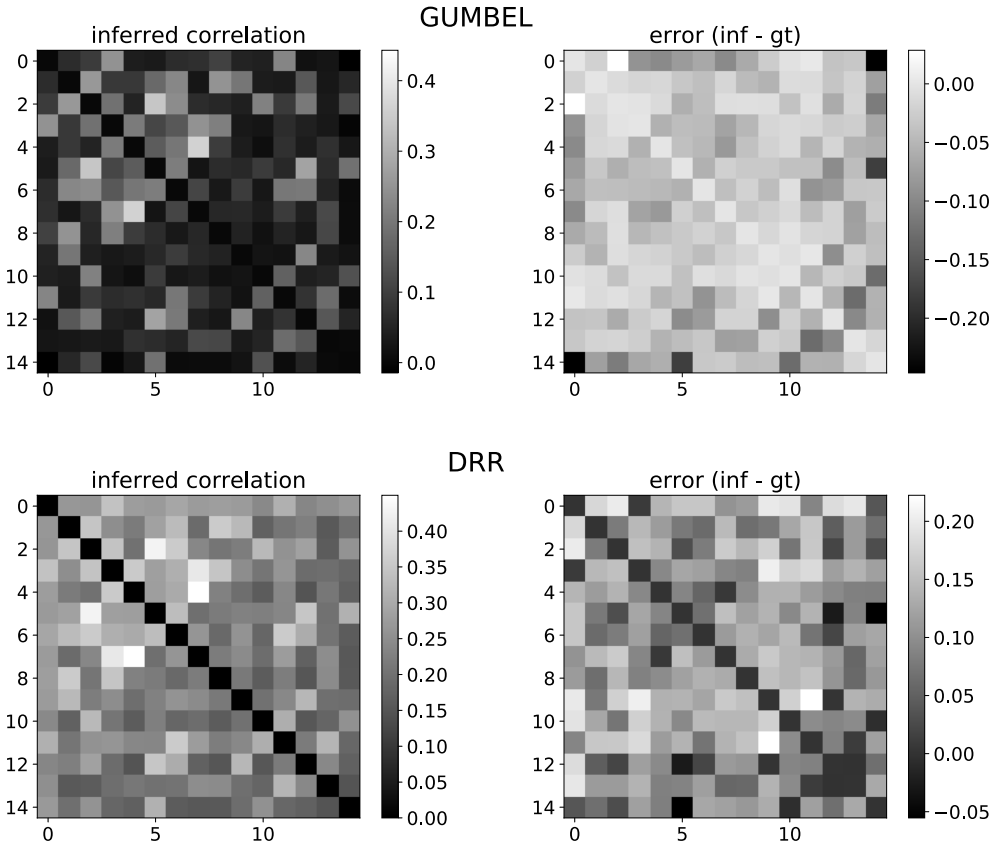


Figure 3.13: High firing rate, moderate noise: Soma-Spine (Input-Output) Correlation and Spine-Spine (Input-Input) Correlation.

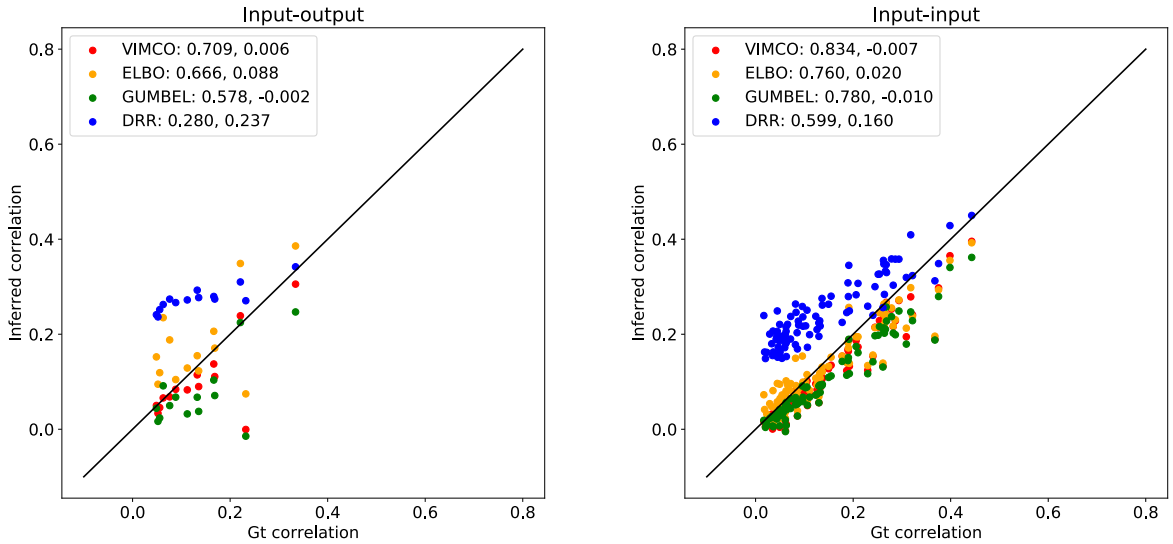
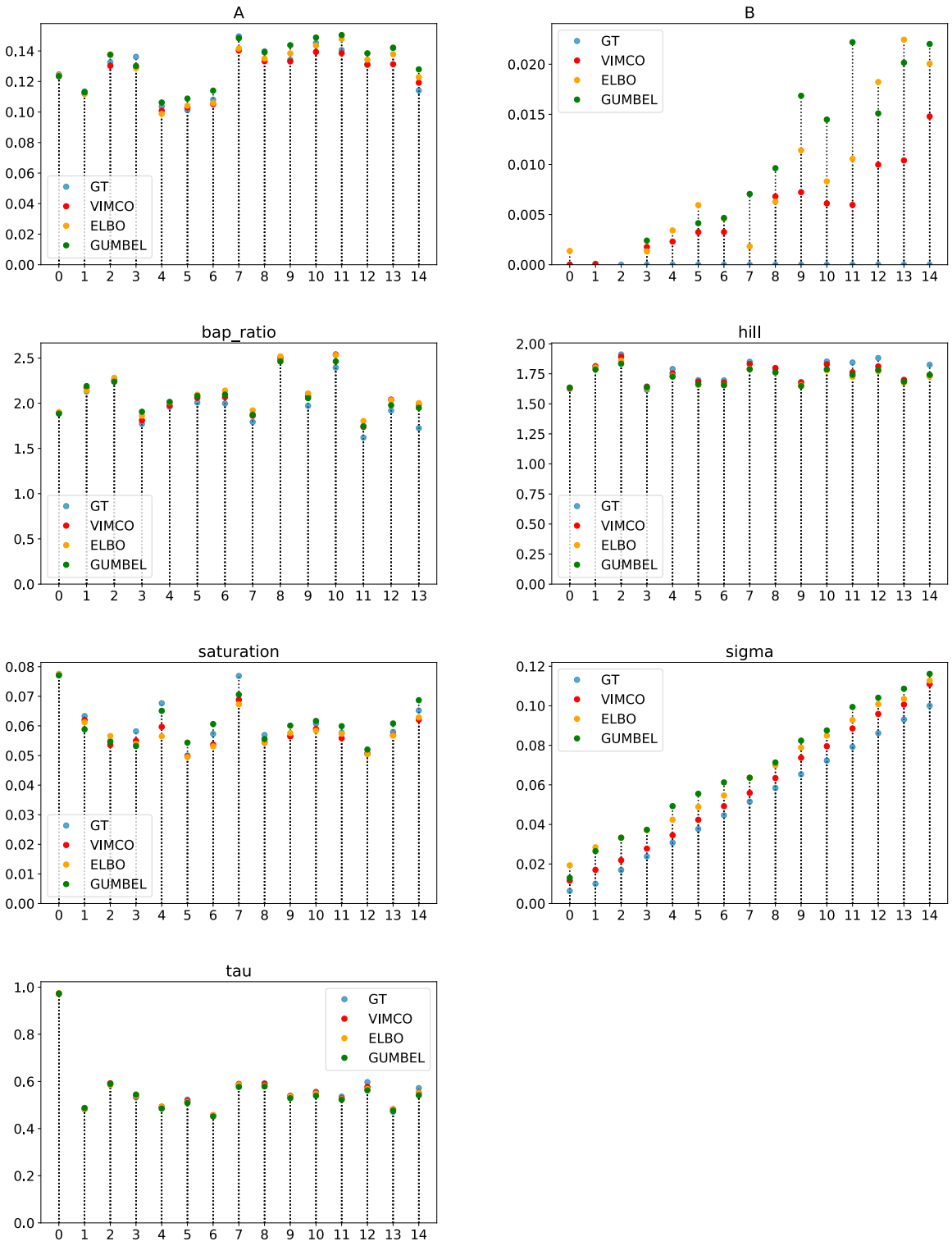


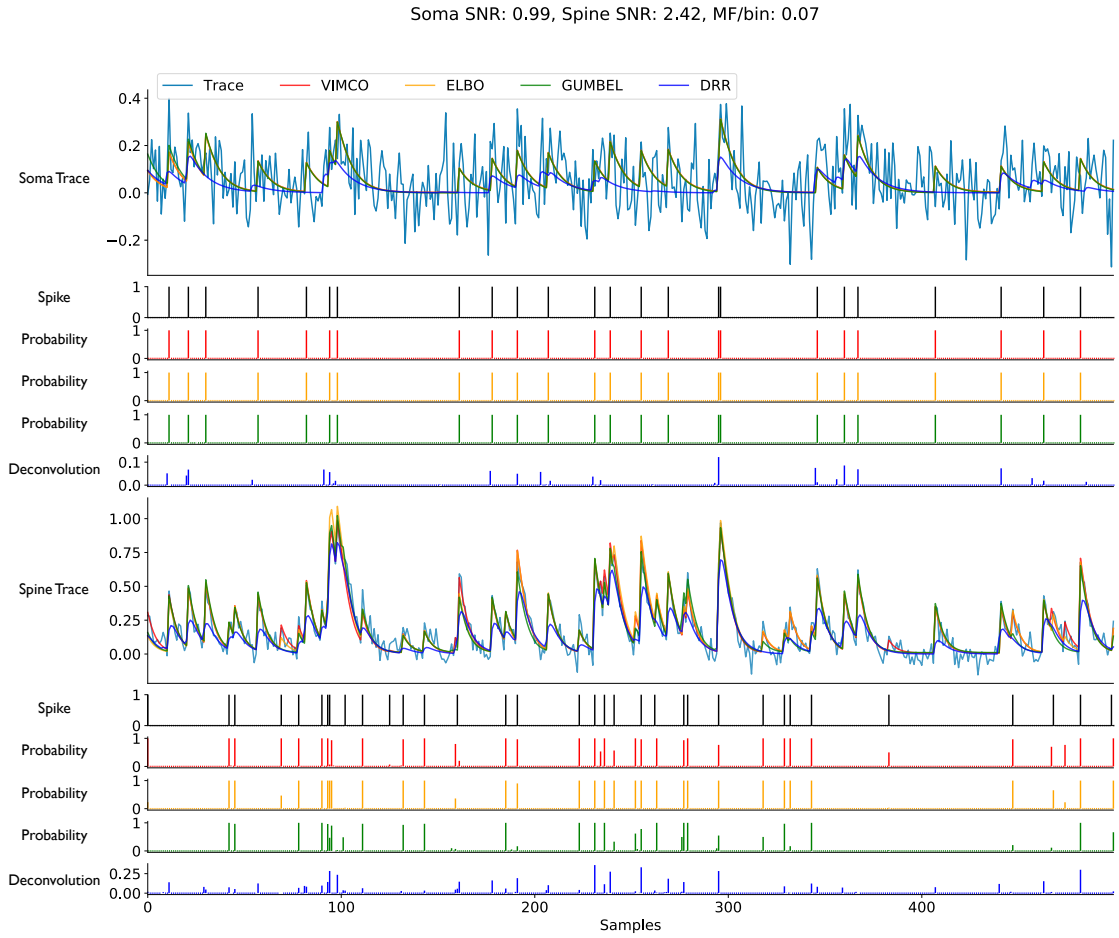
Figure 3.14: High firing rate, moderate noise: Generative Parameters.



3.3 High Firing Rate Cell with Large Noise

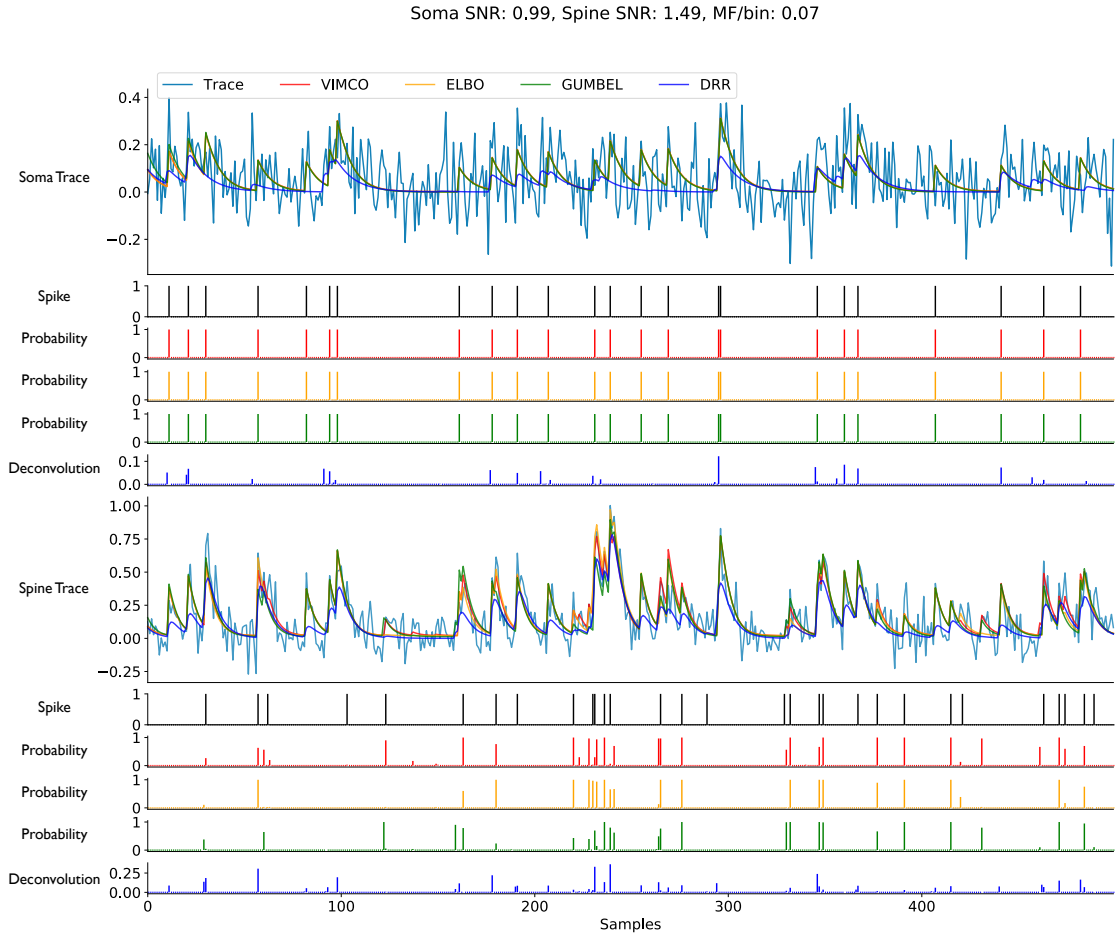
Finally, we tested our model on a cell with large noises at cell body, SNR around 1. At the same time, we have increased the level of the noise at spines: SNR from 5 to 0.80. Fig. 3.15 to Fig. 3.21 show the fitting results.

Figure 3.15: High firing rate, large noise: Inference and reconstruction using different algorithms on synthetic data.



As we can see from the figures, the model has no problem inferring the spikes at soma. This feature is due to the fact that we designed our soma network to utilize all the data, including backpropogated action potential at spines. At the same time, we can see that the inference for the spines is also quite robust to the noise. This is due to the fact that when we fit the model, noises are

Figure 3.16: High firing rate, large noise: Inference and reconstruction using different algorithms on synthetic data.



also leaned from the data. The objective is thus weighted towards less noisy spines, which helps the network to extract information from good data. The advantages of being robust to noise, and automatically adapt to data makes our model suited for use on real data.

Figure 3.17: High firing rate, large noise: : Correlation coefficient and Root mean squared error

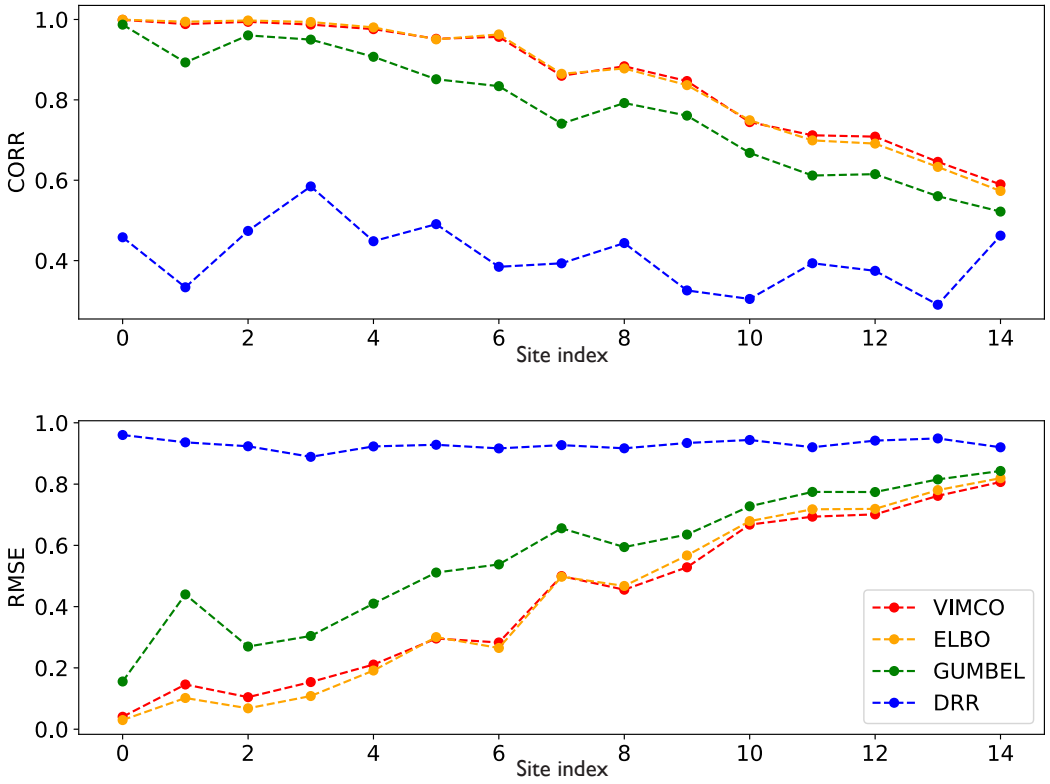


Figure 3.18: High firing rate, large noise: Correlation Matrix: Ground Truth, VIMCO, ELBO

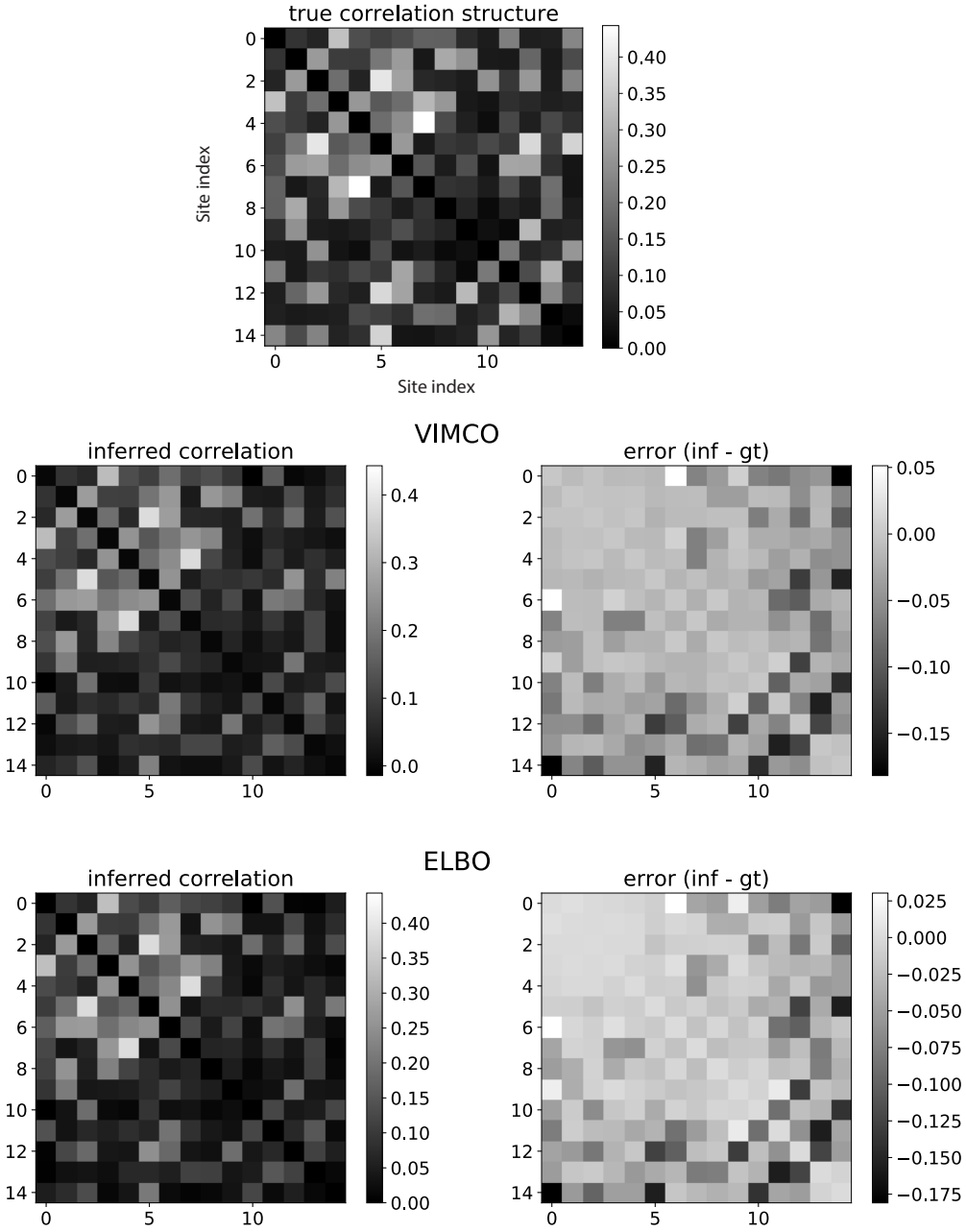


Figure 3.19: High firing rate, large noise: Correlation Matrix: GUMBEL, DRR

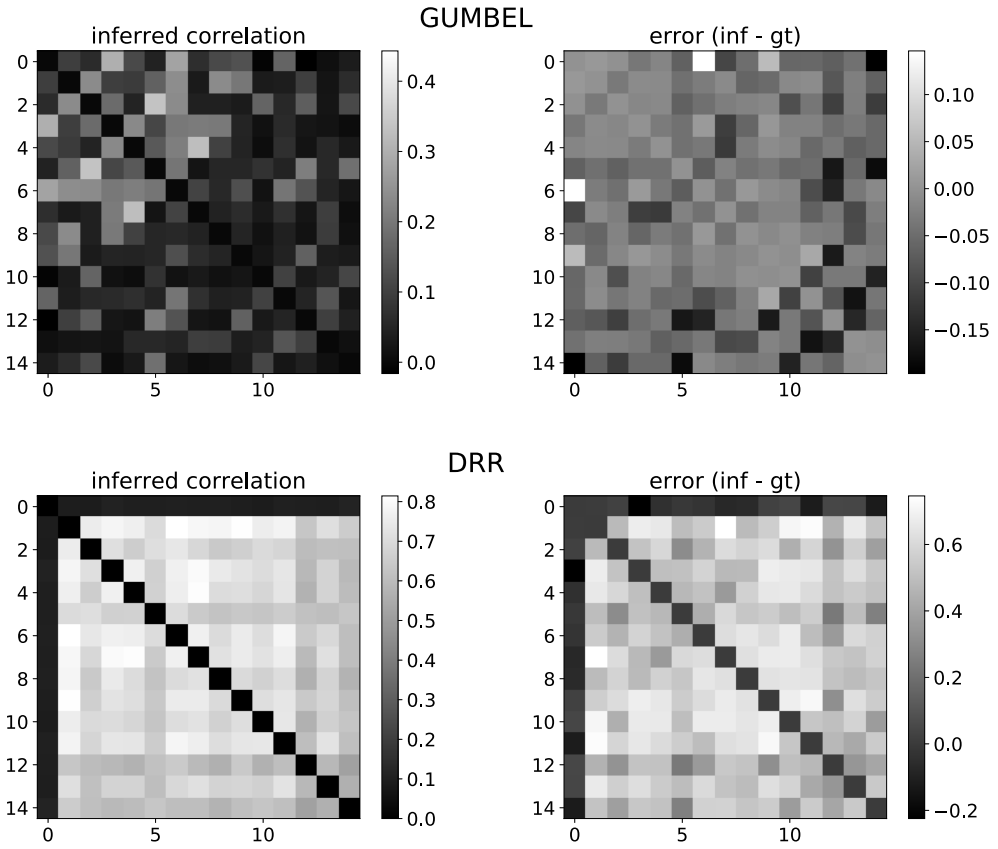


Figure 3.20: High firing rate, large noise: Soma-Spine Correlation and Spine-Spine Correlation.

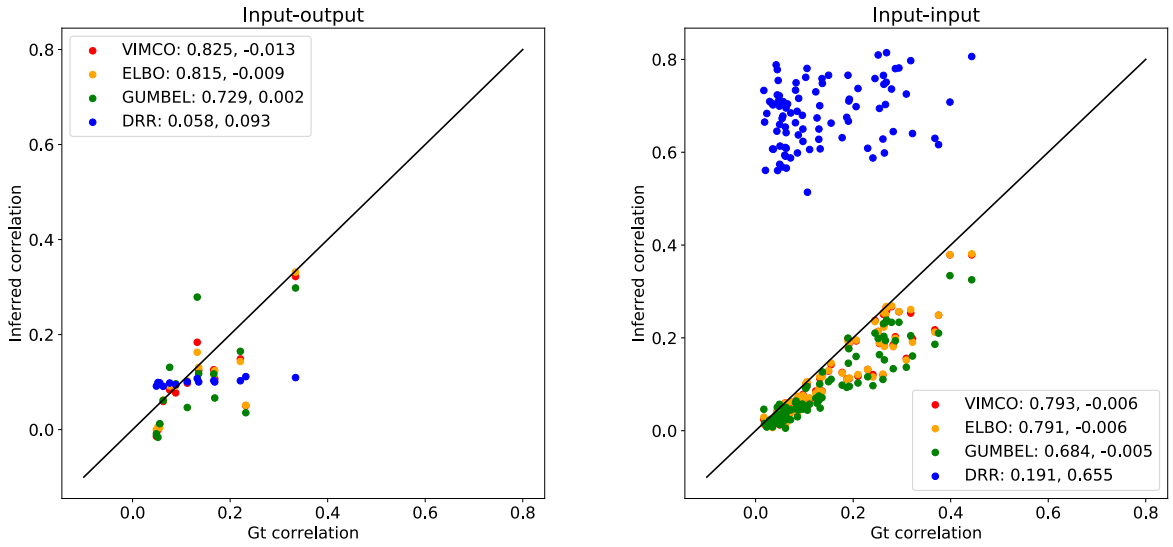
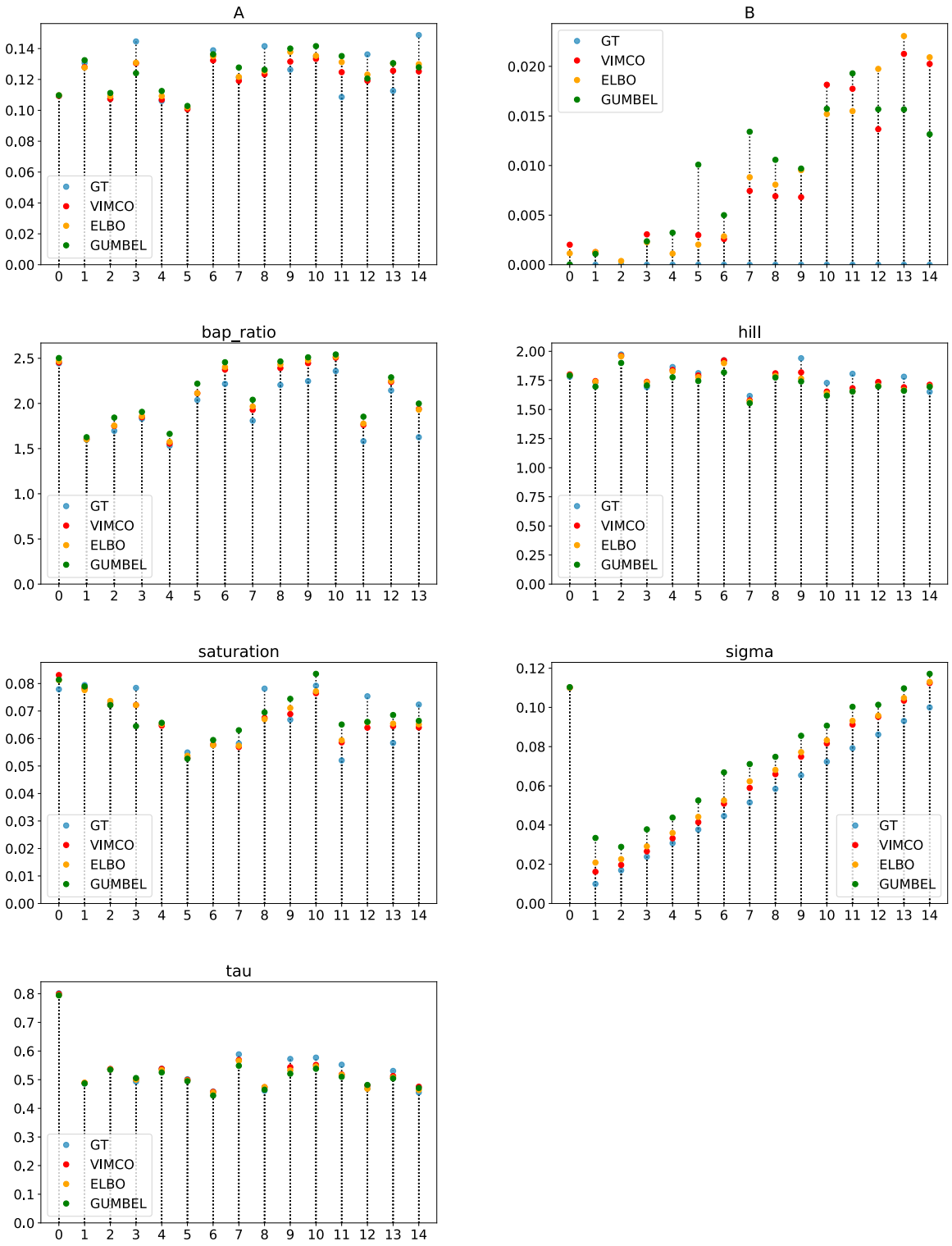


Figure 3.21: High firing rate, large noise: Generative Parameters.



3.4 Real Data

Methods [Kerlin et al., 2017]: Calcium signals were imaged in the dendrites of neurons in the frontal cortex of mice performing a tactile delayed-response task [Guo et al., 2014]. Mouse lines with sparse expression in L2/3 of Cre-recombinase [Gerfen et al., 2013] were crossed with a reporter line (Ai93) expressing the calcium indicator GCaMP6f. Images were collected using a two-photon microscope that allows rapid (approximately 15 Hz) imaging of the soma and up to 300 um of contiguous dendrite, while resolving calcium transients in up to 150 individual dendritic spines. Iterative non-rigid registration was used to correct recordings for motion in three dimensions. Baseline fluorescence was estimated from the median of a 120 second moving window, and then removed from traces. Recording sessions were 40 min to 90 min in duration, which render the discrete data length of 36000 to 81000.

Here we show the fitting results on one dataset that has a data length of 60,000. The input to our algorithm is the fluorescence traces of a matrix of size [60,000, 15], selected 14 spines with soma trace. Fig. 3.22 and Fig. 3.23 show the inferred spikes and reconstructions for the soma and two spines. In Fig. 3.22, the first two rows are the trace, reconstruction and inferred spikes for the soma. The second two rows are the trace, reconstruction and inferred spikes for one spine. The third two rows are for another spine. As we can see from the figure, the algorithm successfully removed back-propagated action potentials from the spine values. The two spines are more active in Fig. 3.23. There are many local events. Our algorithm successfully identified individual input events for these two spines. The inferred spike probabilities are important for biologists for a variety of downstream analyses, for instance, calculating the tuning curve of the cell, understanding cell input-output transformation, for instance.

Figure 3.22: Real Data 1: The first two rows are the trace, reconstruction and inferred spikes for the soma. The second and third two rows are the trace, reconstruction and inferred spikes for two different spines. This plot shows that our algorithm successfully removed back-propagated action potentials.

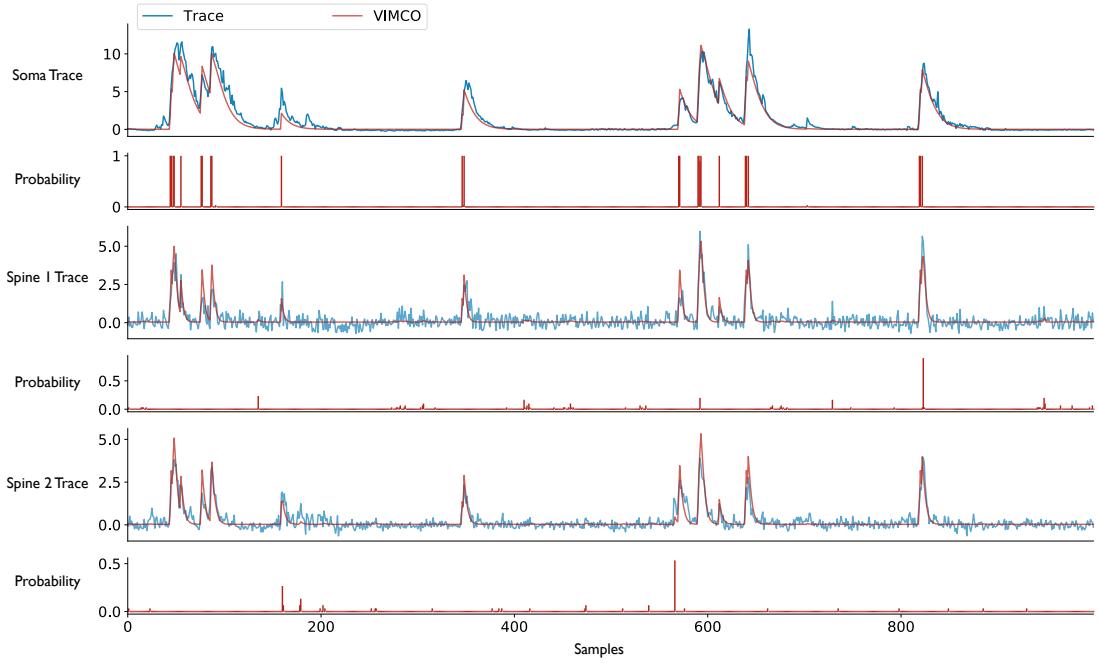
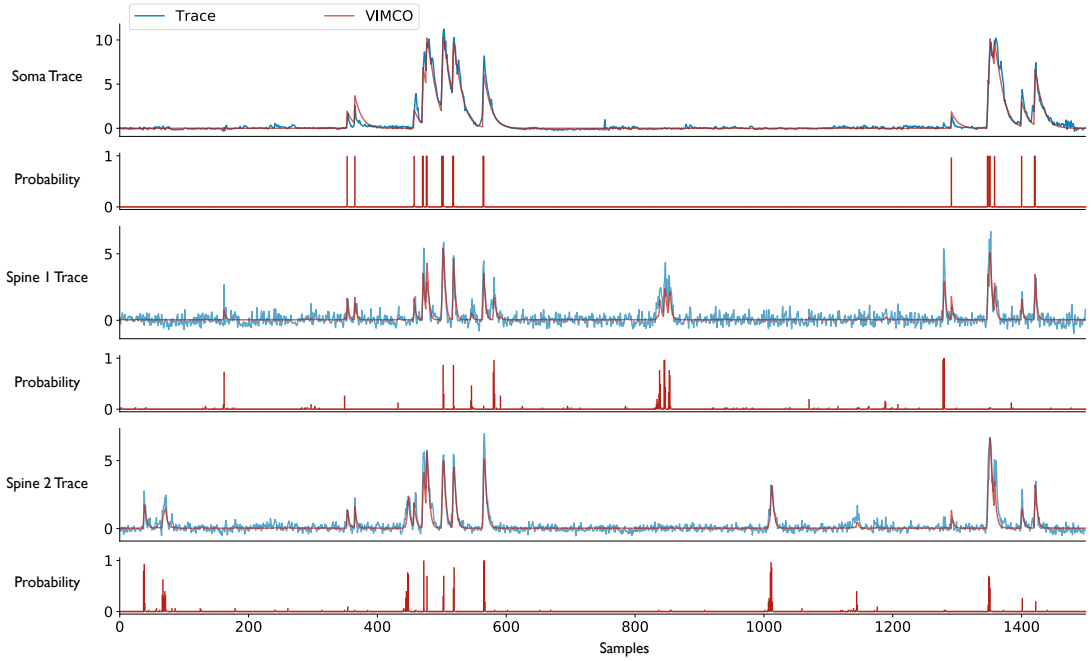


Figure 3.23: Real Data 2: The first two rows are the trace, reconstruction and inferred spikes for soma. The second and third two rows are the trace, reconstruction and inferred spikes for two different spines. The two spines receive many independent events. As we can see, our algorithm identifies independent inputs.



CHAPTER 4

CONCLUSIONS FOR PART I

Advance in imaging technologies and genetic tools has enabled new research in the field of neuroscience. Spiking activity in neurons leads to changes in intra-cellular calcium concentration which can be measured by fluorescence microscopy of genetically encoded calcium indicators such as GCaMP6f. This technology has become popular among neuroscientists since it allows high-resolution measurement of large neural populations. Calcium imaging can also be used to study neural activity at a subcellular resolution. Specifically, we have come upon the important question of inferring pre- and post-synaptic activities from simultaneous somaic and dendritic imaging.

To analyze this dataset, first, spike inference algorithms must be used to infer the underlying neural spiking activity from measured fluorescence dynamics. Second, unique to dendritic imaging, we need to demix calcium transients caused by the pre-synaptic activities from calcium transients caused by back-propagation action potentials. Off-the-shelf algorithms cannot solve such a complicated problem. Besides the biological importance, this dataset also provides us with a playground for testing and developing new machine learning algorithms of practical use.

Specifically, we developed a generative-model-based method for inferring spike trains from fluorescence observations. However, our framework is not only limited to this particular problem. Generally, it can be applied for inference problems associated with any stochastic, discrete, feed-forward state-space dynamical systems. We employ a variational autoencoder (VAE) with stochastic optimization as our computational framework. VAE consists of three parts: generative model, recognition model, and the inference algorithm. We have developed three generative models that are based on biophysical properties of the calcium indicators. In the recognition model, we employ a deep convolutional neural network to perform efficient and fast inference. The inference for discrete latent variables is usually more difficult and less studied compared to continuous latents. We benchmarked our model using three different inference algorithms for discrete latent variables, among which VIMCO and the GUMBEL-Softmax trick with multi-sample objectives shows better

results. We simulated synthetic data of cells with various firing rates and noise levels. Our model has shown superior performance compared to off-the-shelf methods in various metrics. Finally, we applied our model to some real data.

The advantages of our methods are multifold. The result is robust to noise due to our careful design of architecture in the recognition model. The recognition model has two networks, one for the soma, and one for spines. In the somatic network, we input not only the measurement from the soma, but also the measurement from different spines. Thus, the network can use back-propagated action potential information to help infer global events. The spine network also learns shared convolution filters, which results in more robust inference. Second, compared to other spike inference algorithms that produce point estimates of continuous-valued spikes, our method uses a proper binary distribution, which is of great help in dissociating back-propagated action potentials in individual events. Moreover, the spike probability also provides us uncertainty estimation, which can be used for downstream analysis. Third, after training, we have not only inferred spike probabilities, but have also estiamated the generative model. The generative model can be used for generating more simulated data resembling real data. More importantly, since the generative model is biophysically interpretable, a biologist can extract meaningful information from it. Fourth, our model is designed to be fast and scalable. Our package is developed using open source packages, and GPU support. Training the model on a big dataset only takes about 1$\sim$2 hours. The testing time is only minutes. Last but not least, our model and inference framework allows quick modular development. The generative model and recognition model can easily be changed, which makes it especially beneficial for prototyping new models.

4.1 Contributions

In summary, my contributions to this research include:

- Developed an inference framework that uses a stochastic variation autoencoder for solving the inference problem of a stochastic, discrete state-space model.
- Applied the framework to the difficult neuroscience problem of inferring pre- and post-

synaptic activity from calcium imaging.

- Developed a biophysical generative model and its high-speed implementation.
- Developed a deep convolutional neural network-based recognition model for fast and efficient inference.
- Benchmarked three different inference algorithms for discrete latent random variables on extensive synthetic data that show the behavior and characteristics of the algorithms in different scenarios.
- Applied the algorithms to real data and showed promising results.
- Developed a GPU-based software package for the neuroscience community for analyzing dendritic imaging data.

4.2 Limitations of the Approach

Despite many advantages, there are a few limitations of our current approach:

- VI uses a approximate posterior distribution. The power of the inference is thus confined by the flexibility of the approximation. In our method, we used factorized approximate distribution which could limit the performance of the results. Designing fast and richer posterior distribution is an ongoing research, and progress has been made by papers [Oord et al., 2017, Kingma et al., 2016, Rezende and Mohamed, 2015]. We plan to investigate it in the future.
- Like any other gradient-based algorithms, VAE suffers from the problem of local minimum. We employ the common practice in Deep Learning by running experiments with different initialization, and select the best-fit one. It is advised to aslo initialize the model properly so that the algorithm could find the correct modes. In our experiments, we found that it is enough to initialize the generative model with eyeballed estimation and recognition model with rough mean firing rate of spikes (achieved by adjusting the bias of the output layer).

4.3 Future Work

Many interesting points of research can be followed. The future work of this research will be:

- Modify the model to super-resolve the somatic spikes. Different parts of dendrites are imaged sequentially according to their distance from the soma, and this time sequence of data is available. Using this information, we can super-resolve the somatic spikes, and thus achieve better resolution.

- Develop a more biophysically detailed generative model. Dendrites receive massive synaptic inputs from upstream neurons and play an important role in single neuron input-output transformation. They perform local computation and deliver the result to the rest of the neuron through not only a passive/linear cable but also through regenerative/nonlinear mechanisms. By modeling more detailed biophysics of dendritic spikes, we can account for more complicated phenomena in the data.

- Develop richer posterior distributions, for instance, time-wise non-factorized distributions. For fast parallel sampling, we are currently assuming independence among time dimension of the spikes. In reality, there are dependencies between different time bins. Recent papers [Oord et al., 2017, Kingma et al., 2016, Rezende and Mohamed, 2015] have proposed more efficient ways to work with conditional distributions. Increasing the expressiveness of the posterior distribution can make the inference more accurate.

- Develop a new inference algorithm and training procedure. We have tested our algorithm in the super-resolution scenario. However, due to the mode-selection behavior of variational inference, we have some difficulties in fitting the model. This calls for a new procedure for training the model, which we are currently developing.

- Fit more real data with different generative models and characterize in more detail the models' behaviors.

PART 2

Nonlinear System Identification Using a Set-Theoretic Evolutionary Approach

CHAPTER 5

INTRODUCTION

At its core, ***signal processing*** (SP) is not a new discipline. Most of our roughly half-century-long profession is built on a solid foundation of linear mathematics and models that were researched, tested, and refined for several centuries before anyone conceived of an FFT, or a digital model expressed as a computer program. Yet, SP engineers have been brilliantly insightful in shaping, out of this bedrock of mathematics, theories, products, and services that have exploited and synergistically advanced the state of modern networks and digital devices. A rich set of theories and methods based on ***linear, time-invariant*** (LTI) models is now familiar to the SP practitioner and it is these LTI models that have largely supported the spectacular technological change we have witnessed over a few short decades.

LTI model-identification techniques can generally be classified as parametric or nonparametric methods. In the more classical nonparametric approaches, the superposition and homogeneity properties of linear systems are used to characterize the impulse response or equivalent and frequency response function, which is then quantified using correlation or Fourier-domain techniques [Ljung, 1999,Pintelon and Schoukens, 2012]. Nonparametric approaches reflect their origins in continuous-time theory. Interest in parametric models was a natural consequence of the development of the modern computer, which made possible the quick derivation of finite sets of system-characterizing parameters using discrete operations on sampled signals.

Contemporary linear system models represent a blend of seminal efforts arising in mathematics and statistics, systems and control engineering, and signal processing. The structures are often known as time series models, a name used by early developers of the methods prior to the era of modern SP, when the models were primarily viewed as structures to explain the spectrum of random process realizations (e.g., [Kolmogorov, 1941, Box and Jenkins, 1970]). With the advent of laboratory-scale computing machines in the 1970s, time series models began to receive significant attention following the publication of the widely-read text by Box and Jenkins [Box and Jenkins,

1970]. Economists were among the first academicians to have employed these models extensively in their research (e.g., [Hayashi, 2000]), while the nascent field of SP found applications of the methods in speech processing and spectrum estimation [Atal and Schroeder, 1970, Atal and Hanauer, 1971, Burg, 1975, Makhoul, 1975]. Even though the early SP work was principally concerned with linear prediction – hence with a blind input type of model equivalent to the time-series AR model (see below) – the SP applications represent a subtle shift in the view of the model to one of a *system*, rather than strictly as a model of process (signal) generation. The name AR model was not initially employed by the SP community, but, over the decades, time-series models names have been adopted by many signal and systems engineering researchers to refer to variations of canonical linear constant-coefficient difference equations models with, and sometimes without, stochastic signal components. Currently, the autoregressive moving average with exogenous input (ARMAX) model, including the special cases of autoregressive (AR), autoregressive moving average (ARMA), and autoregressive with exogenous input (ARX) models, are the most commonly used representations for linear system identification.

However, the 21$^{\text{st}}$ century SP engineer, is increasingly likely to encounter system analysis and design problems in which LTI models are just not sufficient. Biologically-motivated solutions are but one extremely compelling current example of this trend. Nonlinear and / or time-varying models are difficult platforms around which to design, analyze, and compute solutions. Arduous research over many decades (again based in classical mathematics) has led to a sufficient body of theoretical and applied knowledge in nonlinear systems to support graduate course offerings at many universities (e.g., [Khalil, 2002]). *Ad hoc* applications of nonlinear systems occupy a small but increasing number of pages in the scholarly literature, an SP-based example being the re-emergence of neural networks of massive scale in deep learning of the structure of speech (e.g., [Hinton et al., 2012]). This progress notwithstanding, the vast class of nonlinear models – including every variation that is not LTI – is relatively poorly understood in contrast to the treasury of theoretical and practical LTI knowledge available to the SP practitioner. One approach to accounting for non-LTI system properties, without abandoning the rich SP toolkit for LTI systems, is to employ models

that are ***LTI in parameters*** (LTIiP). For some SP endeavors, nonlinear models that are LTIiP are useful and only marginally more difficult to work with than systems with purely LTI properties. It is upon this bridge class of nonlinear systems that we shall focus in this paper. In particular, the focus of this paper is on the identification of nonlinear parametric models that are LTIiP.

This research introduces a novel evolutionary identification algorithm for LTIiP model identification. An LTIiP-based identification procedure must select sparse and effective model terms (regressor signals) from among a possibly vast number of nonlinear regressors, and estimate the parameters from inputs and output observations in the presence of noise that is generally correlated or has nonlinear dependencies. Towards this end, we develop a biologically-motivated identification framework for both the selection of the correct regressors and estimation of the model parameters. The strategy blends traditional system identification methods with three modeling strategies that are not commonly employed in signal processing: LTIiP models, set-based parameter identification, and evolutionary selection of the model structure. This framework simultaneously addresses selection of the model structure and the parameter estimation. Moreover,a very significant advantage of the algorithm is the lack of need for assumptions about stationarity or distributional characteristics of the noise. The ability to identify the correct model with unbiased parameters under complex noise conditions makes the algorithm transformational for practical biomedical data analysis.

This section of the dissertation begins with a review of optimal bounding ellipsoid (OBE) algorithms. Then the nonlinear evolutionary identification framework is derived under the more general setting of LTIiP models. The problem is further formulated as a multi-objective problem for studying the trade-off between conflicting optimization objectives. The performance of the framework is tested on both simulated systems and practical datasets. Furthermore, the evolutionary identification algorithm is applied for identifying nonlinear, effective brain connectivity.

5.1 Methods for parameter estimation

There are various methods for estimating the model parameters, such as ***minimum squared error*** (MMSE), ***maximum likelihood*** (ML) and ***least squared error*** (LSE) [Graupe, 1989, Haykin,

1995, T. Söderström and P. Stoica, 1989]. Well-known recursive methods - the ***recursive least square*** (RLS) [Haykin, 1995], ***least mean square*** (LMS) [Graupe, 1989], ***instrumental variable*** [T. Söderström and P. Stoica, 1989] and ***optimal bounded ellipsoid*** (OBE) [Dasgupta and Huang, 1987b, Deller, Jr. et al., 1994, Deller, Jr. and Odeh, 1993, Fogel and Huang, 1982a] algorithms are techniques useful in on-line applications. One of the most popular methods in engineering applications is the LSE or its recursive counterpart, RLS, for their simple structure with well-understood convergence performance. RLS has been modified to a weighted RLS (WRLS) using a forgetting factor for tracking time-varying parameters. However, LSE and LMS require the whiteness of the model disturbance, and they fail to perform adequately in colored noise [Haykin, 1995].

Set-membership (SM) estimation and filtering have been widely researched and broadly applied, but have received significantly more interest and attention among the systems and control research communities. SM algorithms are unique in providing a set of feasible parameter vectors (a solution set) instead of a single point estimate. This is achieved through successive refinements of an initial solution set, consistent with *a priori* constraints on the signal or system model. Arising from the SM algorithms, ***Optimal bounding ellipsoid*** (OBE) algorithms belong to the class of recursive SM algorithms and iteratively assign a weight to each incoming data vector that reflects the current observation set's potential to refine the solution set [Deller, Jr. et al., 1994]. Each weight is determined by minimizing a measure of the size of a hyperellipsoidal feasibility set to which the "true" parameter vector must belong. OBE algorithms do not impose any statistical requirements on the disturbance, but require that the sequence squared be pointwise bounded by a known sequence [Lin, 1996].

OBE identification algorithms (e.g., [Deller, Jr. et al., 1994, Deller, Jr. et al., 1993, Fogel and Huang, 1982a]) have strong potential for application to signal processing problems. With respect to conventional least-square-error identification methods (e.g., [Haykin, 1995]), OBE identifiers offer superior adaptation, improved accuracy, efficient use of innovation in the data, improved computational efficiency, robustness to measurement noise, robustness to deviation from the assumed input

model, a set of feasible solutions rather than a single point estimate, and the ability to compute the solution recursively in time without block processing or windows (e.g., [Deller, Jr. and Luk, 1989, Deller, Jr. et al., 1993]).

5.2 Review of OBE algorithms

Schweppe published one of the first OBE-type algorithms in early 1968 [Schweppe, 1968] in the context of estimating state parameters of linear dynamic systems using noisy observations. Assuming bounded inputs and bounded observation error, Schweppe's algorithm estimates the state of the system using bounding ellipsoids. However, as Schweppe notes [Schweppe, 1968], this novel algorithm is presented without a convergence proof, processes all available data, and is computationally impractical.

Witsenhausen in 1968 [Witsenhausen, 1968], and Bertsekas and Rhodes in 1971 [Bertsekas and Rhodes, 1971], tackled the state-estimation problem from a SM approach. Under similar assumptions as those of Schweppe, Bertsekas and Rhodes examined filtering, prediction and smoothing problems. The algorithm of Schweppe, and of Berstekas and Rhodes have a Kalman-Bucy filter structure which is optimal as a state estimator in Gaussian white noise.

In 1979, Fogel [Fogel, 1979] published an OBE identification algorithm for the ARX model (e.g., [Ljung and Södertröm, 1983]) based on *a priori* knowledge of the cumulative error energy. Fogel proves the convergence of the hyperellipsoid central estimator to the true parameter in a deterministic setting, by demonstrating that the hyperellipsoid asymptotically reduces to a point set.

In 1982, Fogel and Huang [Fogel and Huang, 1982a] published an OBE algorithm with selective updating (F-H/OBE) which processes only relevant data, a key feature of modern OBE algorithms. This data-selection process is achieved by assigning weights to each incoming data vector, with a zero weight indicating data rejection. A pre-processing information check $O(m^2)$ (m is the number of parameters) determines the possibility of a non-zero weight, thereby potentially eliminating redundant computations. Fogel and Huang present a sufficient condition for the convergence of the

F-H/OBE hyperellipsoid to a point, provided the observation error is white noise. The validity of this proof remains controversial, and a more recent proof in a stochastic setting with a more general class of OBE algorithms is presented in the paper by Nayeri *et al.* [Nayeri et al., 1994].

The F-H/OBE data selection process was improved to $O(m)$ complexity in 1987 by Dasgupta and Huang [Dasgupta and Huang, 1987b] in their OBE-type algorithm (D-H/OBE). By minimizing κ_n, a scalar not apparently related to the hyperellipsoid volume, Dasgupta and Huang derive a simple but effective algorithm and prove the asymptotic convergence (at an exponential rate) of its central estimator to a region around the true parameter.

Deller *et al.* introduced the ***set-membership weighted recursive least squares*** (SM-WRLS) algorithm in 1989 [Deller, Jr. and Luk, 1989, Deller, Jr. and Picaché, 1989]. SM-WRLS is similar to F-H/OBE but is derived in a much different manner . A major difference between the F-H/OBE algorithm, derived from geometric considerations, and SM-WRLS, derived as an RLS algorithm with special weighting, is in the weighting strategy. With the introduction of SM-WRLS, the relationship between OBE and WRLS was formally established. In 1993, the ***set-membership stochastic approximation*** (SM-SA) was introduced in a paper that unifies all previously known versions of OBE algorithms [Deller, Jr. et al., 1994, Deller, Jr. et al., 1993]. The first stochastic proof of convergence (in probability) of an OBE algorithm is achieved with the SM-SA algorithm in [Deller, Jr. et al., 1994], and the unification in [Deller, Jr. et al., 1994] implies that the convergence result is generally applicable to all published OBE algorithms.

In 1991, Cheung [Cheung, 1991, Cheung et al., 1991] published the ***optimal volume algorithm*** (OVE) based on an affine transformation which reduces the hyperellipsoid volume without imposing the condition that the hyperellipsoid center be equivalent to $\boldsymbol{\theta}_t$. This relaxation on the hyperellipsoid center improves reduction in the hyperellipsoid size with a minimal increase in computational cost.

Gollamudi *et. al.* [Gollamudi et al., 1997] further introduced the quasi-OBE (QOBE) algorithms with its asymptotic convergence properties and selective-updating capabilities. The algorithm features a similar minimization criterion to that of D-H/OBE [Gollamudi et al., 1997, Nagaraj et al., 1997] but with a weighting strategy similar to SM-WRLS and was shown to reduce the

percentage of updates and have excellent tracking performance. The convergence analysis of the QOBE algorithm for identification and filtering was presented by Deller *et al.,* in 1997 [Nagaraj et al., 1997].

Various landmark developments of the OBE algorithm were unified by a framework from Deller [Deller, Jr. et al., 1994]. The framework is based on generalized WRLS with wide classes of 'forgetting factors' and data weights. Different instances of OBE algorithms are distinguished by their weighting policies and the criteria used to determine optimal weight values. This formalism enables the exploration of connections among existing OBE algorithms. We thus will adopt this framework when developing the evolutionary OBE algorithms in the next chapter.

5.3 Motivation for a new algorithm

In spite of significant innovation and effort in system identification algorithms for linear models [e.g., [Walter et al., 1996]], nonlinear model identification still poses many unanswered questions.

A significant body of nonlinear model identification using SM algorithms can generally be partitioned into two categories. The first involves explicitly nonlinear system approaches, many of which result in algorithms of high computational complexity. Many efforts of this type are reported in the seminal literature on set-based methods [Walter et al., 1996, Deller Jr. et al., 1993, Walter and Piet-Lahanier, 1990], and the comprehensive study by [Milanese and Novara, 2004] involving noise and functional gradient bounds provides a more recent example of this type, with a clear contrast to the second category. The second partition of set-based approaches uses more conventional LTI models (ARX-like with various noise models) in which LTIiP results are implicit, but not the focus of the work. The latter work has not featured nonlinearities because, with a predetermined model structure, the nonlinear aspects of a LTIiP model are present only as numerical observations with a cumulative effect in the residuals. Accordingly, results are very nonspecific to the effects of particular nonlinearities, but at the same time, are quite sensitive to them.

In this work, we adopt an approach to account for non-LTI system properties, without entirely abandoning the advantages of LTI identification. This compromise is achieved using models that

are LTIiP, with nonlinearities accounted for in the signal processing by the system. LTIiP nonlinear models are versatile and only marginally more difficult to work with than systems with purely LTI structure. The LTIiP structure presents very specific concerns. Contrary to previous work, the nonlinear structure of the model does not remain static in the identification process.

Two lines of reasoning underlie the methods in this paper. First, as indicated above, it is difficult to assess the effects of a predetermined and static nonlinear structure in a LTIiP model. In general, the most challenging problem in any identification problem is the determination of the model structure. Second, in spite of many interesting and beneficial features of set-bounding algorithms, the information inherent in *set* solutions has not been extensively researched in the LTIiP case for potential exploitation. This work uses measures of model quality reflected in set solutions for guidance in the selection of effective models. This is accomplished through evolutionary strategies for optimization with fitness measures derived from the set solutions. Accordingly, the technique simultaneously solves the model structure identification and parameter-estimation problems, in the presence of unknown noise scenarios.

CHAPTER 6

NONLINEAR EVOLUTIONARY SYSTEM IDENTIFICATION

6.1 Introduction and Overview

A much wider variety of system nonlinearities leads to a much broader class of potential nonlinear model forms. Different insights, approximations, and application requirements have led to many model structures. Historically, nonlinear system identification has focused on a few specific models that can be tightly defined, albeit limited. Early work, based on the Volterra series, generalizes the linear convolution concept to accommodate nonlinear systems [Volterra et al., 1930]. While different identification methods are still actively studied [Levanony and Berman, 2004, Xiao et al., 2013], system identification based on the Volterra (and related Wiener) series remains a challenging problem in general. The identification of Volterra series often requires restrictive inputs like Gaussian white noise [Schetzen, 2006] that are not always consistent with applications. Moreover, the Volterra series can require a very large number of parameters to appropriately represent an output process in terms of inputs, even when the physical model is of relatively low order [Billings, 2013]. More recently, research into nonlinear identification has turned to more constrained nonlinear models, including block-structured nonlinear models, such as the Hammerstein model and the Wiener model [Pintelon and Schoukens, 2012]. Due to the simple structure of such models, their identification can be very efficient, but the model forms are limited, and accurate *a priori* system information is required for satisfactory performance. More results will become available as these models continue to be studied in ongoing work [Pawlak et al., 2007, Yu et al., 2014].

In 1985, the nonlinear-ARMAX (NARMAX) model was introduced by Leontaritis and Billings [Leontaritis and Billings, 1985] as a new representation for a wide class of nonlinear systems. As the extension of the ARMAX linear system, the NARMAX model, of which the nonlinear AR (NAR) and ARX (NARX) models are special cases, is the most concise and comprehensive representation

for nonlinear dynamic system identification [Chen et al., 1991]. The essence of the NARMAX model is that past outputs are included in the expansions. This makes the model more concise, since fewer terms are required to represent systems, but it also means that noise in the output has to be taken into account when estimating the model coefficients [Billings, 2013]. The Volterra series, block-structured models, and many network (e.g., radial basis network, neural network) architectures can be considered special cases of the NARMAX model as well. Identification of NARMAX models, which are applicable to a wide class of nonlinear systems, is of great importance in system modeling.

The NARMAX model can be represented by the LTIiP system form to which we alluded above. The most challenging problem in any LTIiP model is the determination of the model structure. The identification procedure must be designed to select the correct model terms (regressor signals) and estimate the model parameters from measurements of system inputs and outputs in the presence of unknown correlated, possibly multiplicative, noise. A simple approach is to estimate a model which includes all of the nonlinear terms and then to prune (backward elimination) the insignificant components [Sjöberg et al., 1995]. This approach is known to cause numerical and computational problems [Billings, 2013]. Another approach is residual-based selection, in which one term is selected at each time according to some measure of goodness of fit (forward selection) [Sjöberg et al., 1995]. An example of this approach is the FROLS algorithm proposed by Billings *et al.* [Chen et al., 1991, Billings and Zhu, 1994], which is based on the orthogonal least squares (OLS) estimator. The FROLS algorithm determines the model structure according to an index calculated from OLS. However, in order to achieve unbiased parameter estimation in the presence of colored or more complex noise, the algorithm must repeatedly refit the noise model. Moreover, the forward selection is greedy in that it only adds one term at a time [Chen et al., 1998].

In the approach to be presented, the fundamental parameter estimation task (the linear part) uses set-theoretic analysis of the data to deduce feasible sets of solutions in light of certain model assumptions. Several well-known batch and recursive methods can be used to identify *point* estimates of LTIiP systems, but OBE algorithms provide *sets* of feasible parameter vectors rather

than a single point estimate [Walter et al., 1996, Combettes, 1993, Deller Jr. et al., 1993]. This is achieved through successive refinements of an initial solution set, consistent with *a priori* constraints on the signal or system model. In the present work, measurable set solution properties are used to assess the viability of nonlinear regressor functions that compete for survival as components of the model best suited to represent the system [Yan et al., 2013, Yan et al., 2014]. Specifically, we describe an evolutionary approach to the selection of nonlinear regressors. The framework presented simultaneously addresses both selection of the model and the parameter estimation. A very significant advantage of the algorithm is the lack of need for assumptions about stationarity or distributional characteristics of the noise. This feature makes the algorithm especially beneficial for practical model identification.

6.2 Identification framework

Consider a single-input–single-output discrete-time system with input x and output ζ, each typically assumed to belong to some well-behaved space, $\mathfrak{X} \subset \mathbb{R}^{\mathbb{Z}}$ (e.g., ℓ_2). Let $\mathbb{F}_{\boldsymbol{\theta}} : \mathfrak{X} \to \mathfrak{X}$ denote the system operator mapping x to ζ, which is parameterized by a real vector $\boldsymbol{\theta} \in \mathfrak{P} \subset \mathbb{R}^{Q}$,

$$\zeta = \mathbb{F}_{\boldsymbol{\theta}}(x) \tag{6.1}$$

The system is said to be ***linear-in-parameters*** if, for any $x \in \mathfrak{X}$, for all $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \mathfrak{P}$, and for all $\alpha, \alpha' \in \mathbb{R}$,

$$\mathbb{F}_{\alpha\boldsymbol{\theta}+\alpha'\boldsymbol{\theta}'}(x) = \alpha\mathbb{F}_{\boldsymbol{\theta}}(x) + \alpha'\mathbb{F}_{\boldsymbol{\theta}'}(x). \tag{6.2}$$

We assume that $\mathbb{F}_{\boldsymbol{\theta}}$ is a continuous operator so that (6.2) extends to countable additivity (e.g., [Naylor and Sell, 1971]).

The internal processing of the system is based on a subset of a ***candidate set*** of nonlinear regressor functions, $\Xi_\varphi = \{\varphi_q\}$, of size $|\Xi_\varphi|$. Each regressor is a mapping $\varphi_q : \mathbb{R}^{r_q+s_q} \to \mathbb{R}$, operating on a set of r_q past and present system inputs, and s_q past outputs. The LTIiP ***observation***

model, $\mathbb{O}_{\boldsymbol{\theta}*,\boldsymbol{\varphi}*}$, for $t \in \mathbb{Z}$, is given by

$$\begin{aligned} \mathbb{O}_{\boldsymbol{\theta}*,\boldsymbol{\varphi}*} \text{ : } \quad \zeta[t] &= \sum_{q=1}^{Q} \theta_{q*} \varphi_{q*} \left(x_{-\infty}^{t}, \zeta_{-\infty}^{t-1} \right) + e_*[t] \\ &\doteq \boldsymbol{\theta}_*^T \boldsymbol{\varphi}_* \left(x_{-\infty}^{t}, \zeta_{-\infty}^{t-1} \right) + e_*[t], \end{aligned} \tag{6.3}$$

with $\boldsymbol{\theta}_* \in \mathfrak{P}$, and $e_* \in \mathbb{R}^{\mathbb{Z}}$ an error sequence (properties described below) representing uncertainties in the model. The "$*$" subscript indicates a "true," but unknown, quantity associated with the observation model.[1] The arguments, $x_{-\infty}^{t}$ and $\zeta_{-\infty}^{t-1}$, of the regressor signals φ_q (or vector $\boldsymbol{\varphi}$) indicate that a finite number of elements are selected from the subsequences $\{\ldots, x[t-1], x[t]\}$ and $\{\ldots, \zeta[t-2], \zeta[t-1]\}$ by each φ_q for processing at time t. For conservation of space, we define the vectors of signal samples,

$$\boldsymbol{u}_{q*}[t] \doteq \left[\begin{array}{c} \text{column vector of } r_q \text{ inputs from } x_{-\infty}^{t}, \\ \text{and } s_q \text{ outputs from } \zeta_{-\infty}^{t-1}, \text{ used by } \varphi_{q*} \text{ at } t \end{array} \right], \tag{6.4}$$

and the matrix $\boldsymbol{U}_*[t] = \left[\begin{array}{cccc} \boldsymbol{u}_{1*}[t] & \boldsymbol{u}_{2*}[t] & \cdots & \boldsymbol{u}_{Q*}[t] \end{array} \right]$. Given observations of x and ζ sufficient to compute outputs on time interval $t = 1, 2, \ldots, \mathfrak{T}$, we pose an ***estimation model***, formulated as a function of the parameters and regressor signals,

$$\begin{aligned} \mathbb{M}_{\boldsymbol{\theta},\boldsymbol{\varphi}} \text{ : } \quad \zeta^p(t, \boldsymbol{\theta}, \boldsymbol{\varphi}) &= \sum_{q=1}^{Q} \theta_q \varphi_q \left(\boldsymbol{u}_q[t] \right) \\ &\doteq \boldsymbol{\theta}^T \boldsymbol{\varphi}(\boldsymbol{U}[t]) \end{aligned} \tag{6.5}$$

in which each φ_q is drawn from the set Ξ_φ (see Footnote 1), $\boldsymbol{\theta} \in \mathfrak{P}$, and the $\boldsymbol{u}_q[t]$ and $\boldsymbol{U}[t]$ are defined similarly to Eqn. 6.4. The superscript on ζ^p connotes "prediction," as this estimation model corresponds to the classical prediction-error method of Ljung [Ljung, 1999] and others. If the observation equation is modeled stochastically, and $\mathcal{E}\{e_*[t]\} = 0$ for each $t \in \mathbb{Z}$, where $\mathcal{E}\{\cdot\}$ denotes the expected value, then

$$\mathcal{E}\left\{ \zeta[t] \,\middle|\, \boldsymbol{\theta}_*, \boldsymbol{\varphi}_* \right\} = \boldsymbol{\theta}_*^T \boldsymbol{\varphi}_*(\boldsymbol{u}_*[t]) = \zeta^p\left(t \,\middle|\, \boldsymbol{\theta}_*, \boldsymbol{\varphi}_* \right), \tag{6.6}$$

[1] Because the index q in φ_q has been defined as an enumeration of the elements of the candidate set Ξ_φ, the functions in Eqn. 6.3 should be indexed as $\varphi_{q_i*}, i = 1, \ldots, Q$, but we use the slightly abusive notation φ_{q*} for simplicity. It is to be understood that φ_{q*} is the q^{th} element *selected from* Ξ_φ, rather than the q^{th} element of Ξ_φ.

where the "conditioning" notation $\left(\cdots \middle| \boldsymbol{\theta}_*, \boldsymbol{\varphi}_* \right)$ is used in a deterministic function merely to emphasize that the conditioning quantities are to be treated as fixed values. Thus, if the "true" parameters and regressors can be found using the estimation model, the prediction sequence will represent the minimum mean-square-error (MMSE) estimate of the observed sequence. That is, $\zeta^p(t, \boldsymbol{\theta}, \boldsymbol{\varphi})$ is the MMSE predictor of the sequence produced by $\mathbb{O}_{\boldsymbol{\theta}_*, \boldsymbol{\varphi}_*}$. The prediction residual sequence associated with the general estimation model $\mathbb{M}_{\boldsymbol{\theta}, \boldsymbol{\varphi}}$ is

$$\begin{aligned} \varepsilon(t, \boldsymbol{\theta}, \boldsymbol{\varphi}) &= \zeta[t] - \zeta^p(t, \boldsymbol{\theta}, \boldsymbol{\varphi}) \\ &= (\boldsymbol{\theta}_* - \boldsymbol{\theta})^T \boldsymbol{\varphi}_*(\boldsymbol{U}_*[t]) + e_*[t] \\ &\quad + \boldsymbol{\theta}^T \left[\boldsymbol{\varphi}_*(\boldsymbol{U}_*[t]) - \boldsymbol{\varphi}(\boldsymbol{U}[t]) \right] \end{aligned} \tag{6.7}$$

indicating error components due to the possible misadjustment in the parameter estimates as well as the possible improper selection of regressor functions. Assuming that the "true" parameters and regressor signals can be found for use in the estimation model, then, (6.7) reduces to

$$\varepsilon\left(t \middle| \boldsymbol{\theta} = \boldsymbol{\theta}_*, \boldsymbol{\varphi} = \boldsymbol{\varphi}_* \right) = e_*[t], \tag{6.8}$$

the asymptotic best case residual.

The objective of the estimation algorithm to be developed is to determine the appropriate regressor signal forms concomitantly with the estimation of parameters for a particular modeling application. The regressor set will be chosen according to a genetic algorithm based on an evolutionary view of the selection process. The parameters are to be identified using a set-theoretic approach which supports the evolution by contributing to model fitness measures.

6.3 Set-theoretic parameter estimation

6.3.1 Formulation of OBE algorithms

Optimal bounding ellipsoid (OBE) algorithms is an important subclass of recursive ***set-membership*** (SM) algorithms which have significant application potential in engineering – especially in their extended forms in this work which accommodates wide classes of nonlinear systems. The classic (linear-systems) methods developed here provide novel approaches and enhancements to approaches

that are traditionally applied to tasks involving system identification, detection, tracking, and adaptive filtering, and data compression and security. The enhanced procedures developed here have the potential to make advances in methods that identify and process nonlinear dynamics. Moreover, the estimation methods to be presented are particularly attractive to SP researchers due to structures that are familiar and readily interpretable to signal and systems engineers (e.g., [Deller Jr. and Huang, 2002]). In this section, we sketch the basic principles of the OBE class of algorithms in the context of the new identification framework.

We begin by positing the existence of a "true" observation model of form Eqn. 6.3, $\mathbb{O}_{\boldsymbol{\theta}_*,\boldsymbol{\varphi}_*}$,

$$\zeta[t] = \boldsymbol{\theta}_*^T \boldsymbol{\varphi}_* (\boldsymbol{U}_*[t]) + e_*[t], \quad t \in \mathbb{Z},$$

in which, *for the purposes of introducing the OBE algorithm*, we assume that the "true" regressor signals $\boldsymbol{\varphi}_{q*} \left(\boldsymbol{u}_{q*}[t] \right)$ are known. The "true" parameters, however, are unknown and to be estimated based on observed values of the sequences ζ and x. Hence, for development purposes only, we use the conditioned estimation model, $\mathbb{M}_{\boldsymbol{\theta}|\boldsymbol{\varphi}_*}$,

$$\zeta^p \left(t, \boldsymbol{\theta} \,\middle|\, \boldsymbol{\varphi}_* \right) = \boldsymbol{\theta}^T \boldsymbol{\varphi}_* (\boldsymbol{U}_*[t]) \tag{6.9}$$

OBE algorithms are based on the assumption of unknown but bounded disturbances in the observation model [Fogel and Huang, 1982b, Dasgupta and Huang, 1987a]. We assume the existence of a sequence of error bounds,

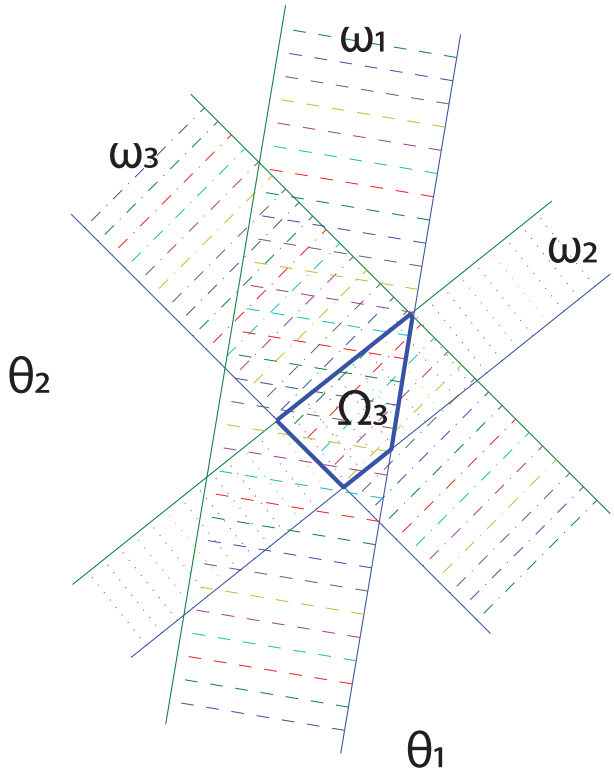
$$| e_*[t] | < \gamma_t, \quad t \in \mathbb{Z}. \tag{6.10}$$

At each t, the observation model of (6.3) in conjunction with the bound γ_t implies the existence of two hyperplanes in the parameter space $\mathfrak{P} \subset \mathbb{R}^Q$,

$$\begin{aligned} \mathcal{H}_t^+ &= \left\{ \boldsymbol{\theta} \in \mathfrak{P} : \zeta[t] = \boldsymbol{\theta}^T \boldsymbol{\varphi}_* (\boldsymbol{U}_*[t]) + \gamma_t \right\}, \quad \text{and} \\ \mathcal{H}_t^- &= \left\{ \boldsymbol{\theta} \in \mathfrak{P} : \zeta[t] = \boldsymbol{\theta}^T \boldsymbol{\varphi}_* (\boldsymbol{U}_*[t]) - \gamma_t \right\} \end{aligned} \tag{6.11}$$

between which $\boldsymbol{\theta}_*$ must lie. The intersection of these pairs of hyperplanes over time forms a convex polytope, say $\overline{\Omega}_t \subset \mathfrak{P}$. The set of points interior to $\overline{\Omega}_t$, say Ω_t, is called the ***feasibility set***, at time

Figure 6.1: Hyperstrips ω_n and their intersection Ω_t as described in (6.12) for a system of order 2 ($m = 2$) at time $t = 3$.



t. $\mathbb{P}_t$ is the exact set of all parameters in $\mathbb{R}^Q$ that could have possibly produced the observed signal using the selected $\mathbb{O}_{\boldsymbol{\theta}_*,\boldsymbol{\varphi}_*}$ model, while adhering to the assumed constraint on the error bounds, γ_t (Fig. 6.3.1),

$$\Omega_t = \cap_{n=1}^{t} \omega_n, \quad \text{where} \quad \omega_n = \{\boldsymbol{\theta} : e_*^2[n] = |\zeta_n - \boldsymbol{\theta}^T \boldsymbol{\varphi}_*(\boldsymbol{U}_*[n])|^2 < \gamma_n^2\}. \tag{6.12}$$

The convex polytopes are difficult to track without adding significant complexity to the recursion on t. However, the existing structure of any OBE-updating procedure readily admits a reasonable set approximation to $\overline{\Omega}_t$ in the form of a hyperellipsoid, say, $\overline{\mathbb{E}}_t$ which overbounds the polytope $\overline{\Omega}_t$. The set of interior points of $\overline{\mathbb{E}}_t$, say $\mathbb{E}_t$, called the ***membership set***, is therefore guaranteed to contain the feasibility set $\mathbb{E}_t \supseteq \Omega_t$. OBE optimization strategies involve minimization of some measure of the "size" of $\mathbb{E}_t$, thus seeking a tight approximation to the feasibility set. Moreover, the updating is very sparse (typically $\ll 10\%$ of the observations are innovative) leading to a very

efficient estimation procedure.

In general, an hyperellipsoidal set in $\mathbb{R}^Q$ is a collection of points

$$\mathbb{E} \doteq \left\{ \boldsymbol{\theta} \in \mathbb{R}^Q \,:\, (\boldsymbol{\theta} - \boldsymbol{\theta}_c)^T \boldsymbol{K} (\boldsymbol{\theta} - \boldsymbol{\theta}_c) < 1 \right\} \tag{6.13}$$

in which $\boldsymbol{\theta}_c$ is the centroid of the set, and $\boldsymbol{K} \in \mathbb{R}^Q$ is a nonnegative-definitive matrix. In an OBE algorithm, the hyperellipsoidal membership set at time $t \geq Q$ is given by (Fig. 6.3.1)

$$\begin{aligned} \mathbb{E}_t &\doteq \left\{ \boldsymbol{\theta} \in \mathbb{R}^Q \,:\, (\boldsymbol{\theta} - \boldsymbol{\theta}_t)^T \kappa_t^{-1} \boldsymbol{C}_t (\boldsymbol{\theta} - \boldsymbol{\theta}_t) < 1 \right\} \\ &= \left\{ \boldsymbol{\theta} \in \mathbb{R}^Q \,:\, (\boldsymbol{\theta} - \boldsymbol{\theta}_t)^T \boldsymbol{C}_t (\boldsymbol{\theta} - \boldsymbol{\theta}_t) < \kappa_t \right\} \end{aligned} \tag{6.14}$$

in which $\boldsymbol{C}_t$ is the weighted sample covariance matrix of the observed regressor signals. $\boldsymbol{C}_t$ is fundamentally defined as the sum of the weighted outer products, and can be computed recursively

$$\begin{aligned} \boldsymbol{C}_t &\doteq \sum_{n=1}^{t} q_{t,n} \boldsymbol{\varphi}_* (\boldsymbol{U}_*[n]) \, \boldsymbol{\varphi}_*^T (\boldsymbol{U}_*[n]) \\ &= \alpha_t \boldsymbol{C}_{t-1} + \beta_t \boldsymbol{\varphi}_* (\boldsymbol{U}_*[t]) \, \boldsymbol{\varphi}_*^T (\boldsymbol{U}_*[t]) \end{aligned}$$

where in the most general case the data weights $q_{t,n}$ may be time-varying in a simple way, e.g.

$$q_{t,n} = \begin{cases} \alpha_{t-1} q_{t-1,n}, & n \leq t-1 \\ \beta_t, & n = t. \end{cases}$$

κ_t is a positive constant

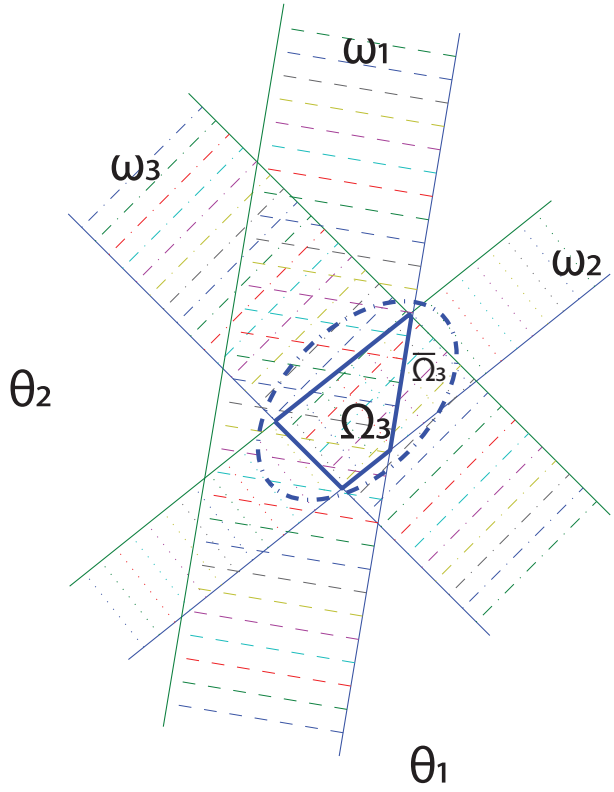
$$\kappa_t = \boldsymbol{\theta}_t^T \boldsymbol{C}_t \boldsymbol{\theta}_t + \sum_{n=1}^{t} q_{t,n} (\gamma_n^2 - \zeta_n^2), \tag{6.15}$$

and $\boldsymbol{\theta}_t$, is the ellipsoid center of $\overline{\Omega}_t$, which can be used as a point estimator of $\boldsymbol{\theta}_*$ if desired,

$$\boldsymbol{\theta}_n = \boldsymbol{P}_t \boldsymbol{c}_t, \quad \text{with } \boldsymbol{P}_t \doteq \boldsymbol{C}_t^{-1} \text{ and } \boldsymbol{c}_t \doteq \sum_{n=1}^{t} q_{n,t} \zeta_n \boldsymbol{\varphi} (\boldsymbol{U}_*[n]) \tag{6.16}$$

Because the hyperellipsoidal membership set is structured using second-order statistics, the updating recursions for the OBE-class algorithms are closely related to the weighted recursive least squares (WRLS) algorithm [Deller Jr. et al., 1994]. The recursive updating equations for a generic

Figure 6.2: The ellipsoid superset $\overline{\Omega}_n \supseteq \Omega_n$ corresponding to the system of Fig. 6.1.



OBE algorithm [Deller Jr. et al., 1994] are as follows. For $t = 1, 2, \ldots$ (functional dependencies other than that on t are suppressed),

$$
\begin{aligned}
G_t &= \boldsymbol{\varphi}_*^T[t]\mathbf{P}_{t-1}\boldsymbol{\varphi}_*[t] \\
\varepsilon_t &= \zeta[t] - \boldsymbol{\theta}_{t-1}^T\boldsymbol{\varphi}_*[t] \\
\mathbf{P}_t &= \frac{1}{\alpha_t}\left[\mathbf{P}_{t-1} - \frac{\beta_t\mathbf{P}_{t-1}\boldsymbol{\varphi}_*[t]\boldsymbol{\varphi}_*^T[t]\mathbf{P}_{t-1}}{\alpha_t + \beta_t G_t}\right] \\
\boldsymbol{\theta}_t &= \boldsymbol{\theta}_{t-1} + \beta_t\mathbf{P}_t\boldsymbol{\varphi}_*[t]\varepsilon_t, \\
\kappa_t &= \alpha_t\kappa_t + \beta_t\gamma_t^2 - \frac{\alpha_t\beta_t\varepsilon_t^2}{\alpha_t + \beta_t G_t},
\end{aligned}
\tag{6.17}
$$

$\{\alpha_t\}$ and $\{\beta_t\}$ are nonnegative weighting sequences. The process is typically initialized with $\boldsymbol{\theta}_0 = 0, \kappa_0 = 1$ and $\mathbf{P}_0 = \frac{1}{\mu}\mathbf{I}$ (μ is a small number).

Different members of the OBE family are distinguished by the forms of the weighting sequences $\{\alpha_t\}$ and $\{\beta_t\}$, and the criteria by which optimal values of these weights are determined. Two of

the three optimization criteria used in the development of OBE algorithms are directly related to the "size" of $\mathbb{E}_t$ at each t. These are:

(i) to minimize the determinant of the inverse ellipsoid matrix, $\mu_t^{\text{vol}} \doteq \det\{\kappa_t \mathbf{P_t}\}$, which is proportional to the square of the volume of the ellipsoid;

(ii) to minimize the trace of the inverse ellipsoid matrix, $\mu_t^{\text{tr}} \doteq \text{tr}\{\kappa_t \mathbf{P_t}\}$ which is proportional to the sum of squares of the ellipsoid's semi-axes;

(iii) to minimize the parameter κ_t which is guaranteed to reduce ellipsoid volume though not to the minimum possible, and which has complex interpretations described in, for example, [Deller Jr. et al., 2007].

The weighting sequences are determined by the optimization criteria. For any of the criteria, there is only one quantity to be optimized at time t, which is dependent upon both α_t and β_t. However, the numbers α_t and β_t are independent of one another, so that any attempt to optimize one of the criterion measures with respect to both α_t and β_t results in an infinity of solutions which is resolved by arbitrarily choosing a value of either weight. Thus, we may either tie the weights together through some functional purpose, or simply eliminate the 'unused' weight altogether by setting it to unity, then seeking the optimal λ_n. The policy of writing the weights α_t and β_t as functions of a single parameter to be optimized are adopted,

$$\alpha_t = \alpha_t(\lambda_t), \text{ and } \beta_t = \beta_t(\lambda_t). \tag{6.18}$$

Optimization in each iteration generally consists of solving a quadratic to determine the existence of optimal weights in the sense of shrinking the ellipsoid size. If such weights do not exist, the observation can be discarded and the effort of updating avoided. Efficient checks for innovation in the data can be used to circumvent the computation involved in direct solution for weights (e.g., [Deller Jr. et al., 1993, Deller Jr. and Huang, 2002, Deller Jr. et al., 1994]). The process is mildly dependent on the particular OBE strategy used. The various methods which are distinguished by the specification of their weighting sequences are summarized in Table 6.1.

Table 6.1: Specification of popular OBE algorithms. The notation of λ_* is used to denote the existence of an optimal weight that minimizes the optimization criterion [Deller Jr. et al., 1994]

Algorithm	$\alpha_t(\lambda_t^*)$	$\beta_t(\lambda_t^*)$	Optimization
F-H/OBE	$1/\kappa_{t-1}$	λ_t^*/γ_t	μ_t^{vol} or μ_t^{tr}
SM-WRLS	1	λ_t^*	μ_t^{vol} or μ_t^{tr}
Dual SM-WRLS	λ_t^*	1	μ_t^{tr} or μ_t^{tr}
D-H/OBE	$1-\lambda_t^*$	λ_t^*	$\kappa(t)$
SM-SA	$\Lambda_{t-1}^*/(\Lambda_{t-1}^*+\lambda_t^*)$ $(\Lambda_t^* \doteq \sum_{n=1}^t \lambda_t^*)$	$\lambda_t^*/(\Lambda_{t-1}^*+\lambda_t^*)$	μ_t^{vol} or μ_t^{tr}
QOBE	1	λ_t	$\kappa(t)$

In addition to the very sparse and efficient updating procedure used by OBE methods, a significant advantage for application is the lack of need for assumptions about stationarity or distributional characteristics of the error sequence e_*. Further, OBE algorithms yield a feasible set of solutions that can be exploited (see below), whereas the centroid of the set provides a conventional estimate which is interpretable as a weighted least square error estimate.

Obtaining the advantages of the OBE algorithm requires knowledge of noise bounds which are unavailable in many applications. This issue is particularly poignant in the present deployment of the method because shifting the sets of regressor signals in $\mathbb{O}_{\boldsymbol{\theta}_*,\boldsymbol{\varphi}_*}$ can have significant effects on the error bounds in the model. In the analysis above, we have assumed known regressor signals. We next introduce innovations into the OBE structure to account for the potential for erroneous error bounds as the model undergoes "evolution."

6.3.2 Technical adjustments for evolving regressor signals

To mitigate the problem of unknown error bounds, we incorporate the ***automatic bound estimation*** (ABE) procedure [Lin et al., 1997]. The *OBE algorithm with automatic bound estimation* (OBE-ABE) developed by Lin in 1996 [Lin, 1996] is the first OBE method to theoretically solve the difficult problem of accurately estimating "true" model error bounds in ARX models with "true" error $\{e_t\}$ and exogenous input $\{u_n\}$ (included as part of the measurements $\{\boldsymbol{\varphi}(\boldsymbol{U}_*)\}$). In theory, OBE-ABE removes the practical roadblock to model identification using OBE algorithms. OBE-

ABE converges consistently under conditions on the "true" model error sequence $\{e_t\}$ that are met by many practical signals, and additionally provides the customary OBE set of feasible solutions.

The basis for the "ABE" in OBE-ABE is Lin's proof [Lin, 1996] that, if the error bounds are overestimated, there exists an interval $\mathbb{I}$ of length $\mathcal{N}$ over which no update takes place for any finite $\mathcal{N}$. Thus, the need to adjust the bound is practically indicated by a sufficiently long period over which no update takes place. When such an interval is found, OBE-ABE invokes its bound re-estimation recursion which depends on $\mathcal{N}$ and an "adjustment constant" ϵ,

$$\Gamma_t = \begin{cases} \Gamma_{t-1} - d_J, & \text{if } d_J > 0 \\ \Gamma_{t-1}, & \text{otherwise} \end{cases} \tag{6.19}$$

$$\text{where,} \quad d_J = (\kappa_{J-1})G_J/Q - \epsilon(2\sqrt{\Gamma_{t-1}} - \epsilon)$$

$$\text{and} \quad J = \arg\max_{t \in \mathbb{I}} \varepsilon_t^2$$

in which ϵ is a small number, and γ_t denotes the sequence of estimated bounds with $\Gamma_0 > \gamma$ (true bound).

Furthermore, an immediate and simple check for the ellipsoid regularity is incorporated into the algorithm. For meaningful parameter estimation, the ellipsoid $\mathbb{E}$ must have non-negative volume. This condition requires that $\kappa_t > 0$.

While the above mentioned two enhancements can be incorporated into any OBE algorithm, here we employ the OBE algorithm version known as set-membership-stochastic approximation (SM-SA). In the SM-SA algorithm, the volume of the ellipsoid (proportional to $\det\{\kappa_t P_t\}$) is minimized at each iteration by letting $\lambda_t = \beta_t$ and $\alpha_t = 1 - \beta_t$ in (6.17), and seeking the optimal β_t in light of the current measurements [Lin et al., 1998]. We will refer our algorithm as ***OBE with evolved regressor signals*** (OBE-ERS), which is given in Algorithm 1.

Several goodness-of-fit measures can be derived from the final parameter membership set. For instance, the conventional prediction error associated with the centroid of the set (point estimator) can be calculated as a quality measure. Additionally, the estimated bounds and the set-related-properties, such as the volume of the ellipsoid and the shape of the ellipsoid, can be used as

Algorithm 1: OBE-ERS

Data: Observation subsequences x, ζ
Result: Parameter membership set
Initialization

① $\boldsymbol{\theta}_0 = \mathbf{0}$, $\kappa_0 = 1$ and $\mathbf{P}_0 = \frac{1}{\mu}\mathbf{I}$;

② γ_0 a overestimated bound;

③ windown length $\mathcal{N}$ and $\epsilon = 0.02$.

```
/* Parameter estimation starts                                    */
for each data pair (x, ζ) do
    if κ_t > 0 then
        if data is innovate then
            execute (6.17);                          /* OBE updates */
        else if 𝕀 > 𝒩 then
            execute (6.19) ;                                /* ABE */
            𝕀 = 0;
        else
            𝕀 = 𝕀 + 1;
        end
    end
end
```

measures of viability. Naturally, the best fitting model should have the minimum estimation error, the smallest ellipsoid volume, and the smallest estimated error bound. A good model will also arise from an ellipsoidal set of "balanced dimensions," so that no redundant regressors are included in the model. In the next section, these properties characterize the performance of models in the estimation model set and are exploited by an evolutionary algorithm to guide the evolution towards better models.

6.4 Evolutionary model selection

6.4.1 "Cell biology" of the LTIiP model

An evolutionary algorithm is a generic population-based meta-heuristic optimization algorithm. Falling under the category of evolutionary algorithms, the genetic algorithm solves optimization and search problems using processes inspired by natural evolution [Mitchell, 1998, Reeves and

Rowe, 2003]. Genetic algorithms often exhibit excellent performance in nonlinear and discrete optimization problems. Here, we use a genetic algorithm to aid in system modeling. The model structure detection problem, which is an NP-hard problem, can be mapped coherently into the standard genetic algorithm framework.

In the present formulation, a LTIiP model is an "***organism***" with a single "***chromosome***." In contrast to the complex chemical structure of chromosomes in living cells, the LTIiP chromosome is a simple, finite, binary sequence in which each bit is treated in the genetic algorithm terminology as a "***gene***." In this case, each gene "codes for" the presence or absence of a particular regressor function in the model.

Let $\Xi_\varphi = \{\varphi_q\}$, of set size $\left|\,\Xi_\varphi\,\right|$ elements, contain the regressor functions available with which to create models. A chromosome is a string of $\left|\,\Xi_\varphi\,\right|$ bits which is in one-to-one correspondence with a model. A unity bit in the ℓ^{th} binary position of the chromosome is a "gene" that codes for regressor φ_q^ℓ in its model. In principle, then, there are $(2^{|\Xi\varphi|} - 1)$ different chromosomes, corresponding to an equal number of distinct models. A viable model is one with an effective set of regressor functions and parameter values that allow it to accurately produce the observed ζ from the observed x. Although they appear in the "***phenotype***," the parameters in some sense represent additional "***regulators of expression***" of the genes, the desired expression being the linear mix of proteins that give the model the highest "***survival potential***." As in biology, survival depends on the inherent suitability of an individual's genetic makeup for meet the challenges of the environment (reflected in x and ζ), and also in the realization of that genetic potential resulting from an effective parameter set (Table 6.2).

To demonstrate, a LTIiP model is represented by a binary string (chromosome) as follows

$$\underbrace{\overbrace{1}^{\text{gene}}\,000\cdots\overbrace{1}^{\text{gene}}\,0\cdots}_{\text{chromosome}} \leftrightarrow \underbrace{\zeta[t] = \theta_1\,\overbrace{x[t]}^{\text{regressor}} + \theta_2\,\overbrace{x[t-1]\zeta[t-1]}^{\text{regressor}}}_{\text{LTIiP model}} \tag{6.20}$$

In this example, the chromosome is a binary string with two genes "on" that together encode the phenotype with only two regressors: $x[t]$, $x[t-1]\zeta[t-1]$. The parameters of the LTIiP model are determined by fitting the model to the observations $\{x[t], \zeta[t]\}$.

Table 6.2: Adaptation of system modeling to a genetic algorithm

Cell biology	**system model**
Chromosome	LTIiP model
Gene	Regressor function
Regulator of gene	Parameter

In practice, specification of the set $\Xi_\varphi = \{\varphi_q\}$ is of great significance since regressor functions in the estimation models are restricted to those in the candidate set. When available, physical knowledge can guide the composition of Ξ_φ. Generally, Ξ_φ may be chosen to include families of functions with desirable approximation characteristics. Popular structures include polynomials, Gaussian functions, and wavelets [Ljung, 1999, Milanese and Novara, 2004].

6.4.2 Survival fitness measures

The set-theoretic aspects of the identification constrain the sets of parameters to those that are feasible in light of the observations and the error constraints. In turn, they determine the range and statistical viability of the phenotype, and ultimately the plight of the chromosome. The algorithm starts with a population $\mathbb{P}$, a set of candidate models typically generated at random (but not necessarily with equal probabilities of 0's and 1's),

$$\mathbb{P}[\tau] = \{\zeta_j^p[\tau] \mid \zeta_j^p \in \mathbb{M}_{\boldsymbol{\theta},\boldsymbol{\varphi}}\} \tag{6.21}$$

where $j = 1, 2, \ldots, |\mathbb{P}|$ represents individuals in the population, and $\tau = 1, 2, \ldots$ is the generation index. The parameters of each model are determined by the OBE-ERS algorithm, and the performance of each model (corresponding to a chromosome) is evaluated via a fitness function derived from the set properties of the resulting ellipsoid.

Different fitness functions can be used for model selection. For instance, a function resembling the Akaike Final Prediction Error (FPE) [Akaike, 1969] can be used as the fitness function for evaluating different models. The FPE, which is broadly used for measuring model quality and determining model order in linear models, is given by

$$\mathcal{F}_1 = \frac{N + \sigma\|\boldsymbol{\theta}\|_0}{N - \sigma\|\boldsymbol{\theta}\|_0} V \tag{6.22}$$

where N is the data length, $\|\boldsymbol{\theta}\|_0$ is the ℓ_0 norm of the vector $\boldsymbol{\theta}$, which also equals the number of model terms, and σ is an adjustable parameter. V is defined as sum of squared one-step prediction errors in the FPE criterion. However,in correspondence with the OBE algorithm, here we define V as the weighted sum of errors with the weights estimated by the OBE algorithm $q_{t,n}$,

$$V = \sum_{n=1}^{t} q_{t,n}(\zeta - \zeta_j^p)^2. \tag{6.23}$$

$\mathcal{F}_1$ represents bias and variance trade-offs in model fitting. A useful interpretation of $\mathcal{F}_1$ is that the criterion provides a measure by simulating the situation in which the model is tested on a different data set. The goal is to minimize the fitness function defined by the FPE criterion.

An alternative model quality measure based on the properties of the ellipsoid set is designed empirically:

$$\mathcal{F}_2 = \|\boldsymbol{\theta}\|_0 \times b^2 \times R \tag{6.24}$$

where $\|\boldsymbol{\theta}\|_0$ is the number of regressor signals, b is the estimated bound of the error sequence, and R is the ratio of the longest and the shortest axes of the ellipsoid. A good model should have a smaller estimated error bound, and the parameter hyperellipsoid should approximate a hypersphere if all the chosen regressors contribute significantly to the model fitting. $\|\boldsymbol{\theta}\|_0$ is included to avoid model overfitting. Other fitness functions can also be designed based on the properties of the membership set (e.g., the volume of the ellipsoid).

Each individual is assigned a fitness value derived from the fitness function which is then used in the selection to bias the new population toward the inclusion of better individuals. Highly-fit individuals (in our case, smaller fitness values) have a high probability of being selected for reproduction. The process continues through subsequent generations. The average fitness of the population improves as more fit individuals appear and interbreed, and the less fit individuals are selected out. The evolutionary algorithm is terminated when a predefined number of generations is reached or the fitness values in the population reaches a prescribed minimum.

6.4.3 Evolutionary operations

The genetic algorithm begins by creating a random initial population. At each step, the algorithm uses the current population to create the offspring that comprise the next generation. The algorithm selects a group of individuals in the current population, called parents, who contribute their genes to their children. The algorithm usually selects individuals that have better fitness values as parents. The operations of selection, crossover, and mutation affect the evolutionary process. Algorithm 2, evolutionary model selection, outlines this process, and the details regarding these operators are described as follows. Further details are found in [Reeves and Rowe, 2003, Goldberg, 2002].

Selection directs the algorithm towards fitter solutions. Selection must favor fitter candidates over weaker ones. Among the various selection schemes, ***tournament selection*** is widely used [Reeves and Rowe, 2003]. The method randomly partitions the population into "tournaments" and selects winners. The winner of each tournament becomes a parent, which is then used by the crossover and mutation operators to produce children.

Crossover is used to vary the chromosomes from one generation to the next. After selection, parent solutions are chosen to produce child solutions with probability p_c. A unity probability indicates that all the selected parent chromosomes are used in reproduction. In the ***uniform crossover*** [Reeves and Rowe, 2003] scheme used here, a random binary vector is first created. Then the child is created by selecting the genes where that vector has a 1 from the first parent, and the genes where that vector is a 0 from the second parent, and the genes that result form the child.

Mutation is used to introduce random alteration of some of a parent's genes to form a child. It is analogous to biological mutation and typically occurs at a relatively low rate. Mutation inverts each bit in a single parent with a low probability p_m per bit.

Replacement determines which children survive into the next generation. We employ an eliteness scheme. Elite individuals are the individuals in a population with the best fitness values. Elite selection operates by combining the parent population with the child population and allowing the elite individuals, up to the size of the population, to survive.

Algorithm 2: Evolutionary model selection

Data: Observation subsequences x, ζ
Result: Best fitting model
Initialization

① population $\mathbb{P}$ of size $|\mathbb{P}|$

② maximum generation $N_{\max}$

③ crossover and mutation rate

④ OBE-ERS algorithm initialization

```
/* Regressor selection starts                                    */
```
for $\tau \leftarrow 1$ to $N_{\max}$ **do**
  for $j \leftarrow 1$ to $|\mathbb{P}|$ **do**
    OBE-ERS ;
    calculate fitness values $\mathcal{F}^j$ of each individual;
  end
  selection;
  crossover and mutation;
  replacement;
end

6.5 Multi-objective optimization

6.5.1 Optimization formulation

The problem of LTiIP model selection and parameter estimation is formulated as a bi-objective optimization problem for which the solution is given by

$$\{\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\varphi}}\} = \underset{\boldsymbol{\theta},\boldsymbol{\varphi}}{\operatorname{argmin}} \quad \left(\sum_{t=1}^{n} q_{t,n} \varepsilon^2(t|\boldsymbol{\theta}, \boldsymbol{\varphi}), \|\boldsymbol{\theta}\|_0 \right) \qquad \text{s.t.} \quad \varphi_q \in \Xi_\varphi \tag{6.25}$$

The objective is a vector, and each coordinate corresponds to an objective function value. The arguments for this optimization problem are $\boldsymbol{\theta}$ and $\boldsymbol{\varphi}$. The optimization returns a vector, $\hat{\boldsymbol{\varphi}}$, of binary decisions concerning whether each regressor should be included in the model, and for the regressors included, the estimated parameters $\hat{\boldsymbol{\theta}}$. Solutions over the set $\Xi = \{\varphi_q\}$ can be ill-conditioned if the number of regressors selected is large. In nonlinear fitting, this is often the case because of the

"fine tuning" required to model nonlinear mappings. Singularity can be avoided by selecting only terms that contribute most significantly to cost reduction. An added benefit of model sparsity is the avoidance of overfitting the requisite finite data records.

The problem Eqn. 6.25 is an NP-hard problem. One way to solve this is to convert it to a single-objective optimization problem using utility functions, which are mappings of objective vectors to real numbers [Bishop, 2006, Deb, 2014, Russell and Norvig, 2010]. For instance, Eqn. 6.22 is one commonly used utility function that forms heuristic combinations of the objective functions [Akaike, 1969]. Though this transformation is effective, the solution could be sensitive to the hyperparameter σ which controls the balance of objectives; thus multiple values are often tested in practice.

To avoid trial-and-error, we use a multi-objective genetic algorithm to directly solve Eqn. 6.25. The bi-objective solution provides a way to study the inherent trade-offs in meeting the two optimization goals.

6.5.2 Multi-objective genetic algorithm

The multi-objective genetic algorithm does not require gradient search, thus allowing application with a discrete optimization functional. Unlike classical optimization algorithms, the evolutionary approach provides multiple optimal solutions from concurrently pursuing multiple solution trajectories. Moreover, the evolutionary approach uses stochastic operators, unlike the deterministic operators used in many classical methods. This allows the algorithms to explore multiple local optima and often to escape local optima and more reliably find global or near-global optima [Deb, 2011].

In particular, we employ the evolutionary multi-objective solver developed by Deb et al. [Deb et al., 2002], the elitist ***non-dominated sorting GA*** (NSGA-II), an efficient GA that works with a population of evolving solutions. The algorithm starts with a random population of chromosomes. Based on the genetic makeup of each chromosome (i.e., a hypothesized model structure), a feasible set of parameters is deduced from the observations using the SM-SA algorithm. Then, the multi-

objective function values of (6.25) are calculated using the centroid of the estimated set of parameters for each individual. The current population is then used to create the offspring that comprise the next generation. In a multi-objective optimization problem, any two solutions have one of two relationships: one dominates the other or neither dominates. In a minimization problem, we say solution a dominates b if the following conditions are satisfied [Loghmanian et al., 2012]:

$$\begin{aligned} &\forall i \in 1, 2, \ldots, g, f_i(a) \leq f_i(b), \text{ and} \\ &\exists i \in 1, 2, \ldots, g, \text{s.t. } f_i(a) < f_i(b). \end{aligned} \tag{6.26}$$

where f_i is the vector of objective values. If the above conditions are not violated, solution a dominates solution b. If there is no solution in the population which dominates b, then b is a ***non-dominated*** solution. The solutions that are non-dominated within the entire search space are called ***Pareto optimal***. Ordinarily there are multiple Pareto optimal solutions. The set of objective value vectors of the Pareto optimal solutions is often called the ***Pareto front***. The goal is to find a Pareto set of solutions that are each on, or near, the Pareto front [Deb, 2014, Boyd and Vandenberghe, 2004].

For each generation, non-dominated individuals are identified and selected from the population using the mathematical partial ordering concept and a niche-preserving operator [Deb et al., 2002]. More specifically, the selection operator first selects all non-dominated individuals (a "front") in the combined parent and child population, to survive, so long as their number does not exceed the population size. Then it eliminates them and repeats the process to find the next front among the remaining population, so long as the size of the front does not exceed the remaining positions in the population size. As soon as a front is too big to fit entirely in the next generation's population, the niche-preserving operator (crowded-comparison-operator) will select individuals that are far from other individuals to maintain diversity in the population [Deb et al., 2002]. The selected individuals in the current population, called parents, contribute their genes to their children. The operators of selection, crossover, mutation, and recombination implement the evolution, together with NSGA-II's non-dominated sorting algorithm to calculate survival.

For the present implementation, tournament selection, scatter crossover, and bit-wise mutation are used [Reeves and Rowe, 2003].

Since multiple non-dominated solutions are maintained, and a measure of diversity [Deb, 2014] is used within the non-dominated set, NSGA-II is able to find multiple, well-distributed, non-dominated solutions. For the model identification, the algorithm is expected to find a set of system models that balance the competing objectives of fitting accuracy and model complexity. A further decision-making criterion is described in next chapter for choosing a single model from the set of solutions.

CHAPTER 7

EXPERIMENTS, APPLICATIONS AND DISCUSSION

7.1 Introduction

The evolutionary model selection algorithm can be applied to a broad class of identification problems spanning many disciplines. In this section, we report on the use of the method in identifying nonlinear (NARMAX) models for both simulated and practical systems to illustrate its application potential. Several factors are common to the experiments. In the initialization of the algorithm, the population size is 50 at each generation,and the maximum number of generations $N_{max} = 100$. The crossover rate is $p_c = 1$ and the mutation rate is $p_m = 0.1$. For each generation, other than two elite parents guaranteed to survive, 80% of the children come from crossover, and 20% from mutation. By design, the candidate set of regressors, $\Xi_{\boldsymbol{\varphi}}$, contains various linear and nonlinear functions. In practice, knowledge of the system can be used to guide the creation of this set, which might include, for example, polynomials and the hyperbolic tangent as functions of the current and past inputs. For the simulation study in this section, our candidate set contains linear and polynomial nonlinear functions (up to order three, including cross-terms) of delayed inputs and outputs with maximum delay of three (i.e. $\zeta[t-1], \zeta[t-2], \zeta[t-3], x[t-1], x[t-2], x[t-3]$), totally 55, for instance, $x[t-1]\zeta^2[t-2]$.Thus, the length of the chromosome is 55 genes, with each bit coding for a regressor, yielding $(2^{55} - 1)$ possible estimation models.

7.2 Results on simulated sequences: Single-objective optimization

7.2.1 IID noise sequence

7.2.1.1 System 1

For simplicity, we begin with a two-parameter NARX system

$$\zeta[t] = x[t-1] + 0.25x[t-2]\zeta[t-3] + e_*[t] \tag{7.1}$$

in which the input sequence x is an uncorrelated Gaussian process, $x \sim \mathcal{G}(0, 2)$, and $e_*[t]$ is uniformly-distributed i.i.d white noise, $e_* \sim \mathcal{U}(-0.1, 0.1)$. The noise is assumed unknown *a priori* and the noise bound is to be estimated. 1024-point signal observations (input-output pairs) are generated. The observation dataset is divided into: a 512-point training set and a 512-point test set. The training set is used to identify the model, while the test set is used to assess the performance of the identified model. The simulated input and output are shown in Fig. 7.1.

Using the $\mathcal{F}_1$ criterion, (6.22) as the fitness function with $\sigma = 1$, the estimated model is

$$\zeta^p[t] = 1.0001x[t-1] + 0.2494x[t-2]\zeta[t-3] - 0.0009\zeta[t-1]\zeta[t-2] \tag{7.2}$$

From the result, we can see that the key structure (the first and second regressor terms) has been detected and the estimated parameters are accurately estimated using the algorithm. The parameter associated with the extra term $\zeta[t-1]\zeta[t-2]$ is so small as to make the term negligible. The residual and the autocorrelation of the residual are shown in Fig. 7.2, where the residual appears to be white in accordance with the true system as in (7.1). Note that since the genetic algorithm is a stochastic optimization algorithm, we have ran the algorithm several times and the results given here are typical of the results obtained.

The Model Predicted Output (MPO) [Billings, 2013] is calculated using the test set. The system output is initialized by a few measured output values and then MPO is calculated from the identified model driven by only the given input. The result is shown in Fig. 7.3. For clear visualization, only 100 samples are shown. The selected model exhibits excellent tracking ability. The Pearson correlation coefficient of the true and estimated output is 0.9996. To further verify that the extra term $-0.0009\zeta[t-1]\zeta[t-2]$ is negligible, the MPO is calculated with the small term excluded. The correlation coefficient is 0.9996 as well which means that the term is insignificant at least to $O(10^{-4})$ in this measure.

Using the same setup with only the fitness function changed, another experiment was imple-

Figure 7.1: Simulated System 1: input and output data.

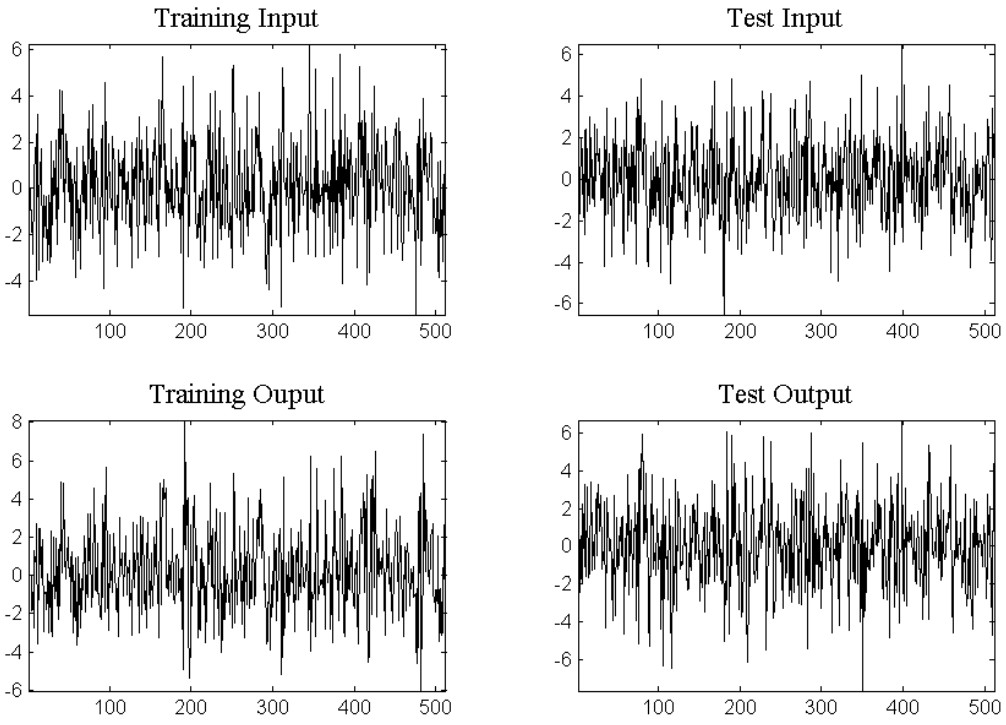


Figure 7.2: System 1: Residual and linear correlation Test: the horizontal line on the second figure is the 95% confidence interval.

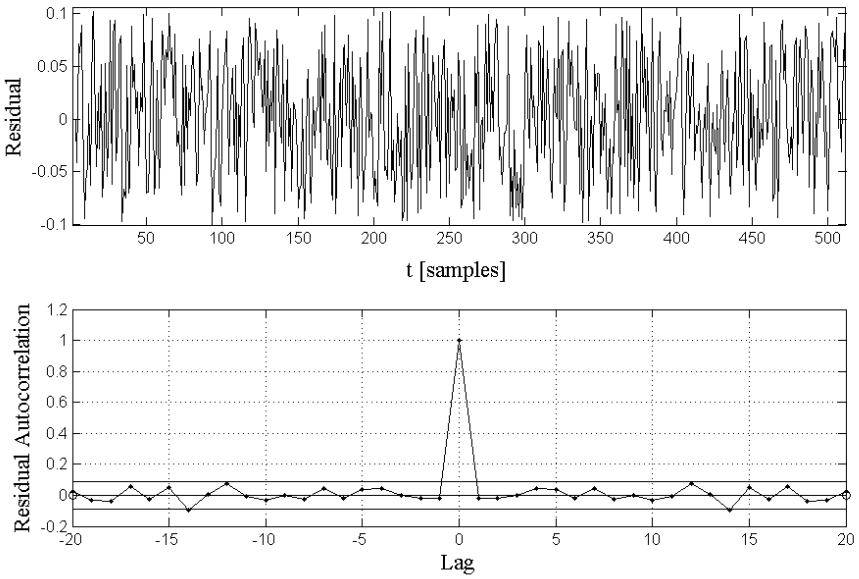
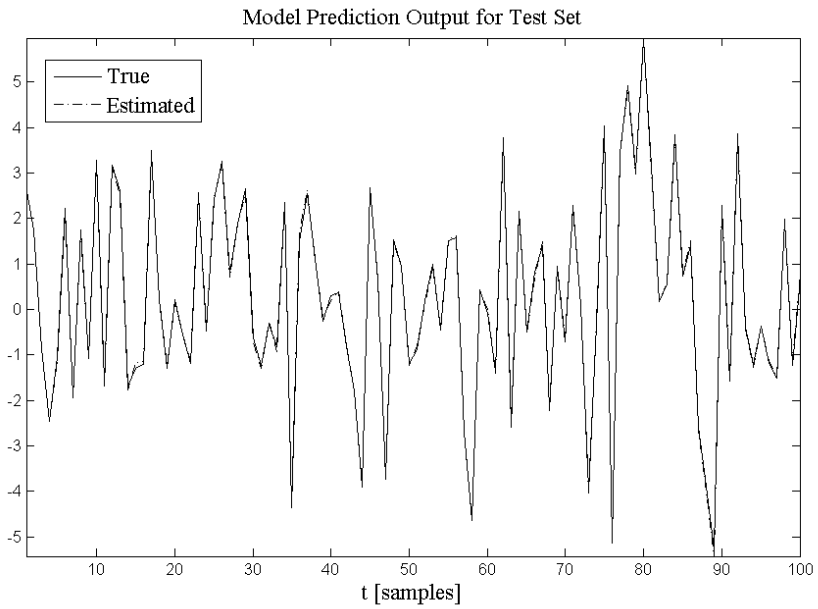


Figure 7.3: System 1: Identification results: True data (continuous curve) and estimated data (dash-dot curve).



mented using the set measure related fitness function of (6.24). The estimated model is

$$\begin{aligned}\zeta^{p}[t] =& 1.0001x[t-1] + 0.2494x[t-2]\zeta[t-3] \\ & - 0.0028\zeta[t-1]\zeta[t-3]\end{aligned} \tag{7.3}$$

The set property related measure is also successful in identifying the nonlinear system. The correlation coefficient of the original and estimated output is 0.9996. The result indicates that in addition to its success when using the prediction error as the conventional point estimate method, the OBE-ERS algorithm provides additional useful options for use as model quality measures and model selection criteria.

7.2.1.2 System 2

A more complex system is next simulated to test the algorithm

$$\begin{aligned}\zeta[t] \;=\; & 0.8\zeta[t-1] - 0.6\zeta[t-2] + x[t-1] - 0.2x^{2}[t-1] \\ & +0.01\zeta[t-1]\zeta[t-2]x[t-2] + e_{*}[t]\,.\end{aligned}$$

As before, 512 points are used for estimation and 512 points for testing. Using the $\mathcal{F}_1$ criterion ($\sigma = 1$) as the fitness function, the estimated model is

$$\begin{aligned}\zeta^p[t] =& 0.8020\zeta[t-1] - 0.6005\zeta[t-2] \\ &+ 0.9973x[t-1] - 0.1984x^2[t-1] \\ &+ 0.0100\zeta[t-1]\zeta[t-2]x[t-2]\end{aligned} \tag{7.4}$$

Again, the algorithm has correctly identified the system model. The correlation coefficient on the test set is 0.9997.

7.2.2 Colored noise sequence

A significant advantage of the OBE estimation is the lack of need for assumptions about stationarity or distributional characteristics of the error sequence. The model selection algorithm can identify the system under different noise conditions, a useful ability for NARMAX model identification with exotic noise conditions.

7.2.2.1 System 3

In this case the system output is contaminated by colored noise and there are no cross products between the noise and the system input and output measurements. A three-parameter system with colored noise is simulated (NARMAX)

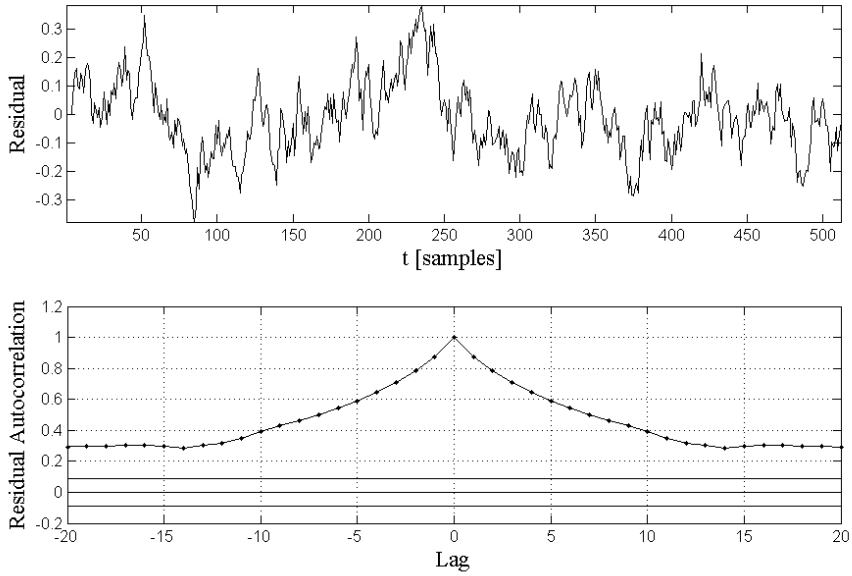
$$\zeta[t] = 0.5\zeta[t-1] + x[t-2] + 0.1x^2[t-1] + e_*[t] \tag{7.5}$$

where colored noise $e_*[t]$ is generated from uniformly-distributed i.i.d. white noise sequence $v[t] \sim U(-0.1, 0.1)$ by lowpass filtering,

$$e_*[t] = -0.9e_*[t-1] + v[t]\,. \tag{7.6}$$

A 1024-point signal observation (input-output pairs) is generated with 512 points for training and 512 points for testing.

Figure 7.4: System 3 with colored noise: Residual and linear correlation test: the horizontal line on the second figure is the 95% confidence interval.



Using the $\mathcal{F}_1$ criterion ($\sigma = 1$) as the fitness function, the estimated model is

$$\begin{aligned}\zeta^p[t] &= 0.5078\zeta[t-1] + 0.9983x[t-2] \\ &\quad +0.0999x^2[t-1] + 0.0009\zeta[t-1]\zeta[t-2]x[t-1]\end{aligned}$$

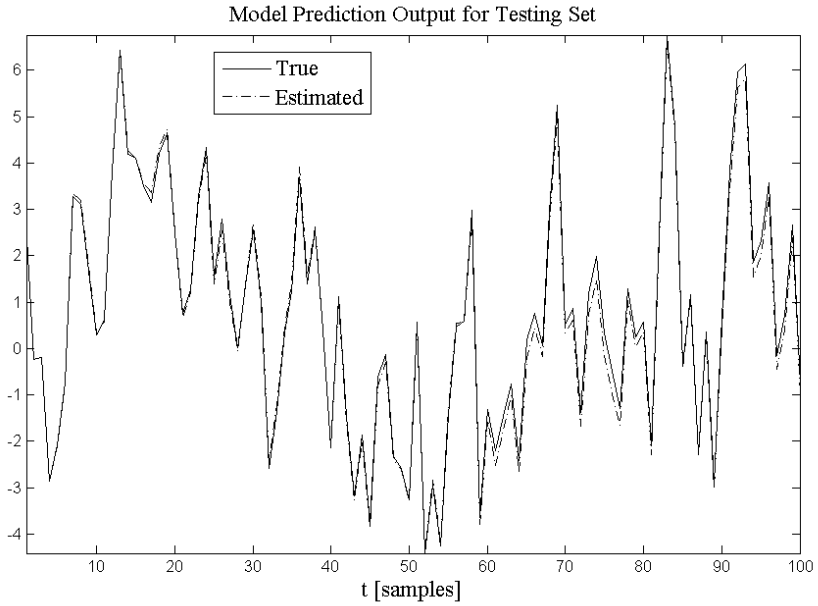
The key structure is detected, and very importantly, the estimated parameters are unbiased in the colored noise environment. The residual, as shown in Fig. 7.4, is colored noise, which corresponds to the true situation. The autocorrelation of the residual measures the linear dependence between the lagged estimated error sequence. The MPO on the 512-point testing dataset is shown in Fig. 7.5. The correlation coefficient is 0.9938.

7.2.3 General noise sequence

7.2.3.1 System 4

In the NARMAX model, it is often the case that the noise is nonlinear and has some cross terms with the input and output [Billings, 2013]. A system is simulated to include nonlinear noise terms

Figure 7.5: System 3 with colored noise: Identification results: True data (continuous curve) and estimated data (dash-dot curve).



as follows

$$\begin{aligned}\zeta[t] =& 0.5\zeta[t-1] + x[t-2] + 0.1x^2[t-1] \\ &+ 0.5e_*[t-1] + e_*[t] + 0.1x[t-1]e_*[t-2]\end{aligned} \tag{7.7}$$

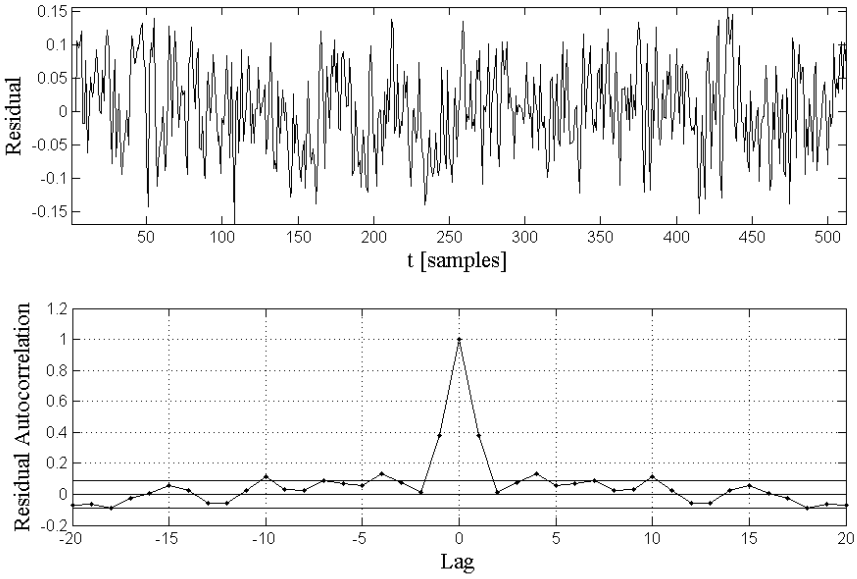
Using the $\mathcal{F}_1$ criterion ($\sigma = 1$) as the fitness function, the algorithm accurately estimated the model

$$\zeta^p[t] = 0.4970\zeta[t-1] + 0.9980x[t-2] + 0.1007x^2[t-1] \tag{7.8}$$

The residual and autocorrelation are shown in the Fig. 7.6. The correlation coefficient on the test set is 0.9992.

The ability to identify the correct model with unbiased parameters without explicit noise modeling makes the algorithm appealing for real system identification, since, in practice, complex noise properties may be unknown.

Figure 7.6: System 4 with general noise: Residual and linear correlation test: the horizontal line on the second figure is the 95% confidence interval.



7.2.4 Comparison study

7.2.4.1 System 5

Finally, to compare the performance of the evolutionary method with some of the most current research, we turn to recent work on Hammerstein systems by Yu, Zhang, and Xie [Yu et al., 2014]. The performance comparison features the following points:

- The disturbance in the Hammerstein model used by Yu *et al.* is restricted to a stationary white output error, whereas the method presented here can identify models with much broader classes of noise with no specific distributional or dependence requirements.

- The Hammerstein model is of fixed parametric form with the objective of the Yu research being the identification of parameter values within this form. The present paper seeks to co-identify the model structure and the parameters within the optimal structure.

- In the present work, there is no need for parameter convergence analysis like that appearing in [Yu et al., 2014, Figures. 4-6], since the parameter convergence is implicit in the set-bounding

algorithm. This convergence has been extensively researched in previous papers [Deller Jr. et al., 1994, Lin et al., 1998].

Each of these factors represents benefits of the present method over the previous work, if the new method can be shown to perform at parity or better on the Hammerstein model relative to the Yu method. This experiment is designed to test that question.

The model studied by Yu *et al.* is a Hammerstein system with a static nonlinearity at the input followed by a linear dynamical system,

$$\begin{aligned} z[t] =& 1 - 0.5x[t] + 0.7x^2[t] \\ \zeta[t] =& 0.3\zeta[t-1] + 0.4\zeta[t-2] \\ &+ z[t] - 2.5z[t-1] + z[t-2] + e_*[t] \end{aligned} \tag{7.9}$$

where the system input $x[t]$ and output $\zeta[t]$ are measurable but the intermediate signal $z[t]$ is unmeasurable. The $x[t]$ is an uncorrelated Gaussian process, $x \sim \mathcal{G}(0, 2)$, and $e_*[t]$ is uniformly-distributed i.i.d white noise. A 512-point signal observation is generated for estimation and 512 points are used for testing. Whereas Yu began with a known model form, a significant difference in the current work is the starting point, at which both the model structure and parameters are unknown, both to be determined from the input, output dataset. With some manipulation, (7.9) can be expressed as the NARX model,

$$\begin{aligned} \zeta[t] =& 0.3\zeta[t-1] + 0.4\zeta[t-2] - 0.5x[t] \\ &+ 0.7x^2[t] + 1.25x[t-1] - 1.75x^2[t-1] \\ &- 0.5x[t-2] + 0.7x^2[t-2] - 0.5 + e_*[t] \end{aligned} \tag{7.10}$$

With (7.10) as the surrogate for Yu's model of (7.9), the evolutionary model selection algorithm is applied to the dataset. The candidate set is modified to contain linear and polynomial nonlinear functions (up to order three, including cross-terms) of present and delayed inputs, and delayed outputs with maximum delay of two (i.e. $\zeta[t-1], \zeta[t-2], x[t], x[t-1], x[t-2]$). A constant regressor is also included for total of 56 possible terms. Using the $\mathcal{F}_1$ criterion ($\sigma = 1$) as the

fitness function, the estimated model is

$$\begin{aligned}\zeta[t] =& 0.2977\zeta[t-1] + 0.3987\zeta[t-2] - 0.5013x[t] \\ &+ 0.6999x^2[t] + 1.2490x[t-1] - 1.7492x^2[t-1] \\ &- 0.4997x[t-2] + 0.6964x^2[t-2] - 0.5056\end{aligned} \tag{7.11}$$

The algorithm has not only correctly estimated the parameter values, but, critically, has properly identified the system model form. The correlation coefficient on the test set is 0.9969.

Beyond demonstrating the critical advantage of automatic determination of the model form with comparable test set correlation, it is difficult to make further quantified comparisons with the results in [Yu et al., 2014]. The previous paper assesses the performance of the featured algorithm on the Hammerstein (output error) model relative to experiments run on an ARX (equation error) model. The method of the present paper, in its focus on the NARMAX model, is certainly amenable to the identification of the ARX special case, but such an experiment would hardly provide a clear comparison with the Hammerstein model results. In contrasting the evolutionary method with the Yu paper, it must also be noted that our method can be applied when the system is corrupted by correlated or even more complex noise. Asymptotic convergence of the parameter estimation is guaranteed by the OBE algorithm [Lin et al., 1998] as part of the evolutionary process.

7.3 Simulation results: Multi-objective optimization

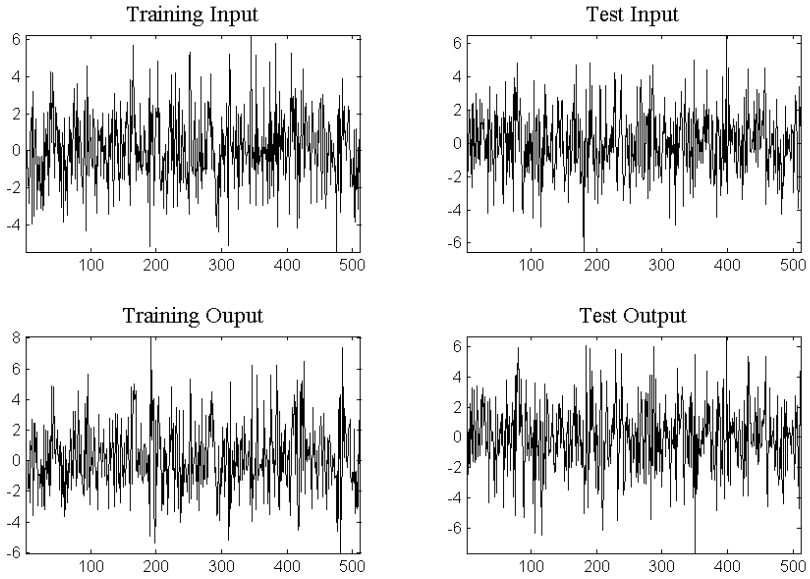
7.3.1 White noise disturbances

Consider the NARX system

$$\begin{aligned}\zeta[t] =& -0.6\zeta[t-1] - 0.16\zeta[t-2] + 0.5x[t-1] \\ &- 0.24x^2[t-2] + 0.1\zeta[t-1]x[t-2] + e_*[t],\end{aligned}$$

in which the input sequence x is an uncorrelated Gaussian process, $x \sim \mathcal{G}(0, 2)$, and $e_*[t]$ is uniformly-distributed i.i.d white noise, $e_* \sim \mathcal{U}(-0.2, 0.2)$. The noise is assumed unknown *a priori* and the noise bound is to be estimated. To assess the performance of the evolutionary identification method, the system (7.3.1) is used to generate 1024 observations (input-output pairs).

Figure 7.7: System 1 input and output signals. The horizontal axis is the sample index, and the vertical axis is the amplitude.



The observation dataset is divided into 512-point training and test sets. The training set is used to identify the model, while the test set is used to assess the performance of the resulting model.

7.3.1.1 Empirical combination of objective functions

To assess the assessment of benefits of bi-objective optimization, we first use a single-objective GA with objective function (6.22) as a fitness measure to model the system (7.3.1). By setting $\lambda = 1$ in (6.22), the two-objective fitness measure is effectively reduced to a single objective. Using a single-objective GA to search for the solution with smallest FPE value, only one estimated model is found (detailed algorithm in [Yan et al., 2013, Yan et al., 2014]),

$$\begin{aligned}\zeta^p[t] = &-0.5964\zeta[t-1] - 0.1568\zeta[t-2] + 0.4948x[t-1] \\ &- 0.2382x^2[t-2] + 0.0979\zeta[t-1]x[t-2] \\ &- 0.0066\zeta[t-2]x[t-1] - 0.0040\zeta[t-1]x^2(t-2)\end{aligned} \tag{7.12}$$

The estimated model contains seven terms, in contrast with the five terms in the "true system" of (7.3.1). From the result, we can see that all five regressor functions in the system have been correctly

identified with accurate parameter estimates. However, there are two relatively superfluous terms in the model with associated parameter values that are an order of magnitude smaller than the key regressors.

7.3.1.2 Multi-objective optimization

To visualize the landscape of the objective space, 500 models are randomly generated and the separate objective values on training dataset inherent in $(V(\boldsymbol{\theta}, \boldsymbol{\varphi}), \|\boldsymbol{\theta}\|_0)$ are plotted in Fig. 7.8 for these experiments. Note that this set of 500 models does not necessarily include the "correct" or best-fit model to fit these particular data. A clear trade-off between the number of parameters (abscissa) and the residual squared error (ordinate) is evident in the plot. The point of best achievable performance if the optimization criteria were independent appears as a heavy dot (the ideal point) at (1, 0.0130) on the plot (near axes intersection).

The multi-objective GA is then applied to the model selection problem. The final efficient points (non-dominated solutions depicted in objective space) are shown in Fig. 7.9. The horizontal axis is the first objective – number of parameters, and the vertical axis is the second objective – residual squared error. For clear visualization, one outlier point is not plotted in the figure, but nevertheless is found by the multi-objective optimization, $(0.0875, 5)$. For convenience, we denote this algorithmically-determined non-dominated (Pareto) set of models by $\mathcal{M}$.

7.3.1.3 Selecting the final model – Regressor significance

It remains to select a particular model from $\mathcal{M}$. A decision-making strategy is based on the reasoning that essential regressors will appear frequently, and with large parameters, among the candidate models. A regressor significance measure, say ω_q, for each $\varphi_q \in \Xi_\varphi$ is defined as the normalized sum of the absolute parameter values for φ_q over all of the models in $\mathcal{M}$. The resulting set of significance measures, $\left\{\omega_q \in [0, 1]\right\}_{q=1}^{55}$, for the current experiment is illustrated in Fig. 7.10. Five significant regressors are apparent on the plot with significance values $\omega_q > 0.1$.The remaining 50 regressors in Ξ_φ have significance values $\ll 0.1$. The regressors corresponding to significant

Figure 7.8: Landscape of the objective space (horizontal axis is the number of parameters, vertical axis is the residual squared error): the '×' points are uniform randomly generated binary strings, and the '•' point is the ideal point.

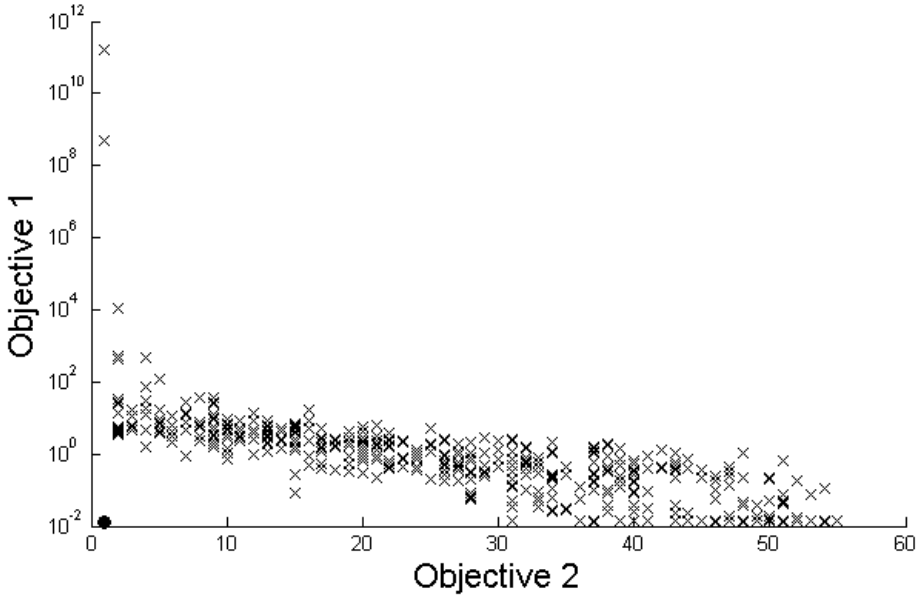


Figure 7.9: Pareto efficient points (horizontal axis is the number of parameters, vertical axis is the residual squared error). The point $(0.0875, 5)$ is not plotted for visualization purpose.

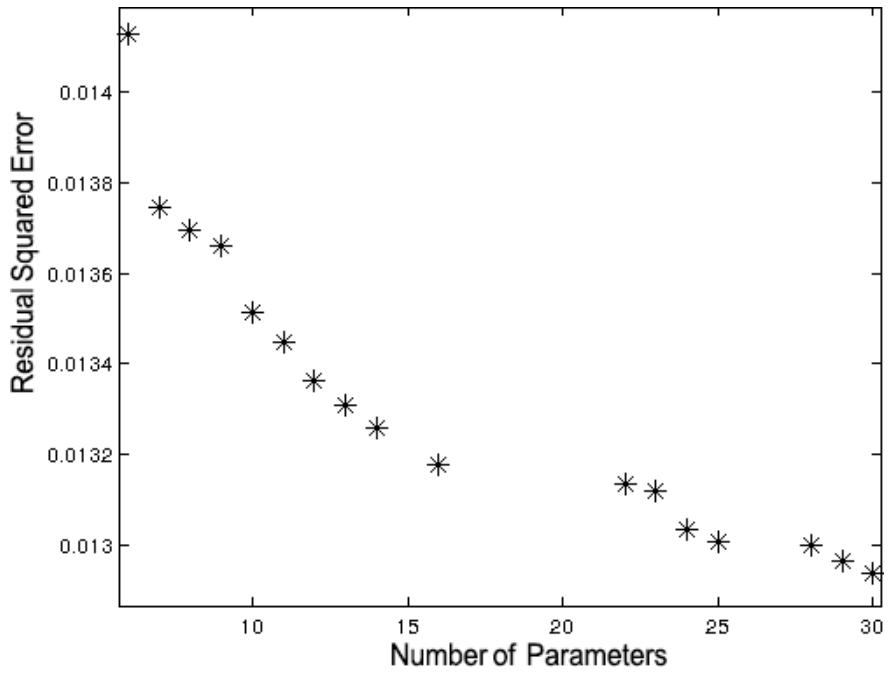
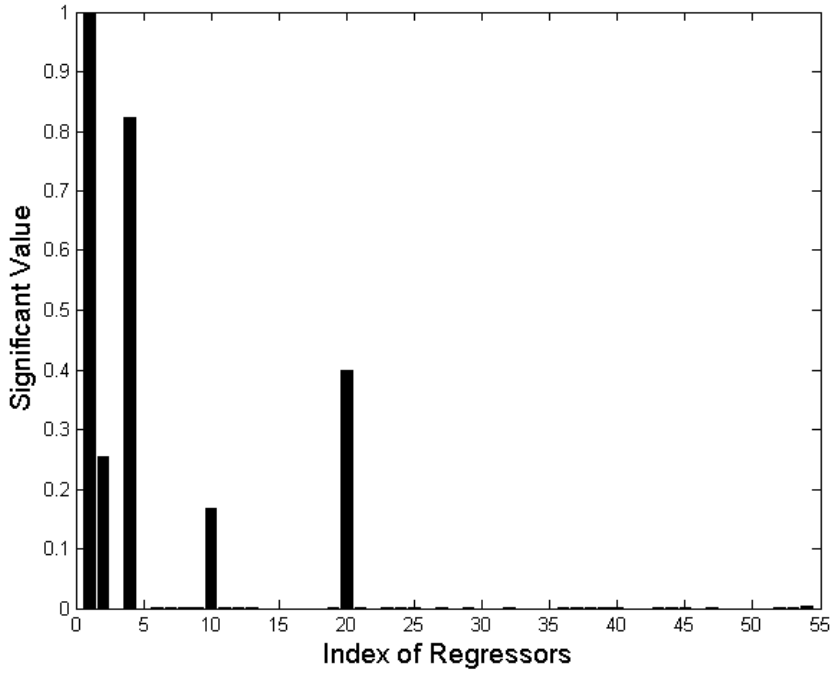


Figure 7.10: The significant value of each term (horizontal axis is the index of regressors, vertical axis is the significant value).



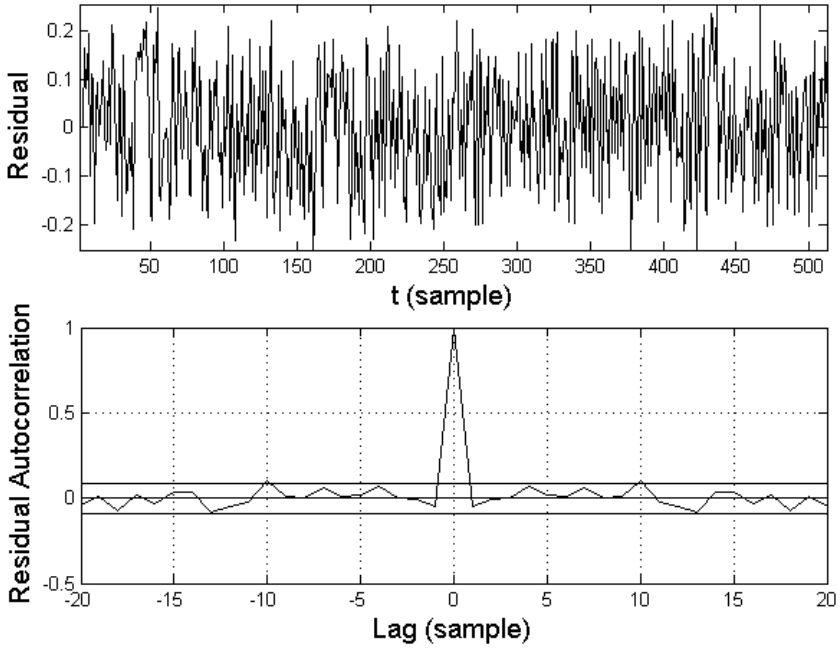
values from left to right on the figure are $\zeta[t-1], \zeta[t-2], x[t-1], \zeta[t-1]x[t-2], x^2[t-2]$, resulting in the final estimated model

$$\begin{aligned}\zeta^p[t] = &-0.5983\zeta[t-1] - 0.1599\zeta[t-2] + 0.5143x[t-1] \\ &- 0.2390x^2[t-2] + 0.0995\zeta[t-1]x[t-2]\end{aligned} \tag{7.13}$$

It is seen that, the structure (5 regressors) has been exactly recovered as in Eqn. 7.3.1. Moreover, the estimated parameters are close to their true values. Multi-objective optimization and the specified decision-making technique produce a superior model structure compared to that obtained using combined single-objective optimization. The superior performance of the multi-objective optimization results from the exploration of multiple solution trajectories with these results integrated to determine a bi-optimal solution. Moreover, the asymptotic convergence of the parameter estimation is guaranteed by the OBE algorithm [Lin et al., 1998] as part of the evolutionary process.

As further evidence of model quality, the residual and its autocorrelation for the designed model are shown in Fig. 7.11. The results are consistent with the white disturbances in the true system. The result is shown in Fig. 7.12. For clear visualization, only 100 samples are shown. The selected

Figure 7.11: System (7.3.1) residual and linear correlation tests. Horizontal lines inn the lower figure show the 95% confidence interval.

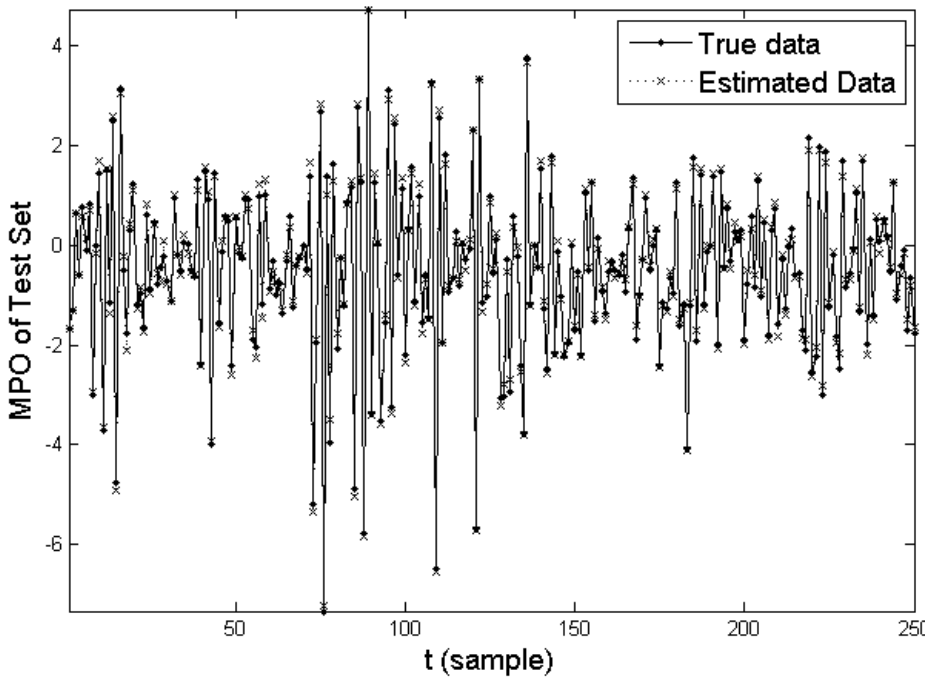


model exhibits excellent tracking ability. The correlation coefficient of the true output and MPO on the test dataset is 0.9953. The model identified is very near to the true system (and true Pareto front or efficient solution).

7.3.1.4 Comparison with classical method

To compare with another sparse linear regression method, we also applied the Orthogonal Matching Pursuit (OMP) algorithm to identify the system model. OMP is a popular greedy algorithm for sparse model identification. The basic principle of the algorithm is to iteratively find the support set of the sparse vector and reconstruct the values of the vector using the restricted support least square estimate. However, since OMP uses least-square-error estimates, the parameter estimation is biased in dependent noise. The number of samples processed is 512. The identified model using

Figure 7.12: System Identification results: True data (continuous curve with '·') and estimated data (dashed curve with '×').

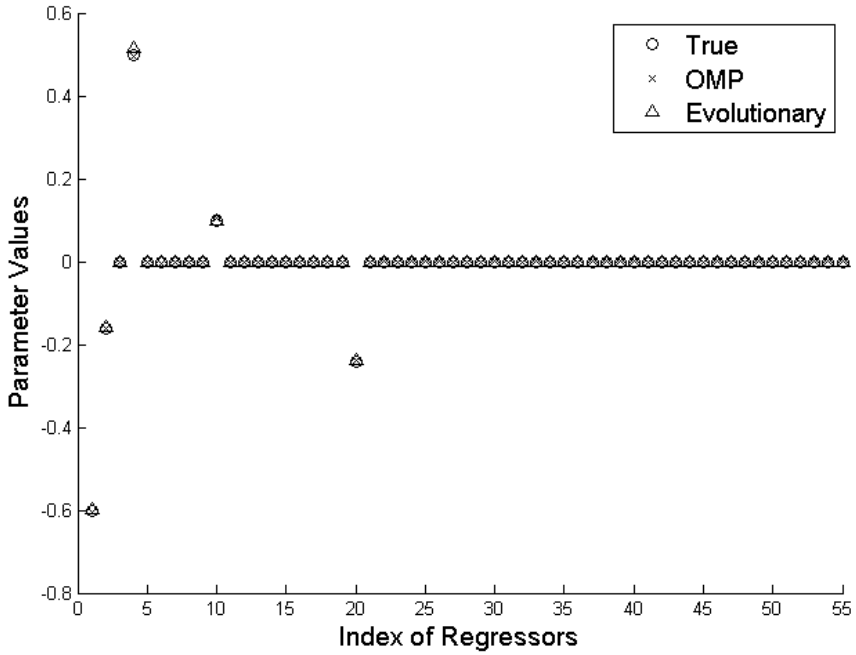


OMP is shown,

$$\begin{aligned}\zeta^{p}[t] = &-0.6059\zeta[t-1] - 0.1619\zeta[t-2] + 0.4941x[t-1] \\ &- 0.2399x^{2}[t-2] + 0.1037\zeta[t-1]x[t-2] \\ &+ 0.0018\zeta^{2}[t-2]\zeta[t-3] + 0.0014\zeta^{2}[t-2]\zeta^{2}[t-3] \\ &+ 0.0014x[t-1]x^{2}[t-2]\end{aligned} \tag{7.14}$$

Terms with arameters less than or equal to 10^{-4} are ignored. It can be seen in Eqn. 7.14 that three extra terms with small parameter values are added to the system model. Compared with this OMP algorithm, the evolutionary multi-objective identification method is superior in the sense that it identifies a sparser and more accurate model. The results are further compared in Fig. 7.13. As seen from the figure, both the evolutionary multi-objective identification algorithm and OMP algorithm perform well in estimating the sparse system model in the white noise condition.

Figure 7.13: System (7.3.1) – Comparison of models. The horizontal axis is the index of regressors, and the vertical axis is the true or estimated parameter values: True parameter values are indicated by ∘, estimated system model using evolutionary algorithm △, and estimated model using OMP ×.



7.3.2 Colored noise disturbances

The evolutionary model selection algorithm produces unbiased model estimates of NARMAX[1] systems with colored noise disturbances. In this second study case, the system is contaminated by colored noise in an equation-error configuration. There are no nonlinear interactions between the noise and signal measurements. The target is a five-parameter NARMAX system with added colored noise,

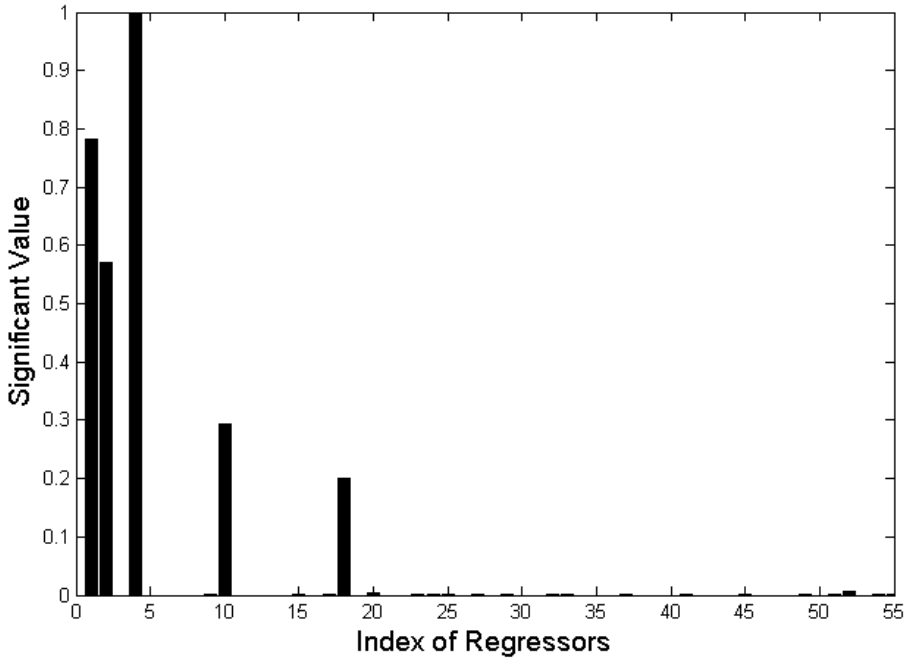
$$\begin{aligned}\zeta[t] = &- 0.8\zeta[t-1] - 0.6\zeta[t-2] + x[t-1] \\ &- 0.2x^2[t-1] + 0.3\zeta[t-1]x[t-2] + e_*[t]\end{aligned} \tag{7.15}$$

in which the exogenous input sequence x is an uncorrelated Gaussian process, $x \sim \mathcal{G}(0, 2)$, and colored noise $e_*[t]$ is generated by filtering an i.i.d. white noise sequence $v[t] \sim \mathcal{U}(-0.5, 0.5)$,

$$e_*[t] = -0.8e_*[t-1] + v[t]\,. \tag{7.16}$$

[1] "NARMAX" is used in a generalized sense in which the disturbances are colored without any attempt to model the dynamics of the noise correlation.

Figure 7.14: The significance value of each term (horizontal axis is the index of regressors, vertical axis is the significance value).



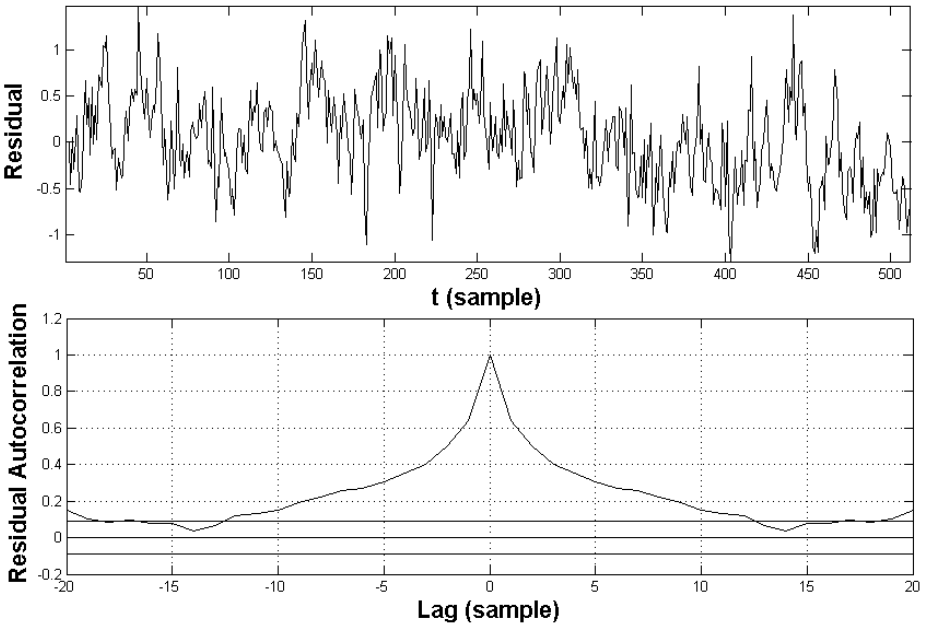
The noise is assumed unknown. A 1024-point signal observation (input-output pairs) is generated with 512 points each for training and testing.

The procedure described for the first experiment is followed; the significant values of regressors are shown in Fig. 7.14. The importance of each regressor can be observed directly from the figure. Regressors that contribute significantly ($\omega_q > 0.1$) are included in the final model. The final estimated model is

$$\zeta[t] = -0.7889\zeta[t-1] - 0.5795\zeta[t-2] + 0.9116x[t-1]$$
$$- 0.2200x^2[t-1] + 0.3042\zeta[t-1]x[t-2]$$

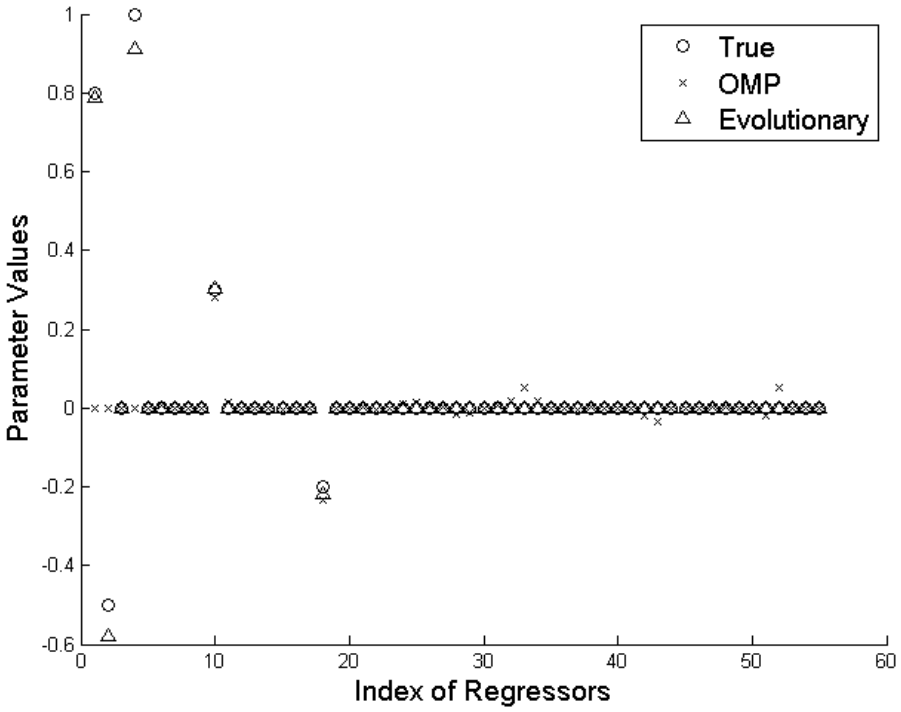
From the result, it can be observed that the structure is detected, and the estimated parameters approximate the true values even in the large, colored noise environment and limited sample size. The residual (Fig. 7.15) is colored noise, corresponding to the true process. The autocorrelation of the residual measures the linear dependence in the lagged estimated error sequence. Moreover, the correlation coefficient between the true and estimated outputs is 0.9931.

Figure 7.15: System (7.15) residual and linear correlation tests. Horizontal lines in the lower figure show the 95% confidence interval.



Comparison with classical method: The OMP algorithm is again used to identify a sparse model from the training dataset. The parameter vector $\boldsymbol{\theta}$ is a 5-sparse (5 nonzero parameters) vector of length 55. The number of samples processed is 512. The OMP algorithm fails to identify a sparse model. Many regressors that are not in the original system are incorrectly added to the model, while several important regressors are missing, such as $\zeta[t-1], \zeta[t-2]$ and $x[t-1]$, as illustrated in Fig. 7.16. As can be seen from the figure, contrary to the OMP algorithm, the proposed evolutionary multi-objective algorithm correctly selects the model structure as well as identifies approximate parameters. In other words, the proposed algorithm is robust to high power and colored noise conditions.

Figure 7.16: System (7.3.2) – Comparison of models. The horizontal axis is the index of regressors, and the vertical axis is the true or estimated parameter values: True parameter values are indicated by ∘, estimated system model using evolutionary algorithm △, and estimated model using OMP ×.

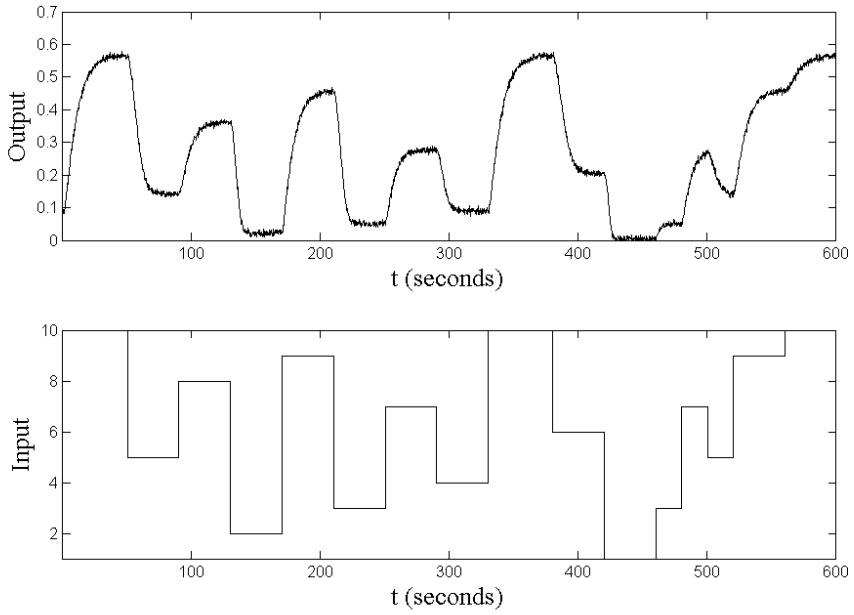


7.4 Practical datasets

To further demonstrate the performance of the proposed algorithm, we apply the method to the classical two-tank system. The cascaded two-tank system is a benchmark for nonlinear system identification [Wigren and Schoukens, 2013]. The system has significant nonlinear dynamics and it is known that the system can be modeled by two coupled nonlinear ordinary differential equations [Wigren, 2006]. In this experiment, we use the dataset from the MATLAB^{TM} System Identification Toolbox developed by [Ljung, 2007]. The dataset contains 3000 input-output observations of a two-tank system generated at a sampling rate of 5 Hz. The input $x(t)$ is the voltage (volt) applied to a pump, which generates an inflow to the upper tank. A small hole at the bottom of this upper tank yields an outflow into the lower tank. The output $y(t)$ of the two-tank system is the liquid level (meter) of the lower tank. The input-output data are plotted in Fig. 7.17.

We split the data into three subsets of equal size. The first subset is for training and the remaining

Figure 7.17: Two-tank system input and output data.



two are for testing. Following the preprocessing procedure suggested by [Ljung, 2007], a linear ARX model is fitted to the training data first to select model orders (the numbers of past outputs and past inputs) using least square error optimization with the Akaike Information Criterion [Akaike, 1969],

$$\text{AIC} = \log\left\{V(1 + 2dN^{-1})\right\}, \tag{7.17}$$

where N is the data length, d is the number of regressor signals, and $V = \frac{1}{N}\sum_1^N (y - y^p)^2$, which is the average of squared one-step prediction errors. The selected model orders are $y(t-1)$, $y(t-2), \cdots, y(t-5)$ and $x(t-3)$. The detail of model order determination can be found in [Ljung, 2007]. Thus, the candidate set is designed to contain linear and polynomial nonlinear functions (up to order three, including cross-terms) of the selected model orders. The set contains 83 regressors and thus yields $(2^{83} - 1)$ possible estimation models.

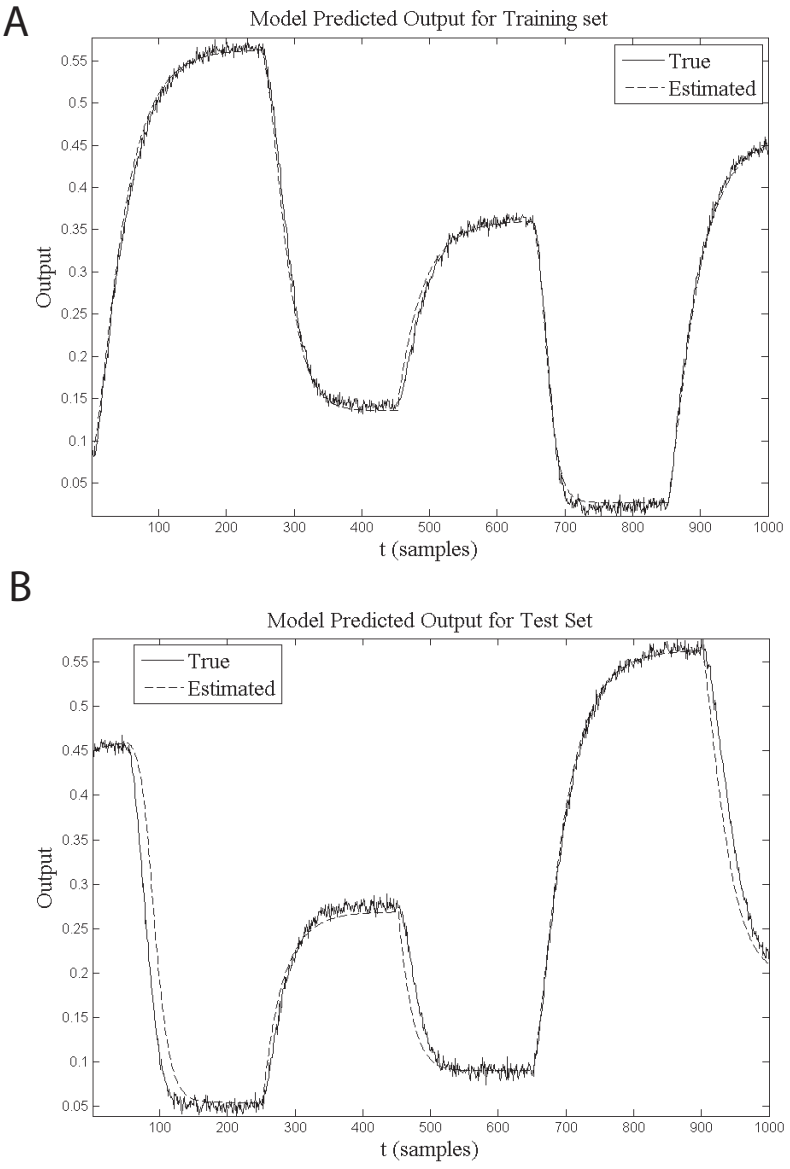
Next, the evolutionary model selection algorithm with (6.22) as fitness function is then applied to identify the system model from the input-output training data. The MPO results on the 1000-point training and testing datasets are shown in Fig. 7.18. The correlation coefficients are 0.9947 and 0.9867 on the two test sets. Moreover, as a goodness-of-fit measure [Ljung, 1999], the normalized

mean squared error (NMSE) is also calculated to evaluate the identified model. The NMSE is defined as

$$\text{NMSE} = 100 \times (1 - \|y - \hat{y}\| \|y - \overline{y}\|^{-1}), \tag{7.18}$$

where $\overline{y}$ is the mean of the sequence, $\hat{y}$ is the MPO of the model. The NMSE on the training set is 94.3137. The NMSEs on test sets one and two are 88.6492 and 84.0115, respectively. The evolutionary model selection algorithm demonstrates good performance on both datasets (Fig. 7.18).

Figure 7.18: Two-tank system identification. True data (continuous curve) and estimated data (dash-dot).



7.5 Application to causality analysis

Advances in neuroimaging and electrophysiological recording have produced a wealth of image and signal data from different brain regions. Research has revealed that cortical areas perform unique elementary functions, but that complex functions require the integrated action of many areas distributed throughout brain regions [Luria, 2012]. A complex function is a system of interrelated processes directed toward the performance of a particular task that is implemented by functionally related cortical areas [Bressler, 1995]. Interconnectivity of cortical areas is essential for high-level functions underlying cognition, and assessment of this interdependence can provide a better understanding of neural systems [Luria, 2012, Bressler, 1995, Sporns et al., 2004, Mao et al., 2011, Berényi et al., 2014, Druckmann and Chklovskii, 2010, Freeman et al., 2014].

Recent research has sought to use these recordings to infer functional and effective connectivity among areas in the brain. The research attempts to identify sets of brain regions that are simultaneously involved in the processing of a task. Specifically, effective connectivity describes how one neural system affects another, including not only a measure of the strength of the connection, but also the direction of information flow [Friston, 2011, Friston et al., 1994]. The main approaches for quantifying effective connectivity of recorded time series at different brain locations are model-based and information-theoretic measures [Pereda et al., 2005, Liu and Aviyente, 2012]. Granger Causality Analysis (GCA), a traditional model-based approach for quantifying effective connectivity, is a widely used measure to describe the directed information flow between two time series [Granger, 2004, Seth, 2010]. GCA describes the causal relationship between two time series by quantifying the degree of predictability of one time series available in the history of the other, and it is ordinarily applied to linear time series models. For instance, Hesse *et al.* [Hesse et al., 2003] used adaptive GCA on EEG data and showed that brains generate dense interactions from posterior to anterior cortical sites when processing conflict situations. Kamiński *et al.* [Kamiński et al., 1997] identified the centers from which EEG activity is spreading during sleep and wakefulness. The influence of subcortical structures was manifested by propagation of activity from the fronto-central region during sleep.

Despite useful results, GCA is fundamentally limited because it can only measure linear causal relationships, since it is based on a linear prediction framework. The approach may be misleading when applied to signals like EEG traces that are known to have nonlinear dependencies. It may ultimately be determined that the brain achieves a quasi-linear system operation in the transmission of signals by a sort of global averaging process. However, given that the transmission is fundamentally nonlinear at the level of individual cell dynamics, it is essential to explore the possibility that information is encoded in highly-nonlinear ways in the brain. There is some existing work which extends GCA to its nonlinear version based on kernel methods [Stramaglia et al., 2011, Marinazzo et al., 2008]; but often, designing an appropriate kernel function can be difficult.

Here we investigate the performance of the evolutionary model selection and estimation algorithm in the problem of determining nonlinear causal connectivity.

7.5.1 Methods

7.5.1.1 Bivariate time series

The fundamental idea of causality can be traced back to Norbert Wiener, who conceived the following notion: if the prediction of one time series could be improved by incorporating the knowledge of a second one, then the second series is said to have a causal influence on the first [Ding et al., 2006]. Later, Granger formalized the prediction idea in the context of linear regression models [Granger et al., 2000].

Linear GCA Let $\{x[t]\}$ and $\{\zeta[t]\}$ be stationary time series. First, we model the dynamics of ζ by an autoregressive model of order p (AR),

$$\zeta[t] = \sum_{i=1}^{p} a_i \zeta[t-i] + e[t]. \tag{7.19}$$

Second, the autoregression is agumented with lagged values of x (ARX),

$$\zeta[t] = \sum_{i=1}^{p} a_i \zeta[t-i] + \sum_{j=1}^{q} b_j x[t-j] + e[t]. \tag{7.20}$$

The lagged x are retained only when their individual t-statics are significant, provided that the coefficients are jointly significant in predicting ζ (determined by F-test) [Altman, 1990, Devore,

2015]. The null hypothesis "x does not Granger-cause ζ" is true only when no lagged values of x are retained in the regression. Exchanging the roles of x and ζ, one can similarly test reverse causality.

GCA may fail to uncover the true relations when there are nonlinear interactions. Moreover, the conventional method assumes the noise $e[t]$ to be white noise. Instead, we use the modeling framework designed in previous chapters for modeling more general connections of time series. The framework greatly generalizes model possibilities. The method not only detects both linear and nonlinear causal relationships, but is also robust to unknown correlated, or nonlinearly dependent noise [Yan and Deller Jr., 2015, Yan et al., 2013, Yan et al., 2014]. These merits makes the algorithm transformational for practical biomedical data analysis.

Nonlinear GCA Instead of fitting AR and ARX models as in Eqn. 7.19 and Eqn. 7.20, we fit a polynomial nonlinear-autoregressive-moving-average (NARMA) and polynomial NARMAX model [Billings, 2013],

$$\zeta[t] = \sum_{i_1=1}^{p} \theta_{i_1} u_{i_1}[t] + \sum_{i_1=1}^{p} \sum_{i_2=i_1}^{p} \theta_{i_1 i_2} u_{i_1}[t] u_{i_2}[t] + \cdots + \sum_{i_1=1}^{p} \cdots \sum_{i_\ell=i_{\ell-1}}^{p} \theta_{i_1 \cdots i_2} u_{i_1}[t] \cdots u_{i_\ell}[t]. \tag{7.21}$$

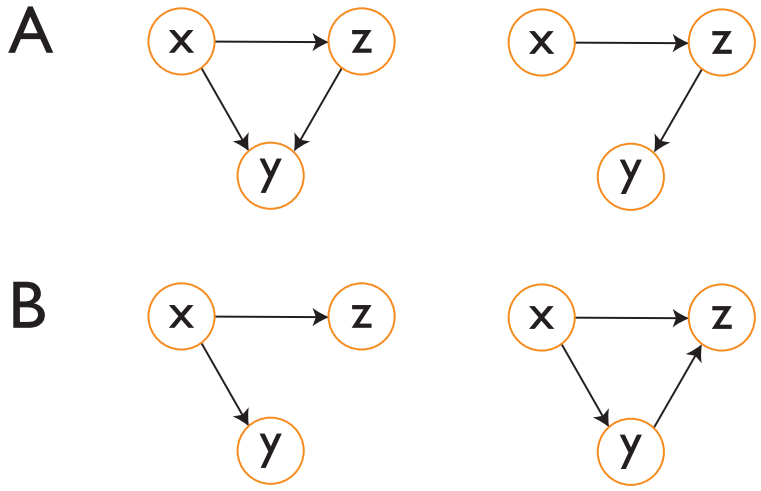
where $u_i = \{\zeta[t-i], e[t-i]\}$ in the reduced model is used for the first step, and $u_i = \{\zeta[t-i], x[t-i], e[t-i]\}$, the full model, is used for the second. Following the same procedure for coefficient significance tests as in linear GCA, the null hypothesis is accepted only when no lagged values of x are retained in the regression.

To identify a sparse model from Eqn. 7.21, we use the developed evolutionary biologically-motivated method for both the selection of the correct regressors and estimation of the model parameters.

7.5.1.2 Multivariate time series

GCA can be readily extended to the multivariate case $X = [x_1, x_2, \ldots, x_n]^T \in \mathbb{R}^n$ by estimating a vector autoregressive model. For instance, x_2 causes x_1 if lagged observations of x_2 help predict

Figure 7.19: The connection regimes that are ambiguous under pair-wise GCA, but can be distinguished by conditional GCA. A: the connection from process X to process Y is directed; right: the connection from process X to Y is mediated through Z. B: process X drives the other two processes Y and Z with differential time delays, while there is no information flow from Y to Z. Right: Y and Z have a common source, but are not directly related.



x_1 when lagged observations of all other variables $x_3 \dots x_n$ are also taken into account,

$$x_i[t] = \sum_{i_1=1}^{p_1} \theta_{1,i} x_1[t - i_1] + \sum_{i_1=1}^{p_2} \theta_{2,i} x_2[t - i_2] \cdots + \sum_{i_n=1}^{p_n} \theta_{n,i} x_n[t - i_n] + e[t]. \tag{7.22}$$

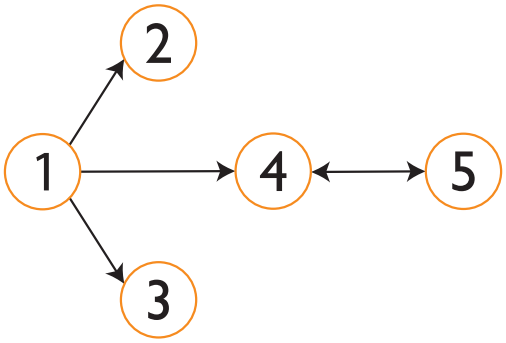
where x_i is a time series. This multivariate extension, sometimes referred to as conditional GCA, is extremely useful because repeated pairwise analyses among multiple variables can sometimes give misleading results. For example, a repeated bivariate analyses would be unable to disambiguate the two connectivity patterns in Fig. 7.19. In the first case, bivariate analyses cannot distinguish whether the connection between the two time series is direct or mediated. Another example is when one process drives the other two processes with differential time delays.

Similar to the bivariate time series case, we have extended the conventional linear model to a nonlinear version by using an LTiIP model (NARMAX),

$$x_i[t] = \sum_{i_1=1}^{p} \theta_{i_1} u_{i_1}[t] + \sum_{i_1=1}^{p} \sum_{i_2=i_1}^{p} \theta_{i_1 i_2} u_{i_1}[t] u_{i_2}[t] + \cdots + \sum_{i_1=1}^{p} \cdots \sum_{i_\ell=i_{\ell-1}}^{p} \theta_{i_1 \cdots i_2} u_{i_1}[t] \cdots u_{i_\ell}[t]. \tag{7.23}$$

where $u_i = \{x_1^{t-i}, \dots x_{j-1}^{t-i}, x_{j+1}^{t-i}, \dots, x_n^{t-i}, e^{t-i}\}$ for the reduced model, and for the full model $u_i = \{x_1^{t-i}, \dots, x_n^{t-i}, e^{t-i}\}$.

Figure 7.20: Schematic illustration of the system: a simulated five-node structurally connected with different time delays.



The evolutionary system modeling algorithm can then be used to select regressors and to estimate the parameters.

7.5.2 Simulation results

7.5.2.1 Colored noise disturbances

As a concrete illustration, we simulated a five-node oscillatory network structurally connected with different delays. The network involves the following multivariate AR model,

$$x_1[t] = 0.95\sqrt{2}x_1[t-1] - 0.9025x_1[t-2] + e_1[t] \tag{7.24}$$

$$x_2[t] = 0.5x_1[t-2] + e_2[t] \tag{7.25}$$

$$x_3[t] = -0.4x_1[t-3] + e_3[t] \tag{7.26}$$

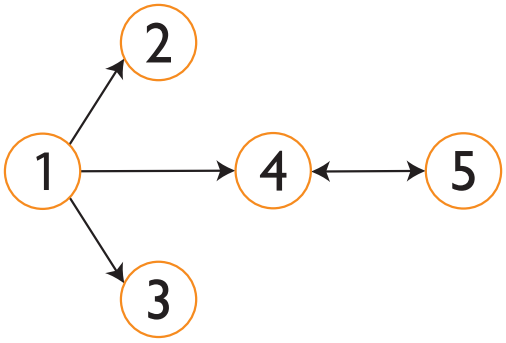
$$x_4[t] = -0.5x_1[t-2] + 0.25\sqrt{2}x_4[t-1] + 0.25\sqrt{2}x_5[t-1] + e_4[t] \tag{7.27}$$

$$x_5[t] = -0.25\sqrt{2}x_4[t-1] + 0.25\sqrt{2}x_5[t-1] + e_5[t] \tag{7.28}$$

where e_1, e_2, e_3, e_4, e_5 are independent noise processes. The structure of the network is illustrated in Fig. 7.20. This example has been studied by Ding *et al.* [Ding et al., 2006], Seth [Seth, 2010], and Baccala and Sameshima [Baccalá and Sameshima, 2001].

Previous work assumes the noise processes to be white. For instance, Seth [Seth, 2010] uses the ordinary least square algorithms (OLS) to identify the model structure. Even though the algorithm uncovers the correct connection schema under a white noise scenario, it cannot identify correct

Figure 7.21: Causal connections identified by conventional OLS for System (7.24). A: Identified system connection using OLS - white noise scenario. The GCA using the OLS algorithm identified the correct connections. Green (red) indicates uni (bi-) directional causality. Width of connector indicates strength. B: Identified system connection using OLS - colored noise scenario. The GCA using OLS algorithm identified incorrect connections.



connections when the noise sources are correlated (Fig. 7.21. The colored noise processes were generated by processing the white noise using five different linear filters,

$$e_1[t] = 0.6e_1[t-1] + v_1[t] \tag{7.29}$$

$$e_2[t] = 0.9e_2[t-1] + v_2[t] \tag{7.30}$$

$$e_3[t] = v_3[t] + 2v_3[t-1] \tag{7.31}$$

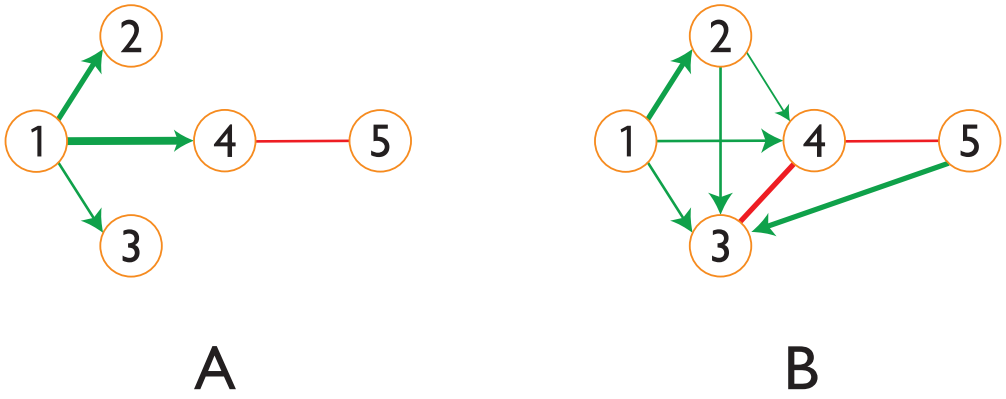
$$e_4[t] = v_4[t] + 1.5v_4[t-1] \tag{7.32}$$

$$e_5[t] = 0.8e_5[t] + v_5[t] \tag{7.33}$$

where $v_1[t], v_2[t], v_3[t], v_4[t], v_5[t]$ are independent white noise.

When the proposed algorithm is applies to this multivariate model, the identified network structure is shown in Fig. 7.22. The proposed method is seen to identify the correct model causality connections even in the colored noise scenario.

Figure 7.22: Identified system connection using proposed algorithm - colored noise scenario. The proposed algorithm identifies the proper connections even in colored noise.



7.5.2.2 Nonlinear network

Further, we added nonlinearity to the system to illustrate the ability to identify nonlinear connections by using our identification framework.

$$x_1[t] = 0.3x_1[t-1] - 0.2x_1[t-2] + e_1[t] \tag{7.34}$$

$$x_2[t] = 0.25x_1[t-2]x_2[t-2] + 0.2x_2[t-1] + e_2[t] \tag{7.35}$$

$$x_3[t] = -0.4x_1[t-2] + e_3[t] \tag{7.36}$$

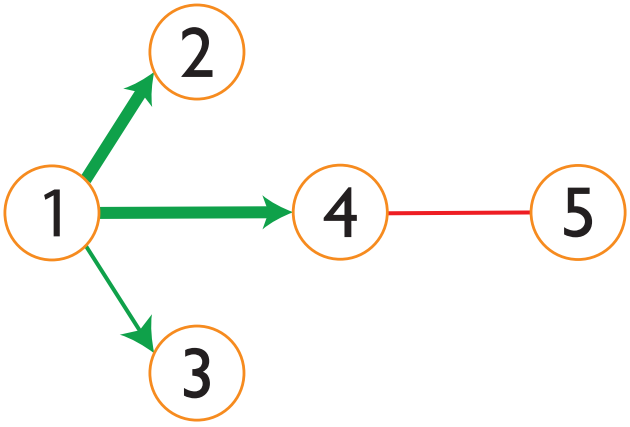
$$x_4[t] = x_1[t-2] + 0.1x_1^2[t-1] + 0.5x_4[t-1] + e_4[t] \tag{7.37}$$

$$x_5[t] = 0.5x_1[t-1] - 0.24x_1[t-2] + 0.1x_1[t-2]x_5[t-1] \tag{7.38}$$

$$- 0.6x_5[t-1] - 0.16x_5[t-2] + e_5[t] \tag{7.39}$$

For comparison, the identified causal connections using OLS and our method are shown in Fig. 7.23. As can be seen from the figure, the conventional OLS method missed the nonlinear connection between time series x_1 and x_2, while our method not only identified linear connections, but also nonlinear connections.

Figure 7.23: Causal connections identified for System 7.34. The proposed algorithm identifies the nonlinear connections. A: Identified system connection using OLS. The GCA using the OLS algorithm incompletely identified the connections, missing nonlinear connections. B: Identified system connection using proposed algorithm. The proposed algorithm identified all connections.



CHAPTER 8

CONCLUSION FOR PART II

In this part of the dissertation, an algorithmic framework has been presented in which nonlinear models can be identified without completely abandoning well-understood and tested methods developed for LTI models. LTIiP models can be arbitrarily nonlinear in signal interactions, while retaining the ability to exploit LTI identification methods.

The LTIiP modeling problem has been addressed with two innovations that lead to sparse models with unbiased parameter estimates, even in dependent noise perturbations. Evolutionary optimization is well adapted to select model structures. Moreover, unbiased parameter solutions in dependent noise are obtained using a set-theoretic identification approach which produces sets of solutions that are consistent with *a priori* model assumptions, rather than conventional point solutions that result from statistical optimization. Properties of set solutions are used to assess the viability of various regressor combinations that are treated as components of the phenotype in the evolutionary competition. Models are placed in evolutionary competition in genetic algorithms adapted for the problem setup. The underlying hypothesis of the work is that nonlinear models that are highly tuned to observed signal data will emerge with sufficient frequency to allow the models to selectively reproduce and mutate within this competition, approaching optimal fits to the data.

While conventional signal processing work in modeling has focused on the estimation of parameters, frequently with white-noise assumptions, the method reported here simultaneously addresses model selection and parameter estimation, with concurrent optimization of model size and fidelity, even in dependent disturbances. The algorithm exhibits excellent performance on simulated studies involving uncorrelated and correlated disturbances and nonlinear signal-noise interactions.

The ability to identify a sparse nonlinear model with unbiased parameters under complex noise conditions makes the algorithm transformational for practical biomedical data analysis. The algorithm is applicable for modeling nonlinear, effective brain connectivity of the cognitive control

networks. Advances in neuroimaging and electrophysiological recording have produced a wealth of image and signal data from different brain regions. One goal is to identify sets of brain regions that are simultaneously involved in the processing of a task. Given that the brain transmission is fundamentally nonlinear at the level of individual cell dynamics, exploring whether information is encoded in highly-nonlinear ways in the brain is essential.

8.1 Contributions

The major contributions of this research are the following. This research has:

1. Introduced an algorithmic framework for simultaneously solving the model-structure determination and parameter estimation problems.

2. Produced a new algorithm that integrates three modeling and identification strategies: LTIiP models, set-based parameter estimation, and evolutionary algorithms for optimization.

3. Achieved an effective balance between model accuracy and dimension by using an evolutionary algorithm with bi-objective optimization. Multiple solutions that are Pareto optimal are studied and used for selecting the best model.

4. Provided extensive simulations that show the behavior and characteristics of the algorithm in the different noise cases and show its performance in these conditions. Applied the algorithms successfully to a real-world problem of dynamical system modeling (two-tank system).

5. Extended the conventional Granger causality analysis to a nonlinear version based on our LTiIP model identification framework. Under this formulation, the algorithm is robust to different noise conditions.

6. Identified a possible application for modeling nonlinear, effective connectivity of the brain cognitive control network.

8.2 Limitations of the Approach

There are a few limitations of this approach:

- It is important to keep in mind of the problem of overfitting when using this method. To avoid overfitting, we separate the data into training dataset and test dataset, and evaluated the model performance using test dataset. In practice, other methods could also we used, for instance, cross-validation.
- The nonlinear autoregressive model could be hard to interpret.

8.3 Future Work

Many interesting points of research can be followed. The future work to follow after this research will be:

- Investigating theoretically the effect of changing regressors on the parameter set when evolving nonlinear models. Exploring the model information contained in the parameter set, and its potential for alternative set measures.
- Test algorithms on modeling causal effective connectivity of the brain cognitive control network, and identifying possible applications on the research of neuronal signal modeling and brain information flow at Janelia Research Campus.

APPENDICES

APPENDIX A

INTEGRATING FACTOR TRICK

Calcium dynamics Eqn. 2.5

$$\frac{dc}{dt} = -\frac{1}{\tau}c(t) + s(t) \tag{A.1}$$

Let's denote $\gamma = \frac{1}{\tau}$

$$\frac{dc}{dt} = -\gamma c(t) + s(t) \tag{A.2}$$

Multiply both sides of Eqn. A.2 by $e^{\gamma t}$. We have

$$\frac{dc}{dt}e^{\gamma t} + e^{\gamma t}\gamma c(t) = s(t)e^{\gamma t} \tag{A.3}$$

$$\frac{d(c(t)e^{\gamma t})}{dt} = s(t)e^{\gamma t} \tag{A.4}$$

$$c(t)e^{\gamma t} = \int_0^t s(x)e^{\gamma x}dx + c_0c(t) = \int_0^t s(x)e^{\gamma(x-t)}dx + c_0e^{-\gamma t} \tag{A.5}$$

Dye dynamics, Eqn. 2.11

$$\frac{dy}{dt} = \frac{1}{\tau_{\text{on}}}(1 + \gamma c^h)(\frac{c^h}{1 + \gamma c^h} - y) \tag{A.6}$$

Let $P(t) = \frac{1}{\tau_{\text{on}}}(1 + \gamma c(t)^h)y$ and $Q(t) = \frac{1}{\tau_{\text{on}}}c(t)^h$ Then Eqn. A.6 becomes,

$$\frac{dy}{dt} + P(t)y(t) = Q(t) \tag{A.7}$$

We multiply both sides of the differential equation by the integration factor $I = e^{\int P(t)dt}$

$$I\frac{dy}{dt} + IP(t)y(t) = IQ(t) \tag{A.8}$$

$$\int (I\frac{dy}{dt} + IPy)dt = \int IQdt \tag{A.9}$$

Since $\frac{d}{dt}(Iy) = I\frac{dy}{dt} + IPy$, we have

$$Iy = \int IQdy \tag{A.10}$$

$$y = I^{-1}\int IQdt \tag{A.11}$$

The trick above is called the integrating factor trick.

APPENDIX B

INFERENCE ALGORITHMS

B.1 GUMBEL

There are two main steps. First, we need to represent our discrete random variable as a series of Gumbel distributions. Second we need to relax the discreteness of the random variable.

B.1.1 Gumbel representation

Random variable G is distributed according to a standard Gumbel distribution if $G = -\log(-\log(U))$, where $U \sim \text{Uniform}[0, 1]$. We can convert any discrete distribution into a series of Gumbel random variables.

If our discrete random variable X has $P(X = k) \propto \alpha_k$, then

$$X = \text{argmax}_k(\log \alpha_k + G_k) \tag{B.1}$$

where G_k is an i.i.d. standard Gumbel random variable.

B.1.2 Discreteness Relaxation

Let's represent the output of the argmax function as vector x with one hot representation. Possible outputs of argmax correspond to the vertices of the simplex $\Delta^{n-1} = [x \in \mathbb{R}^n | x_k \in [0, 1], \sum_{k=1}]$. We can relax this discreteness to allow any vector which sums to 1. We can use the softmax function to smoothly approximate the argmax, as temperature $\tau \in (0, \infty)$ approaches 0. Using these two ideas, the elements of relaxed random variable X can be written as [Maddison et al., 2016]:

$$X_k = \frac{\exp((\log \alpha_k + G_k)/\tau)}{\sum_{i=1}^{n} \exp((\log \alpha_i + G_i)/\tau)} \tag{B.2}$$

The distribution of X from the above equation has a closed form solution for the probability density.

$$p_{\alpha,\tau}(x) = (n-1)!\tau^{n-1} \prod_{k=1}^{n} \left(\frac{\alpha_k x_k^{-\tau-1}}{\sum_{i=1}^{n} \alpha_i x_i^{-\tau}} \right) \tag{B.3}$$

B.1.3 Bernoulli Case

Let's parametrize our two-state discrete variable D, where $D_1 + D_2 = 1$ with $\alpha \in (0, \infty)^2$:

$$P(D_1 = 1) = \frac{\alpha_1}{\alpha_1 + \alpha_2} \tag{B.4}$$

According to the Gumbel representation

$$P(D_1 = 1) = P(G_1 + \log \alpha_1 > G_2 + \log \alpha_2) = P(\log U - log(1 - U) + \log \alpha > 0) \tag{B.5}$$

where $\alpha = \frac{\alpha_1}{\alpha_2}$ and $U \sim Uniform(0, 1)$.

Similarly to the general case, we can relax the discrete variable:

$$X = \frac{1}{1 + \exp(-(\log \alpha + L)/\tau)} \tag{B.6}$$

where $L \sim Logistic$

B.2 ELBO

B.2.1 Objective

$$L(x) = \mathbb{E}_{q(h|x)} \left[\log \frac{p(x, z}{q(z|x)} \right] \tag{B.7}$$

B.2.2 Gradients

Using the reparametrization trick

$$\nabla_\phi L = \mathbb{E}_{u(\epsilon)} \left[\nabla_\phi g(t(\epsilon, \phi), \theta, \phi) \right] \tag{B.8}$$

where

$$\epsilon = Logistic(f_\phi(x)/\tau, 1/\tau) \tag{B.9}$$

$$t(\epsilon) = \frac{1}{1 + exp(-\epsilon)} \tag{B.10}$$

$$g(z, \theta, \phi) = \log p_\theta(x, z) - \log q(z; \phi) \tag{B.11}$$

B.3 GUMBEL

B.3.1 Objective

$$L^K(\theta, \phi) = \mathbb{E}_{Z^i \ q_\phi(z|x)} \left[\log \left(\frac{1}{K} \sum_{k=1}^{K} \frac{p_\theta(Z^k, x)}{q_\phi(Z^k|x)} \right) \right] \tag{B.12}$$

B.3.2 Gradients

$$\nabla_\phi L = \mathbb{E}_{u(\epsilon)} \left[\nabla_\phi g(t(\epsilon, \phi), \theta, \phi) \right] \tag{B.13}$$

where

$$\epsilon = Logistic(f_\phi(x)/\tau, 1/\tau) \tag{B.14}$$

$$t(\epsilon) = \frac{1}{1 + exp(-\epsilon)} \tag{B.15}$$

$$g(z, \theta, \phi) = \log \left(\frac{1}{m} \sum_{i=1}^{m} \frac{p_\theta(Z^i, x)}{q_\phi(Z^i|x)} \right) \tag{B.16}$$

B.4 VIMCO

B.4.1 Objective

$$L_k(x) = \mathbb{E}_{q(z_{1:k}|x)} \left[\log \frac{1}{k} \sum_{i=1}^{k} \frac{p(x, z_i)}{q(z_i|x)} \right] = \mathbb{E}_{q(z_{1:k}|x)} \left[\log \hat{I}(z_{1:k}) \right] \tag{B.17}$$

B.4.2 Gradients

Define

$$f(x, z_i) = \frac{p(x, z_i)}{q(z_i|x)} \tag{B.18}$$

$$\tilde{w_i} = \frac{f(x, z_i)}{\sum_{i=1}^{k} f(x, z_i)} \tag{B.19}$$

Its gradient can be expressed as

$$\nabla_\xi L_k(x) = \mathbb{E}_{q(z_{1:k}|x)} \left[\sum_{i=1}^{k} \log \hat{I}(z_{1:k}) \nabla_\xi \log q(z_i|x) \right] + \mathbb{E}_{q(z_{1:k}|x)} \left[\sum_{i=1}^{k} \tilde{w_i} \nabla_\xi \log f(x, z_i) \right] \tag{B.20}$$

The second term is easy to estimate. The first term is much harder. We call $\log \hat{I}(z)$ the learning signal.

Since,

$$\mathbb{E}_q \left[\nabla_\phi \log q_\phi(z|x) \right] = \mathbb{E}_q \left[\frac{\nabla_\phi q_\phi(z|x)}{q_\phi(z|x)} \right] = \nabla \mathbb{E}_q[1] = 0 \tag{B.21}$$

We can subtract from $\log \hat{I}(z)$ any constant value without affecting the expectation of the first term. To reduce the variance, we want to subtract a value that is very close to $\log \hat{I}(z)$. The VIMCO uses $f(x, z_{j \neq i})$ ($K - 1$ of them) to estimate $f(x, z_i)$ [Mnih and Rezende, 2016].

$$\hat{f}(x, z_{-i}) = \frac{1}{k-1} \sum_{j \neq i} f(x, z_j) \tag{B.22}$$

This gives the following local learning signals:

$$\hat{L}(z_i|z_{-i}) = \log \frac{1}{k} \sum_{j=1}^{k} f(x, z_j) - \log \frac{1}{k} \left(\sum_{j \neq i} f(x, z_j) + \hat{f}(x, z_{-i}) \right) \tag{B.23}$$

The final gradient estimator has the form:

$$\nabla_\xi L_k(x) \approx \sum_{i=1}^{k} \hat{L}(z_i|z_{-i}) \nabla_\xi \log q(z_i|x) + \sum_{i=1}^{k} \tilde{w}_i \nabla_\xi \log f(x, z_i) \tag{B.24}$$

BIBLIOGRAPHY

BIBLIOGRAPHY

[Aitchison et al., 2017] Aitchison, L., Russell, L., Packer, A. M., Yan, J., Castonguay, P., Hausser, M., and Turaga, S. C. (2017). Model-based bayesian inference of neural activity and connectivity from all-optical interrogation of a neural circuit. In *Advances in Neural Information Processing Systems*, pages 3486–3495.

[Akaike, 1969] Akaike, H. (1969). Fitting autoregressive models for prediction. *Annals of the Institute of Statistical Mathematics*, 21(1):243–247.

[Altman, 1990] Altman, D. G. (1990). *Practical Statistics for Medical Research*. CRC press.

[Atal and Hanauer, 1971] Atal, B. and Hanauer, L. (1971). Speech analysis and synthesis by linear prediction of the speech wave. *The Journal of the Acoustical Society of America*, 50:637–655.

[Atal and Schroeder, 1970] Atal, B. and Schroeder, M. (1970). Adaptive predictive coding of speech signals. *Bell System Technical J.*, 49:1973–1976.

[Baccalá and Sameshima, 2001] Baccalá, L. A. and Sameshima, K. (2001). Partial directed coherence: a new concept in neural structure determination. *Biological cybernetics*, 84(6):463–474.

[Berényi et al., 2014] Berényi, A., Somogyvari, Z., Nagy, A. J., Roux, L., Long, J. D., Fujisawa, S., Stark, E., Leonardo, A., Harris, T. D., and Buzsáki, G. (2014). Large-scale, high-density (up to 512 channels) recording of local circuits in behaving animals. *Journal of neurophysiology*, 111(5):1132–1149.

[Bertsekas and Rhodes, 1971] Bertsekas, D. P. and Rhodes, I. B. (1971). Recursive state estimation for a set-membership description of uncertainty. *IEEE Trans. on Automatic Control*, AC-16:117–128.

[Billings, 2013] Billings, S. A. (2013). *Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-temporal Domains*. John Wiley & Sons.

[Billings and Zhu, 1994] Billings, S. A. and Zhu, Q. M. (1994). A structure detection algorithm for nonlinear dynamic rational models. *International Journal of Control*, 59(6):1439–1463.

[Bishop, 2006] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

[Box and Jenkins, 1970] Box, G. and Jenkins, G. (1970). *Time Series Analysis: Forecasting and Control*. Holden-Day, 4$^{\text{th}}$ edition.

[Boyd and Vandenberghe, 2004] Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.

[Bressler, 1995] Bressler, S. L. (1995). Large-scale cortical networks and cognition. *Brain Research Reviews*, 20(3):288–304.

[Burda et al., 2015] Burda, Y., Grosse, R., and Salakhutdinov, R. (2015). Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*.

[Burg, 1975] Burg, J. (1975). *Maximum Entropy Spectral Analysis*. PhD thesis, Stanford U.

[Byrne et al., 2014] Byrne, J. H., Heidelberger, R., and Waxham, M. N. (2014). *From molecules to networks: an introduction to cellular and molecular neuroscience*. Academic Press.

[Chen et al., 1991] Chen, S., Cowan, C. F., and Grant, P. M. (1991). Orthogonal least squares learning algorithm for radial basis function networks. *Neural Networks, IEEE Transactions on*, 2(2):302–309.

[Chen et al., 1998] Chen, S. S., Donoho, D. L., and Saunders, M. A. (1998). Atomic decomposition by basis pursuit. *SIAM J. Scientific Computing*, 20(1):33–61.

[Chen et al., 2013] Chen, T.-W., Wardill, T. J., Sun, Y., Pulver, S. R., Renninger, S. L., Baohan, A., Schreiter, E. R., Kerr, R. A., Orger, M. B., Jayaraman, V., et al. (2013). Ultrasensitive fluorescent proteins for imaging neuronal activity. *Nature*, 499(7458):295.

[Cheung, 1991] Cheung, M. F. (1991). *On Optimal Algorithms for parameter set estimation*. PhD thesis, Ohio State University, Columbus.

[Cheung et al., 1991] Cheung, M. F., Yurkovich, S., and Passino, K. M. (1991). An optimal volume ellipsoid algorithm for parameter set estimation. In *Proc. 30th IEEE Conf. Decision and Control*, pages 969–974, Brighton.

[Combettes, 1993] Combettes, P. L. (1993). The foundations of set theoretic estimation. *Proc. IEEE*, 81:182–208.

[Dasgupta and Huang, 1987a] Dasgupta, S. and Huang, Y. (1987a). Asymptotically convergent modified recursive least-squares with data-dependent updating and forgetting factor for systems with bounded noise. *Information Theory, IEEE Transactions on*, 33(3):383–392.

[Dasgupta and Huang, 1987b] Dasgupta, S. and Huang, Y.-F. (1987b). Asymptotically convergent modified recursive least squares with data dependent updating and forgetting factor for systems with bounded noise. *IEEE Trans. on Information Theory*, 33:383–392.

[Deb, 2011] Deb, K. (2011). Multi-objective optimisation using evolutionary algorithms: An introduction. In *Multi-objective Evolutionary Optimisation for Product Design and Manufacturing*, pages 3–34. Springer.

[Deb, 2014] Deb, K. (2014). Multi-objective optimization. In *Search Methodologies*, pages 403–449. Springer.

[Deb et al., 2002] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on*, 6(2):182–197.

[Deller, Jr. and Luk, 1989] Deller, Jr., J. and Luk, T. C. (1989). Linear prediction analysis of speech based on set-membership theory. *Computer Speech and Language*, 3:301–327.

[Deller, Jr. et al., 1994] Deller, Jr., J., Nayeri, M., and Liu, M. S. (1994). Unifying the landmark developments in optimal bounding ellipsoid identification. *Int. J. Adaptive Control and Signal Processing*, 8:48–63.

[Deller, Jr. et al., 1993] Deller, Jr., J., Nayeri, M., and Odeh, S. F. (1993). Least-square identification with error bounds for real-time signal processing and control. *Proc. IEEE*, 81(6):813–849.

[Deller, Jr. and Odeh, 1993] Deller, Jr., J. and Odeh, S. F. (1993). Adaptive set-membership identification in O(m) time for linear-in-parameters models. *IEEE Trans. on Signal Processing*, 41(5).

[Deller, Jr. and Picaché, 1989] Deller, Jr., J. and Picaché, G. P. (1989). Advantages of a Givens rotation approach to temporally recursive linear prediction analysis of speech. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 37(3):429–431.

[Deller Jr. et al., 2007] Deller Jr., J. R., Gollamudi, S., Nagaraj, S., et al. (2007). Convergence analysis of the quasi-OBE algorithm and related performance issues. *International J. Adaptive Control and Signal Processing*, 21:499–527.

[Deller Jr. and Huang, 2002] Deller Jr., J. R. and Huang, Y. F. (2002). Set-membership identification and filtering for signal processing applications. *Circuits, Systems, and Signal Processing*, 21:69–82.

[Deller Jr. et al., 1994] Deller Jr., J. R., Nayeri, M., and Liu, M. S. (1994). Unifying the landmark developments in optimal bounding ellipsoid identification. *International J. Adaptive Control and Signal Processing*, 8:43–60.

[Deller Jr. et al., 1993] Deller Jr., J. R., Nayeri, M., and Odeh, S. F. (1993). Least-square identification with error bounds for real-time signal processing and control. *Proc. IEEE*, 81:815–849.

[Deneux et al., 2016] Deneux, T., Kaszas, A., Szalay, G., Katona, G., Lakner, T., Grinvald, A., Rózsa, B., and Vanzetta, I. (2016). Accurate spike estimation from noisy calcium signals for ultrafast three-dimensional imaging of large neuronal populations in vivo. *Nature communications*, 7:12190.

[Devore, 2015] Devore, J. (2015). *Probability and Statistics for Engineering and the Sciences*. Cengage Learning.

[Ding et al., 2006] Ding, M., Chen, Y., and Bressler, S. (2006). Granger causality: basic theory and application to neuroscience. 2006. *arXiv preprint q-bio/0608035*.

[Druckmann and Chklovskii, 2010] Druckmann, S. and Chklovskii, D. B. (2010). Over-complete representations on recurrent neural networks can support persistent percepts. In *Advances in Neural Information Processing Systems*, pages 541–549.

[Fogel, 1979] Fogel, E. (1979). System identification via membership set constraints with energy constrained noise. *IEEE Trans. on Automatic Control*, 24:752–758.

[Fogel and Huang, 1982a] Fogel, E. and Huang, Y.-F. (1982a). On the value of information in system identification - Bounded noise case. *Automatica*, 18:229–238.

[Fogel and Huang, 1982b] Fogel, E. and Huang, Y. F. (1982b). On the value of information in system identification: Bounded noise case. *Automatica*, 18:229–238.

[Freeman et al., 2014] Freeman, J., Vladimirov, N., Kawashima, T., Mu, Y., Sofroniew, N. J., Bennett, D. V., Rosen, J., Yang, C., Looger, L. L., and Ahrens, M. B. (2014). Mapping brain activity at scale with cluster computing. *Nature methods*, 11(9):941–950.

[Friedrich and Paninski, 2016] Friedrich, J. and Paninski, L. (2016). Fast active set methods for online spike inference from calcium imaging. In *Advances In Neural Information Processing Systems*, pages 1984–1992.

[Friston, 2011] Friston, K. J. (2011). Functional and effective connectivity: a review. *Brain Connectivity*, 1(1):13–36.

[Friston et al., 1994] Friston, K. J. et al. (1994). Functional and effective connectivity in neuroimaging: a synthesis. *Human Brain Mapping*, 2(1-2):56–78.

[Gerfen et al., 2013] Gerfen, C. R., Paletzki, R., and Heintz, N. (2013). Gensat bac cre-recombinase driver lines to study the functional organization of cerebral cortical and basal ganglia circuits. *Neuron*, 80(6):1368–1383.

[Gershman and Goodman, 2014] Gershman, S. and Goodman, N. (2014). Amortized inference in probabilistic reasoning. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 36.

[Goldberg, 2002] Goldberg, D. E. (2002). *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*. Kluwer Academic Publishers.

[Gollamudi et al., 1997] Gollamudi, S., Nagaraj, S., and Huang, Y. (1997). SMART: A toolbox for set-membership filtering. In *Proc. 1997 European Conf. on Circuit Theory and Design*, Budapest.

[Granger, 2004] Granger, C. W. (2004). Time series analysis, cointegration, and applications. *American Economic Review*, pages 421–425.

[Granger et al., 2000] Granger, C. W., Huangb, B.-N., and Yang, C.-W. (2000). A bivariate causality between stock prices and exchange rates: evidence from recent asianflu. *The Quarterly Review of Economics and Finance*, 40(3):337–354.

[Graupe, 1989] Graupe, D. (1989). *Time Series Analysis, Identification, and Adaptive Systems*. Krieger, Malabar, Florida.

[Guo et al., 2014] Guo, Z. V., Li, N., Huber, D., Ophir, E., Gutnisky, D., Ting, J. T., Feng, G., and Svoboda, K. (2014). Flow of cortical activity underlying a tactile decision in mice. *Neuron*, 81(1):179–194.

[Hayashi, 2000] Hayashi, F. (2000). *Econometrics*. Princeton U. Press.

[Haykin, 1995] Haykin, S. (1995). *Adaptive Filter Theory*. Prentice-Hall, Englewood-Cliffs, New Jersey.

[Hesse et al., 2003] Hesse, W., Möller, E., Arnold, M., and Schack, B. (2003). The use of time-variant eeg granger causality for inspecting directed interdependencies of neural assemblies. *Journal of neuroscience methods*, 124(1):27–44.

[Hill, 1910] Hill, A. V. (1910). The possible effects of the aggregation of the molecules of haemoglobin on its dissociation curves. *j. physiol.*, 40:4–7.

[Hinton et al., 2012] Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T., et al. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE*, 29(6):82–97.

[Jang et al., 2016] Jang, E., Gu, S., and Poole, B. (2016). Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

[Jordan, 1998] Jordan, M. I. (1998). *Learning in graphical models*, volume 89. Springer Science & Business Media.

[Jordan et al., 1999] Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233.

[Jordan et al., 2001] Jordan, M. I., Sejnowski, T. J., and Poggio, T. A. (2001). *Graphical models: Foundations of neural computation*. MIT press.

[Kamiński et al., 1997] Kamiński, M., Blinowska, K., and Szelenberger, W. (1997). Topographic analysis of coherence and propagation of eeg activity during sleep and wakefulness. *Electroencephalography and clinical neurophysiology*, 102(3):216–227.

[Kerlin et al., 2017] Kerlin, A., Mohar, B., Flickinger, D., MacLennan, B., Davis, C., Ji, N., Spruston, N., and Svoboda, K. (2017). Clustering of dendritic activity during decision making. In *Society for Neuroscience Annual Meeting Abstracts 2017-S-40722-SfN*, pages 1368–1383.

[Khalil, 2002] Khalil, H. (2002). *Nonlinear Systems*. Prentice-Hall, $3rd$ edition.

[Kingma et al., 2016] Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., and Welling, M. (2016). Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems*, pages 4743–4751.

[Kingma and Welling, 2013] Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

[Klambauer et al., 2017] Klambauer, G., Unterthiner, T., Mayr, A., and Hochreiter, S. (2017). Self-normalizing neural networks. In *Advances in Neural Information Processing Systems*, pages 971–980.

[Kolmogorov, 1941] Kolmogorov, A. (1941). Stationary sequences in Hilbert space. *Bulletin of Moscow U.*, 2:1–40.

[Leontaritis and Billings, 1985] Leontaritis, I. J. and Billings, S. A. (1985). Input-output parametric models for non-linear systems part i: deterministic non-linear systems. *International journal of control*, 41(2):303–328.

[Levanony and Berman, 2004] Levanony, D. and Berman, N. (2004). Recursive nonlinear system identification by a stochastic gradient algorithm: stability, performance, and model nonlinearity considerations. *Signal Processing, IEEE Transactions on*, 52(9):2540–2550.

[Lin et al., 1997] Lin, T., Nayeri, M., and Deller Jr., J. (1997). Automatic bound estimation: A practical development in optimal bounding ellipsoid processing. *IEEE Signal Processing Letters*, 4:236–239.

[Lin, 1996] Lin, T. M. (1996). *Optimal bounding ellipsoid algorithms with automatic bound estimation*. PhD thesis, Michigan State University, East Lansing.

[Lin et al., 1998] Lin, T. M., Nayeri, M., and Deller Jr., J. R. (1998). A consistently convergent OBE algorithm with automatic estimation of error bounds. *International J. Adaptive Control and Signal Processing*, 12:305–324.

[Liu and Aviyente, 2012] Liu, Y. and Aviyente, S. (2012). Quantification of effective connectivity in the brain using a measure of directed information. *Computational and mathematical methods in medicine*, 2012.

[Ljung, 1999] Ljung, L. (1999). *System Identification: Theory for the User*. Prentice-Hall PTR, 2^{nd} edition.

[Ljung, 2007] Ljung, L. (2007). *System Identification Toolbox for Use with {MATLAB}*. The MathWorks, Inc.

[Ljung and Söderström, 1983] Ljung, L. and Söderström, T. (1983). *Theory and Pratice of Recursive Identification*. MIT Press, Cambridge, Mass.

[Loghmanian et al., 2012] Loghmanian, S. M. R., Yusof, R., Khalid, M., and Ismail, F. S. (2012). Polynomial NARX model structure optimization using multi-objective genetic algorithm. *International Journal of Innovative Computing, Information and Control (IJICIC)*, 8(10B):7341–7362.

[Luria, 2012] Luria, A. R. (2012). *Higher Cortical Functions in Man*. Springer Science & Business Media.

[Macke et al., 2009] Macke, J. H., Berens, P., Ecker, A. S., Tolias, A. S., and Bethge, M. (2009). Generating spike trains with specified correlation coefficients. *Neural computation*, 21(2):397–423.

[Maddison et al., 2016] Maddison, C. J., Mnih, A., and Teh, Y. W. (2016). The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*.

[Makhoul, 1975] Makhoul, J. (1975). Linear prediction: A tutorial review. *Proceedings of the IEEE*, 63:561–580.

[Mao et al., 2011] Mao, T., Kusefoglu, D., Hooks, B. M., Huber, D., Petreanu, L., and Svoboda, K. (2011). Long-range neuronal circuits underlying the interaction between sensory and motor cortex. *Neuron*, 72(1):111–123.

[Marinazzo et al., 2008] Marinazzo, D., Pellicoro, M., and Stramaglia, S. (2008). Kernel-granger causality and the analysis of dynamical networks. *Physical Review E*, 77(5):056215.

[Milanese and Novara, 2004] Milanese, M. and Novara, C. (2004). Set membership identification of nonlinear systems. *Automatica*, 40:957–975.

[Mitchell, 1998] Mitchell, M. (1998). *An introduction to genetic algorithms*. MIT press.

[Mnih and Gregor, 2014] Mnih, A. and Gregor, K. (2014). Neural variational inference and learning in belief networks. *arXiv preprint arXiv:1402.0030*.

[Mnih and Rezende, 2016] Mnih, A. and Rezende, D. J. (2016). Variational inference for monte carlo objectives. *arXiv preprint arXiv:1602.06725*.

[Morse and Feshbach, 1946] Morse, P. M. and Feshbach, H. (1946). *Methods of theoretical physics*. Technology Press.

[Nagaraj et al., 1997] Nagaraj, S., Gollamundi, S., Deller, Jr., J., Huang, Y.-F., and Kapoor, S. (1997). Performance studies on a "Quasi-OBE" algorithm for real-time signal processing. In *Proc. 40th Annual Midwest Symposium on Circuits and Systems*, volume 2, pages 770–773, Sacramento.

[Nayeri et al., 1994] Nayeri, M., Liu, M., and Deller, Jr., J. (1994). Do interpretable optimal bounding ellipsoid algorithms converge? parts I & II. In *Proc. 10th IFAC/IFORS Symp. on System Identification (SYSID '94)*, volume 3, pages 389–400, Copenhagen.

[Naylor and Sell, 1971] Naylor, A. and Sell, G. (1971). *Linear Operator Theory*. Holt-Rinehart-Winston.

[Oord et al., 2016] Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*.

[Oord et al., 2017] Oord, A. v. d., Li, Y., Babuschkin, I., Simonyan, K., Vinyals, O., Kavukcuoglu, K., Driessche, G. v. d., Lockhart, E., Cobo, L. C., Stimberg, F., et al. (2017). Parallel wavenet: Fast high-fidelity speech synthesis. *arXiv preprint arXiv:1711.10433*.

[Pawlak et al., 2007] Pawlak, M., Hasiewicz, Z., and Wachel, P. (2007). On nonparametric identification of Wiener systems. *Signal Processing, IEEE Transactions on*, 55(2):482–492.

[Pereda et al., 2005] Pereda, E., Quiroga, R. Q., and Bhattacharya, J. (2005). Nonlinear multivariate analysis of neurophysiological signals. *Progress in Neurobiology*, 77(1):1–37.

[Pintelon and Schoukens, 2012] Pintelon, R. and Schoukens, J. (2012). *System identification: a frequency domain approach*. John Wiley & Sons.

[Pnevmatikakis et al., 2014] Pnevmatikakis, E. A., Gao, Y., Soudry, D., Pfau, D., Lacefield, C., Poskanzer, K., Bruno, R., Yuste, R., and Paninski, L. (2014). A structured matrix factorization framework for large scale calcium imaging data analysis. *arXiv preprint arXiv:1409.2903*.

[Ranganath et al., 2014] Ranganath, R., Gerrish, S., and Blei, D. (2014). Black box variational inference. In *Artificial Intelligence and Statistics*, pages 814–822.

[Reeves and Rowe, 2003] Reeves, C. R. and Rowe, J. E. (2003). *Genetic Algorithms: Principles and Perspectives: A Guide to GA Theory*, volume 20. Springer.

[Rezende and Mohamed, 2015] Rezende, D. J. and Mohamed, S. (2015). Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*.

[Russell and Norvig, 2010] Russell, S. and Norvig, P. (2010). *Artificial Intelligence: A Modern Approach*. Series in Artificial Intelligence. Prentice Hall, Upper Saddle River, NJ, third edition.

[Schetzen, 2006] Schetzen, M. (2006). *The Volterra and Wiener Theories of Nonlinear Systems*. Krieger Publishing Company.

[Schweppe, 1968] Schweppe, F. C. (1968). Recursive state estimation: Unknown but bounded errors and system inputs. *IEEE Trans. on Automatic Control*, AC-13:22–28.

[Seth, 2010] Seth, A. K. (2010). A matlab toolbox for granger causal connectivity analysis. *Journal of Neuroscience Methods*, 186(2):262–273.

[Sjöberg et al., 1995] Sjöberg, J., Zhang, Q. H., Ljung, L., et al. (1995). Nonlinear black-box modeling in system identification: A unified overview. *Automatica*, 31:1691–1724.

[Speiser et al., 2017] Speiser, A., Yan, J., Archer, E. W., Buesing, L., Turaga, S. C., and Macke, J. H. (2017). Fast amortized inference of neural activity from calcium imaging data with variational autoencoders. In *Advances in Neural Information Processing Systems*, pages 4024–4034.

[Sporns et al., 2004] Sporns, O., Chialvo, D. R., Kaiser, M., and Hilgetag, C. C. (2004). Organization, development and function of complex brain networks. *Trends in Cognitive Sciences*, 8(9):418–425.

[Stramaglia et al., 2011] Stramaglia, S., Angelini, L., Pellicoro, M., and Marinazzo, D. (2011). Nonlinear granger causality for brain connectivity. In *Medical Measurements and Applications Proceedings (MeMeA), 2011 IEEE International Workshop on*, pages 197–201. IEEE.

[T. Söderström and P. Stoica, 1989] T. Söderström and P. Stoica (1989). *System Identification*. Prentice-Hall, Englewood-Cliffs, New Jersey.

[Volterra et al., 1930] Volterra, V., Fantappiè, L., and Long, M. (1930). *Theory of functionals and of integral and integro-differential equations*. Blackie & Son Limited.

[Wainwright et al., 2008] Wainwright, M. J., Jordan, M. I., et al. (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305.

[Walter et al., 1996] Walter, E., Norton, J., Piet-Lahanier, H., and Milanese, M., editors (1996). *Bounding Approaches to System Identification*. Perseus Publishing.

[Walter and Piet-Lahanier, 1990] Walter, E. and Piet-Lahanier, H. (1990). Estimation of parameter bounds from bounded-error data: a survey. *Mathematics and Computers in Simulation*, 32(5):449–468.

[Wigren, 2006] Wigren, T. (2006). Recursive prediction error identification and scaling of nonlinear state space models using a restricted black box parameterization. *Automatica*, 42(1):159–168.

[Wigren and Schoukens, 2013] Wigren, T. and Schoukens, J. (2013). Three free data sets for development and benchmarking in nonlinear system identification. In *Control Conference (ECC), 2013 European*, pages 2933–2938. IEEE.

[Witsenhausen, 1968] Witsenhausen, H. S. (1968). Sets of possible state of linear systems given perturbed observation. *IEEE Trans. on Automatic Control*, AC-13:556–568.

[Xiao et al., 2013] Xiao, Z., Jing, X., and Cheng, L. (2013). Parameterized convergence bounds for volterra series expansion of NARX models. *Signal Processing, IEEE Transactions on*, 61(20):5026–5038.

[Yan and Deller Jr., 2015] Yan, J. and Deller Jr., J. R. (2015). Narmax model identification using a set-theoretic evolutionary approach. *Signal Processing, An International Journal of*.

[Yan et al., 2013] Yan, J., Deller Jr., J. R., Fleet, D. B., et al. (2013). Evolutionary identification of nonlinear parametric models with a set-theoretic fitness criterion. In *Proc. 2013 IEEE China Summit and International Conf. Signal and Information Processing*, volume 1, pages 44–48.

[Yan et al., 2014] Yan, J., Deller Jr., J. R., Yao, M., and Goodman, E. D. (2014). Evolutionary model selection for identification of nonlinear parametric systems. In *Proc. 2014 IEEE China Summit and International Conf. Signal and Information Processing*, pages 693–697.

[Yan et al., 2018] Yan, J., Kerlin, A., Aitchison, L., Svoboda, K., and Turaga, S. C. (2018). Infering pre and post-synaptic activity from dendritic calcium imaging. In *Computational and System Neuroscience (COSYNE)*.

[Yu et al., 2014] Yu, C., Zhang, C., and Xie, L. (2014). A new deterministic identification approach to Hammerstein systems. *Signal Processing, IEEE Transactions on*, 62(1):131–140.