UNIVERSITY OF CALIFORNIA,
IRVINE

Estimation of Sparse State Transition Parameters in State Space Models

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Electrical Engineering

by

Misagh Khayambashi

Dissertation Committee:
Professor A. Lee Swindlehurst, Chair
Professor Ramesh Srinivasan
Professor Zoran Nenadic

2018

DEDICATION

To Dr. Swindlehurst, for his patience, support, faith, kindness, and guidance; and for giving me the opportunity to become who I am today...

To my wife, Ronak, for her unwavering company and love, and for bearing with me as I worked on this thesis...

To my parents, without whose sacrifices and support I would not be here.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Swindlehurst, for his faith in me, and for giving me the freedom to explore and grow. This work would not exist without his guidance and support.

ABSTRACT OF THE DISSERTATION

Estimation of Sparse State Transition Parameters in State Space Models

By

Misagh Khayambashi

Doctor of Philosophy in Electrical Engineering

University of California, Irvine, 2018

Professor A. Lee Swindlehurst, Chair

Motivated by the many applications associated with sparse multivariate models, the estimation of the directional interactions between imperfectly measured nodes of a network is studied.

First, the node dynamics and interactions are assumed to follow a linear multivariate autoregressive (MVAR) model. The observations consist of noisy linear combinations of the underlying node activities. Maximum a posteriori (MAP) criterion is adopted for parameter estimation. Due to the intractability of the MAP problem, the Expectation Maximization (EM) algorithm is used to iteratively solve the MAP problem. To impose sparsity on state transition parameters, the EM algorithm is augmented with an ℓ_1 regularization of the connectivity matrix. Multiple techniques have been used to lower the computational complexity. Importantly, an efficient coordinate descent algorithm utilizing a closed-form solution is designed to solve the ℓ_1-regularized EM. For noise covariance estimation, the Cholesky factors of the unknown covariance matrices are used directly in the optimization process in order to impose positive definiteness and guarantee the functionality of the ℓ_1 optimization.

The algorithm is first applied to synthetic data to evaluate the estimation accuracy. Comparison with previous work over an extensive set of configurations shows that our method is superior under moderate to high sparsity.

The algorithm is then evaluated on real data for two different applications: temperature prediction and estimation of effective brain connectivity. Applied to real temperature data obtained from 98 stations across the U.S. mainland, the algorithm is able to identify the predictive interactions between the time series that not only are consistent with previous work, but also reveal predictive power for coastal stations.

The algorithm, however, does not perform well in the estimation of effective brain connectivity from real electroencephalography (EEG) data. We show that this shortcoming is due to the inflexibility of the linear model to capture EEG dynamics. The Neural Mass Model (NMM) is then adopted to realistically model the underlying mechanisms of EEG signals. The estimation algorithm is tailored to the nonlinearity of the NMM model. The modified algorithm is then applied to a simple synthetic system, and it is observed that the results are insensitive to the source of the connection. The root cause of the problem is then analyzed and the challenges facing the future work are discussed.

Chapter 1

Introduction

1.1 Applications of Sparse Multivariate Autoregressive Models

A very large number of applications are associated with multivariate autoregressive (MVAR) or equivalent models. Example applications include, but are not limited to, identifying the effective connectivity in the brain [63], multipath wireless communication channels in sensor networks [14], echo cancellation [31], determining dominant predictors of atrial fibrillation [125], gene and protein interaction analysis [11][47], and determining the main risk factors for certain pathologies [72][114]. Many of these applications lead to models with extremely large dimensions, with a huge number of possible parameters, and often there is not enough observed data to reliably estimate such a huge set of parameters. Fortunately, there are many situations in which the MVAR parameters can be assumed to be sparse.

Analysis of molecular mechanisms underlying important biological processes is yet another application of sparse MVAR models [83]. In [43], the sparse MVAR model is applied to estimate gene regulatory networks based on gene expression profiles obtained from time series microarray experiments. Both simulated data and real HeLa cell cycle gene expression data are used to validate the results of the sparse MVAR inference even when the number of samples is smaller than the number of genes. A similar approach applied to a first-order Markovian time series model of gene interactions is presented in [1].

Sparse MVAR models have also been applied to wind power prediction [30], solar power forcasting [19], and causal modeling of unstructured temperature data [81]. In [30], a two stage method similar to [26] was used. First, the partial spectral coherence matrix of the time series is used to trim away the autoregressive elements that are suspected to be negligible. Then, the values of the predetermined set of elements are determined by maximum likelihood estimation. A graph-based sparse model more parsimonious than the traditional MVAR model is adopted in [81] to discover the predictive relations between the temperature recordings over 365 days at 150 cities around the continental United States.

Sparse high dimensional MVAR models are widely used in econometrics [34] to analyze the joint evolution of macroeconomic time series, and to provide structural information about the model [105]. Example applications include optimal sparse portfolio selection [35], housing market prediction [96], identifying demand effects in a large network of product categories and modeling the market response [116], and analysis of causality in stock market data [60].

1.2 Sparse Brain Connectivity

A major aspect of the complexity of nervous systems relates to their intricate morphology, especially the inter-connectivity of their neuronal processing elements. Neural connectivity patterns have long attracted the attention of neuro-anatomists [120, 107] and play crucial roles in determining the functional properties of neurons and neuronal systems.

Neural activity, and by extension neural codes, are constrained by connectivity. Understanding brain connectivity is thus crucial to elucidating how neurons and neural networks process information. In addition to revealing the underlying mechanisms of information processing in the brain, connectivity analyses have found clinical applications [9, 99, 106] since certain cognitive disorders such as autism, schizophrenia, and attention-deficit/hyperactivity are hypothesized to be at least partially attributed to abnormal connectivity patterns in brain.

In more highly evolved nervous systems, brain connectivity can be described at several levels of scale [102]. These levels include individual synaptic connections that link individual neurons at the microscale, networks connecting neuronal populations at the mesoscale, as well as brain regions linked by fiber pathways at the macroscale. At the microscale, detailed anatomical and physiological studies have revealed many of the basic components and interconnections of microcircuits in the mammalian cerebral cortex. At the mesoscale, they are arranged into networks of columns and minicolumns. At the macroscale, which is the focus of this thesis, very large numbers of neurons and neuronal populations forming distinct brain regions are interconnected by inter-regional pathways, forming large-scale patterns of anatomical connectivity.

Broadly speaking, the concept of connectivity is defined in three different ways: Anatomical, Functional, and Effective [102, 63] (see Fig. 1.1). In this thesis, we focus on effective connectivity. *Anatomical (structural) connectivity* refers to the anatomical connections between regions [95]. An anatomical connection is necessary for communication between regions. However, not all anatomical connections are used to perform certain tasks. In addition to invasive methods, Diffusion Tensor Imaging (DTI) and Magnetic Resonance Imaging (MRI) are among the non-invasive methods used to estimate anatomical pathways in different scales.

Functional connectivity is a statistical measure of connectivity between regions of brain [69]. Regardless of the presence of a direct anatomical or causal connection between two regions, two regions are said to be functionally connected if their joint activity shows significant levels

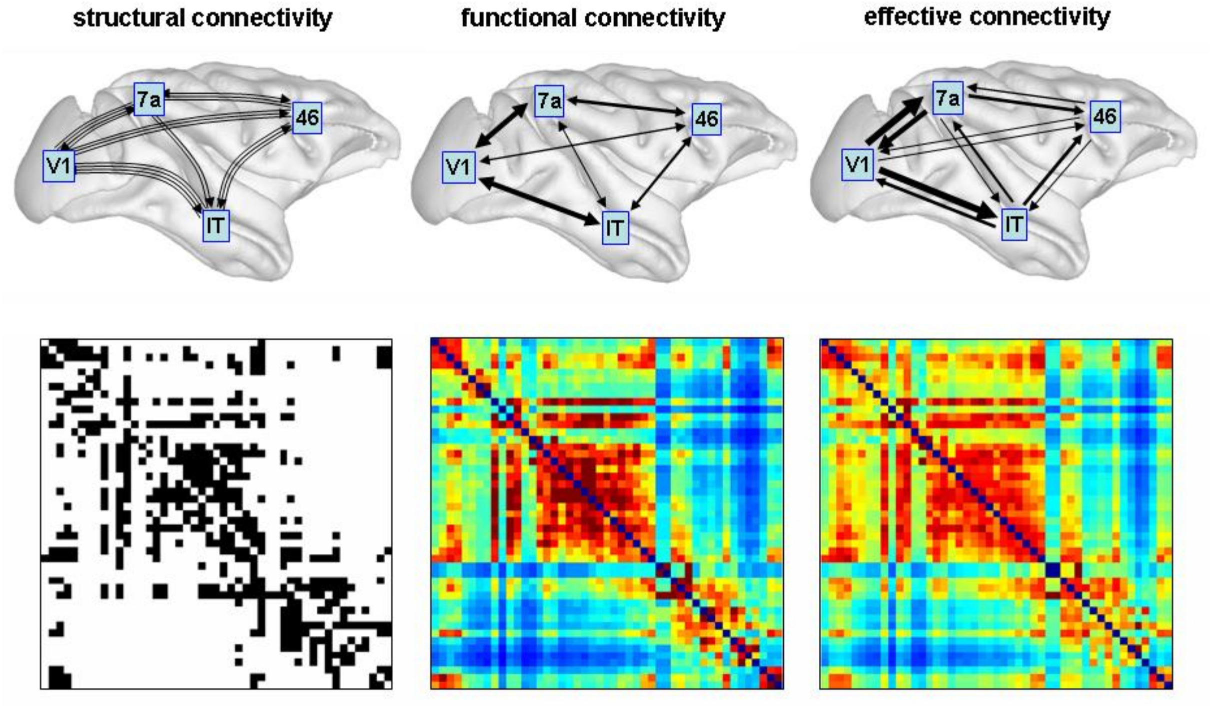


Figure 1.1: Modes of brain connectivity; Sketches at the top illustrate structural connectivity (fiber pathways), functional connectivity (correlations), and effective connectivity (information flow) among four brain regions in macaque cortex. Matrices at the bottom show binary structural connections (left), symmetric mutual information (middle) and non-symmetric transfer entropy (right). Data was obtained from a large-scale simulation of cortical dynamics [62]

of correlation. By definition, functional connectivity has no sense of direction. The concept is applicable to any imaging modality that records the brain activity, including functional MRI (fMRI) and Electroencephalography (EEG).

In this thesis, we focus on the *effective* or *directional* connectivity, which refers to any directional and causal connectivity measure between two regions. An accurate characterization of effective connectivity in the human brain requires a comprehensive mapping of the connectome as well as an electrophysiological specification of how neurons or populations of neurons interact to process information. Despite recent advances in mapping the human connectome and in measuring activity of neural populations [93, 101], our knowledge is still far from being comprehensive because of the extraordinary complexity of the human brain. Most of brain research is still dedicated to brain-behaviour interactions, and to measuring changes in functional and effective connectivity in a response to behavioural stimulation. Developing new methods for inferring effective connectivity within the brain is still an underdeveloped topic. Furthermore, the interpretation and the value of the inferred measures of effective connectivity directly depend on the underlying model assumed to govern the dynamics of human brain, as well as the capability of the measurement modality in accurately exposing the underlying system.

The estimation of connectivity in the human brain is only one instance of the more general problem of identifying the interaction among multiple nodes of a network given some time series observation. Consequently, the same methodology may be used in other applications requiring the estimation of intra-network interactions.

In the area of brain connectivity analysis, the necessity of incorporating sparsity into the estimation has been studied and various methods have been suggested to estimate the parameters of the sparse underlying system [56, 112, 17, 58, 38, 104, 16, 39]. In [104], the high degree of clustering and the short characteristic length of the small-world topology of the macaque and cat (visual) cortex are reproduced by considering less than 1% of the possible connections and properly distributing them. Moreover, it has been shown in [103] that dynamics with high complexity are supported by graphs whose units are organized into densely intra-linked groups that are sparsely and reciprocally inter-linked. A sparse MVAR model is used in [10] for electrocorticography (ECoG) modeling and the group Lasso method [123] is used to estimate the sparse directional network connectivity. A realistic anatomical network topology obtained from tract-tracing studies of a macaque brain [121] with 71 nodes and 746 connections (15% sparsity) is used. Sparsity levels higher than the anatomical sparsity are considered, as not all physically connected brain regions are actively communicating. In [18], the functional connectivity of the human brain is estimated by exploiting the underlying sparsity to set most MVAR parameters to zero.

1.3 Outline of the Thesis

This thesis focuses on the parameter estimation for MVAR models postulated to govern the dynamics of a network of interacting nodes. We consider both linear systems and a special case of nonlinear systems appropriate for modeling neural dynamics.

The proposed estimation procedure is evaluated on synthetic data as well as two different applications (temperature prediction and brain effective connectivity estimation) using real data. The strengths and the shortcomings of the proposed method are discussed and analyzed. For the special application of brain connectivity, the estimation procedure is tailored to a more realistic nonlinear model, namely the Neural Mass Model (NMM), and evaluated on real and synthetic EEG data.

In the linear model, the directional interaction of the nodes is encoded in the elements of a *connectivity matrix*. Furthermore, unlike much of the previous work, the underlying network is assumed to be only imperfectly measured through a noisy linear combination of the node activities. We utilize a maximum a posteriori (MAP) framework to infer the strength and delay of the connectivities embedded in the model. A critical aspect of our approach is that the sparseness of the connectivity pattern is incorporated into the estimation procedure. Furthermore, we develop an efficient algorithm to solve the estimation problem. We show that our algorithm performs better than previously proposed approaches, and the performance gap widens as the sparseness of the underlying system increases.

While previous work has separately examined some of the individual contributions that follow, our work combines them together to make the solution applicable to more general problems in a computationally tractable way. A summary of the contributions of this work is provided below:

1. With the exception of [57, 58, 22], the previous work assumes that the time series whose directional interactions are to be estimated are observed directly. We generalize this to cases where the observed time series are linearly superimposed, and thus only observed as a mixture. An example of this type of scenario is when electrodes on the scalp take EEG measurements of the electrical activity of the brain under the skull. In this case, each electrode simultaneously measures the electrical contribution of multiple brain regions.

2. Even in cases where previous work has considered linear mixtures of the target time series, the estimation of directional interactions follows a two-stage approach [57, 58]. In the first stage, the target time series are estimated from the observed time series without considering the structure of the underlying signal dynamics (i.e., the sparsely connected MVAR model). In the second stage, the interactions are determined from the estimated time series. As discussed in later chapters and illustrated in the simulations, the two-stage approaches yield suboptimal performance. A joint optimization that estimates both the state vector and the structure of the interactions (connectivity) simultaneously will be closer to optimal. Cheung et al. [22] utilize such an EM-based

joint optimization, but without considering sparsity.

3. ℓ_1-regularization of the EM algorithm has been applied in prior work [10, 123, 83, 43, 116, 60]. However, the numerical methods used to solve the problem are computationally complex for the problem dimensions that we consider. We have proposed to use coordinate descent for this problem and derived the analytical solution to each step of the algorithm to reduce the computational complexity. A number of additional algorithm modifications have been adopted in our approach to further lower the complexity; these modifications are unique to our approach and have not been considered elsewhere.

4. Although it is not the main focus of the thesis, accurate estimation of the positive definite innovation and noise covariance matrices is critical for the connectivity estimation algorithm to work. Previous work on this problem has not considered estimating the Cholesky factors of these matrices to address this issue. Rather, they limit the space of possible covariance matrices by adding an additional regularization or by constraining the covariance matrices to have a specific structure (e.g., diagonal). The Cholesky factors, on the other hand, can describe and parameterize all positive definite matrices.

The thesis is outlined as follows. Chapter 2 reviews the previous work on sparse parameter estimation in linear MVAR models and provides the problem statement. The proposed estimation method is then discussed. The chapter concludes with remarks on hyperparameter selection and computational complexity of the proposed algorithm.

In Chapter 3, the algorithm is first evaluated on a comprehensive set of synthetic system configurations and is shown to outperform the previous work under moderate to high ground truth sparsity. The algorithm is also applied on real temperature data to find the predictive powers of different weather stations. Finally, the applicability of the algorithm to real EEG data for brain effective connectivity estimation is examined. The shortcomings of the model motivate the following chapters.

Chapter 4 serves as an introduction to the Neural Mass Model (NMM). The NMM was adopted to overcome the shortcomings of the linear MVAR model of the earlier chapters. In Chapter 5, the estimation algorithm is derived from scratch for a network of neural masses. Finally, Chapter 6 explores the intrinsic limitations of a MAP-based estimation approach toward connectivity estimation for a network of neural masses. The future work section in the final chapter presents some candidate solutions.

Chapter 2

Parameter Estimation for Linear State Space Models

2.1 Associating Causal Influence with Model Parameters

Defining a true measure of causality requires an accurate, physically meaningful, and interpretable model of the underlying mechanisms of a given system. When such a model is unavailable, computationally prohibitive, or uninterpretable, mathematical measures of causality exist to capture the interactions. These measures may or may not coincide with the true causality depending on the presumed model and the application.

Granger and Geweke [50, 48] were among the first to formulate the concept of 'Granger causality'. In his seminal work, Granger defined the causal effect of phenomenon A on phenomenon B as equivalent to an improvement in accuracy of prediction of B, given A and all other relevant parameters, compared to the accuracy of prediction without A. Other measures of connectivity have been proposed, including those based on the Directed Transfer Function (DTF) [64], Partial Directed Coherence (PDC) [6], Directed DTF [66], and Phase Slope Index [87]. The performance of these methods in estimating effective connectivity using EEG data is compared in [57].

Model-free information-theoretic measures of causality have also gained popularity. Schreiber et al. [98] introduced Transfer Entropy (TE), in which the time series are approximated by Markov processes and the Kullback-Leibler divergence is used to verify the assumption of independence between two time series. Later, Branett et al. [7] proved the equivalence of Granger causality and TE for Gaussian variables. Chavez et al. [20] and Liu et al. [73] compared different information theoretic causality measures including TE [98], Mutual Information and its delayed variants [65], Directed Information (DI) [79], Conditional and time-lagged DI [73], and Directed Trans-Information [78].

Linear multivariate autoregressive (MVAR) models and variants have also been utilized to describe the dynamics of network interaction and connectivity. Non-linear and modulational effects in effective connectivity have also been examined in the neuroscience and engineering literature. Buchel et al. [15] focused on the time variance and non-linear attentional-modulation effects in connectivity. Freiwald et al. [41] proposed Local Linear Non-linear AR models that could describe both linear and non-linear interactions. The methodology was compared with other non-linear interaction models including generic non-linear MVAR, linear MVAR with past-dependent coefficients, and locally weighted polynomial non-parametric regression [70]. Marinazzo et al. [77] proposed a non-linear extension of Granger causality, called kernel Granger causality, by taking a geometric approach to the regression problem.

Most of the methods mentioned thus far require stationarity. A range of methods have been proposed to overcome this issue for applications where stationarity does not hold. Ding et al. [29] examined the non-stationarity of brain activity and applied an adaptive MVAR framework to stationary overlapping epochs. Moller et al. [84] approached non-stationarity by applying Recursive Least Squares with a forgetting factor. Sato et al. [97] expanded the time varying coefficients of an AR process in a wavelet basis and iteratively estimated the wavelet coefficients. Gao et al. introduced the concept of the “evolutionary state space” by allowing the parameters to evolve across epochs in order to model the non-stationarity of the data [46].

2.2 Previous Work on MVAR Parameter Estimation

Statistical frameworks have been successfully applied to estimation of effective connectivity in MVAR models as well as others. Harrison et al. [54] proposed an iterative Bayesian estimation of MVAR parameters by assuming certain priors on the MVAR coefficients and hyper-parameters. Ho et al. [61] used Kalman filtering and the EM algorithm to find the ML estimate of the state space parameters. A more flexible extension of [61] was proposed by Gao et. al [46]. Lenz et al. [67] have studied the problem of recovering connectivity from joint EEG-fMRI data with Kalman filtering and EM. Smith et al. [100] and Rajapakse et al. [92] were among the first to employ Dynamic Bayesian Networks (DBN) to recover connectivity, implicitly handling non-linearities and direct causality. More recently, Mutlu [86] used discrete DBN with multinomial distributions to capture non-linearities, and proposed learning the DBN structure by a combination of Markov chain Monte Carlo (MCMC) methods and a greedy graph search. Zheng et al. [126] used Bayesian networks to model brain activity in one snapshot by assuming Gaussian interaction between the nodes. A maximum a posteriori (MAP) estimate of the structure of the graph was obtained through a greedy search algorithm. In [88], the sparsity-promoting priors are replaced by super-Gaussian lower bounds, and the coefficients of interaction are estimated as the mean of the resulting Gaussian posterior. With an emphasis on high-dimensional scenarios, the authors of [89] propose solving the optimization problem using the alternating direction method of multipliers.

The majority of available methods assume that the available time series directly represent the activity of the interacting entities. However, this information is often available only imperfectly through some mapping from the node activity space to the measurement space. In the context of source reconstruction, Haufe has proposed two-stage strategies in [55] and [58]; first, node activities are directly estimated from the observations, independently at each time instant, by solving an inverse problem. A sub-optimum result is expected because the inverse problem neglects the connectivity structure of the underlying system. Second, the estimated time series are analyzed to estimate the connectivity patterns. A recent work by Cheung et al. [22] reveals some disadvantages of using two-stage methods such as bias in activity and connectivity estimates. Single-stage methods based on EM to find ML estimates of MVAR coefficients directly from the observations are proposed in [71, 22].

Recently, profound physiological evidence has been found to support sparse connectivity models in describing the brain's effective connectivity [59, 3]. Nevertheless, estimation of sparse connections between interacting nodes of a network is a generic problem with potential applications outside neuroscience. Chen et al. [21] imposes this sparsity on fMRI data by assuming sparsity promoting priors on the AR coefficients and updates the posterior distribution of the coefficients. Estimates of the AR coefficients are found from the expectation of the resulting posterior, which is calculated by a Reversible-Jump Markov Chain Monte Carlo (RJMCMC) approach [51].

Motivated by previous work in [21, 22, 61], we propose a MAP-based approach that jointly estimates the activities and the connectivity pattern and incorporates the prior information on the sparsity of the underlying system. The details are provided in the next sections.

2.3 Related Work on ℓ_1-Regularized Optimization

A common approach to imposing sparsity in estimation problems is to use ℓ_1-regularization. The application of ℓ_1-regularization has a relatively long history [113]. Tibshirani et al. [108] proposed one approach, referred to as the Least Absolute Shrinkage and Selection Operator (or LASSO) technique. The emergence of the Least-Angle Regression (LARS) algorithm provided an efficient solution to the optimization problem underlying LASSO for linear regression [33]. More recently, path-wise coordinate descent methods have been proposed for solving LASSO problems [42, 118, 110, 111]. While LARS exploits the linearity of the regularization path to calculate the exact path of solutions against the regularization parameter, the coordinate descent method focuses on the efficiency of finding the solution for high-dimensional problems on a (usually) equi-spaced grid of the parameters.

2.4 Model and Problem Statement

The system under study is assumed to have interacting nodes whose activities can only be measured indirectly and imperfectly. Section 2.4.1 explains the model postulated to describe the node interactions. The measurements are assumed to be a stochastic mapping from the node activity space to the measurement space. Section 2.4.2 describes the postulated measurement model.

2.4.1 Node activity model

Consider a network of M interacting nodes. Denote the instantaneous activity of node $m \in \{1, \cdots, M\}$ at discrete time index k by $v_m[k]$. Similar to the previous work [29, 97, 54, 61, 22, 71], individual node activities are postulated to be a linear combination of a local innovation $w_m[k]$ and the past activity of other regions as well as the target region:

$$v_m[k] = w_m[k] + \sum_{\tau=1}^{D} \sum_{i=1}^{M} a_{i,m}[\tau] v_i[k-\tau] \ , \tag{2.1}$$

where D is the maximum anticipated delay, and $a_{i,m}[\tau]$ models the influence of node i on node m at delay τ, or equivalently the strength of the connection from node i to node m with delay τ. A proper value for D may be found through standard model selection techniques, as discussed in Section 2.9, or from domain-specific knowledge. The innovation w_m serves as the driving force of the network: assuming a stable system, the system converges to zero activity in the absence of the driving force.

The model in (2.1) can be written in vector form as

$$\mathbf{v}_k = \mathbf{w}_k + \sum\nolimits_{\tau=1}^{D} \mathbf{A}_\tau \mathbf{v}_{k-\tau} \tag{2.2}$$

by defining $\mathbf{v}_k \triangleq [v_1[k], v_2[k], \cdots, v_M[k]]^T$ and $\mathbf{w}_k \triangleq [w_1[k], w_2[k], \cdots, w_M[k]]^T$. The matrix $\mathbf{A}_\tau$, representing the connectivity between nodes at delay τ, is defined per element by $(\mathbf{A}_\tau)_{i,j} = a_{j,i}[\tau]$. The innovation w_m is assumed to be a temporally white zero-mean multivariate normal random vector with unknown covariance. As shown in Section 2.5, temporally colored noise may be easily included by augmenting the state vector with the parameters representing the dynamics of w_m.

A fundamental prior incorporated in our work is that the collection of connectivity matrices has relatively few non-zero entries. In other words, the vector $[\text{vec}^T(\mathbf{A}_1)\ \text{vec}^T(\mathbf{A}_2) \cdots \ \text{vec}^T(\mathbf{A}_D)]^T$ is sparse. This definition does not distinguish between the sparsity of the spatial and temporal connections. The spatial sparsity refers to the sparsity of connections between nodes, while the temporal sparsity refers to the existence of a few dominant delays between two regions *if* a connection between them exists.

2.4.2 Measurement model

At each discrete time index k, N scalars $y_1[k], \cdots, y_N[k]$ are recorded as measurements. Each $y_i[k]$ is assumed to be a linear combination of the underlying node activities plus some uncertainty or interference originating from unmodeled effects and/or measurement noise. Define the vector of measurements as $\mathbf{y}_k \triangleq [y_1[k], y_2[k], \cdots, y_N[k]]^T$. We assume that

$$\mathbf{y}_k = \mathbf{C}\mathbf{v}_k + \mathbf{u}_k \tag{2.3}$$

where $\mathbf{C}$ is an $N \times M$ *gain* matrix, and $\mathbf{u}_k$ is the vector of measurement noise/interference at time k. Similar to the process noise term $\mathbf{w}_k$, we assume that $\mathbf{u}_k$ is a temporally white zero-mean Gaussian with unknown covariance.

2.4.3 Problem Statement

Given a set of T measurements $\mathcal{Y} = \{\mathbf{y}_1, \cdots, \mathbf{y}_T\}$, the system model, and priors on the unknown parameters, we wish to estimate the connectivity matrices $\mathbf{A} = [\mathbf{A}_1, \cdots, \mathbf{A}_D]$ as well as the second order statistics of $\mathbf{u}_k$ and $\mathbf{w}_k$. We should emphasize that estimation of the connectivity matrix is the main goal of this work and hence that the interference statistics are of secondary importance. Henceforth, $\boldsymbol{\theta}$ will denote the collection of all unknowns. Certain priors should be added to avoid unrealistic outcomes. For instance, the connectivity matrix $\mathbf{A}$ is sparse and should be chosen such that the system is stationary and stable.

2.5 Likelihood-Based Parameter Estimation

As mentioned in Section 2.4, the noise terms $\mathbf{w}_k$ and $\mathbf{u}_k$ are assumed to be temporally-white zero-mean normally distributed multivariates with unknown covariances. Consequently, (infinitely) many choices of the unknown parameters can explain the observations if the proper value of $\mathbf{w}_k$ and $\mathbf{u}_k$ is assumed. Nevertheless, different values of $\mathbf{u}_k$ and $\mathbf{w}_k$ are not equally likely, a fact that can help define the merit of a given choice of parameters.

In MAP estimation, the optimum parameter is the one that maximizes the *parameter posterior* for a given set of observations [85]. Mathematically, denoting the set of possible parameters by Θ, the MAP estimate is given by:

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}\in\Theta} p(\boldsymbol{\theta}|\mathcal{Y}) = \arg\max_{\boldsymbol{\theta}\in\Theta} p(\mathcal{Y}|\boldsymbol{\theta})p(\boldsymbol{\theta}) \ . \tag{2.4}$$

The term $p(\mathcal{Y}|\boldsymbol{\theta})$ is the data likelihood, and the parameter prior $p(\boldsymbol{\theta})$ encodes the prior probability of $\boldsymbol{\theta}$ (e.g. the sparsity of the connectivity matrix). Note that the optimization result is the same if any monotonically increasing function, such as logarithm, is applied to the posterior $p(\boldsymbol{\theta}|\mathcal{Y})$.

2.5.1 Derivation of Data Likelihood

The data likelihood is calculated by marginalizing over the hidden states:

$$p(\mathcal{Y}|\theta) = \int d\mathcal{V} p(\mathcal{V}|\theta)p(\mathcal{Y}|\mathcal{V},\theta) \tag{2.5}$$

where $\mathcal{V} = \{\mathbf{v}_1, \cdots, \mathbf{v}_T\}$. The integral arguments are calculated, multiplied, and then integrated in $\mathcal{V}$ plane to obtain the data likelihood.

Calculating $p(V|\theta)$

Given the dynamic model of Equation 2.2, it is helpful to think of past activity as system state. Specifically, with $\mathbf{V}_{t_2:t_1} = [\mathbf{v}_{t_2}^T \ \cdots \ \mathbf{v}_{t_1}^T]^T$ $(t_2 > t_1)$, and $\mathbf{A} = [\mathbf{A}_1 \cdots \mathbf{A}_D]$:

$$\mathbf{v}_k = \mathbf{w}_k + \mathbf{A}\mathbf{V}_{(k-1):(k-D)} \tag{2.6}$$

To calculate $p(\mathcal{V}|\theta)$, we utilize the system dynamic given in Equation 2.2 to factor the distribution into smaller components. Dropping the θ dependence for notational convenience:

$$p(\mathcal{V}) = p(\mathbf{V}_{D:1}) \prod_{k=D+1}^{T} p(\mathbf{v}_k|\mathbf{V}_{(k-1):1}) \tag{2.7}$$

Using Equation 2.6, each $p(\mathbf{v}_k|\mathbf{V}_{(k-1):1})$ term can be rewritten as $p(\mathbf{v}_k|\mathbf{V}_{(k-1):(k-D)})$, or equivalently $p(\mathbf{w}_k = \mathbf{v}_k - \mathbf{A}\mathbf{V}_{(k-1):(k-D)}|\mathbf{V}_{(k-1):1})$ (with some abuse of notation). To make this product tractable, we impose a Gaussian distribution on the initial state and the nuisance $\mathbf{w}_k$. Furthermore, we assume that $\mathbf{w}_k$ is independent of $\mathbf{V}_{(k-1):1}$, or equivalently $\mathbf{w}_k$ is temporally white.

Using Gaussian statistics, the product is calculated recursively as follows. Let's start by assuming that the first $t \geq D$ hidden activities are jointly Gaussian $\mathbf{V}_{t:1} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{\mathbf{V}_{t:1}})$, and that $\mathbf{w}_{t+1}$ is zero-mean Gaussian with covariance $\mathbf{\Sigma}_{\mathbf{w}}$ and independent of $\mathbf{V}_{t:1}$. Using the dynamic Equation 2.6, and defining $\mathbf{A}_{|t} = [\mathbf{A} \ \mathbf{0}_{m\times m(t-D)}]$, we conclude that $\mathbf{V}_{t+1:1}$ is also zero mean Gaussian with the following covariance function:

$$\boldsymbol{\Sigma}_{\mathbf{V}_{t+1:1}} = \begin{bmatrix} E[\mathbf{v}_{t+1}\mathbf{v}_{t+1}^T] & E[\mathbf{v}_{t+1}\mathbf{V}_{t:1}^T] \\ E[\mathbf{V}_{t:1}\mathbf{v}_{t+1}^T] & E[\mathbf{V}_{t:1}\mathbf{V}_{t:1}^T] \end{bmatrix} \tag{2.8}$$

$$E[\mathbf{v}_{t+1}\mathbf{v}_{t+1}^T] = \boldsymbol{\Sigma}_{\mathbf{v}_{t+1}} = \mathbf{A}_{|t}\boldsymbol{\Sigma}_{\mathbf{V}_{t:1}}\mathbf{A}_{|t}^T + \boldsymbol{\Sigma}_{\mathbf{w}} \tag{2.9}$$

$$E[\mathbf{v}_{t+1}\mathbf{V}_{t:1}^T] = \mathbf{A}_{|t}\boldsymbol{\Sigma}_{\mathbf{V}_{t:1}} \tag{2.10}$$

$$E[\mathbf{V}_{t:1}\mathbf{V}_{t:1}^T] = \boldsymbol{\Sigma}_{\mathbf{V}_{t:1}} \tag{2.11}$$

which results in the following recursive equation:

$$\boldsymbol{\Sigma}_{\mathbf{V}_{t+1:1}} = \begin{bmatrix} \mathbf{A}_{|t}\boldsymbol{\Sigma}_{\mathbf{V}_{t:1}}\mathbf{A}_{|t}^T + \boldsymbol{\Sigma}_{\mathbf{w}} & \mathbf{A}_{|t}\boldsymbol{\Sigma}_{\mathbf{V}_{t:1}} \\ (\mathbf{A}_{|t}\boldsymbol{\Sigma}_{\mathbf{V}_{t:1}})^T & \boldsymbol{\Sigma}_{\mathbf{V}_{t:1}} \end{bmatrix} \tag{2.12}$$

Given that $p(\mathcal{V}|\theta) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{V}_{-(D-1):N}})$, this recursive equation can be used to calculate the hidden variable posterior given parameters. However, the resulting covariance matrix is very large, highly nonlinear in connectivity parameters, and worst of all, needs to be inverted. Alternatively, it is possible to formulate a recursion on the inverse of covariance matrix directly. To do this, note that $p(\mathbf{V}_{t+1:1}) = p(\mathbf{V}_{t:1})p(\mathbf{v}_{t+1}|\mathbf{V}_{t:1})$, where $p(\mathbf{V}_{t:1}) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{V}_{t:1}})$ and $p(\mathbf{v}_{t+1}|\mathbf{V}_{t:1}) \sim \mathcal{N}(\mathbf{A}_{|t}\mathbf{V}_{t:1}, \boldsymbol{\Sigma}_{\mathbf{w}})$. Then, multiplying the two distributions:

$$\begin{aligned} p(\mathbf{V}_{t+1:1}) = & \frac{1}{\sqrt{(2\pi)^{m(t+1)}|\boldsymbol{\Sigma}_{\mathbf{w}}||\boldsymbol{\Sigma}_{\mathbf{V}_{t:1}}|}} \times \\ & \exp[-\frac{1}{2}(\mathbf{v}_{t+1} - \mathbf{A}_{|t}\mathbf{V}_{t:1})^T\boldsymbol{\Sigma}_{\mathbf{w}}^{-1}(\mathbf{v}_{t+1} - \mathbf{A}_{|t}\mathbf{V}_{t:1}) \\ & \quad - \frac{1}{2}\mathbf{V}_{t:1}^T\boldsymbol{\Sigma}_{\mathbf{V}_{t:1}}^{-1}\mathbf{V}_{t:1}] \end{aligned} \tag{2.13}$$

Now, if we can write the exponent in terms of $\mathbf{V}_{t+1:1}$, we can find the expression for $\boldsymbol{\Sigma}_{\mathbf{V}_{t+1:1}}$.

The exponent is rewritten as:

$$[(\mathbf{v}_{t+1} - \mathbf{A}_{|t}\mathbf{V}_{t:1})^T, \mathbf{V}_{t:1}^T] \begin{bmatrix} \mathbf{\Sigma}_{\mathbf{w}}^{-1} & \\ & \mathbf{\Sigma}_{\mathbf{V}_{t:1}}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{v}_{t+1} - \mathbf{A}_{|t}\mathbf{V}_{t:1} \\ \mathbf{V}_{t:1} \end{bmatrix} \tag{2.14}$$

Given that:

$$\begin{aligned} \begin{bmatrix} \mathbf{v}_{t+1} - \mathbf{A}_{|t}\mathbf{V}_{t:1} \\ \mathbf{V}_{t:1} \end{bmatrix} &= \begin{bmatrix} \mathbf{I}_{m\times m} & -\mathbf{A}_{|t} \\ \mathbf{0}_{mt\times m} & \mathbf{I}_{mt\times mt} \end{bmatrix} \mathbf{V}_{t+1:1} \\ &\triangleq \mathbf{F}_t\mathbf{V}_{t+1:1} \end{aligned} \tag{2.15}$$

we can write:

$$\begin{aligned} p(\mathbf{V}_{t+1:1}) = & \frac{1}{\sqrt{(2\pi)^{m(t+1)}|\mathbf{\Sigma}_{\mathbf{w}}||\mathbf{\Sigma}_{\mathbf{V}_{t:1}}|}} \times \\ & \exp[-\frac{1}{2}\mathbf{V}_{t+1:1}^T\mathbf{F}_t^T \begin{bmatrix} \mathbf{\Sigma}_{\mathbf{w}}^{-1} & \\ & \mathbf{\Sigma}_{\mathbf{V}_{t:1}}^{-1} \end{bmatrix} \mathbf{F}_t\mathbf{V}_{t+1:1}] \end{aligned} \tag{2.16}$$

Since the determinant of the upper triangular matrix $\mathbf{F}$ is 1, we arrive at the following recursive equation in terms of the inverse of covariance matrix:

$$\mathbf{\Sigma}_{\mathbf{V}_{t+1:1}}^{-1} = \mathbf{F}_t^T \begin{bmatrix} \mathbf{\Sigma}_{\mathbf{w}}^{-1} & \\ & \mathbf{\Sigma}_{\mathbf{V}_{t:1}}^{-1} \end{bmatrix} \mathbf{F}_t \tag{2.17}$$

After partitioning $\mathbf{F}$ and $\mathbf{A}_{|t}$ and some algebraic manipulation, we find it convenient to define:

$$\begin{aligned}
\mathbf{D} &\triangleq \begin{bmatrix} \mathbf{I} \\ -\mathbf{A}^T \end{bmatrix} \mathbf{\Sigma}_{\mathbf{w}}^{-1}[\mathbf{I}, -\mathbf{A}] \\
&\triangleq \begin{bmatrix} \mathbf{D}_{1,1} & \mathbf{D}_{1,2} & \cdots & \mathbf{D}_{1,D+1} \\ \mathbf{D}_{2,1} & \mathbf{D}_{2,2} & \cdots & \mathbf{D}_{2,D+1} \\ \vdots & & & \vdots \\ \mathbf{D}_{D+1,1} & \mathbf{D}_{D+1,2} & \cdots & \mathbf{D}_{D+1,D+1} \end{bmatrix} \\
\mathbf{D}_{i,j} =& \mathbf{A}_{i-1}^T \mathbf{\Sigma}_{\mathbf{w}}^{-1} \mathbf{A}_{j-1} \\
&, \mathbf{A}_0 \triangleq -\mathbf{I}_m \\
&, \mathbf{A}_{-k} = \mathbf{A}_k
\end{aligned} \tag{2.18}$$

Then it follows that $\mathbf{\Sigma}_{\mathbf{V}_{t+1:1}}^{-1}$ is simply formed by creating two all-zero square $m(t+1) \times m(t+1)$ matrices, filling the top left corner of the first with $\mathbf{D}$, filling the bottom right corner of the second with $\mathbf{\Sigma}_{\mathbf{V}_{t:1}}^{-1}$, and finally adding the two matrices. As t gets larger and larger, an implication is that the covariance of initial state will only manifest itself at the bottom right $MD \times MD$ corner of $\mathbf{\Sigma}_{\mathbf{V}_{t:1}}^{-1}$, and the rest of $\mathbf{\Sigma}_{\mathbf{V}_{t:1}}^{-1}$ will be independent of initial state covariance. In what follows, we will ignore the small portion of $\mathbf{\Sigma}_{\mathbf{V}_{t:1}}^{-1}$ occupied by initial state covariance.

Starting with $t = D+1$, and through induction, it is easy to show that $\mathbf{\Sigma}_{\mathbf{V}_{K:1}}^{-1}$ for any $K \geq D$ is a banded $Km \times Km$ matrix consisting of $m \times m$ blocks $\mathbf{S}_{i,j}$ $(i, j \in \{1 \cdots K\})$:

$$\mathbf{\Sigma}_{\mathbf{V}_{K:1}}^{-1} = \begin{bmatrix} \mathbf{S}_{1,1} & \cdots & \mathbf{S}_{1,K} \\ \vdots & & \vdots \\ \mathbf{S}_{K,1} & \cdots & \mathbf{S}_{K,K} \end{bmatrix} \tag{2.19}$$

Ignoring the effect of initial state covariance, for any row index $r \in \{1 \cdots K\}$, and any

$\Delta \in \{0 \cdots K-r\}$:

$$\begin{aligned}\mathbf{S}_{r,r+\Delta} =& \left(\sum_{\ell=\min(0,r-(K-D))+1}^{\min(r,D+1-\Delta)} \mathbf{D}_{\ell,\ell+\Delta} \right) + \\ & \left[\mathbf{1}(r \geq K-D+1) \right] \mathbf{s}_{r-(K-D),r-(K-D)+\Delta} \\ \mathbf{S}_{r+\Delta,r} =& \mathbf{S}^T_{r,r+\Delta} \end{aligned} \tag{2.20}$$

Overall, it is observed that the hidden state V is a zero mean Gaussian whose precision matrix consists of $m \times m$ blocks, and each block is a sum of $\mathbf{A}$-quadratic terms of the form $\mathbf{A}_i^T \mathbf{\Sigma}_{\mathbf{w}}^{-1} \mathbf{A}_j$.

Calculating $p(\mathcal{Y}|V,\theta)p(V|\theta)$

Assuming a temporally white measurement noise $\mathbf{u}$, the observation posterior given hidden states and system parameters is given by:

$$\begin{aligned} p(\mathcal{Y}|\mathcal{V},\theta) =& \prod_{t=1}^{T} \frac{1}{(2\pi)^{N/2}\sqrt{|\mathbf{\Sigma_u}|}} \times \\ & \exp[-\frac{1}{2}(\mathbf{y}_t - \mathbf{C}\mathbf{v}_t)^T \mathbf{\Sigma}_{\mathbf{u}}^{-1} (\mathbf{y}_t - \mathbf{C}\mathbf{v}_t)] \\ =& \frac{1}{(2\pi)^{NT/2}|\mathbf{\Sigma_u}|^{T/2}} \times \\ & \exp[-\frac{1}{2} \sum_{t=1}^{T} \mathbf{y}_t^T \mathbf{\Sigma}_{\mathbf{u}}^{-1} \mathbf{y}_t - \mathbf{y}_t^T \mathbf{\Sigma}_{\mathbf{u}}^{-1} \mathbf{C}\mathbf{v}_t - \\ & \qquad \mathbf{v}_t^T \mathbf{C}^T \mathbf{\Sigma}_{\mathbf{u}}^{-1} \mathbf{y}_t + \mathbf{v}_t^T \mathbf{C}^T \mathbf{\Sigma}_{\mathbf{u}}^{-1} \mathbf{C}\mathbf{v}_t] \\ =& c_1 \exp[-\frac{1}{2}(\boldsymbol{\alpha} - Y^T \mathbf{J} V - V^T \mathbf{J}^T Y + V^T \mathbf{L} V)] \end{aligned} \tag{2.21}$$

with:

$$
\begin{aligned}
c_1 =& \frac{1}{(2\pi)^{NT/2}|\mathbf{\Sigma}_\mathbf{u}|^{T/2}} &(2.22)\\
\boldsymbol{\alpha} =& \sum_{t=1}^{T} \mathbf{y}_t^T \mathbf{\Sigma}_\mathbf{u}^{-1} \mathbf{y}_t &(2.23)\\
Y =& [\mathbf{y}_T^T \cdots \mathbf{y}_1^T]^T &(2.24)\\
\mathbf{J} =& \mathbf{I}_T \otimes (\mathbf{\Sigma}_\mathbf{u}^{-1}\mathbf{C}) &(2.25)\\
\mathbf{L} =& \mathbf{I}_T \otimes (\mathbf{C}^T\mathbf{\Sigma}_\mathbf{u}^{-1}\mathbf{C}) &(2.26)
\end{aligned}
$$

Now, because:

$$p(\mathcal{V}|\theta) \sim \mathcal{N}(0, \mathbf{\Sigma}_{\mathbf{V}_{T:1}}) = c_2 \exp[-.5V^T \mathbf{\Sigma}_{\mathbf{V}_{T:1}}^{-1} V] \tag{2.27}$$

and using the fact that any Gaussian PDF on $\mathbb{R}^d$ with mean $\boldsymbol{\mu}$ and covariance $\mathbf{\Sigma}$ can be written as:

$$
\begin{aligned}
p(\mathbf{x}) =& \exp[\xi + \boldsymbol{\eta}^T\mathbf{x} - \frac{1}{2}\mathbf{x}^T\mathbf{\Lambda}\mathbf{x}]\\
\mathbf{\Lambda} =& \mathbf{\Sigma}^{-1}, \boldsymbol{\eta} = \mathbf{\Sigma}^{-1}\boldsymbol{\mu}\\
\xi =& -\frac{1}{2}(d\log 2\pi - \log|\mathbf{\Lambda}| + \boldsymbol{\eta}^T\mathbf{\Lambda}^{-1}\boldsymbol{\eta}) &(2.28)
\end{aligned}
$$

we rewrite $p(\mathcal{V}|\theta)p(\mathcal{Y}|\mathcal{V},\theta)$ in this standard form. The result is a non-normalized PDF in $\mathcal{V}$ plane. Since the integral of a normalized PDF is equal to 1, the integral $p(\mathcal{Y}|\theta) = \int d\mathcal{V} p(\mathcal{Y}|\mathcal{V},\theta)p(\mathcal{V}|\theta)$ is equal to the normalization constant. The natural logarithm of the integral is then proportional to:

$$
\begin{aligned}
\log p(\mathcal{Y}|\theta) \propto& - NT\log 2\pi - T\log|\mathbf{\Sigma}_\mathbf{u}| - \boldsymbol{\alpha} + \log|\mathbf{\Sigma}_{\mathbf{V}_{T:1}}^{-1}|\\
& - \log|\mathbf{L} + \mathbf{\Sigma}_{\mathbf{V}_{T:1}}^{-1}| + Y^T\mathbf{J}(\mathbf{L} + \mathbf{\Sigma}_{\mathbf{V}_{T:1}}^{-1})^{-1}\mathbf{J}^TY &(2.29)
\end{aligned}
$$

Intractability of the Data Likelihood

For temporally white $\mathbf{u}_k$ and $\mathbf{w}_k$ with covariance matrices $\boldsymbol{\Sigma}_{\mathbf{u}}$ and $\boldsymbol{\Sigma}_{\mathbf{w}}$, it was shown that the log-likelihood can be represented as:

$$\begin{aligned}\log p(\mathcal{Y}|\boldsymbol{\theta}) \propto & -T \log |\boldsymbol{\Sigma}_{\mathbf{u}}| - \boldsymbol{\alpha} + \log |\boldsymbol{\Sigma}_{\mathbf{V}_{T:1}}^{-1}| \\ & - \log |\mathbf{L} + \boldsymbol{\Sigma}_{\mathbf{V}_{T:1}}^{-1}| + Y^T \mathbf{J} (\mathbf{L} + \boldsymbol{\Sigma}_{\mathbf{V}_{T:1}}^{-1})^{-1} \mathbf{J}^T Y\end{aligned} \tag{2.30}$$

with:

$$\boldsymbol{\alpha} = \sum_{t=1}^{T} \mathbf{y}_t^T \boldsymbol{\Sigma}_{\mathbf{u}}^{-1} \mathbf{y}_t \tag{2.31}$$

$$Y = [\mathbf{y}_T^T \cdots \mathbf{y}_1^T]^T \tag{2.32}$$

$$\mathbf{J} = \mathbf{I}_T \otimes (\boldsymbol{\Sigma}_{\mathbf{u}}^{-1} \mathbf{C}) \tag{2.33}$$

$$\mathbf{L} = \mathbf{I}_T \otimes (\mathbf{C}^T \boldsymbol{\Sigma}_{\mathbf{u}}^{-1} \mathbf{C}) \tag{2.34}$$

$$\mathbf{V}_{t_2:t_1} = [\mathbf{v}_{t_2}^T \ \cdots \ \mathbf{v}_{t_1}^T]^T \tag{2.35}$$

where the symbol $\otimes$ denotes the Kronecker product and $\boldsymbol{\Sigma}_{\mathbf{V}_{T:1}}$ represents the covariance matrix of $\mathbf{V}_{T:1}$. The precision matrix $\boldsymbol{\Sigma}_{\mathbf{V}_{T:1}}^{-1}$ is made up of blocks, where each block is quadratic in the connectivity coefficients. The intractability of the log likelihood criterion originates from the terms $\log |\boldsymbol{\Sigma}_{\mathbf{V}_{T:1}}^{-1}|$, $\log |\mathbf{L}+\boldsymbol{\Sigma}_{\mathbf{V}_{T:1}}^{-1}|$, and $(\mathbf{L}+\boldsymbol{\Sigma}_{\mathbf{V}_{T:1}}^{-1})^{-1}$. Rather than directly optimizing this function, we will use the expectation maximization (EM) algorithm.

2.6 The Expectation Maximization Algorithm

The EM algorithm relies on Jensen's inequality, which states that for a concave function ϕ, $\phi(E[X]) \geq E[\phi(X)]$. EM also exploits the fact that the joint observation-state likelihood is typically more tractable than the marginals and conditionals. The EM starts with an assumed value of the parameters at iteration t, namely $\boldsymbol{\theta}^{(t)}$. With $\mathcal{L}(.) \triangleq \log p(.)$, the following function of θ is calculated in the expectation step:

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = E_{p(\mathcal{V}|\mathcal{Y},\boldsymbol{\theta}^{(t)})}[\mathcal{L}(\mathcal{V},\mathcal{Y};\boldsymbol{\theta})] \tag{2.36}$$

The maximization step exploits the tractability of Q by choosing $\boldsymbol{\theta}^{(t+1)}$ such that $Q(\boldsymbol{\theta}^{(t+1)}|\boldsymbol{\theta}^{(t)}) > Q(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}^{(t)})$. It can be shown that increasing Q at each iteration also increases the likelihood function $\log p(Y|\boldsymbol{\theta})$. Furthermore, if the parameter $\boldsymbol{\theta}$ is split into k subsets $\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_k$, the same strategy can be applied to the subsets. First, all subsets other than $\boldsymbol{\theta}_i$ are fixed and Q is optimized over $\boldsymbol{\theta}_i$. Then, the process is repeated for other subsets.

In EM-MAP [68], the Q function is augmented with the log-prior of the parameter $p(\boldsymbol{\theta})$ (or some generic penalty function) and the maximization step is carried out over the sum of Q and the log-prior.

2.6.1 Expectation Step

The first step is to calculate the joint state-observation log-likelihood function $\mathcal{L}$. Assuming a temporally white Gaussian $\mathbf{u}_k$ and $\mathbf{w}_k$, and ignoring the initial state distribution, the parameter dependent portion of $\mathcal{L}$ in Eq. (2.36) is proportional to

$$\begin{aligned} \mathcal{L}(\mathcal{V},\mathcal{Y};\theta) \quad \propto \quad & -T\ln|\boldsymbol{\Sigma}_{\mathbf{u}}| - T\ln|\boldsymbol{\Sigma}_{\mathbf{w}}| \\ & -\sum\nolimits_{t=1}^{T}(\mathbf{y}_t - \mathbf{C}\mathbf{v}_t)^T\boldsymbol{\Sigma}_{\mathbf{u}}^{-1}(\mathbf{y}_t - \mathbf{C}\mathbf{v}_t) \\ & -\sum\nolimits_{t=1}^{T}(\mathbf{v}_t - \mathbf{A}\mathbf{V}_{t-1})^T\boldsymbol{\Sigma}_{\mathbf{w}}^{-1}(\mathbf{v}_t - \mathbf{A}\mathbf{V}_{t-1}) \end{aligned} \tag{2.37}$$

with $\mathbf{V}_t = [\mathbf{v}_t^T, \mathbf{v}_{t-1}^T, \cdots, \mathbf{v}_{t-(D-1)}^T]^T$. The assumption of temporally white noise is common (e.g. see [29, 97, 54, 61, 22, 71]), but the extension to temporally colored noise is possible by augmenting the state vector to include the noise dynamics. For simplicity, we consider the case of temporally white noise.

Next, $\mathcal{L}(\mathcal{V}, \mathcal{Y}; \boldsymbol{\theta})$ is expanded to carry out the expectation step in iteration j. Using the identity $\mathbf{a}^T\mathbf{M}\mathbf{b} = \text{trace}[\mathbf{M}\mathbf{b}\mathbf{a}^T]$ for column vectors $\mathbf{a}$ and $\mathbf{b}$, and replacing $E_{p(\mathcal{V}|\mathcal{Y},\boldsymbol{\theta}^{(j)})}$ with E_j for notational convenience, we have:

$$\begin{aligned} E_j\{\mathcal{L}(\mathcal{V}, \mathcal{Y}; \theta)\} \propto & - T \ln |\boldsymbol{\Sigma}_{\mathbf{u}}| - T \ln |\boldsymbol{\Sigma}_{\mathbf{w}}| - \sum_{t=1}^{T} \Bigg(\mathbf{y}_t^T \boldsymbol{\Sigma}_{\mathbf{u}}^{-1} \mathbf{y}_t \\ & - 2\mathbf{y}_t^T \boldsymbol{\Sigma}_{\mathbf{u}}^{-1} \mathbf{C} E_j\{\mathbf{v}_t\} + \text{tr}\left[\mathbf{C}^T \boldsymbol{\Sigma}_{\mathbf{u}}^{-1} \mathbf{C} E_j\{\mathbf{v}_t \mathbf{v}_t^T\}\right] \\ & + \text{tr}\left[\boldsymbol{\Sigma}_{\mathbf{w}}^{-1} E_j\{\mathbf{v}_t \mathbf{v}_t^T\}\right] - 2\text{tr}\left[\mathbf{A}^T \boldsymbol{\Sigma}_{\mathbf{w}}^{-1} E_j\{\mathbf{v}_t \mathbf{V}_{t-1}^T\}\right] \\ & + \text{tr}\left[\mathbf{A}^T \boldsymbol{\Sigma}_{\mathbf{w}}^{-1} \mathbf{A} E_j\{\mathbf{V}_{t-1} \mathbf{V}_{t-1}^T\}\right] \Bigg) \end{aligned} \tag{2.38}$$

Therefore, it suffices to calculate the following expectations:

$$\mathbf{M}_{j,t} \triangleq E_j\{\mathbf{V}_t\} \tag{2.39}$$

$$\mathbf{F}_{j,t} \triangleq E_j\{\mathbf{V}_t \mathbf{V}_t^T\} \tag{2.40}$$

$$\mathbf{G}_{j,t} \triangleq E_j\{\mathbf{V}_t \mathbf{V}_{t-1}^T\} \tag{2.41}$$

To do this, the model is rewritten in state space form:

$$\mathbf{V}_t = \mathbf{A}_e \mathbf{V}_{t-1} + \mathbf{W}_t \tag{2.42}$$

$$\mathbf{y}_t = \mathbf{C}_e \mathbf{V}_t + \mathbf{u}_t \tag{2.43}$$

with e representing *extension*, and:

$$\mathbf{A}_e = \begin{bmatrix} \multicolumn{2}{c}{\mathbf{A}} \\ \mathbf{I}_{M(D-1)} & \mathbf{0}_{M(D-1)\times M} \end{bmatrix} \tag{2.44}$$

$$\mathbf{A} = [\mathbf{A}_1, \cdots, \mathbf{A}_D] \tag{2.45}$$

$$\mathbf{W}_t = [\mathbf{w}^T \ \mathbf{0}_{1\times M(D-1)}]^T \tag{2.46}$$

$$\mathbf{C}_e = [\mathbf{C} \ \mathbf{0}_{N\times M(D-1)}]. \tag{2.47}$$

While only temporally white noises are used for the derivations here, the state space model may be modified to take colored noise into account as well. A colored noise $\mathbf{w}_k$ may be in general realized by the state space form:

$$\mathbf{w}_k = \mathbf{S}\mathbf{z}_k, \ \mathbf{z}_k = \mathbf{T}\mathbf{z}_{k-1} + \mathbf{x}_k \tag{2.48}$$

where $\mathbf{x}_k$ is a temporally white noise and $\mathbf{z}_k$ is the hidden state vector whose state equation determines the dynamics of $\mathbf{w}_k$. After some algebraic manipulation, the state equation in Eq. (2.43) may be modified as:

$$\mathbf{V}_k = \begin{bmatrix} \mathbf{A} & \mathbf{ST} \\ \mathbf{I}_{M(D-1)} & \mathbf{0}_{M(D-1)\times M} \\ \mathbf{0}_{M\times M(D-1)} & \mathbf{T} \end{bmatrix} \mathbf{V}_{k-1} + \begin{bmatrix} \mathbf{0}_{MD\times 1} \\ \mathbf{x}_{k-1} \end{bmatrix} \tag{2.49}$$

with $\mathbf{V}_k = [\mathbf{v}_k, \mathbf{v}_{k-1}, \cdots, \mathbf{v}_{k-D+1}, \mathbf{z}_k]^T$.

Assuming temporally white Gaussian processes $\mathbf{u}_k$ and $\mathbf{w}_k$, and a Gaussian initial distribution, all states are normally distributed, and the expectations can be calculated using a Kalman smoother (KS) [94]. The Kalman smoother is an offline procedure that requires a forward pass (the Kalman filter) through the data, storing the results, and then a backward pass. If complexity or real-time operation is an issue, the Kalman filter (KF) may replace the smoother at the cost of suboptimality in the mean and covariance estimates. The KF does not require storing the outputs and the estimates are calculated online. The sub-optimality of the KF compared to the KS is due to the fact that the KS uses all observations while the KF only uses the past observations. However, the suboptimality has negligible influence on our estimation results because, as will be shown shortly, the parameter estimates depend on the temporal averages of the expectations, and this averaging absorbs the difference between the KF and KS state estimates. Therefore, we use the KF instead of the KS in the remainder of the paper to calculate the expectations.

The KF calculates the mean $E_j\{\mathbf{V}_t\} = \mathbf{M}_{j,t}$ and the following covariances directly:

$$\begin{aligned} \mathbf{P}_{j,t} &= E_j\{(\mathbf{V}_t - \mathbf{M}_{j,t})(\mathbf{V}_t - \mathbf{M}_{j,t})^T\} && (2.50) \\ \mathbf{R}_{j,t} &= E_j\{(\mathbf{V}_t - \mathbf{M}_{j,t})(\mathbf{V}_{t-1} - \mathbf{M}_{j,t-1})^T\}\ . && (2.51) \end{aligned}$$

The variables $\mathbf{P}_{j,t}$ and $\mathbf{R}_{j,t}$ are then used to calculate the expectations in Eq. (2.38). Using MATLAB™ matrix indexing notation:

$$\begin{aligned} \mathbf{m}_{j,t} &= E_j\{\mathbf{v}_t\} = \mathbf{M}_{j,t}(1:m) \\ \mathbf{F}_{j,t} &= E_j\{\mathbf{V}_t\mathbf{V}_t^T\} = \mathbf{P}_{j,t} + \mathbf{M}_{j,t}\mathbf{M}_{j,t}^T \\ \mathbf{G}_{j,t} &= E_j\{\mathbf{V}_t\mathbf{V}_{t-1}^T\} = \mathbf{R}_{j,t} + \mathbf{M}_{j,t}\mathbf{M}_{j,t-1}^T \\ \mathbf{f}_{j,t} &= E_j\{\mathbf{v}_t\mathbf{v}_t^T\} = \mathbf{F}_{j,t}(1:m, 1:m) \\ \mathbf{g}_{j,t} &= E_j\{\mathbf{v}_t\mathbf{V}_{t-1}^T\} = \mathbf{G}_{j,t}(1:m, :) && (2.52) \end{aligned}$$

2.6.2 Maximization step

Suppose that parameters $\mathbf{A}^{(j)}$, $\mathbf{\Sigma}_{\mathbf{u}}^{(j)}$, and $\mathbf{\Sigma}_{\mathbf{w}}^{(j)}$ at step j are given. The maximization step at iteration $j+1$ is given by the following sequence of maximizations:

$$\begin{aligned}
\mathbf{A}^{(j+1)} &= \arg\max_{\mathbf{A}\in\mathcal{S}} Q(\mathbf{A}|\mathbf{A}^{(j)}, \mathbf{\Sigma}_{\mathbf{u}}^{(j)}, \mathbf{\Sigma}_{\mathbf{w}}^{(j)}) - C_A(\mathbf{A}) \\
\mathbf{\Sigma}_{\mathbf{u}}^{(j+1)} &= \arg\max_{\mathbf{\Sigma}_{\mathbf{u}}\in\mathcal{P}_N} Q(\mathbf{\Sigma}_{\mathbf{u}}|\mathbf{A}^{(j+1)}, \mathbf{\Sigma}_{\mathbf{u}}^{(j)}, \mathbf{\Sigma}_{\mathbf{w}}^{(j)}) \\
\mathbf{\Sigma}_{\mathbf{w}}^{(j+1)} &= \arg\max_{\mathbf{\Sigma}_{\mathbf{w}}\in\mathcal{P}_M} Q(\mathbf{\Sigma}_{\mathbf{w}}|\mathbf{A}^{(j+1)}, \mathbf{\Sigma}_{\mathbf{u}}^{(j+1)}, \mathbf{\Sigma}_{\mathbf{w}}^{(j)})\ , \qquad (2.53)
\end{aligned}$$

where $\mathcal{S}$ is the set of all $\mathbf{A}$'s that lead to a stable and stationary system, and $\mathcal{P}_M$ ($\mathcal{P}_N$) is the set of symmetric positive definite (covariance) matrices in $\mathbb{R}^M$ ($\mathbb{R}^N$). The term C_A is a penalty function that encourages sparse connectivity matrices. An example of such a penalty function is $C_A(\mathbf{A}) = \lambda||\mathbf{A}||_1$ which results from applying a Laplacian prior on $\mathbf{A}$ [5]. The hyper-parameter λ should be selected properly by cross-validation, as discussed later.

2.7 Exploiting Sparsity in Estimation of the Connectivity Matrix

The non-differentiability of C_A makes the maximization step for $\mathbf{A}$ the most complicated among all parameters. The log-likelihood function consists of the Q function, which is quadratic in $\mathbf{A}$, and the ℓ_1 penalty function.

While many packages are available to solve the LASSO problem [122], all of them are formulated in terms of the vectorized version of the regression weights ($\mathbf{A}$ in our problem) to solve

$$\min_{\mathrm{vec}(\mathbf{A})} ||\mathbf{b} - \mathbf{B}\mathrm{vec}(\mathbf{A})||_2^2 + \lambda||\mathrm{vec}(\mathbf{A})||_1 \ . \tag{2.54}$$

The vector $\mathbf{b} \in \mathbb{R}^{M^2D\times 1}$ and matrix $\mathbf{B} \in \mathbb{R}^{M^2D\times M^2D}$ are inputs to the algorithm. To discuss the dimensionality of this problem, consider the application of our method to the brain connectivity estimation problem. We usually consider at least $M = 20$ regions of interest (ROI) and an upper bound for D can be calculated by estimating the maximum distance between the ROI and typical neural transmission speeds. With an action conduction speed of 1-10 m/sec and an ROI separation of 5-10 cm, a reasonable upper-bound for the delay is about 20 msec. For an EEG system sampled at 1kHz, this translates to $D = 20$. Although $\mathbf{b}$ and $\mathbf{B}$ can be calculated in terms of the parameters of our problem, the dimensionality ($M^2D \approx 80,000$) makes it impossible to use available solvers. Consequently, a more efficient alternative should be designed.

To mitigate this issue, we propose to use coordinate descent to solve the optimization problem. Coordinate descent optimization updates one coordinate (or block of dependent coordinates) at a time in an iterative fashion, leaving the others fixed, until convergence is reached. In some cases, this update can be carried out analytically, yielding a fast and simple algorithm. Typically, the problem is solved for a large value of the regularization parameter and the solution is used as a warm start for solving the problem with a smaller regularization parameter. The coordinate descent optimization is computationally efficient to the extent that the coordinate updates are efficient.

The optimization problem in Eq. (2.53) can be rewritten as

$$\mathbf{A}^{(j+1)} {=} \operatorname*{argmax}_{\mathbf{A}} \left[-\lambda'|\mathbf{A}|_1 - E_j \left\{ \sum_{t=1}^{T} \mathbf{w}_t^T \mathbf{\Gamma}_{\mathbf{w}}^{(j)} \mathbf{w}_t \right\} \right] \tag{2.55}$$

with $\mathbf{w}_t = \mathbf{v}_t - \mathbf{A}\mathbf{V}_{t-1}$, $\mathbf{\Gamma}_{\mathbf{w}} = \mathbf{\Sigma}_{\mathbf{w}}^{-1}$, and $\lambda' \triangleq T\lambda$ to factor out the scaling of the regularization parameters with the data size T. After expansion:

$$\begin{aligned}\mathbf{A}^{(j+1)} &= \arg\max_{\mathbf{A}} \Bigg[-\lambda|\mathbf{A}|_1 + \left\langle E\left\{\mathbf{v}_t^T \mathbf{\Gamma}_{\mathbf{w}}^{(j)} \mathbf{A}\mathbf{V}_{t-1}\right\}\right\rangle \\ &\quad + \left\langle E\left\{\mathbf{V}_{t-1}^T \mathbf{A}^T \mathbf{\Gamma}_{\mathbf{w}}^{(j)} \mathbf{v}_t\right\}\right\rangle - \left\langle E\left\{\mathbf{V}_{t-1}^T \mathbf{A}^T \mathbf{\Gamma}_{\mathbf{w}}^{(j)} \mathbf{A}\mathbf{V}_{t-1}\right\}\right\rangle \Bigg] \\ &\triangleq \arg\max_{\mathbf{A}} \left\{-\lambda|\mathbf{A}|_1 + K(\mathbf{A})\right\} \end{aligned} \tag{2.56}$$

with $\langle x_t \rangle \triangleq (\sum_{t=1}^{T} x_t)/T$. Due to the non-differentiability of the ℓ_1 norm, the first order optimality condition is given in terms of the matrix gradient of K and matrix sub-differential of the ℓ_1 norm [12]:

$$\mathbf{0} \in \boldsymbol{\nabla}_{\mathbf{A}} K(\mathbf{A}) - \lambda \boldsymbol{\partial}_{\mathbf{A}} |\mathbf{A}|_1 \ . \tag{2.57}$$

Since coordinate descent operates component-wise, we can rewrite the optimality condition for a single element of $\mathbf{A}$, namely a_{pq}, while fixing all other elements:

$$0 \in \nabla_{a_{pq}} K(\mathbf{A}) - \lambda \partial_{a_{pq}} |\mathbf{A}|_1 \ . \tag{2.58}$$

The optimum a_{pq} is then found through conditional analysis:

$a_{pq} \neq 0$: Assume that the optimum a_{pq} is not zero. Then, $\partial_{a_{pq}}|\mathbf{A}|_1 = \text{sign}(a_{pq})$. Also, using the following identities (small and capital letters represent vectors and matrices respectively):

$$\frac{\partial\ \mathbf{a}^T\mathbf{M}\mathbf{b}}{\partial\ \mathbf{M}} = \mathbf{b}\mathbf{a}^T \tag{2.59}$$

$$\frac{\partial\ \mathbf{a}^T\mathbf{M}^T\mathbf{b}}{\partial\ \mathbf{M}} = \mathbf{a}\mathbf{b}^T \tag{2.60}$$

$$\frac{\partial\ \mathbf{b}^T\mathbf{X}^T\mathbf{D}\mathbf{X}\mathbf{c}}{\partial\ \mathbf{X}} = \mathbf{c}\mathbf{b}^T\mathbf{X}^T\mathbf{D} + \mathbf{b}\mathbf{c}^T\mathbf{X}^T\mathbf{D}^T \tag{2.61}$$

and assuming that $\mathbf{\Gamma}_{\mathbf{w}}^{(j)}$ is symmetric, we get

$$
\begin{aligned}
\boldsymbol{\nabla}_{\mathbf{A}} K(\mathbf{A}) &= \left\langle E[\mathbf{V}_{t-1}\mathbf{v}_t^T]\right\rangle 2\mathbf{\Gamma}_{\mathbf{w}}^{(j)} - \left\langle E[\mathbf{V}_{t-1}\mathbf{V}_{t-1}^T]\right\rangle \mathbf{A}^T 2\mathbf{\Gamma}_{\mathbf{w}}^{(j)} \\
&= \left\langle \mathbf{g}_{j,t}^T \right\rangle 2\mathbf{\Gamma}_{\mathbf{w}}^{(j)} - \langle \mathbf{F}_{j,t-1}\rangle \mathbf{A}^T 2\mathbf{\Gamma}_{\mathbf{w}}^{(j)} \ , \qquad (2.62)
\end{aligned}
$$

which is linear in the elements of $\mathbf{A}$. If $\mathbf{\Gamma}_{\mathbf{w}}^{(j)}$ is not symmetric, all the terms $2\mathbf{\Gamma}_{\mathbf{w}}^{(j)}$ should be replaced with $\mathbf{\Gamma}_{\mathbf{w}}^{(j)} + (\mathbf{\Gamma}_{\mathbf{w}}^{(j)})^T$. Now, by definition, $\nabla_{a_{pq}} K(\mathbf{A}) = [\boldsymbol{\nabla}_{\mathbf{A}} K(\mathbf{A})]_{qp}$. Using the definition of matrix multiplication and fixing all elements of $\mathbf{A}$ except for a_{pq}, the negated sub-differential for a_{pq} can be written as:

$$
-\nabla_{a_{pq}} K(\mathbf{A}) + \lambda \partial_{a_{pq}} |\mathbf{A}|_1 = r_{pq} a_{pq} + \tilde{s}_{pq} + \lambda \mathrm{sign}(a_{pq}) \qquad (2.63)
$$

with

$$
\begin{aligned}
r_{pq} &= 2\mathbf{\Gamma}_{\mathbf{w}}^{(j)}{}_{pp} \langle \mathbf{F}_{j,t-1}\rangle_{qq} \\
\tilde{s}_{pq} &= \left[2\mathbf{\Gamma}_{\mathbf{w}}^{(j)} \tilde{\mathbf{A}}_{\backslash pq} \langle \mathbf{F}_{j,t-1}\rangle\right]_{pq} - \left[2\mathbf{\Gamma}_{\mathbf{w}}^{(j)} \langle \mathbf{g}_{j,t}\rangle\right]_{pq} \ , \qquad (2.64)
\end{aligned}
$$

where $\tilde{\mathbf{A}}_{\backslash pq}$ is the current estimate of $\mathbf{A}$ with element (p, q) set to zero.

For the first-order optimality (applied to the negated differential) to truly represent a minimum, the second derivative should be positive; this means that r_{pq} should be positive. In other words, the diagonal elements of the precision matrix should be positive. A sufficient condition for this is to guarantee the positive definiteness of the covariance estimates as discussed in Section 2.8. If a diagonal element is negative, the first order optimality condition can lead to a maximum of the negated score function rather than a minimum, and thus decreases the score function rather than increasing it. As a workaround, it is possible to limit the covariance matrix to the space of diagonal matrices with positive elements, or to incorporate the positive definiteness of the covariance matrix into the estimation (as will be done by using the Cholesky factor in Section 2.8).

For now, assume that r_{pq} is positive. Fig. 2.1 shows how to solve the equation $ra + s + \lambda \text{sign}(a) = 0$ for positive r. The non-zero solution exists only when $\lambda < |\tilde{s}_{pq}|$. With this condition, the first order optimality condition is met at:

$$a_{pq} \leftarrow \text{sign}(\tilde{s}_{pq}) \frac{\lambda - |\tilde{s}_{pq}|}{r_{pq}} \tag{2.65}$$

$a_{pq} = 0$: Suppose that the optimum a_{pq} is actually zero. The first order optimality condition is now

$$0 \in \nabla_{a_{pq}} K(\mathbf{A}) - \lambda[-1, 1] \tag{2.66}$$

or

$$0 \in r_{pq} a_{pa} + \tilde{s}_{pq} + \lambda[-1, 1] = \tilde{s}_{pq} + \lambda[-1, 1] \ , \tag{2.67}$$

which is equivalent to $\lambda \geq |\tilde{s}_{pq}|$. Combining the two conditions, the optimum a_{pq} can be written compactly as a soft thresholding operator:

$$a_{pq} \leftarrow \text{sign}(\tilde{s}_{pq}) \min \left(0, \frac{\lambda - |\tilde{s}_{pq}|}{r_{pq}} \right) \ . \tag{2.68}$$

The process is repeated element-by-element (or block-by-block of elements) until convergence is reached. Unlike the original formulation of Eq. (2.54) in terms of vec($\mathbf{A}$), this approach only requires knowledge of $\langle \mathbf{F}_{j,t-1} \rangle$, $\mathbf{\Gamma}_{\mathbf{w}}^{(j)}$, $\langle \mathbf{g}_{j,t} \rangle$. This makes the problem tractable, and reduces the number of required parameters from $(M^2 D)^2$ to $M^2(D^2+D+1)$, an improvement

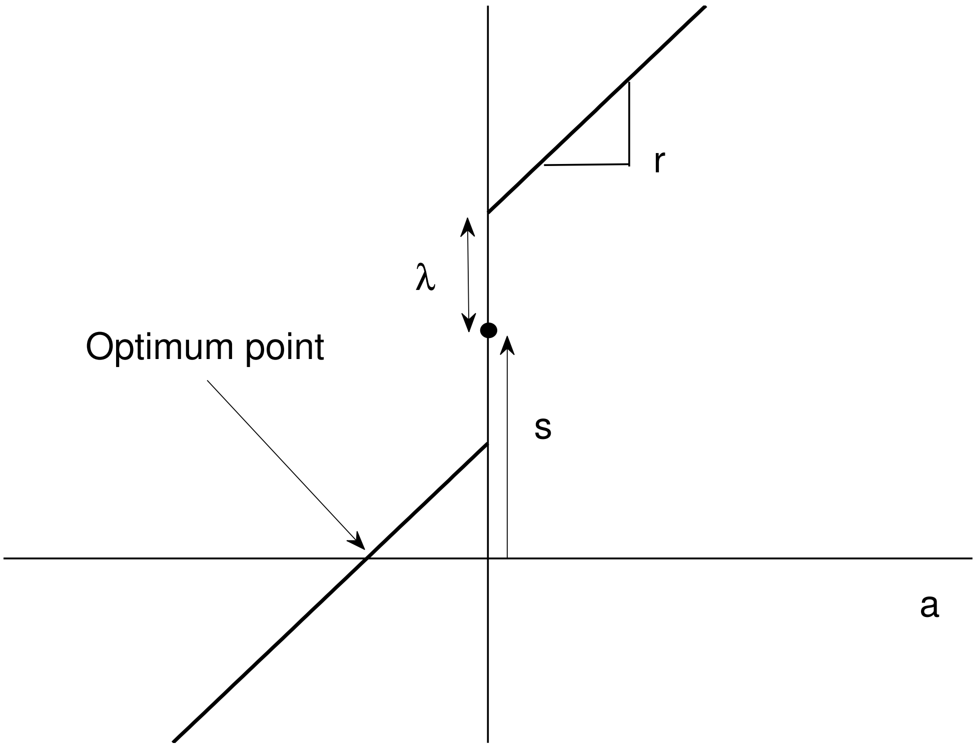


Figure 2.1: Finding the first-order optimal point in an ℓ_1 regularization problem.

of order M^2.

In order to further speed up the process, it is possible to update multiple elements simultaneously. However, a new update equation should be designed such that the simultaneous update of multiple elements is efficient. This is not possible with the structure of $\tilde{\mathbf{A}}_{\backslash pq}$, since the matrix should be constructed for every element separately. The trick is to replace $\tilde{\mathbf{A}}_{\backslash pq}$ with $\tilde{\mathbf{A}}$ and cancel out the extra contribution of $\tilde{a}_{pq}$ by subtracting an auxiliary term:

$$\tilde{s}_{pq} = \left[2\boldsymbol{\Gamma}_{\mathbf{w}}^{(j)}\tilde{\mathbf{A}}\left\langle\mathbf{F}_{j,t-1}\right\rangle\right]_{pq} - 2\left[\boldsymbol{\Gamma}_{\mathbf{w}}^{(j)}\right]_{pp}\tilde{a}_{pq}\left\langle\mathbf{F}_{j,t-1}\right\rangle_{qq} - \left[2\boldsymbol{\Gamma}_{\mathbf{w}}^{(j)}\left\langle\mathbf{g}_{j,t}\right\rangle\right]_{pq} \tag{2.69}$$

To simultaneously update the elements of $\mathbf{A}$ with row indices in $\phi \subset \{1 \cdots M\}$ and column indices in $\psi \subset \{1 \cdots MD\}$, the scalar equation $ra + s + \lambda\text{sign}(a) = 0$ is rewritten in matrix form as:

$$\begin{aligned} &R_{\phi\psi} \odot \mathbf{A}_{\phi,\psi} + \tilde{S}_{\phi,\psi} + \lambda\text{sign}(\mathbf{A}_{\phi,\psi}) = \mathbf{0}_{|\phi|\times|\psi|} \\ &R_{\phi\psi} = \text{diag}_{\phi}\left(2\boldsymbol{\Gamma}_{\mathbf{w}}^{(j)}\right)\text{diag}_{\psi}^{T}\left(\left\langle\mathbf{F}_{j,t-1}\right\rangle\right) \\ &\tilde{S}_{\phi\psi} = \left[2\boldsymbol{\Gamma}_{\mathbf{w}}^{(j)}\tilde{\mathbf{A}}\left\langle\mathbf{F}_{j,t-1}\right\rangle - 2\,\boldsymbol{\Gamma}_{\mathbf{w}}^{(j)}\left\langle\mathbf{g}_{j,t}\right\rangle\right]_{\phi\psi} - R_{\phi\psi} \odot \tilde{\mathbf{A}}_{\phi\psi} \end{aligned} \tag{2.70}$$

with $(\mathbf{X}_{\phi,\psi})_{i,j} = \mathbf{X}_{\{\phi\}_i,\{\psi\}_j}$, $\odot$ representing the Hadamard product, and $\text{diag}_{\phi}(\mathbf{X})$ a column vector whose ith element is the $\{\phi\}_i$th diagonal element of $\mathbf{X}$. Then, the update equation will be:

$$\mathbf{A}_{\phi\psi} \leftarrow \text{sign}(\tilde{S}_{\phi\psi}) \odot \min\left(0, \frac{\lambda - |\tilde{S}_{\phi\psi}|}{R_{\phi\psi}}\right) \tag{2.71}$$

with all operations performed element-wise.

Finally, to enforce the stability of the system, the elements of $\mathbf{A}$ at each iteration are shrunk by a factor proportional to the largest eigenvalue of the current estimate of $\mathbf{A}$. The two-step procedure of updating $\mathbf{A}$ and enforcing stability is repeated until convergence is reached.

2.8 Estimating Positive Definite Noise Covariances

Rather than calculating the derivative w.r.t. $\mathbf{\Sigma_w}$, we take the derivative of the joint likelihood function w.r.t. $\mathbf{\Gamma_w}$. To do this, we exploit the following identities:

$$\frac{\partial \ln|\mathbf{X}|}{\partial \mathbf{X}} = \mathbf{X}^{-1}, \ \ln|\mathbf{X}^{-1}| = -\ln|\mathbf{X}|, \ \frac{\partial \, \mathbf{z}^T\mathbf{M}\mathbf{z}}{\partial \mathbf{M}} = \mathbf{z}\mathbf{z}^T \ . \tag{2.72}$$

Then, with $\mathbf{w}_t^{(j)} = \mathbf{v}_t - \mathbf{A}^{(j)}\mathbf{V}_{t-1}$:

$$\begin{aligned}\frac{\partial \ E_j\mathcal{L}}{\partial \,\mathbf{\Gamma_w}} =& E_j\frac{\partial \ \mathcal{L}}{\partial \,\mathbf{\Gamma_w}} \\ =& E_j\frac{\partial}{\partial \,\mathbf{\Gamma_w}}\left[-T\ln|\mathbf{\Sigma_w}| - \sum_{t=1}^{T}\mathbf{w}_t^{(j)^T}\mathbf{\Gamma_w}\mathbf{w}_t^{(j)}\right] \\ =& T\mathbf{\Sigma_w} - \sum_{t=1}^{T} E_j\{\mathbf{w}_t^{(j)}\mathbf{w}_t^{(j)^T}\} = \mathbf{0}_{M\times M} \ . \end{aligned} \tag{2.73}$$

After replacing the expression for $\mathbf{w}_t^{(j)}$ and calculating the expectations, we get the following update equation for $\mathbf{\Sigma_w}$:

$$\mathbf{\Sigma}_{\mathbf{w}}^{(j+1)} = \langle \mathbf{f}_{j,t}\rangle - \langle \mathbf{g}_{j,t}\rangle \, \mathbf{A}^{(j)^T} - \mathbf{A}^{(j)} \, \langle \mathbf{g}_{j,t}\rangle^T + \mathbf{A}^{(j)} \, \langle \mathbf{F}_{j,t-1}\rangle \, \mathbf{A}^{(j)^T} \tag{2.74}$$

with $\langle x_t\rangle \triangleq (\sum_{t=1}^{T} x_t)/T$ representing the temporal average of the argument. Note that this update equation does not guarantee the positive definiteness of the covariance matrix, which can cause problems in estimation of the connectivity matrix, as explained in Sec. 2.7. Apart from the more general Cholesky factorization discussed below, this problem can be addressed by restricting the covariance matrix to be diagonal. In this case, $\mathbf{\Sigma_w} = \mathrm{diag}(\sigma_{w,1}^2, \cdots, \sigma_{w,M}^2)$. Choosing $\sigma_{w,n}^{-2}$ as the parameters of interest and setting the derivative to 0, it is straightforward to show that the update equation is:

$$(\sigma^2_{w,n})^{(j+1)} = \left\langle E_j \left| (\mathbf{w}_t)_n \right|^2 \right\rangle = (\left\langle E_j\{\mathbf{w}_t \mathbf{w}_t^T\}\right\rangle)_{n,n} = (\mathbf{\Sigma}_{\mathbf{w}}^{(j+1)})_{n,n} \tag{2.75}$$

In the simple case of $\mathbf{\Sigma_w} = \sigma^2_w \mathbf{I}_M$, the update equation reduces to:

$$(\sigma^2_w)^{(j+1)} = \mathrm{tr}(\mathbf{\Sigma}_{\mathbf{w}}^{(j+1)})/M \ . \tag{2.76}$$

Similar equations can be derived for the $\mathbf{\Sigma_u}$ update. With $\mathbf{u}_t^{(j)} = \mathbf{y}_t - \mathbf{C}\mathbf{v}_t$:

$$\begin{aligned} \frac{\partial\ E\mathcal{L}}{\partial\ \mathbf{\Gamma_u}} &= E\frac{\partial\ \mathcal{L}}{\partial\ \mathbf{\Gamma_u}} = E\frac{\partial}{\partial\ \mathbf{\Gamma_u}}\left[-T\ln|\mathbf{\Sigma_u}| - \sum_{t=1}^{T} \mathbf{u}_t^{(j)^T} \mathbf{\Gamma_u} \mathbf{u}_t^{(j)} \right] \\ &= T\mathbf{\Sigma_u} - \sum_{t=1}^{T} E[\mathbf{u}_t^{(j)} \mathbf{u}_t^{(j)^T}] = \mathbf{0}_{N\times N} \end{aligned} \tag{2.77}$$

which results in:

$$\mathbf{\Sigma}_{\mathbf{u}}^{(j+1)} = \left\langle \mathbf{y}_t \mathbf{y}_t^T \right\rangle - \left\langle \mathbf{y}_t \mathbf{m}_{j,t}^T \right\rangle \mathbf{C}^T - \mathbf{C} \left\langle \mathbf{y}_t \mathbf{m}_{j,t}^T \right\rangle^T + \mathbf{C} \left\langle \mathbf{f}_{j,t} \right\rangle \mathbf{C}^T \ . \tag{2.78}$$

For the case of a diagonal $\mathbf{\Sigma_u}$, update equations similar to those for a diagonal $\mathbf{\Sigma_w}$ are used.

Unfortunately, the covariance update Eqs. 2.74 and 2.78 do not guarantee the positive definiteness of the result. Estimation of positive definite covariance matrices has been extensively studied in the literature [90, 119, 53]. In what follows below, we exploit the uniqueness of the Cholesky decomposition of symmetric positive definite matrices.

Any positive definite matrix is readily parameterized in the space of its Cholesky factor. Furthermore, the Cholesky factors have real and positive diagonal elements. The uniqueness facilitates casting an unconstrained optimization problem in the space of Cholesky factor in order to keep the estimate of covariance positive definite at all iterations.

Given the log likelihood function, it is more convenient to formulate the problem in terms of the Cholesky factor of precision matrix rather than covariance matrix. To exploit the Cholesky factorization in our optimization, note that the $\boldsymbol{\Gamma}$-dependent terms in the log-likelihood are of the following form:

$$\mathbf{a}^T \boldsymbol{\Gamma} \mathbf{b} \tag{2.79}$$

$$\ln |\boldsymbol{\Gamma}| \ . \tag{2.80}$$

Define the Cholesky decomposition of $\boldsymbol{\Gamma} \in \mathbb{R}^{k \times k}$ as $\boldsymbol{\Gamma} = \mathbf{L}\mathbf{L}^T$, where $\mathbf{L}$ is a lower triangular real matrix with positive diagonal elements.

Derivative of $\ln |\boldsymbol{\Gamma}|$

The derivative of $\ln |\boldsymbol{\Gamma}|$ w.r.t $\mathbf{L}$ is calculated as follows. First, $|\boldsymbol{\Gamma}| = |\mathbf{L}||\mathbf{L}^T| = |\mathbf{L}|^2$. Consequently, $\ln |\boldsymbol{\Gamma}| = 2 \ln |\mathbf{L}|$. Also, since $\mathbf{L}$ is triangular, its determinant is equal to the product of its diagonal elements. Then:

$$\begin{aligned}
\frac{\partial \ln |\mathbf{L}|}{\partial \operatorname{vech}(\mathbf{L})} &= \frac{\partial \sum_{i=1}^{k} \ln L_{ii}}{\partial \operatorname{vech}(\mathbf{L})} = \frac{\partial \sum_{i=1}^{k} \ln L_{ii}}{\partial \operatorname{vec}(\mathbf{L})} \frac{\partial \operatorname{vec}(\mathbf{L})}{\partial \operatorname{vech}(\mathbf{L})} \\
&= [L_{11}^{-1}, \mathbf{0}_{1 \times k}, L_{22}^{-1}, \mathbf{0}_{1 \times k}, \cdots, L_{kk}^{-1}] S_k^T \\
&= [L_{11}^{-1}, \mathbf{0}_{1 \times k-1}, L_{22}^{-1}, \mathbf{0}_{1 \times k-2}, \cdots, \mathbf{0}_{1 \times 1}, L_{kk}^{-1}]
\end{aligned} \tag{2.81}$$

where $S_k \in \mathbb{R}^{k(k+1)/2 \times k^2}$ is the elimination matrix of order k [75], and vech (.) is the half-vectorization operator [44].

Derivative of $\mathbf{a}^T \mathbf{\Gamma} \mathbf{b}$

Applying the chain rule [76] to the identity $\mathbf{a}^T \mathbf{\Gamma} \mathbf{b} = (\mathbf{a}^T \mathbf{L})(\mathbf{L}^T \mathbf{b})$:

$$\frac{\partial \ \mathbf{a}^T \mathbf{\Gamma} \mathbf{b}}{\partial \ \mathrm{vech}\,(\mathbf{L})} = (\mathbf{b}^T \mathbf{L})(\mathbf{I}_k \otimes \mathbf{a}^T) S_k^T + (\mathbf{a}^T \mathbf{L})(\mathbf{I}_k \otimes \mathbf{b}^T) S_k^T \tag{2.82}$$

which is linear in the elements of $\mathbf{L}$. To emphasize this linearity, we can factor out S_k^T and use the following identities:

$$\mathrm{vec}(\mathbf{AXB}) = (\mathbf{B}^T \otimes \mathbf{A})\mathrm{vec}(\mathbf{X}) \tag{2.83}$$

$$\mathrm{vec}(\mathbf{L})^T = \mathrm{vech}\,(\mathbf{L})^T \, S_k \tag{2.84}$$

$$\mathbf{x}^T \otimes \mathbf{y} = \mathbf{y}\mathbf{x}^T \tag{2.85}$$

for generic matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{X}$ and column vectors $\mathbf{x}$ and $\mathbf{y}$ to write the derivative as:

$$\frac{\partial \ \mathbf{a}^T \mathbf{\Gamma} \mathbf{b}}{\partial \ \mathrm{vech}\,(\mathbf{L})} = \mathrm{vech}\,(\mathbf{L})^T \, S_k [\mathbf{I}_k \otimes (\mathbf{a}\mathbf{b}^T + \mathbf{b}\mathbf{a}^T)] S_k^T \ . \tag{2.86}$$

Derivative of the log-likelihood

For the following prototypical expression:

$$f(\mathbf{L}) = -T \ln |\mathbf{\Sigma}| - \sum_{t=1}^{T} E[\mathbf{a}_t^T \mathbf{\Gamma} \mathbf{a}_t] \ , \tag{2.87}$$

setting the transpose of the derivative to 0 leads to the following nonlinear equation in $\mathrm{vech}\,(\mathbf{L})$:

$$\left(\frac{\partial\, f}{\partial\ \mathrm{vech}\,(\mathbf{L})}\right)^T \propto S_k\mathcal{D}(\mathbf{L}) - S_k[\mathbf{I}_k \otimes \mathbf{\Lambda}]S_k^T\mathrm{vech}\,(\mathbf{L}) = \mathbf{0}_{\frac{k(k+1)}{2}\times 1}$$

$$\mathbf{\Lambda} \triangleq \left\langle E[\mathbf{a}\mathbf{a}^T]\right\rangle$$

$$\mathcal{D}(\mathbf{L}) \triangleq [L_{11}^{-1}, \mathbf{0}_{1\times k}, L_{22}^{-1}, \mathbf{0}_{1\times k}, \cdots, \mathbf{0}_{1\times k}, L_{kk}^{-1}]^T \ . \tag{2.88}$$

The special structure resulting from the combination of the Kronecker product and the elimination matrix helps solve this equation efficiently without resorting to non-linear solvers. To see this, define $\mathbf{e}_i = [1/L_{ii}, \mathbf{0}_{1\times(k-i)}]^T$, $\mathbf{G}_i = \mathbf{\Lambda}_{(i:k),(i:k)}$, and $\mathbf{h}_i = [L_{i,i}, L_{i+1,i}, \cdots, L_{k,i}]^T$, so that the system of non-linear equations can be re-written as:

$$\begin{bmatrix} \mathbf{G}_1 & & & \\ & \mathbf{G}_2 & & \\ & & \ddots & \\ & & & \mathbf{G}_k \end{bmatrix} \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \vdots \\ \mathbf{h}_k \end{bmatrix} = \begin{bmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \\ \vdots \\ \mathbf{e}_k \end{bmatrix} , \tag{2.89}$$

which is a decomposition into k independent systems of equations. The structure of the problem is also shown in Fig. 2.2. For the ith sub-system $\mathbf{G}_i\mathbf{h}_i = \mathbf{e}_i$:

$$\mathbf{\Lambda}_{(i:k),(i:k)}\mathbf{L}_{(i:k),i} = \begin{bmatrix} 1/L_{ii} \\ \mathbf{0}_{(k-i)\times 1} \end{bmatrix} . \tag{2.90}$$

Denote the solution of this problem by $\mathbf{L}^{\mathrm{nonlin}}_{(i:k),i}$. Now, if $1/L_{ii}$ on the right hand side is replaced with 1, the result is a linear system in the elements of $\mathbf{L}_{i:k,i}$. Denote the solution of this linear system by $\mathbf{L}^{\mathrm{lin}}_{i:k,i}$. In a Gaussian elimination procedure, by eliminating the variables bottom-up, the final equation is of the form $cL^{\mathrm{lin}}_{ii} = 1$ for some constant c which is independent of the

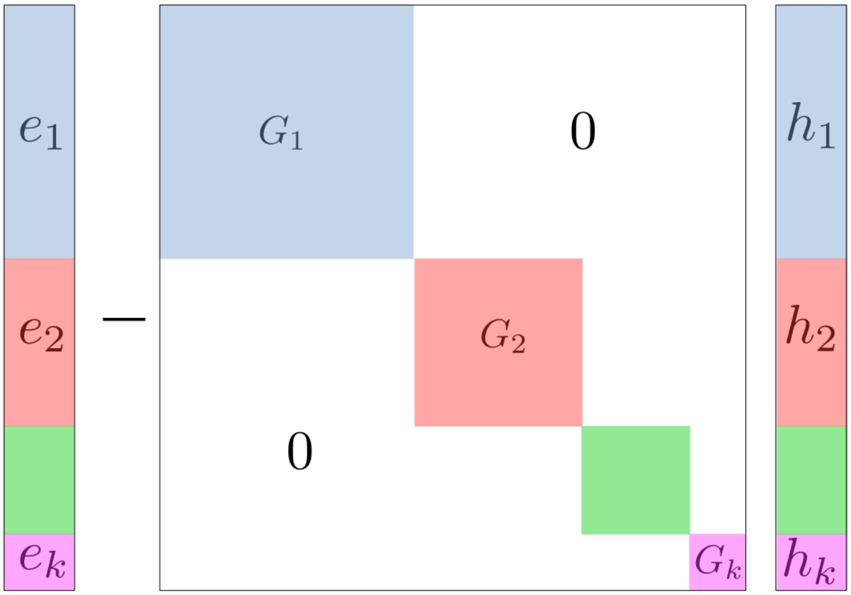


Figure 2.2: structure of the nonlinear equation $f(\mathbf{L}) = 0$

right hand side of the equation. Consequently, $L_{ii}^{\text{lin}} = 1/c$. Repeating the same procedure for the nonlinear system, the final equation is $cL_{ii}^{\text{nonlin}} = 1/L_{ii}^{\text{nonlin}}$, which leads to $L_{ii}^{\text{nonlin}} = 1/\sqrt{c}$. Therefore, the nonlinear solution is related to the linear solution by $L_{ii}^{\text{nonlin}} = \sqrt{L_{ii}^{\text{lin}}}$. The rest of the variables are scaled by the same factor $L_{ii}^{\text{nonlin}}/L_{ii}^{\text{lin}} = 1/\sqrt{L_{ii}^{\text{lin}}}$. Therefore:

$$\mathbf{L}_{i:k,i}^{\text{nonlin}} = \mathbf{L}_{i:k,i}^{\text{lin}} \times \frac{1}{\sqrt{L_{ii}^{\text{lin}}}} \ . \tag{2.91}$$

Finally, this result can be applied to update $\mathbf{L}_{\mathbf{u}}$ and $\mathbf{L}_{\mathbf{w}}$ respectively by using:

$$\begin{aligned}
\mathbf{\Lambda}_{\mathbf{w}} &= \langle \mathbf{f}_{j,t} \rangle - \langle \mathbf{g}_{j,t} \rangle \mathbf{A}^{(j)^T} - \mathbf{A}^{(j)} \langle \mathbf{g}_{j,t} \rangle^T + \mathbf{A}^{(j)} \langle \mathbf{F}_{j,t-1} \rangle \mathbf{A}^{(j)^T} \\
\mathbf{\Lambda}_{\mathbf{u}} &= \left\langle \mathbf{y}_t \mathbf{y}_t^T \right\rangle - \left\langle \mathbf{y}_t \mathbf{m}_{j,t}^T \right\rangle \mathbf{C}^T - \mathbf{C} \left\langle \mathbf{y}_t \mathbf{m}_{j,t}^T \right\rangle^T + \mathbf{C} \langle \mathbf{f}_{j,t} \rangle \mathbf{C}^T
\end{aligned} \tag{2.92}$$

which is the same as Eqs. (2.74) and (2.78) for unconstrained covariance updates.

2.9 Hyperparameter Selection

The proper value of the hyperparameters λ and D are selected by k-fold cross validation. Specifically, $k = 10$ is used for the evaluations. The dataset is partitioned into k non-overlapping subsets. For any given candidate value of D and λ, the estimation algorithm is repeated k times, each time using $k - 1$ out of the k data subsets as the training set for parameter estimation, and the remaining one data subset as the test set. The validation cost is the mean one-step prediction error evaluated over the test set, averaged over all k possible data partitions. The hyperparameters resulting in the smallest validation cost are then selected. The candidate values for the hyperparameters may be obtained from a grid over $\log \lambda$ and D. More efficient approaches such as Bayesian Hyperparameter Optimization [37] may be used to obtain the candidate hyperparameters. An example of hyperparameter selection is provided in Section 3.1.

2.10 Computational Complexity

The EM algorithm for the estimation of the system parameters is summarized in Algorithm 1. The connectivity matrix is initialized with small values ($\ll 1$) drawn from a zero-mean normal distributions with a variance of 10^{-4}. Moreover, the covariance matrices are initialized as identity matrices. Although not presented here, our simulations show that the initialization does not affect the convergence of the algorithm.

If implemented naïvely, each step of the Kalman filter results in a complexity of order $O((MD)^3 + (MD)N^2 + (MD)^2N + N^3)$ which is equivalent to $O((MD)^3 + N^3)$. However, the special structure of $\mathbf{A}_e$ in Eq. (2.47) may be exploited to reduce the complexity to $O(M^3D^2 + N^3)$. If the Kalman filter is implemented for all time samples, then the overall complexity of the Kalman filter is given by $O(T(M^3D^2 + N^3))$. As will be discussed shortly,

Algorithm 1 EM algorithm for sparse connectivity estimation

```
1: Input: $\{\mathbf{y}_1 \cdots \mathbf{y}_T\}$, initial guess of parameters ($\mathbf{A}$, $\boldsymbol{\Sigma}_\mathbf{u}$, $\boldsymbol{\Sigma}_\mathbf{w}$), hyper-parameter $\lambda$ and $D$
2: Apply KF to find $\mathbf{M}_t, \mathbf{P}_t, \mathbf{R}_t$ (Eq. (2.51))
3: Calculate all $\langle . \rangle$ in Eq. (2.92)
4: repeat
5:    update $\boldsymbol{\Sigma}_\mathbf{u}$ using Eqs. (2.90,2.91,2.92), run KF, update all $\langle . \rangle$ in Eq. (2.92)
6:    update $\boldsymbol{\Sigma}_\mathbf{w}$ using Eqs. (2.90,2.91,2.92), run KF, update all $\langle . \rangle$ in Eq. (2.92)
7:    repeat
8:       select the set of rows $\phi$ and set of columns $\psi$ of $\mathbf{A}$ to be updated
9:       update $\mathbf{A}$ using Eq. (2.71), run KF, update all $\langle . \rangle$ in Eq. (2.64)
10:   until convergence of $\mathbf{A}$
11: until convergence of $\mathbf{A}$, $\boldsymbol{\Sigma}_\mathbf{u}$, $\boldsymbol{\Sigma}_\mathbf{w}$
```

the algorithm does not need to apply the Kalman filter over the entire dataset, leading to the elimination of factor T.

The time complexity of updating the noise covariance matrices is found by analyzing Eqs. (2.90), (2.91), and (2.92). The calculation of $\boldsymbol{\Lambda}_\mathbf{w}$ and $\boldsymbol{\Lambda}_\mathbf{u}$ in Eq. (2.92) imposes a complexity of $O(M^3D^2)$ and $O(NM^2 + N^2M)$ respectively. Furthermore, an extra $O(M^4)$ and $O(N^4)$ is added for solving the set of equations in Eq. (2.89) for $\boldsymbol{\Lambda}_\mathbf{w}$ and $\boldsymbol{\Lambda}_\mathbf{u}$ respectively. Therefore, the overall complexity of updating the estimate of $\boldsymbol{\Lambda}_\mathbf{w}$ and $\boldsymbol{\Lambda}_\mathbf{u}$ is given by $O(M^4 + M^3D^2)$ and $O(N^4 + NM^2 + N^2M)$. The complexity of updating the connectivity matrix is similarly calculated; an analysis of Eq. (2.71) reveals a complexity of $O(M^3D)$ for Line 9 of Algorithm 1.

The Kalman filtering is performed after updating each of the parameters $\boldsymbol{\Sigma}_\mathbf{u}, \boldsymbol{\Sigma}_\mathbf{w}, \mathbf{A}$ to calculate the temporal averages ($\langle . \rangle$). Profiling Algorithm 1 reveals that the repetitive execution of the Kalman filter and the iterative nature of coordinate descent for the update of $\mathbf{A}$ are the main computational bottlenecks. The complexity, however, can be reduced considerably under certain conditions and approximations described below. The results reported in Section 3.3 use an efficient implementation of the algorithm that includes all the following performance guidelines.

Fewer KF runs

If we assume that the shape of the Q function in Eq. (2.36) (specifically, the position of the optimum point in the parameter space) does not change significantly after a single parameter update (Lines 5, 6, and 9 of Algorithm 1), it is possible to run the KF less frequently. For example, all the KF steps in Algorithm 1 may be replaced by one KF step after Line 10, when all parameters have been updated once. This assumption is realistic because the parameter update equations use temporal averages of the KF outputs and the temporal averaging smooths out the variations in the time-averaged KF outputs caused by small changes to the parameters.

Generalized EM

Using Generalized EM [36, 80], the iteration-within-iteration bottleneck introduced by the update of $\mathbf{A}$ can be avoided. A single (or a few) step(s) of the $\mathbf{A}$ update can replace the coordinate descent loop (Lines 7 through 10 of Algorithm 1). This strategy works because a single update of $\mathbf{A}$ increases the objective function Q even if it does not maximize it.

Asymptotic Approximations

The update equations use temporal averages rather than the direct KF outputs. Since the temporal averages converge after a large enough number of time steps, the steady-state behavior of the KF can simplify the update of the temporal averages and possibly eliminate the need to run the KF. One approach is to find analytic expressions for the temporal averages and thus avoid the KF altogether. Another approach is to monitor the change in temporal averages and stop the KF as soon as convergence is reached. We use the latter approach in the examples presented next.

Stochastic Gradient Descent

The MAP algorithm builds on the joint likelihood function in Eq. (2.38) which takes all the observations into account. This form of likelihood eventually leads to the repetitive application of the Kalman filter on the same interval of the dataset. Alternatively, the cost function may be approximated by the cost of a single example as is done in Stochastic Gradient Descent (SGD) [13]. In particular, a single step of the Kalman filter is executed upon the arrival of a new data point. The output of the Kalman filter then substitutes the temporal averages in Algorithm 1 and the parameters are updated accordingly. In addition to the significant reduction in complexity, the SGD scheme allows for on-line estimation of time-varying parameters. Finally, momentum and averaging techniques may be applied to improve the performance of SGD [91].

To provide realistic estimates of the run-time, the algorithm was implemented in MATLAB and applied to a dataset with 10000 time samples for different problem dimensions on an Intel Core i-5-7200U CPU (2.5 GHz × 4) with 8 gigabytes of RAM. The results are summarized in Table 2.1.

$N = 10$			
	$D = 10$	$D = 30$	$D = 90$
$M = 10$	.9	1.8	16.5
$M = 30$	2.2	17	152
$M = 90$	22.5	203.5	
$N = 30$			
	$D = 10$	$D = 30$	$D = 90$
$M = 10$	1.3	2.5	16
$M = 30$	2.8	17	149.2
$M = 90$	22	188	
$N = 90$			
	$D = 10$	$D = 30$	$D = 90$
$M = 10$	1.7	2.9	16.8
$M = 30$	3.2	18.1	154
$M = 90$	24.1	204.4	

Table 2.1: Execution time per time sample of input dataset in miliseconds. The algorithm is implemented in MATLAB and executed on an Intel Core i-5-7200U CPU (2.5 GHz × 4) with 8 gigabytes of RAM. The $M = D = 90$ requires more system memory than is available on the target hardware.

Chapter 3

Evaluation of the Sparse Linear MVAR Parameter Estimation Algorithm

3.1 Demonstration

For the initial simulation study, a synthetic system consisting of 20 nodes ($M = 20$) and 20 dimensional observations ($N = 20$) with $D = 20$ is considered. The evaluation is extended to other problem dimensions in Section 3.3. For the first set of results, the 20×400 connectivity matrix is assumed to have 40 non-zero values selected randomly, and stability is imposed by shrinking the connectivity coefficients until the conditions of the Gershgorin circle theorem are met [8]. The covariance matrices $\mathbf{\Sigma_u}$ and $\mathbf{\Sigma_w}$ and the measurement matrix $\mathbf{C}$ are generated randomly. The actual values of $\mathbf{\Sigma_u}$ and $\mathbf{\Sigma_w}$ were observed to have negligible influence on the estimation of the connectivity matrix; both the estimated covariance and the ground truth covariance were used in the connectivity estimation and the results did not show any

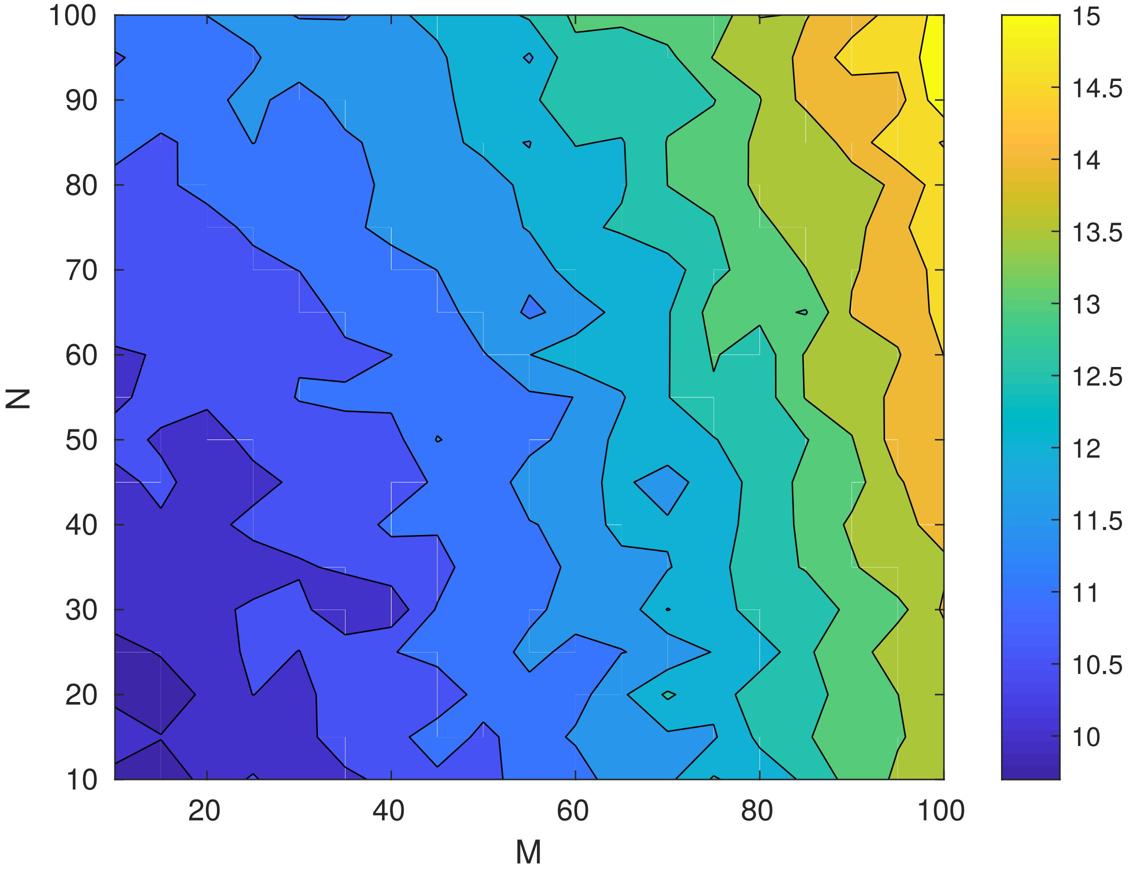


Figure 3.1: Colormap of the relative error (in percentage) between the estimated and the ground-truth covariance matrices for different problem dimensions. For each problem dimensions, the estimation is repeated for 100 different ground-truth covariances and the results are averaged. The estimation performance slightly degrades with the problem dimension and is more sensitive to M.

significant sensitivity. Nevertheless, a summary of the covariance estimate performance is provided in Fig. 3.1 for different problem dimensions.

The proper value of the hyperparameters λ and D are calculated per the cross–validation procedure discussed in Section 2.9. The search space is logarithmic in λ and linear in D. In order to reduce the computational complexity of a brute-force grid search, a Bayesian Hyperparameter Optimization approach [37] is used to choose candidate values. Fig. 3.2 shows the validation cost in the $\lambda - D$ plane. It is observed that the true value of D is almost at the center of the lowest cost contour; thus, the validation cost surface is capable of estimating D with a good accuracy.

Fig. 3.3 compares the estimated and true connectivity coefficients at different values of λ for an instance of ground truth $\mathbf{\Sigma_u}$, $\mathbf{\Sigma_w}$, $\mathbf{C}$, and $\mathbf{A}$. As expected, by increasing the strength of the regularization (larger λ), fewer non-zero connections emerge. The metrics used for evaluating the algorithm performance are described below.

3.2 Evaluation Metrics

Intuitively, the quality of the estimate is high if a truly strong (weak) connection is estimated as a strong (weak) connection. The simplest embodiment of this quality measure is to classify the connections as either *strong* or *weak*. Defining the *strength threshold* $\mathcal{T}$ as a fraction of the maximum estimated strength, all connection strengths larger (smaller) than the threshold are said to be strong (weak) connections, which leads to a binary classification. Based on this definition, classical measures such as *true positive rate* (TPR), *positive prediction value* (PPV), *true negative rate* (TNR), *negative prediction value* (NPV), and the area under the ROC curve can be used to quantitatively evaluate the performance of the algorithm. In a general classification problem, let N_{pn} denote the number of *p*ositive samples that are classified as *n*egative. If N_{pp}, N_{np}, and N_{nn} are defined similarly, the four detection performance measures are given by:

$$TPR = \frac{N_{pp}}{N_{pp} + N_{pn}} \tag{3.1}$$

$$PPV = \frac{N_{pp}}{N_{pp} + N_{np}} \tag{3.2}$$

$$TNR = \frac{N_{nn}}{N_{nn} + N_{np}} \tag{3.3}$$

$$NPV = \frac{N_{nn}}{N_{nn} + N_{pn}} \; . \tag{3.4}$$

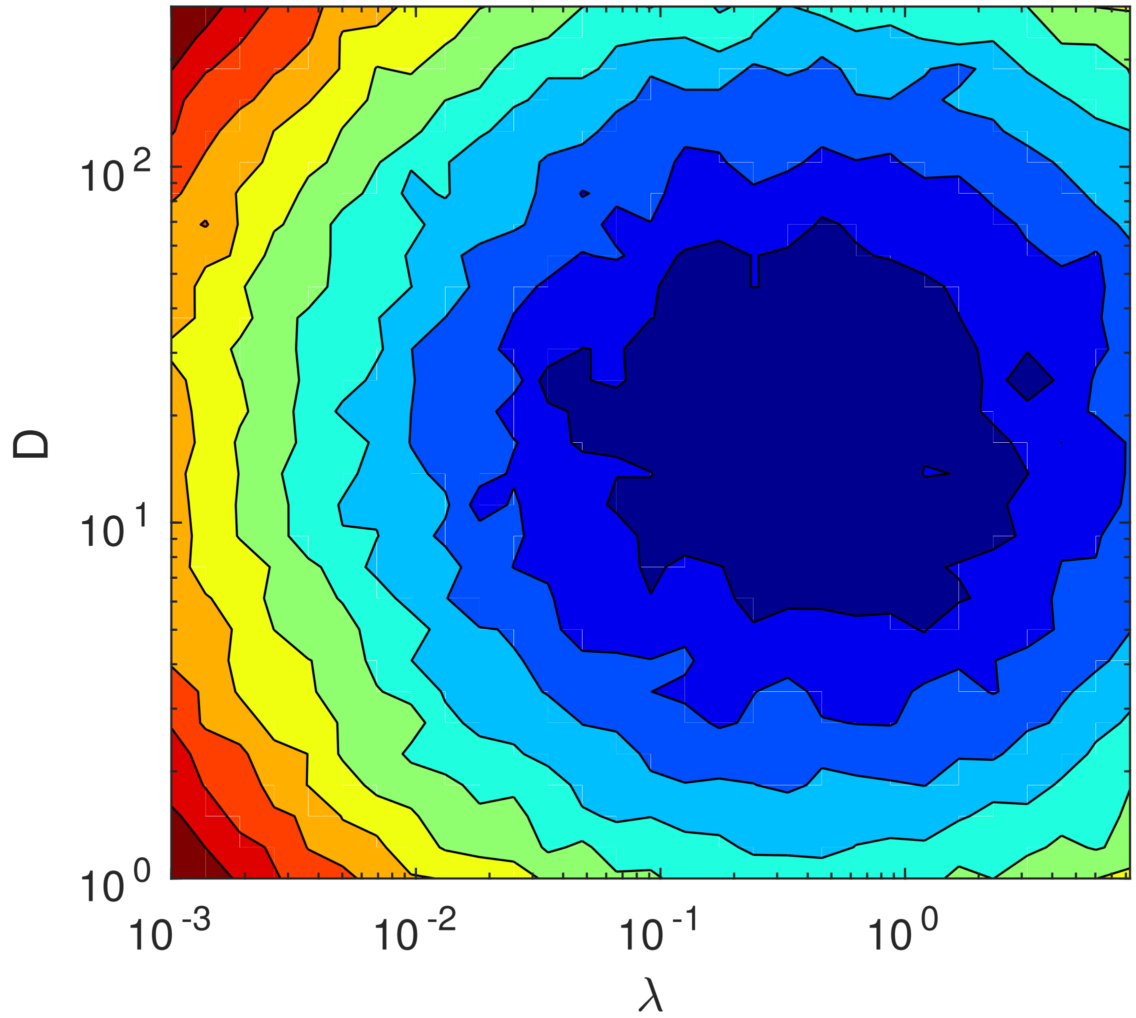


Figure 3.2: Colormap of the validation cost for the example discussed in Section 3.1. Blue represents low cost and red represents high cost. Although the D domain search is linear, the results are shown in log domain for a better visual representation.

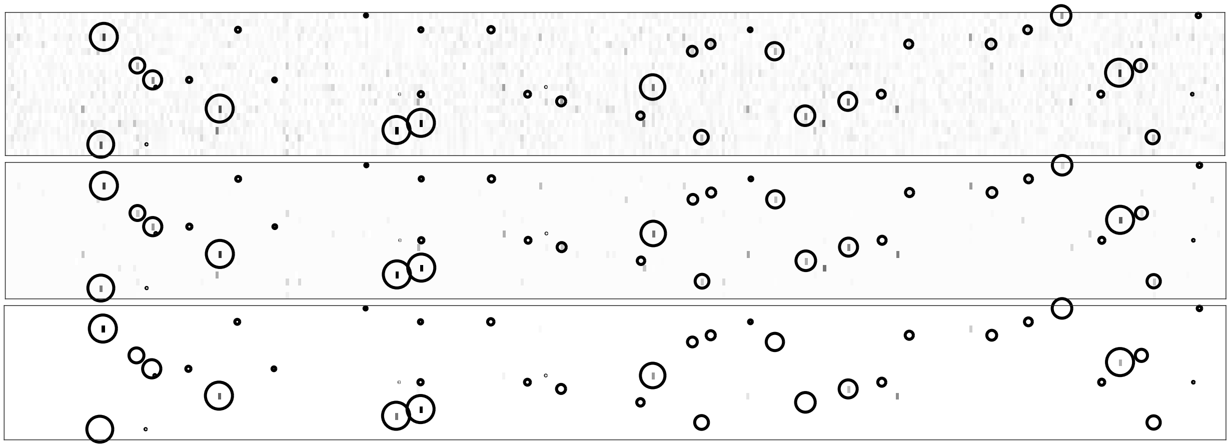


Figure 3.3: Estimated connectivity coefficients and actual connectivity coefficients for different regularization parameters ($\lambda = 0, .6, 2.2$ from top to bottom). The black circles show the location of actual connections and their radius is proportional to the strength of the actual connection at that location. The gray-map represents the strength of the estimated connectivity, with white and black representing the minimum and maximum values, respectively.

As an example, consider measuring the accuracy of the algorithm in detecting all the connections stronger than $.9A_{\max}$, where $A_{\max}$ is the value of the strongest connection. To do this, we set $\mathcal{T} = .9$ and calculate TPR, PPV, TNR, and PPV from the estimation output.

Fig. 3.4a illustrates the PPV against the regularization parameter λ for different detection thresholds and $M = D = N = 20$. The results are averaged over 1000 different system realizations (connectivity patterns and interference statistics), but all with the same number of non-zero connections. While similar curves may be obtained for TPR, TNR, and NPV, some of these curves do not provide much insight. For instance, NPV and TNR are very close to 1 and do not carry much information about the performance, a phenomenon primarily due to the sparseness of the connectivity matrix. Because of the trade-off between TPR and PPV, it is more meaningful to focus on the performance of the algorithm in terms of an acceptable TPR-PPV pair or the area under the TPR-PPV curve. Fig. 3.4b shows a plot of PPV vs. TPR. As expected intuitively, a high PPV is not achievable without sacrificing TPR. However, if a smaller subset of the strongest connections is of interest (a threshold of 0.9 compared to 0.5), a higher TPR is achievable for any given PPV and vice versa. For

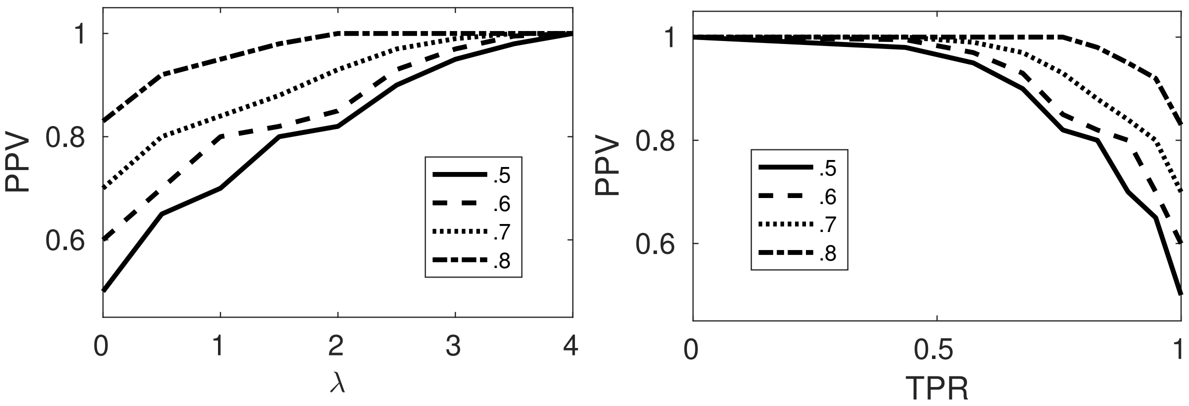


Figure 3.4: a) (LHS) detection performance vs regularization parameter for different detection thresholds. b) (RHS) Positive prediction value vs. true positive rate for different detection thresholds.

instance, it is observed that for a detection threshold of 0.8, both a TPR and PPV larger than 90% is achievable.

3.3 Comparison with Previous Work

The following methods have been selected from the literature for comparison with the algorithm proposed in this article. In most cases, the algorithms have been modified/augmented to be as consistent as possible with our system model:

Granger causality for multiple time series

In [74], node activities (as well as measurement noise statistics) are first estimated from the observations by solving the inverse problem. A similar multivariate autoregressive model is assumed to govern the interaction of node activities. The weights of the linear combination, $\{\mathbf{A}_1, \cdots, \mathbf{A}_D\}$ (together with the process noise statistics) are estimated from the known activity estimates by minimizing the resulting one-step prediction error. Since the weights denote the causal connections, all weights stronger than the threshold $\mathcal{T}$ are chosen as strong

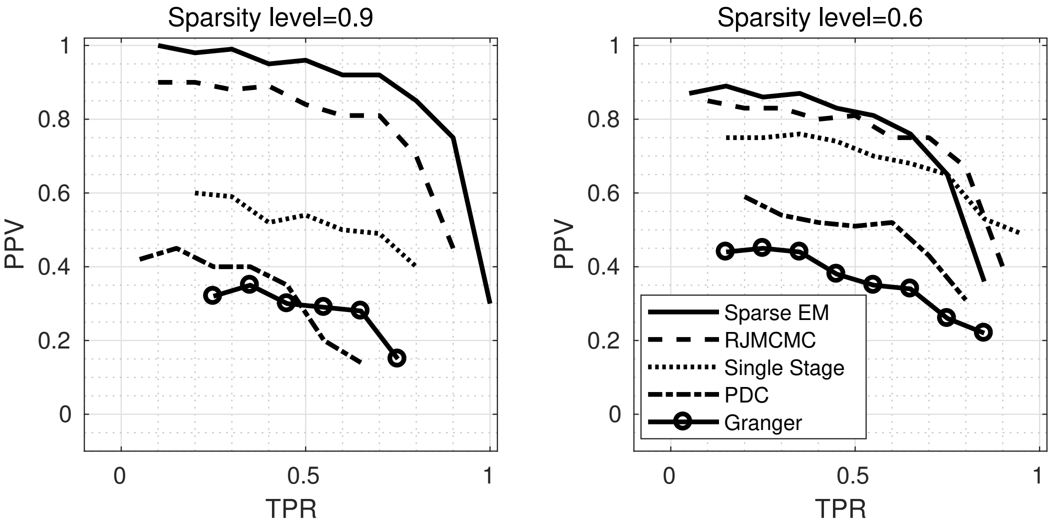


Figure 3.5: Comparison of our work (Sparse EM) with previous work (RJMCMC [21], Single Stage [22], PDC [6], Granger [74]) at two different levels of sparsity. The curves demonstrate achievable PPV-TPR pairs as $\mathcal{T}$ varies. Problem dimensions are $M = D = N = 20$.

connections. Since there exists no regularization parameter here, the TPR and PPV depend only on $\mathcal{T}$. The TPR-PPV curve is then calculated by eliminating $\mathcal{T}$ between TPR($\mathcal{T}$) and PPV($\mathcal{T}$).

Partial Directed Coherence

Baccala et al. [6] introduced the concept of PDC as a frequency domain implementation of Granger causality. Once the AR coefficients $\mathbf{A}_1, \cdots, \mathbf{A}_D$ have been estimated by minimizing the one step prediction error, a frequency transform $\mathbf{A}(\omega) = \sum_{d=1}^{D} \mathbf{A}_d e^{-j\omega d}$ is calculated. The PDC measure between any pair of nodes is then calculated from $\mathbf{A}(\omega)$, followed by an averaging over ω. If the PDC measure $\kappa_{i,j}$ between nodes i and j is greater than $\mathcal{T}$ (i.e. the connection is strong enough), the corresponding delay is estimated as the location of the peak of $[(\mathbf{A}_1)_{ij}, \cdots, (\mathbf{A}_D)_{ij}]$.

Conditional Granger and Single-Stage Non-Sparse Connectivity Estimation

Cheung et al. [22] avoid a two-stage method by jointly estimating the activity and state transition matrix using EM. However, the objective function does not incorporate connection sparsity. The conditional Granger causality metric [48] is applied to the estimation result to determine the connectivity. Connections stronger than $\mathcal{T}$ are considered for calculating the TPR-PPV performance metric.

RJMCMC-aided Minimum-Variance Estimate From Parameter Posterior

The authors of [21] assume that the node activities are available. Therefore, their algorithm should be preceded by estimating the node activities by solving the corresponding inverse problem. With the node activities available, the (scaled) parameter posterior $p(\mathbf{A}|Y)$ is calculated using a Laplacian prior on the connectivity elements to impose sparsity. Then, the connectivity is estimated using $E_{p(\mathbf{A})}\{p(\mathbf{A}|Y)\}$. Since the integrand of the expectation is highly complex in the parameters, RJ-MCMC is used to perform numerical integration and calculate the estimate. Note that unlike [21], the simulation is repeated for different regularization parameters rather than considering a non-informative prior on λ and averaging using RJMCMC.

Even after the modifications, the residual inconsistencies preclude a direct algorithm comparison using curves similar to Fig. 3.4. First, it is not possible to add the regularization parameter to the framework of the first three methods [74, 6, 22]. Second, the algorithms in [74, 6, 22] are not designed to take sparsity into account. Thus, comparison with a method tailored to sparsity is not fair. We suggest the following remedies to resolve these issues; First, for our algorithm and for [21], the free variable λ is replaced with an *optimum* value found by 10-fold cross validation. Second, different ground truth sparsity is assumed, ranging from very sparse systems to non-sparse. For each given level of sparsity, 1000 different

	Sparse EM	RJMCMC [21]	Single Stage [22]	PDC [6]	Granger [74]
$\bar{s} = 0.99$	0.7960	0.6495	0.3150	0.2060	0.1465
$\bar{s} = 0.90$	0.6820	0.6365	0.3530	0.2400	0.1775
$\bar{s} = 0.80$	0.6275	0.6105	0.5510	0.2980	0.2530

Table 3.1: Comparison of our work (Sparse EM) with previous work (RJMCMC [21], Single Stage [22], PDC [6], Granger [74]). The ground truth sparsity level $\bar{s}$ is varied as .99, .9, and .8. The table data represent the area under ROCs in Fig. 3.5 .

ground truth connectivity patterns are considered. The performance of the algorithm is then averaged over these 1000 systems. As a result, TPR and PPV will be functions of $\mathcal{T}$ and the sparsity level $\bar{s}$, where $\bar{s} = 1 - s$ and s is the fraction of non-zero ground truth connections.

To plot TPR vs. PPV, an independent variable should be selected and then eliminated. Given the dependence of both PPV and TPR on a) sparsity level $\bar{s}$, b) strength threshold $\mathcal{T}$, and c) algorithm choice, it is possible to compare the performance of different algorithms at a fixed $\bar{s}$ by varying $\mathcal{T}$ as the independent variable. Fig. 3.5 uses this strategy to summarize the relative performance of the algorithms at different levels of sparsity. Note that different algorithms are not guaranteed to attain the same bounds on TPR and PPV as $\mathcal{T}$ varies. To facilitate the comparison of the algorithms under different levels of sparsity, Table 3.1 summarizes the TPR-PPV curves of Fig. 3.5 by calculating the area under the curves. Similar to the area under a *receiver operating curve* (ROC), a higher value generally indicates a superior overall performance.

The results in Fig. 3.5 and Table 3.1 reveal that the proposed algorithm achieves better TPR-PPV pairs compared to the previous work especially for high sparsity levels. The method of [21] performs better than the others as it takes the sparsity directly into account. The relatively superior performance of our algorithm compared to [21] can be attributed to the single-stage joint estimation utilized in our work compared with the two-stage method approach of [21]. It is also observed that the single-stage method of [22] performs better than other non-sparse methods that resort to two-stage estimation. Overall, at medium to high levels of sparsity, the superior performance of our algorithm is consistent with expectation since the other methods (except for [21]) are designed for connectivity patterns without any sparsity prior.

With a decreasing level of sparsity, it is observed that our algorithm and [21] slightly degrade. This is a consequence of the inconsistency between the assumption of a sparse system and the reality of a non-sparse system. Nevertheless, the performance degradation of the sparse methods with decreasing ground truth sparsity is still much less than that of the non-sparse methods with increasing sparsity since the sparse methods can partially adapt themselves to the ground truth level of sparsity. In other words, the cross-validation can automatically choose the correct value of λ that best fits the ground truth sparsity level. For a less sparse system, the cross validation chooses a smaller value of λ. Although this hyper-parameter selection slows down the performance degradation with decreasing sparsity, it cannot mitigate the degradation altogether.

The results presented thus far indicate superior performance of the proposed algorithm, henceforth referred to as Sparse EM, with a problem dimension of $M = N = D = 20$ under different levels of sparsity. However, more extensive simulations are required for other problem dimensions. To this end, we consider parameters pairs (M, D) in $\{20, 40, 60, 80\} \times \{10, 20, 30, 40\}$. For each (M, D), the value of N is set equal to M, since the value of N was observed to have negligible effect on the estimation performance as long as $N \geq M$.

Three different levels of sparsity ($\bar{s} \in \{.9, .8, .5\}$) were considered for the simulations. For each parameter tuple $(M, D, \bar{s})$ and for each algorithm, 100 different test systems were generated and the parameters were estimated. The ROC curves were calculated from the estimation results and averaged over the 100 test systems. Finally, the area under the averaged ROC curves was used as the performance metric.

Fig. 3.7 illustrates the simulation results. The colormaps represent the averaged area under the ROC curves. The sparsity level is fixed for each column and the two axes of each colormap represent the values of M and D. The vertical layers represent the proposed method, RJMCMC [21], Single Stage [22], PDC [6], and Granger [74] from top to bottom respectively. In order to improve the visual interpretability of the results, the original colormaps have been processed by histogram equalization [4] and the colorbars have been modified accordingly.

Regardless of the sparsity level and the algorithm choice, the estimation performance decays with increasing M and D. It is also observed that the performance gap between the Sparse EM and the other methods is more significant at higher levels of sparsity.

At a sparsity level of $\bar{s} = .9$, the Sparse EM and the RJMCMC methods perform noticeably better than the other three methods. Furthermore, the Sparse EM method outperforms the RJMCMC method by an approximate margin of .2. The results also reveal that the performance of Sparse EM is more robust to an increasing problem dimension. Notice that a sparsity of $\bar{s} = .9$ on the entire connectivity matrix (of dimension $M \times MD$) is equivalent to a lower level of sparsity in terms of the *node-to-node spatial connectivity*, which defines two nodes to be spatially connected if at least one of the delay lines between the nodes is active. Assuming that the locations of the non-zero elements of $\mathbf{A}$ are selected randomly and independently, the probability that two nodes are not connected may be approximated by $P_{\text{disconnect}} = \bar{s}^D$ which is always smaller than $\bar{s}$. As shown in Fig. 3.6, even at an apparently high sparsity level of $\bar{s} = .9$, the probability that two nodes are spatially disconnected is low. Therefore, the proposed Sparse EM algorithm can outperform the other methods not only under highly sparse systems, but also under relatively spatially dense systems. The performance gap, however, decreases as the system becomes more dense.

At lower sparsity levels of $\bar{s} = .8$ and .5, the performance of Single Stage [22], PDC [6], and Granger [74] improve, but still underperform the Sparse EM and the RJMCMC method. At $\bar{s} = .5$, RJMCMC performs on par with Sparse EM and can even outperform our method at low values of M. Still, the proposed method is more robust to increasing D and M compared

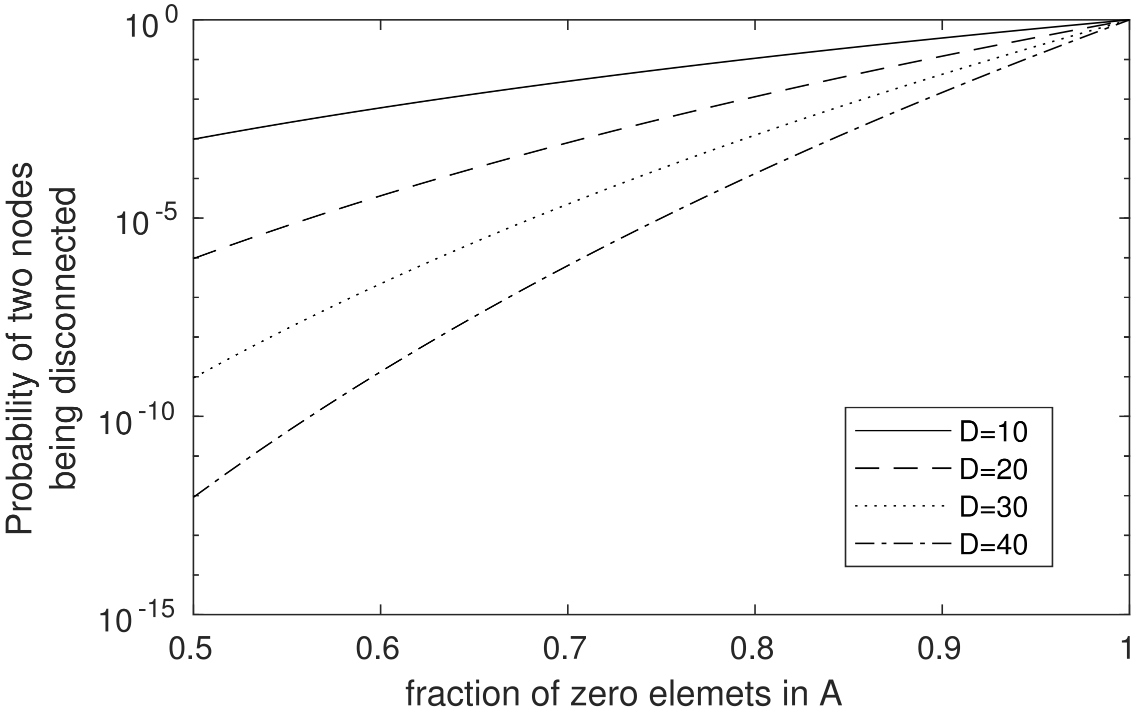


Figure 3.6: The relationship between $P_{\text{disconnect}}$ (y axis) and $\bar{s}$ (x axis).

to RJMCMC.

3.4 Application to Temperature Prediction

To evaluate the performance of the algorithm on real data, we consider the problem of identifying the predictive directional interactions between the daily averaged temperatures recorded at weather stations across the mainland U.S.[1]. This problem is one example where ground-truth is not available: without accurate knowledge of the complex climactic mechanisms that underlie the evolution of global temperatures, it is not possible to determine how the temperature at one location at one point in time influences the temperature value measured at another location and time sample. Therefore, no physical meaning should be associated with the inferred interactions. Rather, the interactions only provide a measure of

[1] The dataset is available at ftp://ftp.ncdc.noaa.gov/pub/data/gsod/.

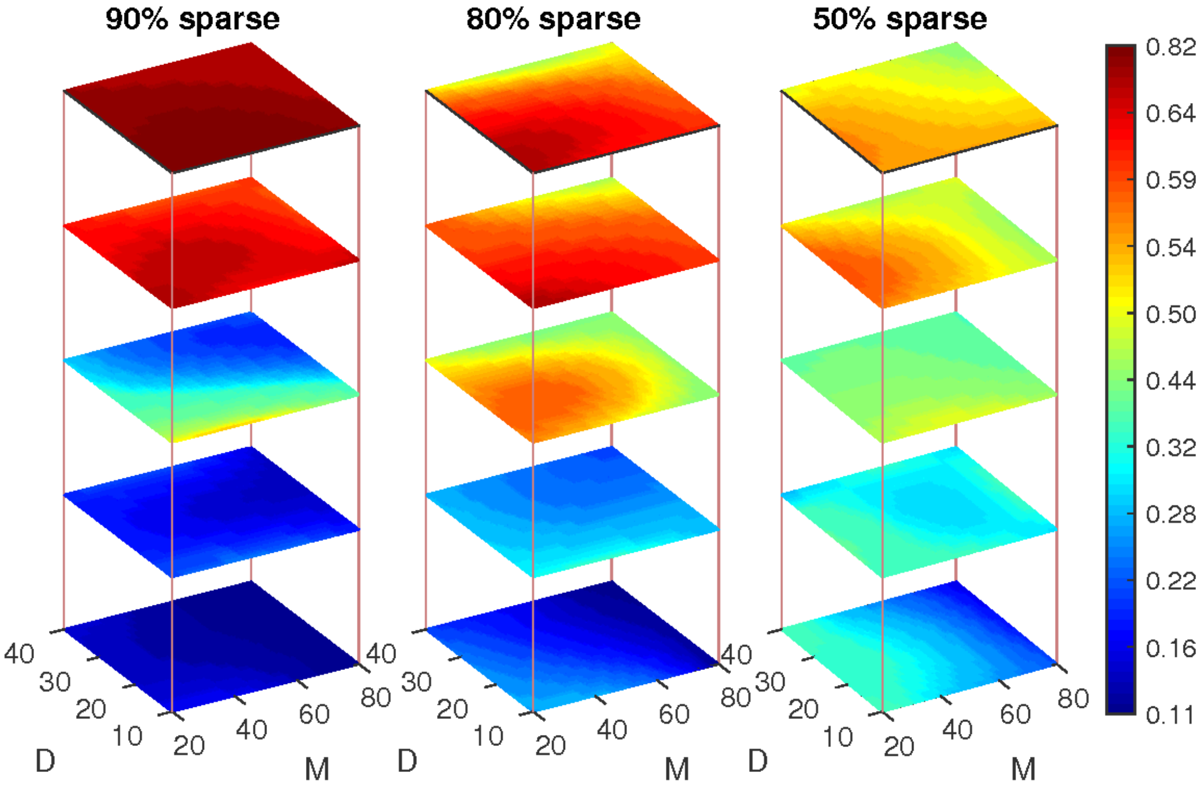


Figure 3.7: The area under the ROC curve for different algorithms, sparsity levels, and problem dimensions. Layer from top to bottom: The proposed method, RJMCMC [21], Single Stage [22], PDC [6], and Granger [74]

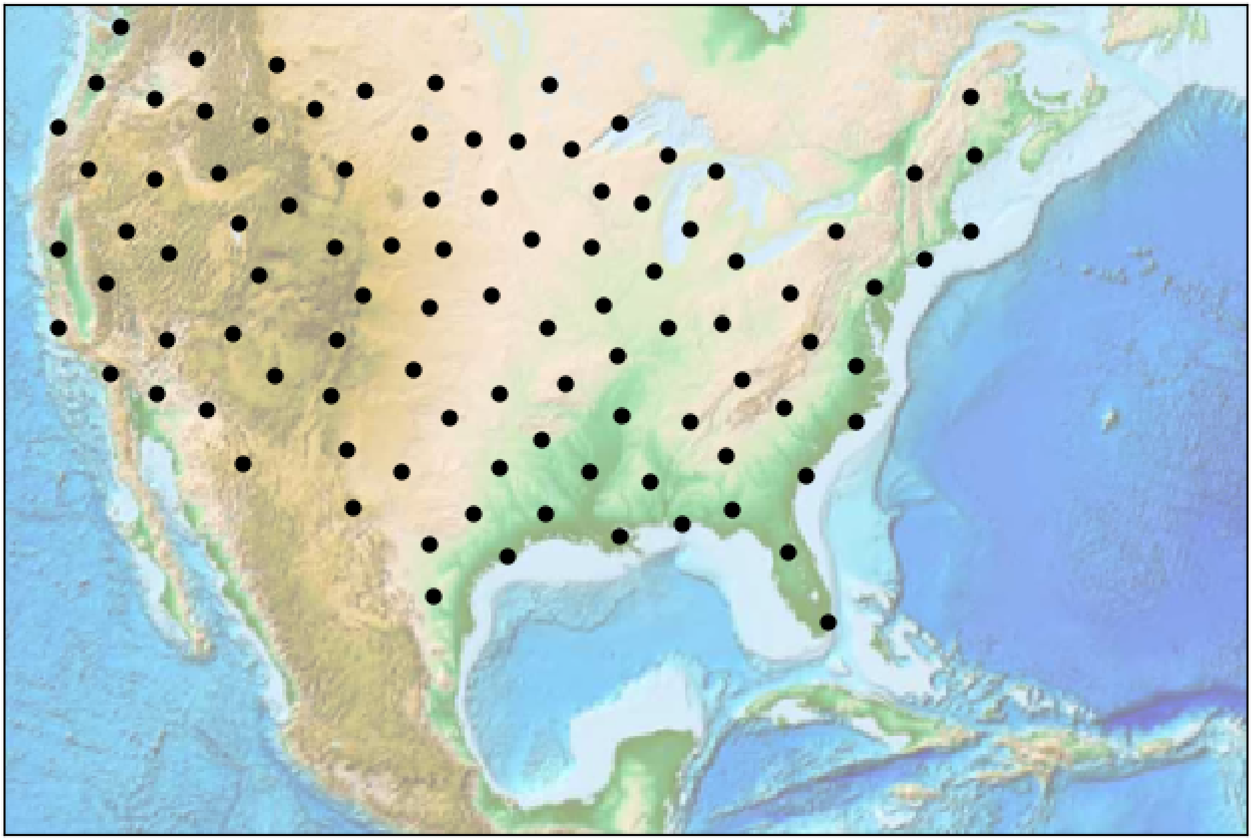


Figure 3.8: Locations of the weather stations included in the temperature causality analysis.

predictive power between the temperatures. Despite the absence of ground-truth, the estimation should be consistent with expected observations such as strong interactions between nearby points.

The data from 2016 and 2017 are preprocessed to include stations that are at least 150 miles apart and whose data covers the entire year. The stations included in the analysis are shown in Fig. 3.8. The time series are detrended at each station using polynomials of order 6 and then normalized to have zero mean and unit variance.

The time series of interest are recorded directly at each station. Therefore, the measurement matrix $\mathbf{C}$ is set to identity. As the temperature measurements are presumed to be accurate, the measurement noise $\mathbf{u}_k$ is assumed to have i.i.d components with a power equal to one tenth of the signal power. The cross-validation approach discussed previously is used to

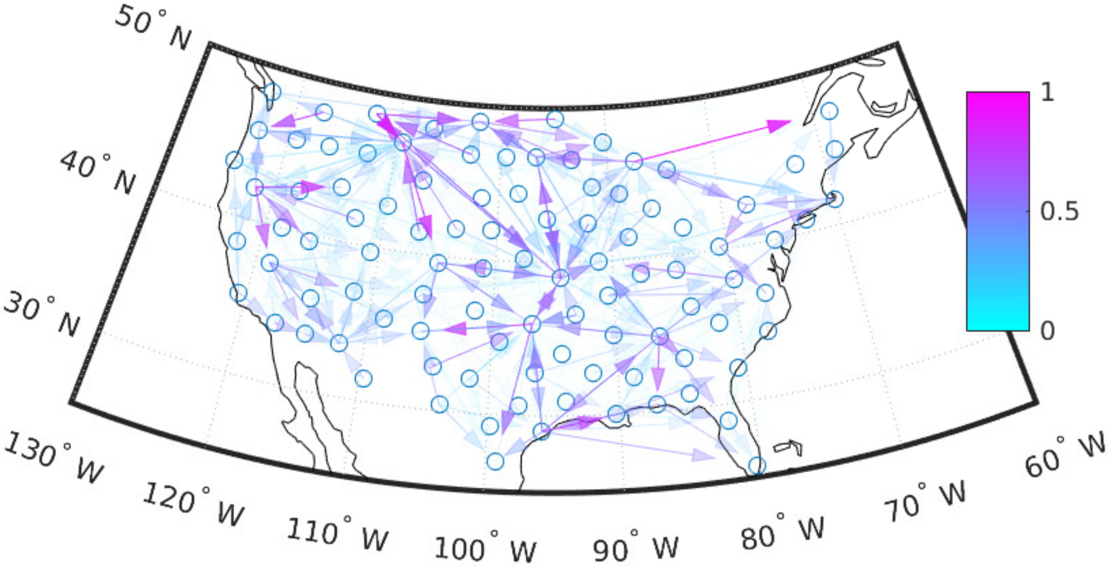


Figure 3.9: The predictive interaction among temperature data. The strength of interaction is encoded by color and transparency.

determine the optimal choice of D and λ based on the available data.

Figure 3.9 depicts the results of the connectivity estimation. According to Eq. (2.2), the influence of node i on node j may be expressed by the equation $v_j[k] = \sum_{\tau=1}^{D} a_{i,j}[\tau]v_i[k-\tau]$. The strength values used to produce Fig. 3.9 are calculated as $\sum_{k=1}^{D} |a_{i,j}[k]|$ to summarize the interaction over multiple time delays.

To gain insight beyond that provided by Fig. 3.9, we investigate the correlation between the predictive strength and the distance between the stations. Fig. 3.10 shows the correlation between the strengths of interactions and the distance between the stations. To bring out the pattern, the sparse reverse Cuthill-McKee ordering algorithm [23] is used to reorder the original distance matrix such that nearby stations are grouped closely together. The causal strength matrix is reordered similarly. As expected, all significant interactions only exist between nearby stations and no interaction is observed between distant stations.

Another approach to verify the results is to examine the predictive power of different stations and match it with expected climactic phenomena. Fig. 3.11 plots the predictive power of

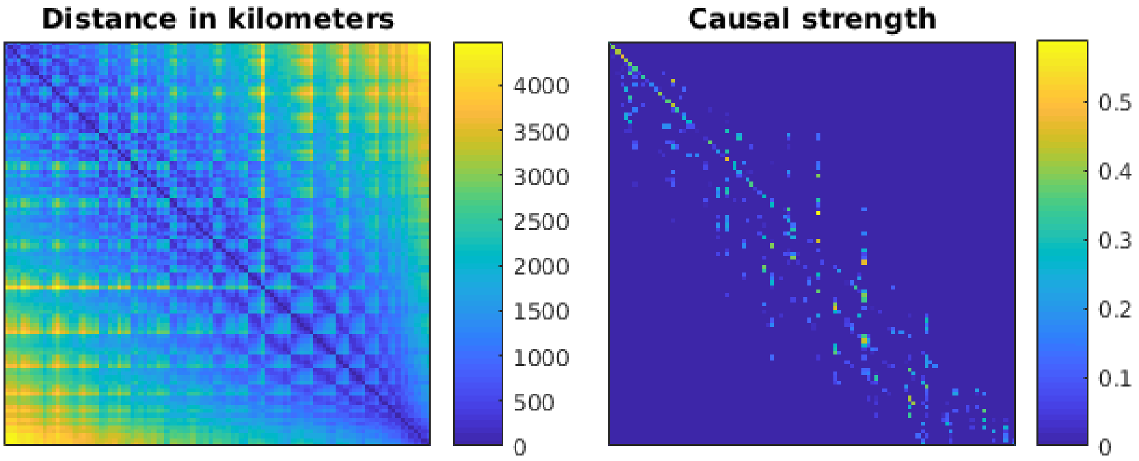


Figure 3.10: The correlation between the strength of interaction and station distance after applying sparse reverse Cuthill-McKee reordering [23] of the matrices to group together nearby stations.

the stations in the U.S., where the predictive power of a station i is defined as $\sum_{\tau,j} |a_{i,j}[\tau]|$. The results are consistent with those reported in [82], which showed that the locations along longitudes of 110 degrees west and 90 degrees west have the highest predictive power. Moreover, our method also detects relatively high predictive power for coastal stations located on the east and west cost as well as near the Gulf of Mexico. Although a physically meaningful association is not possible due to the use of a "physics–ignorant" model, the predictive power of the coastal stations coincides with the expected influence of the ocean on the mainland temperatures, which is consistent with the known influence of major ocean currents such as the California, Caribbean, Gulf stream, and north Atlantic currents on the mainland temperature. The ocean–land interaction mechanisms include, but are not limited to the ocean serving as a heat-retaining panel, distributing the temperature, generating surface winds, and producing rain.

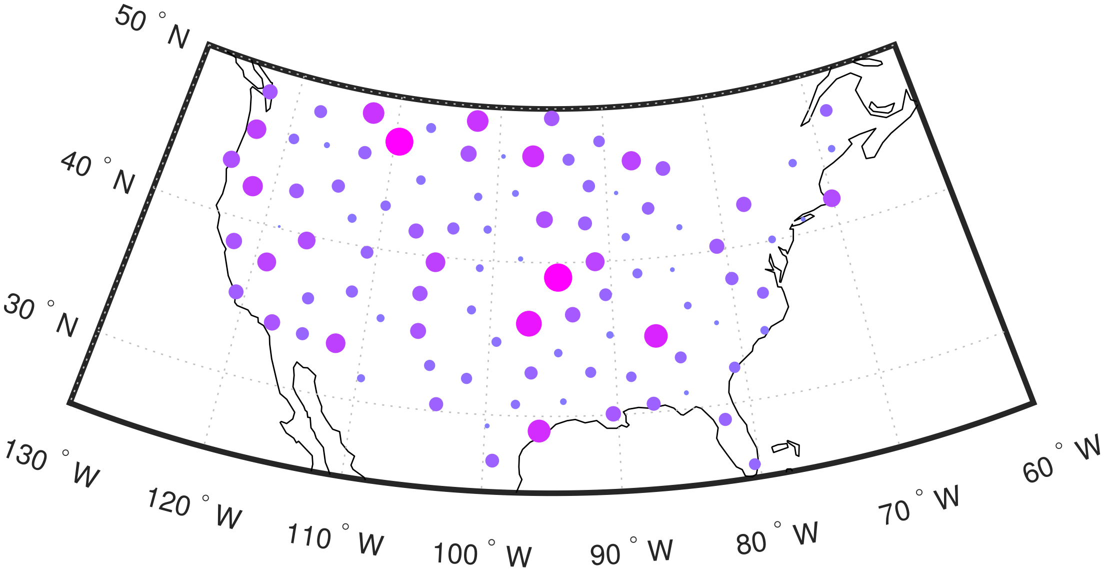


Figure 3.11: The predictive power of the stations. Larger circles encode higher predictive power.

3.5 Application to EEG Data

In this section, we investigate the challenges in applying the proposed estimation framework for identifiying the effective macroscale brain connectivities from EEG recordings.

3.5.1 Dataset Description

The Recordings and the Geometry

The EEG dataset consists of the electrode recordings and the geometry of the cortex, scalp, and electrodes. The EEG signals are recorded over 128 electrodes for 50 seconds with a sample rate of 1024 Hz. The recording are corrected for eye movement. The subject are in resting state, with eyes open for one dataset and eye closed for the other dataset.

The electrode locations are given as polar coordinates on a unit sphere. To map these to

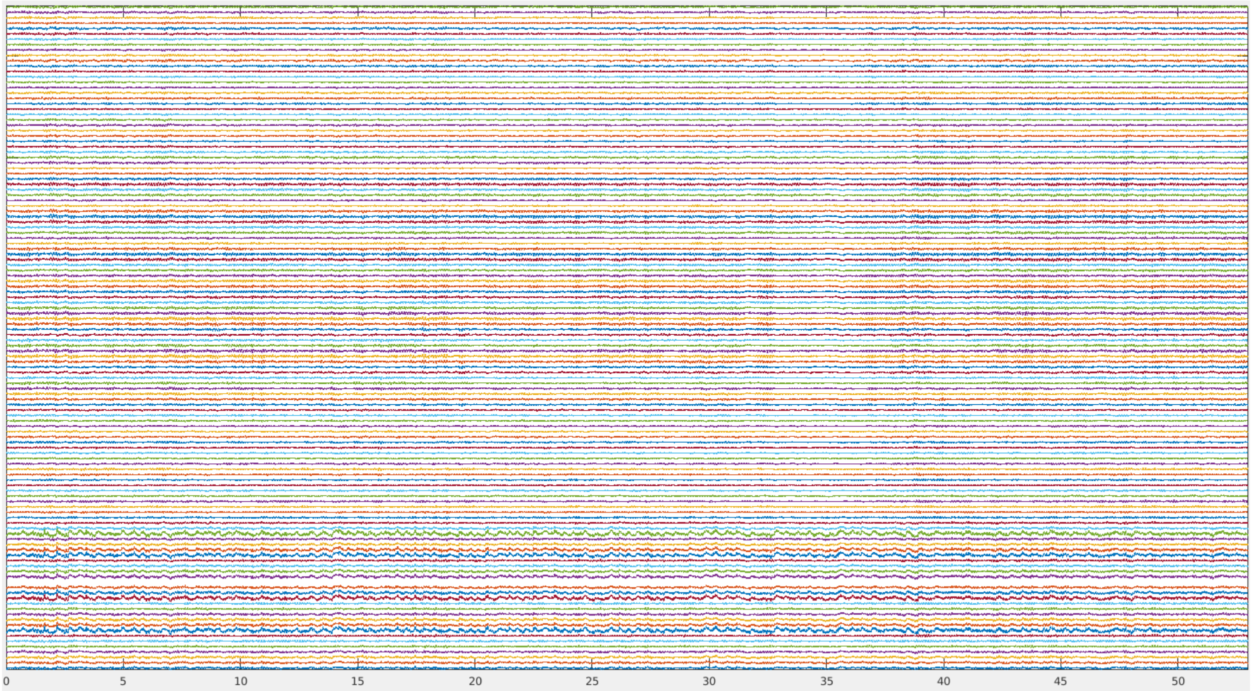


Figure 3.12: Processed EEG recordings for closed eye resting state

locations on the scalp, the original locations are extended radially until they coincide with the surface of the scalp. 18 of the electrodes are excluded from the analysis due to either lack of interpretation or low SNR. Fig. 3.12 shows the corrected EEG recordings.

The geometry of the brain and the scalp is encoded as a set of vertex coordinates and faces, where each face connects 3 vertices. Fig. 3.13 illustrates the geometry of the scalp and the cortex. The electrode locations are shown in Fig. 3.14.

Lead Field Matrix

For EEG analysis (e.g., localization of neuronal activity on the cortex), the problem of determining the measurement matrix, alse referred to as the gain matrix and the lead field matrix (LFM), is a well-studied problem, and numerous methods have been developed to address it. For example, MRI images can be used to reconstruct the geometry of the cortex and the scalp, and then finite boundary element modeling can be used to reconstruct the

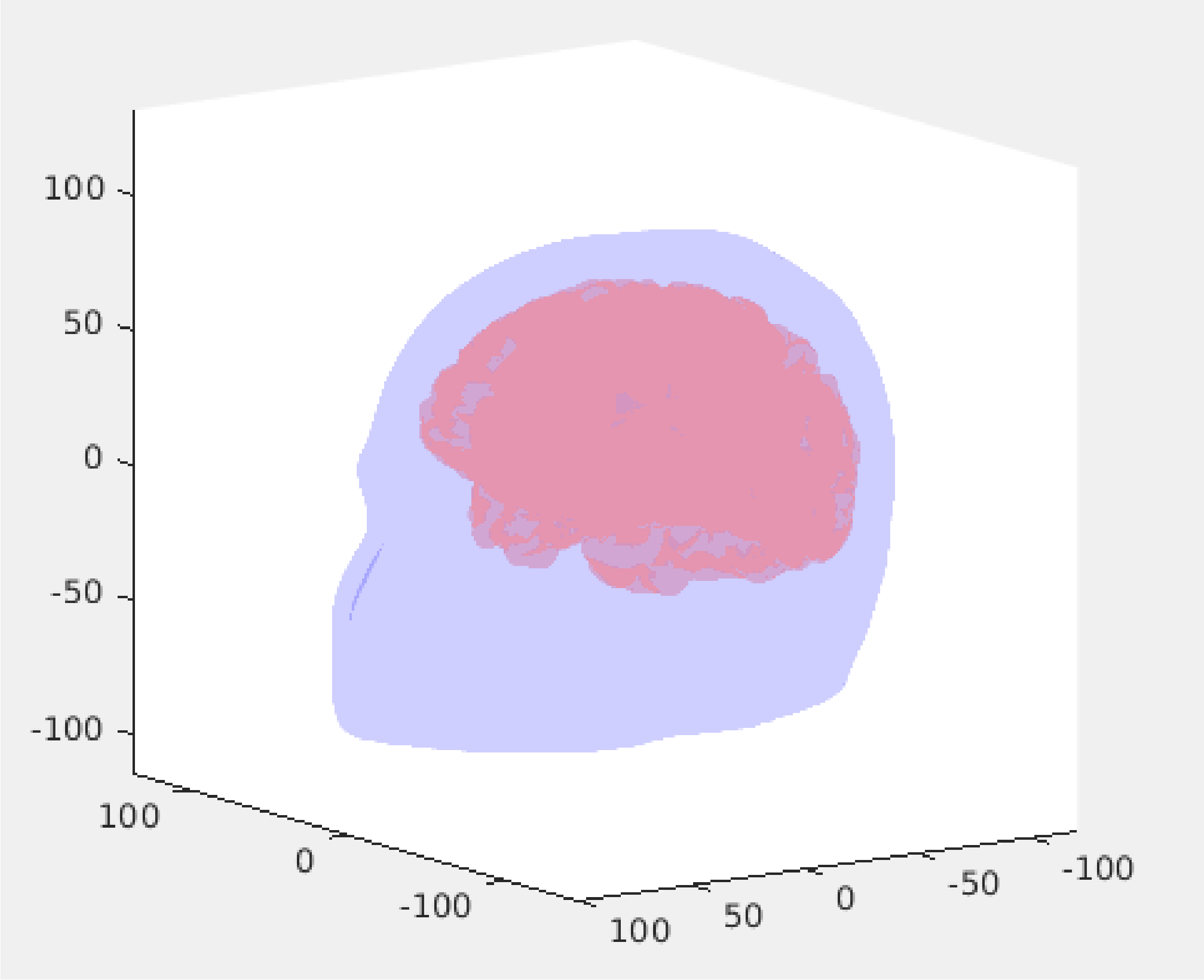


Figure 3.13: The geometry of the scalp and the cortex. The dimensions are in milimeters.

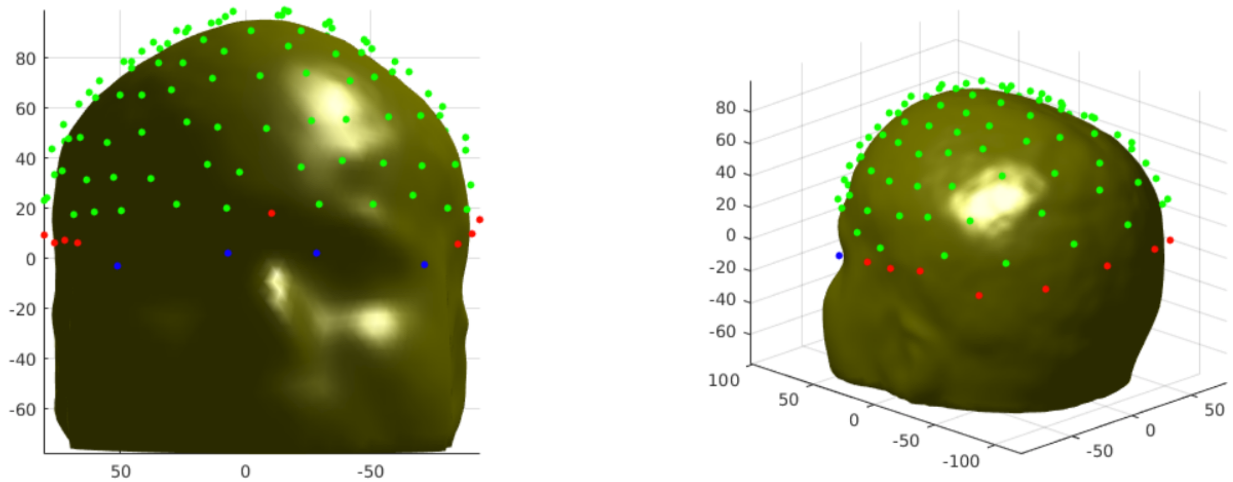


Figure 3.14: The location of electrodes w.r.t the scalp. The green electrodes are used in the analysis. Red and blue electrodes are excluded due to either low SNR or lack of interpretability.

electromagnetic propagation from the brain, through the skull, and to the electrodes on the scalp (see [2] for an example of this kind of approach).

3.5.2 Dataset Characteristics

Ill-Conditioned LFM

The gain matrix resulting from the three layer head model is fat due to the high number of patches required to capture the geometry of the cortex. Figs. 3.15 through 3.18 show samples of the strongest 20 modes of the LFM obtained by Singular Value Decompositon (SVD). These modes capture more than 99% of the variability in the LFM. Even after performing a cortex segmentation [28], the resulting region-to-electrode gain matrix is low rank.

The high condition number of the resulting LFM may be attributed to the geometrically wide impulse responses of individual cortex patches over the scalp, which is a result of the current-spreading caused by the three layer head model. In this case, it is true that the

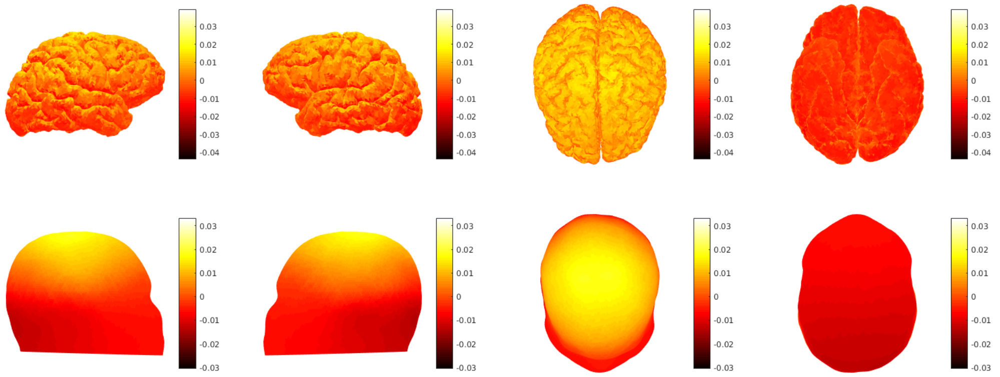


Figure 3.15: The strongest mode of the LFM

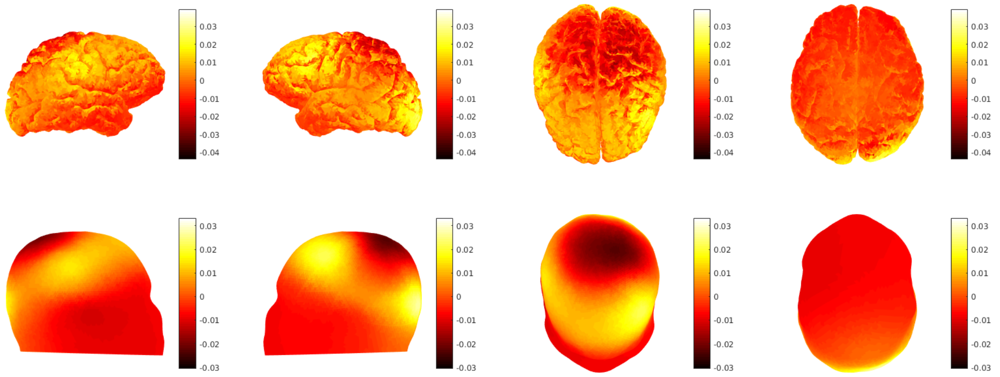


Figure 3.16: The 6th strongest mode of the LFM

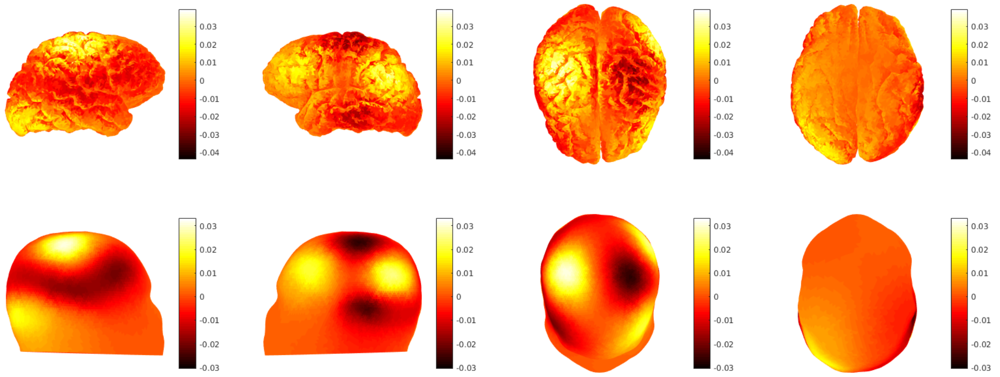


Figure 3.17: The 11th strongest mode of the LFM

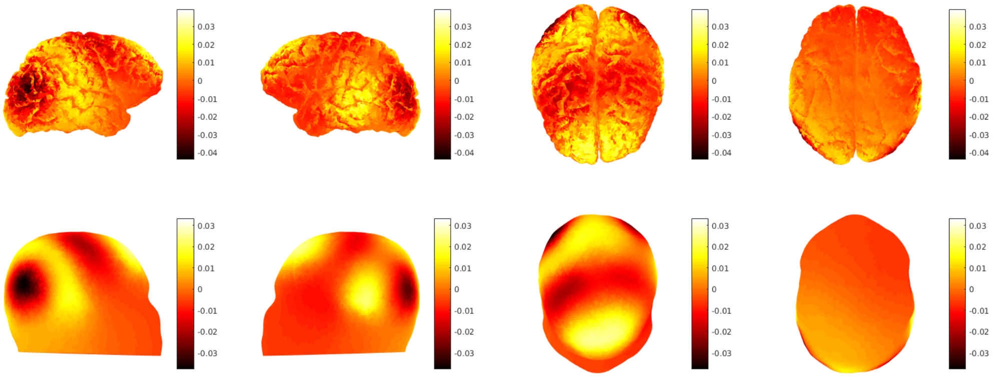
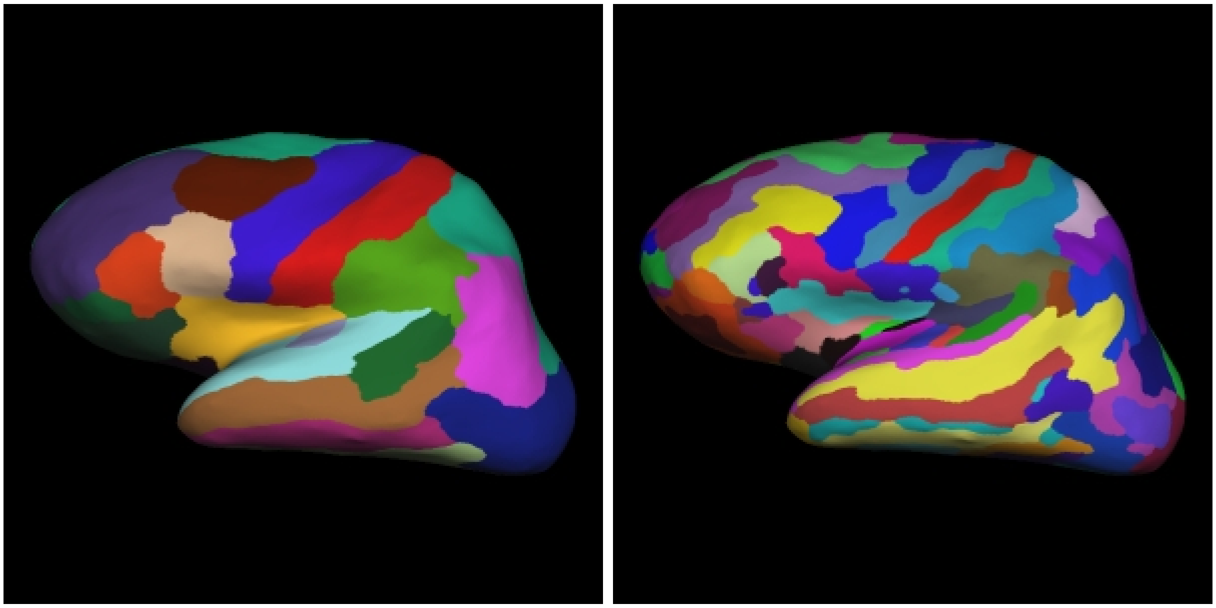


Figure 3.18: The 16th strongest mode of the LFM

connections ending in or originating from regions with similar steering-vectors (columns of the lead-field matrix) are fundamentally unidentifiable.

A standard solution to this problem is to interpolate the EEG recordings over the entire scalp surface and then apply a spatial filter [117] over the scalp and at the sensor locations to minimize the contributions of the regions which are not located directly under the target sensor. A similar spatial filter is also applied to the columns of the ill-conditioned LFM in order to create a well-conditioned linear model. Nevertheless, this approach is not implemented here; The linear state space model suffers from other shortcomings that will be discussed shortly.

For the macro-scale inter-region connectivity analysis, two brain segmentations are considered. The Desikan-Killiany Atlas [27] is a 34 area cortical atlas that is based on gyral morphology. The Destrieux Atlas [28] provides a finer granulated parcellation as it parcellates each hemisphere into 74 regions of interest. Figs. 3.19(a) and 3.19(b) show these brain atlases. To account for the spatially nonuniform activity of the relatively large regions, SVD is applied to the subset of the LFM that belongs to a given region of interest, and the strongest mode is selected as a representative of the region activity. The final gain matrix consists of columns that show the electrode responses to individual region activities.



(a) The Desikan brain atlas.

(b) The Destrieux brain atlas.

Figure 3.19: The two brain atlases used for constructing the region-to-electrode gain matrix.

Spatial Characteristics of the Recordings

A corollary of the ill-conditioned LFM is that the electrode recordings are highly correlated. Fig. 3.20 shows the correlation between the recordings. As a result, the information provided by the 128 electrodes is highly redundant and indeed far less informative than expected.

The informative part of the recordings may be extracted by applying principal component analysis (PCA) to the data by considering the recording at each instant as a point in a 128-dimensional space. Fig. 3.21 shows that more than 99% of the variability is explained by considering only the first 10 principal components. Similarly, the LFM should be reduced by multiplying the LFM on the left by the whitening matrix resulting from the PCA.

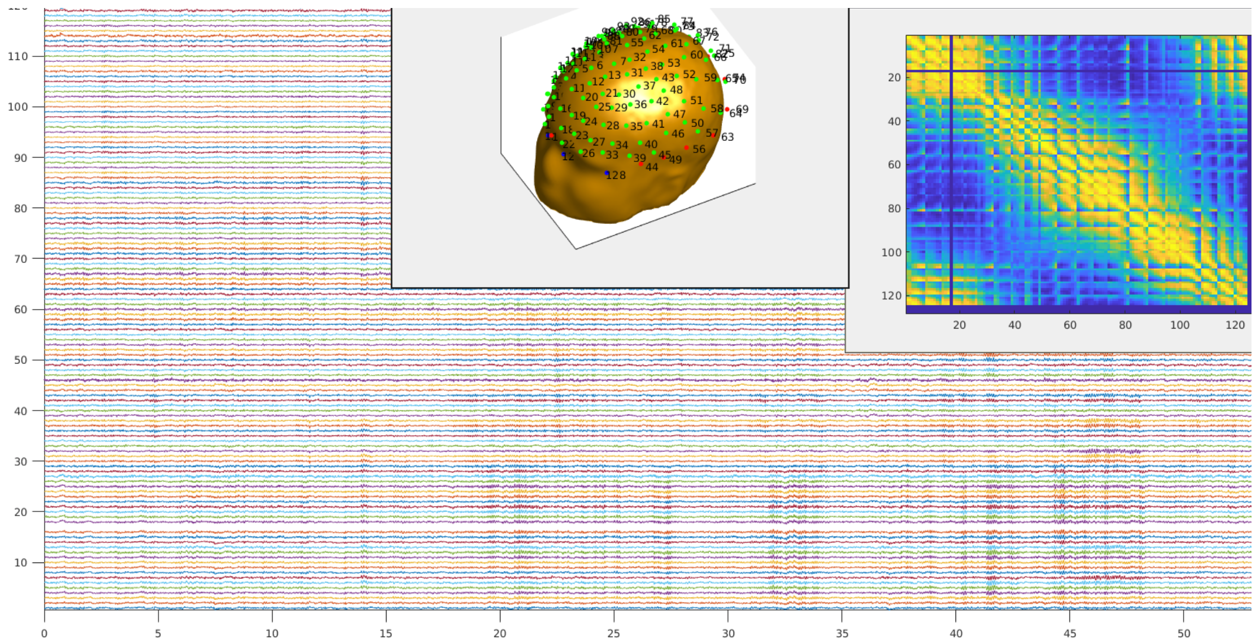


Figure 3.20: The correlation between electrode recordings. Electrode indices are shown on the scalp. A very high correlation exists between nearby electrodes.

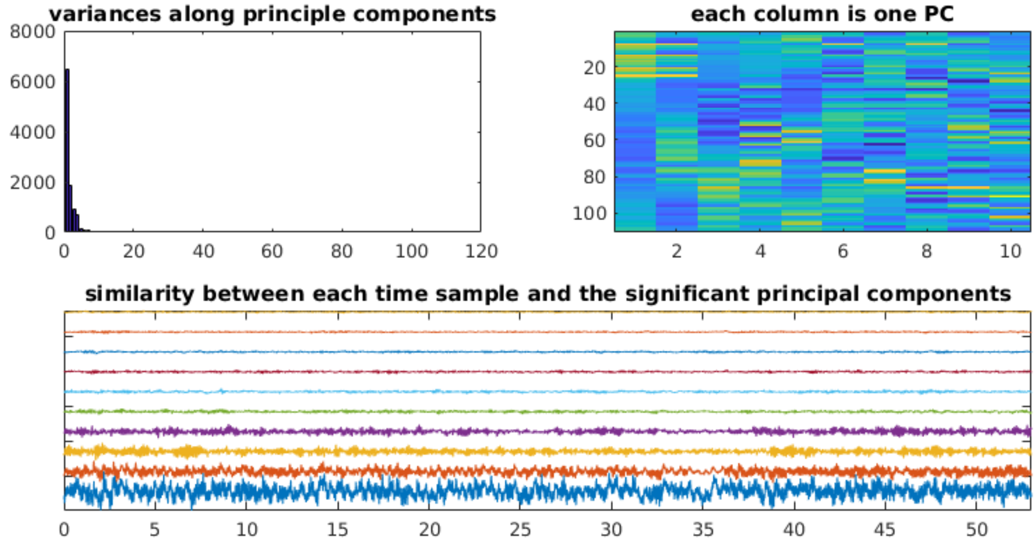


Figure 3.21: PCA analysis of electrode recordings. Top left: More than 99% of the variability is explained by the first 10 principal components. Top right: The first 10 principal components. Bottom: The recordings are projected into the subspace spanned by the first 10 principal components.

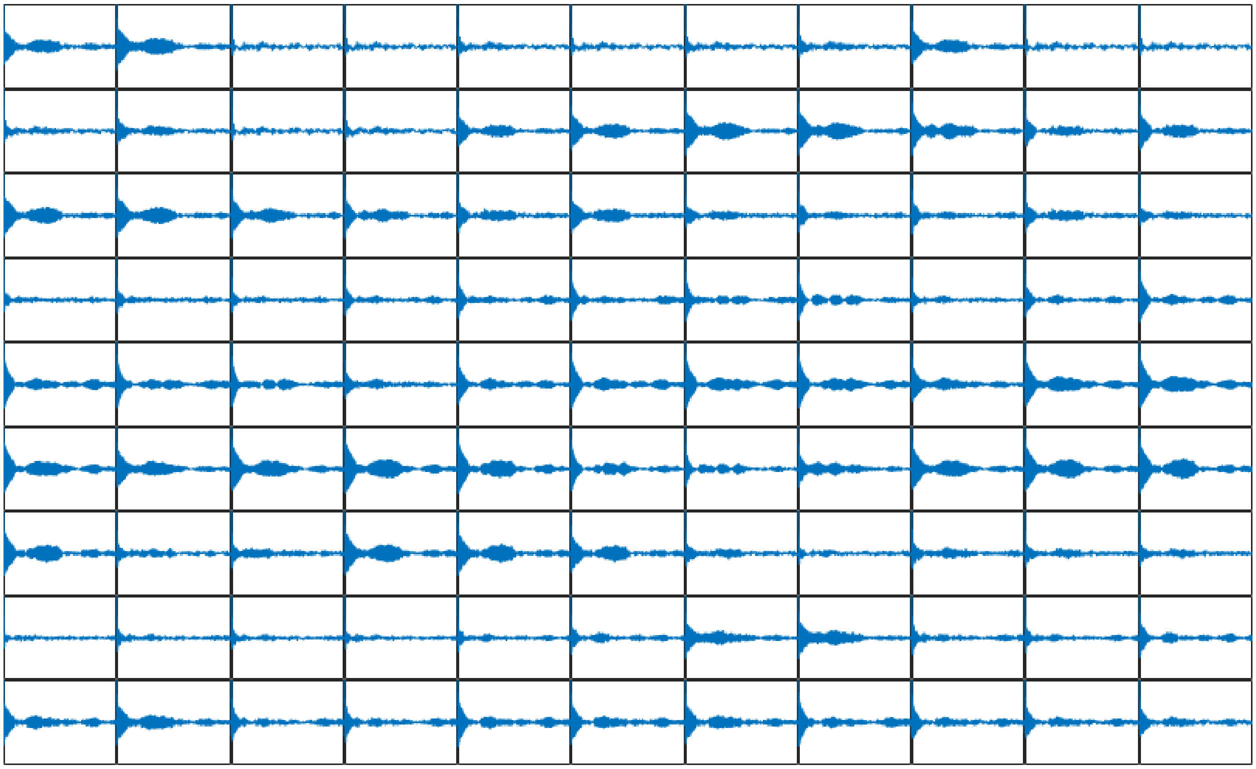


Figure 3.22: The autocorrelation of each electrode recording. Each subplot is associated with one electrode. The maximum autocorrelation lag is 25000 samples, or approximately 25 seconds.

Spectrotemporal Characteristics of the Recordings

Even after applying PCA in the spatial domain, the transformed recordings still show significant redundancy in time domain. This is evident from the temporal auto-correlations depicted in Fig. 3.22. From an equivalent perspective, one may investigate the spectrum of the recordings as shown in Fig. 3.23. Most of the signal energy is concentrated at frequencies below 4 Hz and around 10 Hz.

As will be discussed shortly, the colored spectrum of EEG may not be well captured by the linear model even after modifying the model to incorporate colored noise.

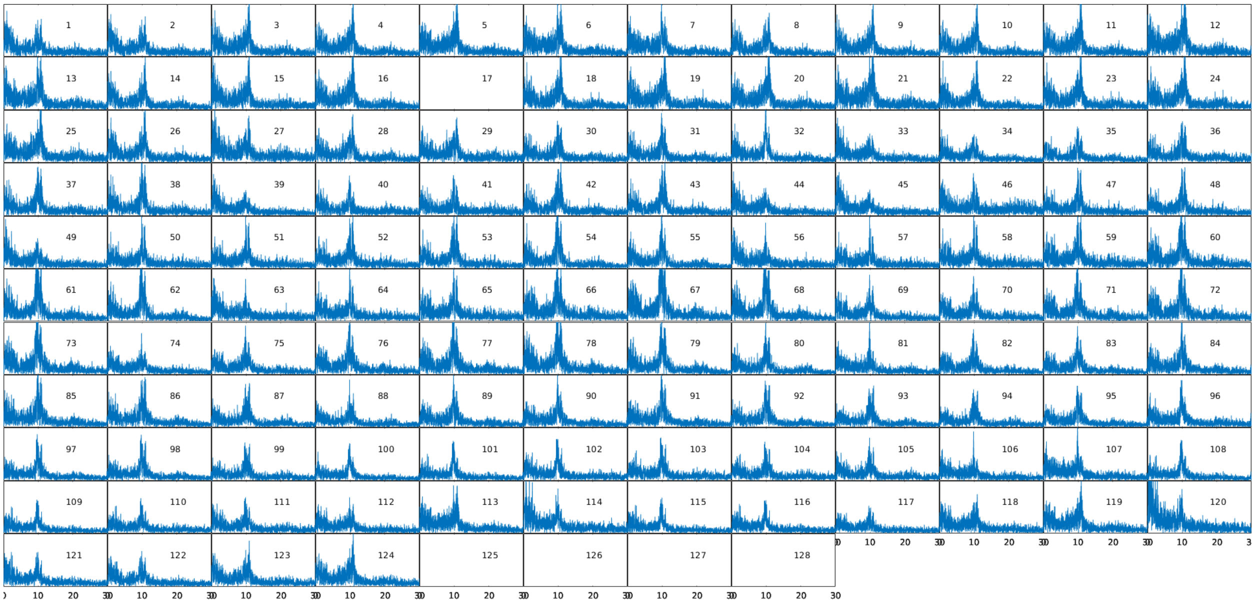


Figure 3.23: The spectrum of each electrode recording. Each subplot is associated with one electrode. The horizontal axis spans 0 Hz to 30 Hz.

3.5.3 Evaluation

Inflexibility of the White-Noise Driven Linear Model

The parameter estimation algorithm is applied to the described dataset. For the Desikan atlas, only 25 regions of interest with significant contributions are considered. A maximum delay of $D = 10$ and a regularization of $\lambda = .18$ was selected by the hyperparameter selection approach discussed in Section 2.9.

Our simulations show that the convergence point is slightly different between the runs and depends on both the starting point of the EM and the order in which the connections are estimate. Fig. 3.24 shows an example of the estimated connectivity matrix. To bring out the pattern, the estimated connectivity matrix may be represented as a 25×25 grid of plots, where the subplot (i, j) summarizes the connections into region i originating from region j at different delays. Fig. 3.25 shows two examples of this representation.

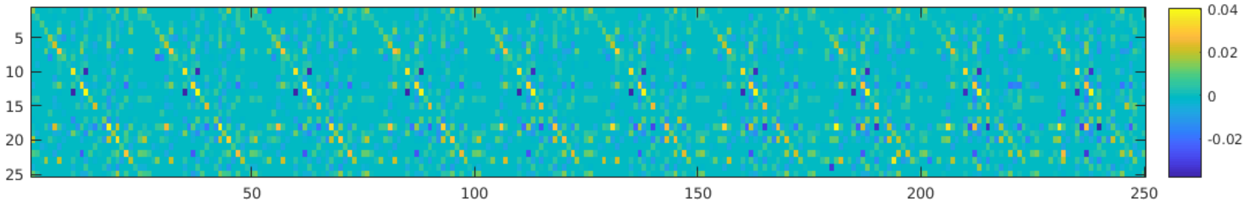


Figure 3.24: The connectivity matrix estimated from EEG signals. 25 regions of interest with a maximum delay of $D = 10$ is considered.

The results consistently show the following two characteristics:

- The algorithm is biased toward contributing the dynamics of a region into the history of the same region. Furthermore, the dependence on the self past activity manifests as a simple low pass filtering. The same feature is observed even after increasing D to large values such as $D = 100$. Both the bias of the algorithm in favor of self-connections and the oversimplified form of this dependence signal the inconsistency of the white-noise-driven linear model with the true system underlying the brain dynamics. In particular, the model should be capable of attributing 'high enough' importance to the region-to-region interactions. Furthermore, the estimated dynamics should be rich enough to capture the spectral and temporal features of EEG signals. In fact, if the model is used to generate synthetic EEG signals, the resulting recordings appear to be temporally white and posses none of the features of real EEG signals.
- The algorithm's behaviour in estimating the temporally-smeared connections is relatively consistent and independent of the inherent randomness. However, the estimated single-delay spiky connections between different regions vary significantly from simulation to simulation. Therefore, assuming that spiky connections do exist between brain regions, the algorithm is not capable of estimating them correctly. We may conclude that the model cannot resolve fine temporal connection information from EEG signals.

The inflexibility of the model may be viewed from other perspectives as well. First, if the model is truly valid, the measurement prediction error obtained from the Kalman filter should

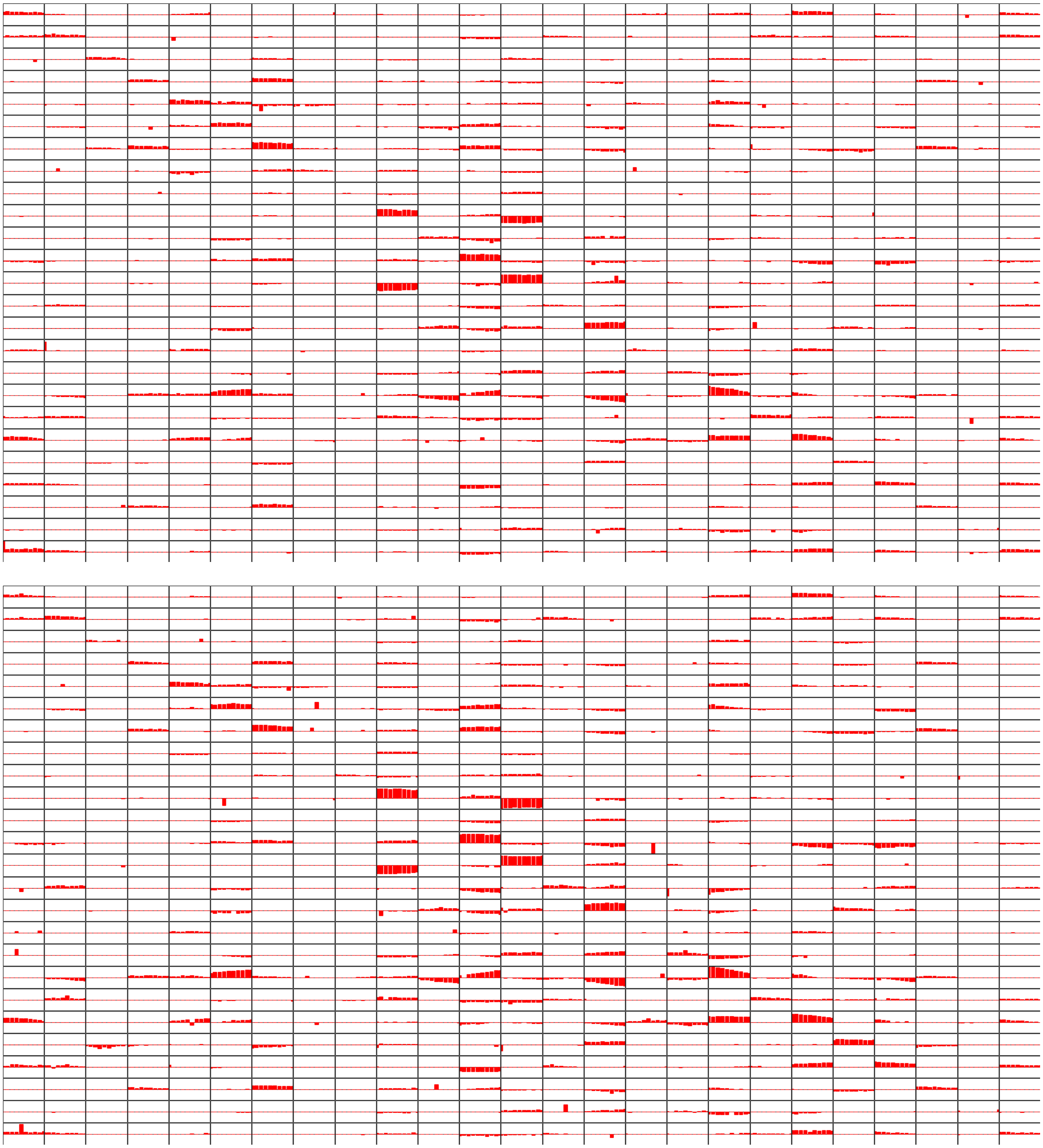


Figure 3.25: Two examples of the estimated connectivity. Subplot (i, j) summarizes the connections into region i originating from region j at different delays.

be temporally white. Although not presented here, the autocorrelation of the residue time series was calculated and was observed to have a very long oscillating tail. Formal statistical tests such as the Ljung-Box test were also applied to the residues and it was observed that the residues are not white.

Second, the system should be stable so the Kalman filter works correctly. Therefore, the elements of the $\mathbf{A}$ matrix should be small ($\ll 1$). As shown below, this condition implies that the model cannot reproduce the spectrum of EEG.

Starting from Eqs. 2.2 and 2.3, the Fourier transforms may be combined to obtain the spectrum of the measurement in terms of the process and measurement noise spectrum:

$$\mathbf{y}(e^{j\omega}) = \mathbf{C}[\mathbf{I} - \mathbf{A}(e^{j\omega})]^{-1}\mathbf{w}(e^{j\omega}) + \mathbf{u}(e^{j\omega}) \tag{3.5}$$

with:

$$\mathbf{A}(e^{j\omega}) = \sum_{d=1}^{D} \mathbf{A}_d e^{-j\omega d} \tag{3.6}$$

Given $|\mathbf{A}(e^{j\omega})| \ll 1$ due to stability, we can use the approximation $[\mathbf{I}-\mathbf{A}(e^{j\omega})]^{-1} \approx \mathbf{I}+\mathbf{A}(e^{j\omega})$. Then:

$$\mathbf{y}(e^{j\omega}) = \mathbf{C}\mathbf{w}(e^{j\omega}) + \mathbf{C}\mathbf{A}(e^{j\omega})\mathbf{w}(e^{j\omega}) + \mathbf{u}(e^{j\omega}) \tag{3.7}$$

Therefore, if $|\mathbf{A}(e^{j\omega})| \ll 1$ and if the spectrum of process and measurement noise are flat, the spectrum of $\mathbf{y}$ will be hardly affected by the small connectivity coefficients and will not reproduce the well defined spectrum of EEG demonstrated in Section 3.5.2. In fact, the synthetic EEG signals generated by the white-noise driven linear model produce an effectively white spectrum and waveforms that do not match the EEG waveforms. However, a colored process noise $\mathbf{w}$ might be able to mitigate this issue.

The Challenges of Incorporating Real EEG Dynamics

To take the dynamics of EEG into account, one possibility is to consider a colored noise as the driving force of the process. The colored noise should be written in the state space form to enable Kalman filtering. A second approach is to attribute the dynamics to the dependence of a node activity upon its own past. The connections may affect the dynamics as well. In general, a combination of these two approaches may be used.

To test the applicability of either of the mentioned approaches, a reasonable first step is to simplify the problem by assuming that there is no measurement layer on top of the process: If the time series of interest are directly measured, can we detect the connectivities reliably? If so, the next step would be to add the measurement layer. To answer this question, we take an incremental approach by starting from simplified systems and examine the challenges.

Modeling the Dynamics of a Single Electrode EEG Recording: Assume that the connections do not exist and there is no measurement layer. What kind of dynamical model can reproduce time series similar to a single electrode recordings and simultaneously fit into the problem formulation?

How could such a simplification be useful? Assuming that the LFM model is correct, the observation dynamics is a linear combination of the process dynamics. Therefore, a good dynamical model for each electrode activity might be a good candidate for the underlying process as well.

One possible candidate is the auto-regressive moving-average (ARMA) model as it may be written in state space form. The AR model is in particular promising, as the required states, the past history of a node activity, are already included in the problem formulation.

The first challenge is the required ARMA complexity. The examination of the 1024 Hz

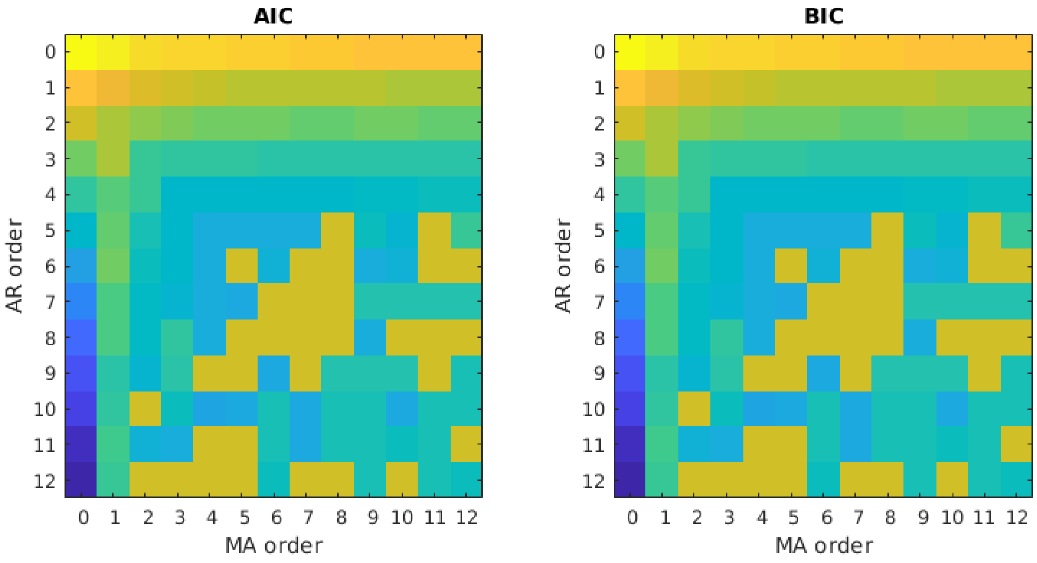


Figure 3.26: The AIC and BIC scores for ARMA model selection. Cool colors represent higher model quality. The 'dents' are caused by MATLAB not finding a stable fit.

data reveals that both short-range and long-range temporal features exist in the time series. Therefore, a large number of AR or MA terms should be expected to take both the short-range and long-range dependencies into account.

Both the Akaike information criterion (AIC) and the Bayesian information criterion (BIC) are used for selecting the order. The results are shown in Fig. 3.26. Jusding from both the AIC and BIC scores, it seems that AR leads to a more parsimonious model and MA terms are redundant.

It is noteworthy that MATLAB's ARMA fitting algorithms focus on minimizing the one-step ahead prediction error, without considering other metric such as spectral fit and whiteness of the residues. In other words, it is possible to find estimated models with a very low residue power that do not mimic the real system.

A second challenge caused by the combination of the high sampling rate of 1024 Hz and the dominant low frequency EEG components is that the AR fitting algorithm regresses the response against a set of highly correlated samples. Therefore, the predictive power may be relatively insensitive to the choice of regressors. In other words, different combinations of the regressors may lead to the same predictive power.

Fig. 3.27 shows the spectral fitness of the AR models estimated with orders ranging from 1 to 64. The smallest order that leads to a reasonable spectral fit is 21. Importantly, the dynamic range of the estimated AR coefficients are reported on each subplot. As expected, the dynamic range explodes with an increasing AR order. Therefore, if the AR coefficients are to be estimated together with other model parameters such as connections and noise covariances, it is very unlikely to achieve a stable system that mimics the spectrum of the EEG recordings and returns temporally white resides.

In fact, while the normality of the residues holds for all case, we found through an exhaustive search that no AR model leads to a temporally white residue as tested by the Ljung-Box test and other similar tests. As shown in Fig. 3.28, the residue autocorrelations are much larger that the whiteness confidence interval. Therefore, the AR model fails to correctly model the EEG signal at a sample rate of 1024 Hz.

A potential reason for the failure of the AR in modeling EEG is the high sampling rate. In other words, not all the past points are informative. The underlying signal is continuous time, and the dynamics are governed by the interaction of individual systems that follow simple differential equations. Also, by Shannon's sampling theorem, many of the intermediate samples do not help in the prediction and may be considered redundant data. It is true that the ARMA class may not be a good model for EEG. However, if we persist on using ARAM because it fits readily into the linear MVAR model, downsampling the data could open up new possibilites.

Fig. 3.29 compares the whiteness fitness and prediction fitness of different AR orders for various levels of temporal downsampling. A trade-off between the two fitness criteria is observed; while downsampling can whiten the residues, the one-step predictive power of the

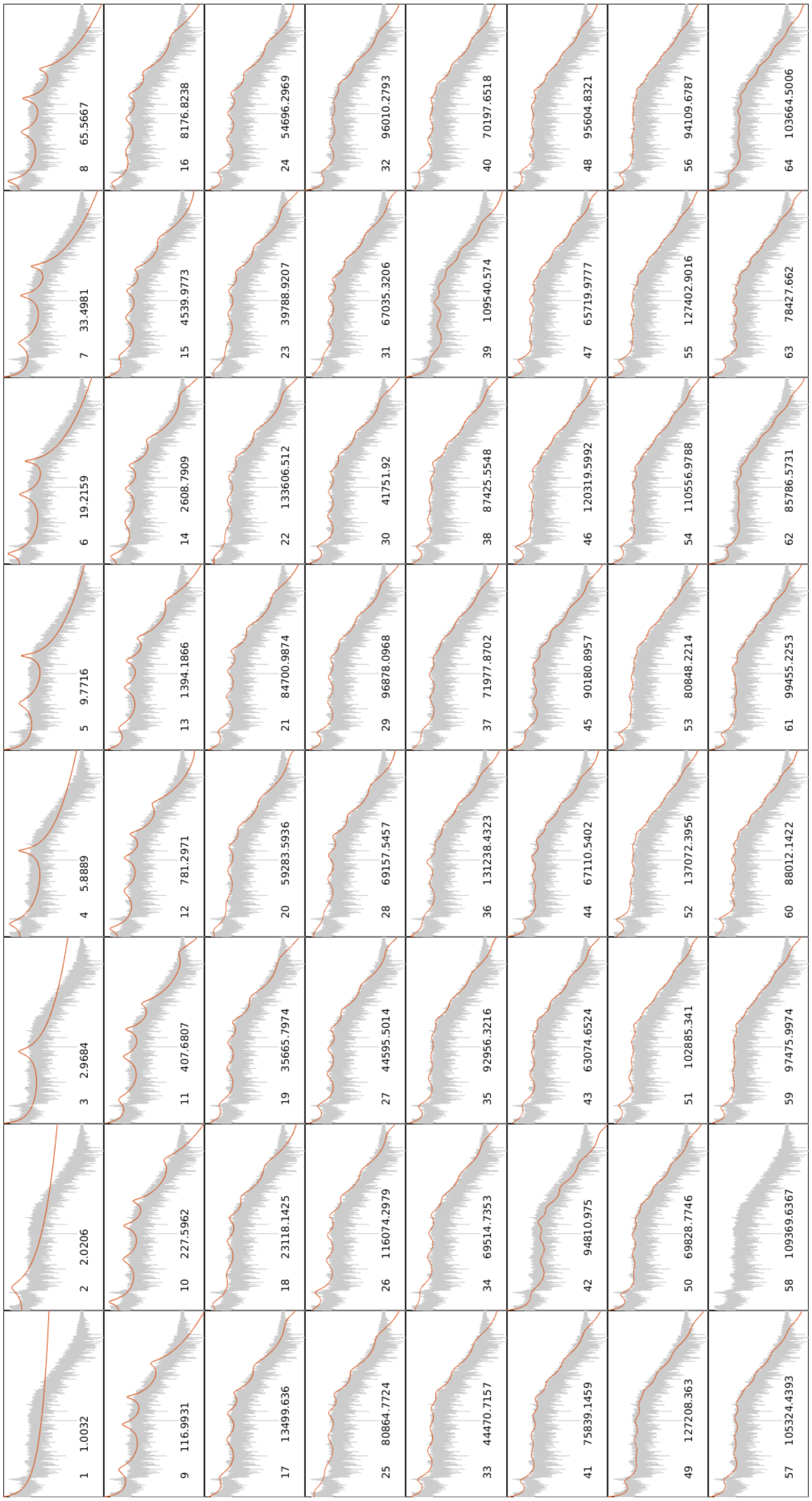


Figure 3.27: The spectral fit of the fitted AR models. The gray spectrum is extracted from real EEG. The red spectrum is obtained from the estimated AR model. The annotations on each subplot show the AR order and the dynamic range of the estimated AR coefficients.

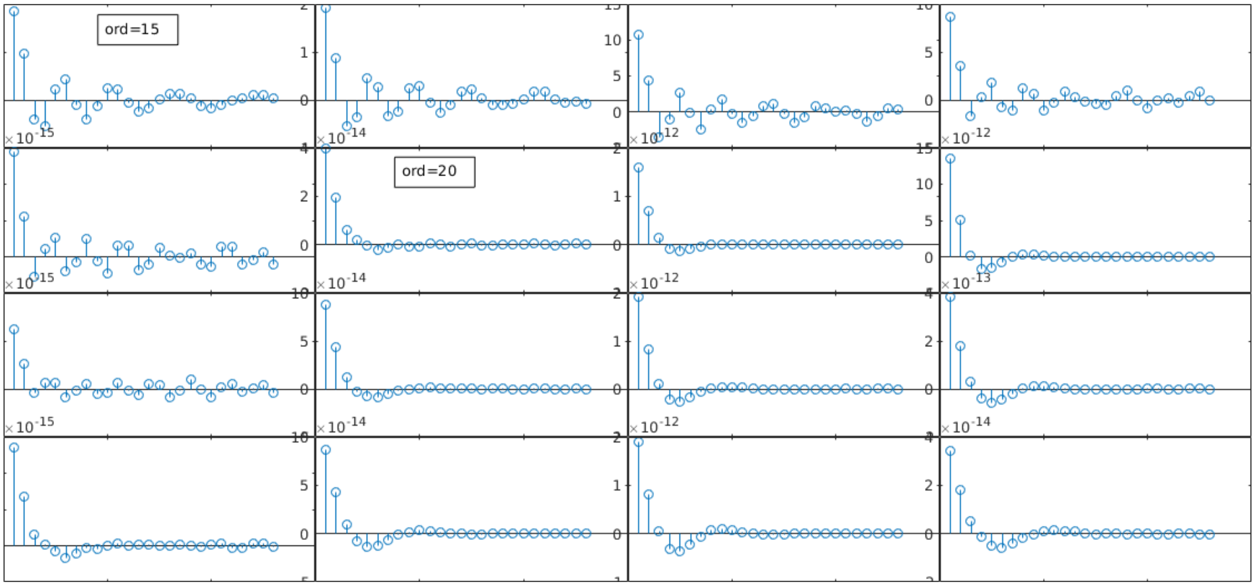


Figure 3.28: The stem plots represent the residue autocorrelation for different AR order (top left: 15, bottom right: 30. Orders increasing horizontally). The red lines that almost overlap with the x axis show the confidence region for the whiteness test.

model deteroriates with lower sampling rate.

A good compromise is achievable with a downsampling factor of 4 (sampling frequency of 256 Hz) and an AR order of 30. Fig. 3.30 illustrates different fitness criteria for this configuration (prediction accuracy, residue whiteness, spectral fitness, similarity of temporal waveform features).

Despite the reasonable fit of the AR model, the fitness is still highly sensitive to the accuracy of the AR coefficients. In fact, for the AR model presented in Fig. 3.30, a dynamic range as low as 10^4 results in a highly distorted spectrum and highly correlated residues. This sensitivity is especially problematic when the AR coefficients should be estimated together with other system parameters. The slightest deviation from the 'ideal' coefficients results in an unstable or low quality model.

The sensitivity of the model to AR coefficients is best justified by examining the pole-zero diagram of the resulting filter [52]. Fig. 3.31 shows the location of the poles and zeros of

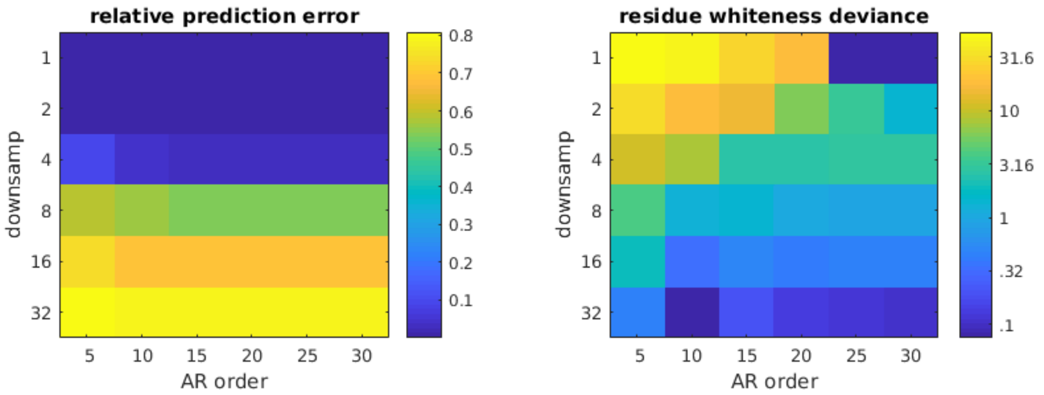


Figure 3.29: Left: colormap of relative prediction error. Right: colormap of residue whiteness deviance, defined as the ratio of the maximum value of the autocorrelation sequence to the whiteness confidence value.

the AR model presented in Fig. 3.30. The two pairs of conjugate poles on the far right are very close to the unit circle. Therefore, these poles account for both the dominant oscillatory modes of EEG. Due to the high model order, very small errors in the AR coefficients translate to large deviations in the locations of the dominant poles, resulting in either instability or lack of model fitness in terms of the similarity of the spatiotemporal features and whiteness of the residue.

As an alternative to the prediction-based AR parameter estimation, spectral-based filter design methods may be used to model the EEG recording as the output of a filter with a white noise input. Although fundamentally equivalent to the prediction-based AR parameter estimation methods, the spectral based methods provide the possibility of using the second-order-section (SOS) realization of the filter. The SOS realization has the advantage of robustness to the accuracy of the coefficients. However, the SOS formulation does not fit well into the linear MVAR model and is not pursued here. Specifically, the log likelihood function will include nonlinear polynomial terms in the coefficients of each section.

In summary, the primary challenges of modeling a single EEG recording are the numerical sensitivity and the complexity of the cost function if AR parameter estimation and SOS filter design are used respectively. Even if a high quality and tractable method exists for

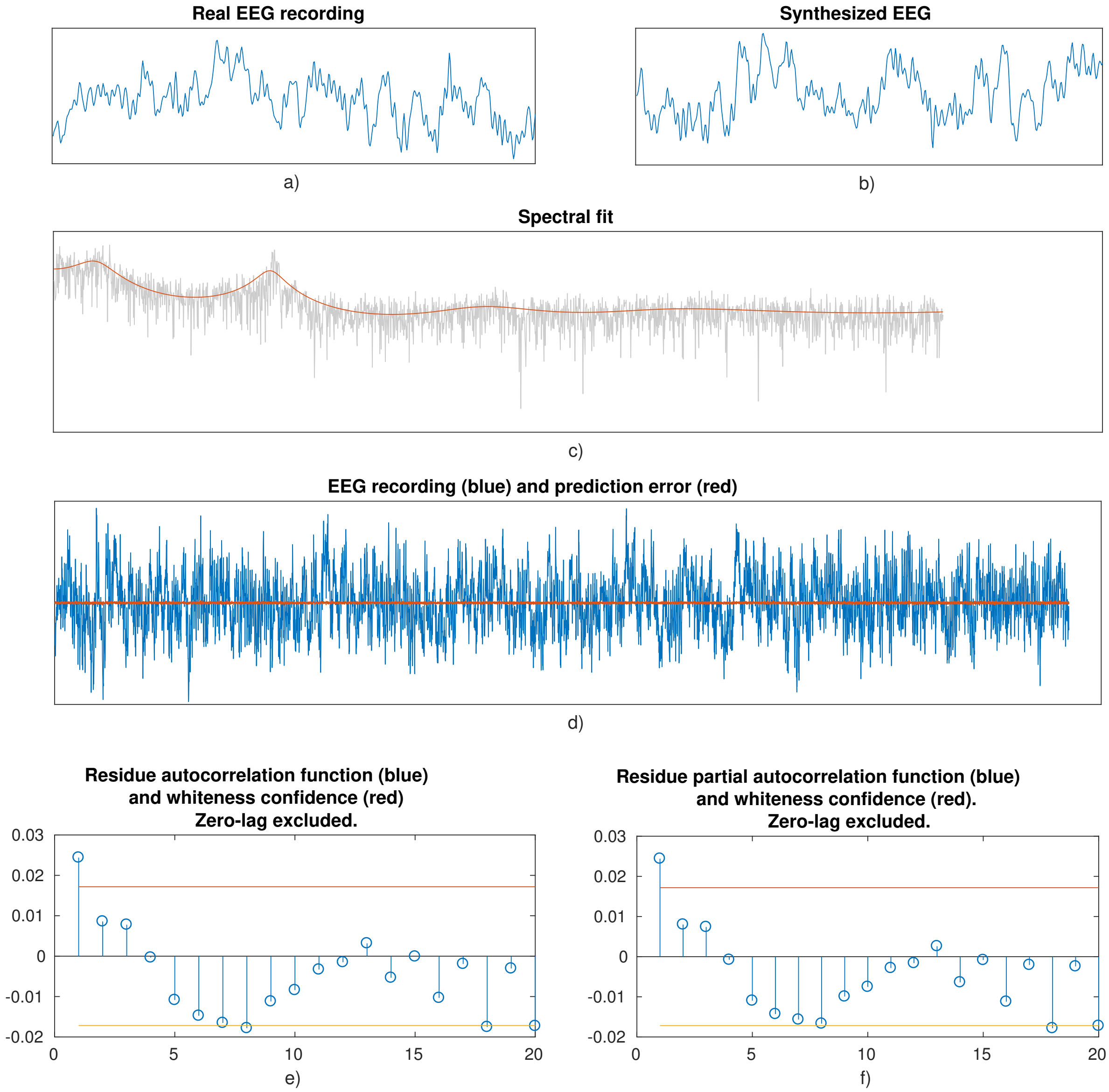


Figure 3.30: AR of order 30 fitted to 256 Hz real EEG signal. a) and b): the synthesized and real EEG share the same tempppral features. c) the AR spectrum matches the EEG spectrum. d) The prediction error is much smaller than the reference EEG signal (34 dB). e) and f) both the autocorrelation function (ACF) and partial ACF (PACF) fall within the whiteness confidence interval. The zero-lag is excluded as it bears no information about the whiteness.

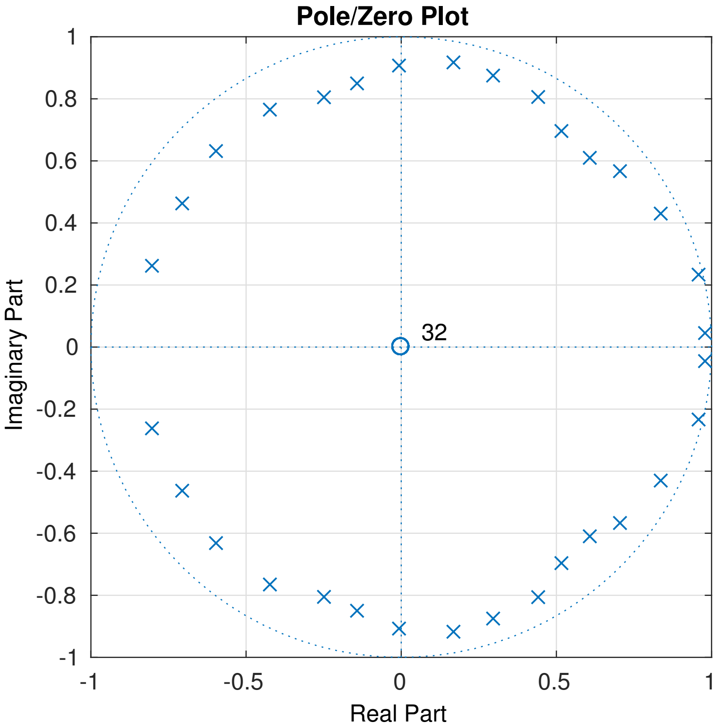


Figure 3.31: Pole and zero locations of the AR model of order 30 fitted to 256 Hz EEG.

modeling a single EEG recording, other issues remain to be addressed:

- The paramters of the model are not physiologically interpretable. In other words, the ARMA modeling is a black-box exploratory approach toward EEG modeling.
- Examination of EEG signals reveal that a combination of multiple slow and fast features constitutes the waveform. It is possible that the mechanisms generating the slow waves is different from the one generating the faster faster. Therefore, trying to find a single ARMA model that explains the combination might not be a reasonable approach. For instance, the low frequency content might be explained well by a small sample density corresponding to its bandwidth, and all the information in between maybe interpolated following Shannon's sampling theorem. The faster waves, on the other hand, will require a different sample rate.

Physiologically models, such as the neural mass model (NMM) discussed in the following chapters, may be able to provide a parsimonious, interpretable, and numerically stable alternative to the MVAR model . Models such as NMM suggest an interaction of second order dynamic with different time constants. Therefore, explaining dynamics of multiple time constants with a single ARMA model is probably not the best solution. The wide dynamic range of the AR coefficients is related to the wide dynamic range of the time constants that appear in the NMM model.

Modeling the Dynamics of Multi-Electrode EEG Recording: In the following paragraphs, we attempt to find a black-box linear MVAR model that explains the dynamics of multi-electrode recordings. Each electrode recording is regressed against the past activity of that electrode and other electrodes as well. To identify the challenges, the underlying regions-of-interest are ignored for simplicity: if the challenges facing the simplified model cannot be addressed, it is unlikely that the more complex model including the regions-of-interest can be addressed effectively.

To examine the influence of adding cross-electrode connections on the dynamics, we fix the dependence of a single EEG recording on its own past according to the AR model discussed above. The simple pathological example of Fig. 3.32 involving only two electrode recordings are considered. To examine the influence of the connections on system dynamics, the transfer function of the system is derived below.

In the frequency domain, we have:

$$y_1 = H_1(w_1 + c_{2,1}D_{2,1})y_2 \tag{3.8}$$

$$y_2 = H_2(w_2 + c_{1,2}D_{1,2})y_1 \tag{3.9}$$

Define $C_1 = c_{1,2}D_{1,2}$ and $C_2 = c_{2,1}D_{2,1}$. By eliminating y_2, the transfer function of y_1 is given

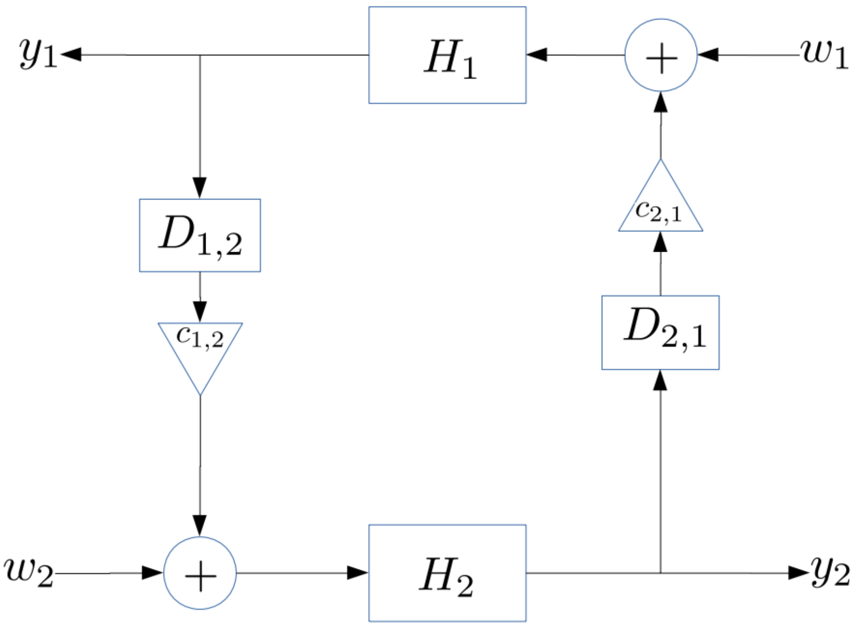


Figure 3.32: The MVAR model with connections between the two time series. w_1 and w_2 are the white noises driving the system. H_1 and H_2 are the single electrode transfer functions discussed previously (found by AR estimation or SOS realization). D blocks represent delay. $c_{1,2}$ and $c_{2,1}$ represent the strength of connections. y_1 and y_2 are the electrode recordings.

by:

$$y_1 = \frac{H_1}{1 - H_1C_1H_2C_2}w_1 + \frac{H_1C_2H_2}{1 - H_1C_1H_2C_2}w_2 \tag{3.10}$$

For simplicity, assume that $H_1 = H_2$. Then, defining $Q = 1/H$, it is easy to show that the poles of the transfer function are the roots of the polynomial $Q^2 - C_1C_2$. Therefore, compare to a system without any connections, the C_1C_2 terms modifies the root locations. The larger the magnitude of C_1C_2, the higher will be the influence of the connections on the dynamics of the system. Assuming zero-delay connections for simplicity, the root locus parameterized by C_1C_2 is illustrated in Fig. 3.33. It is observed that an increasing connection strength leads to lower frequency dominant modes of oscillation. Higher connection strengths push one of the dominant weakly damped poles across the unit circle boundary (at $C_1C_2 = 25 \times 10^{-6}$) and cause instability. Importantly, the numerical value of the connection strength that leads to significant changes in the dynamics, namely $c < 5 \times 10^{-3}$, is much smaller than the

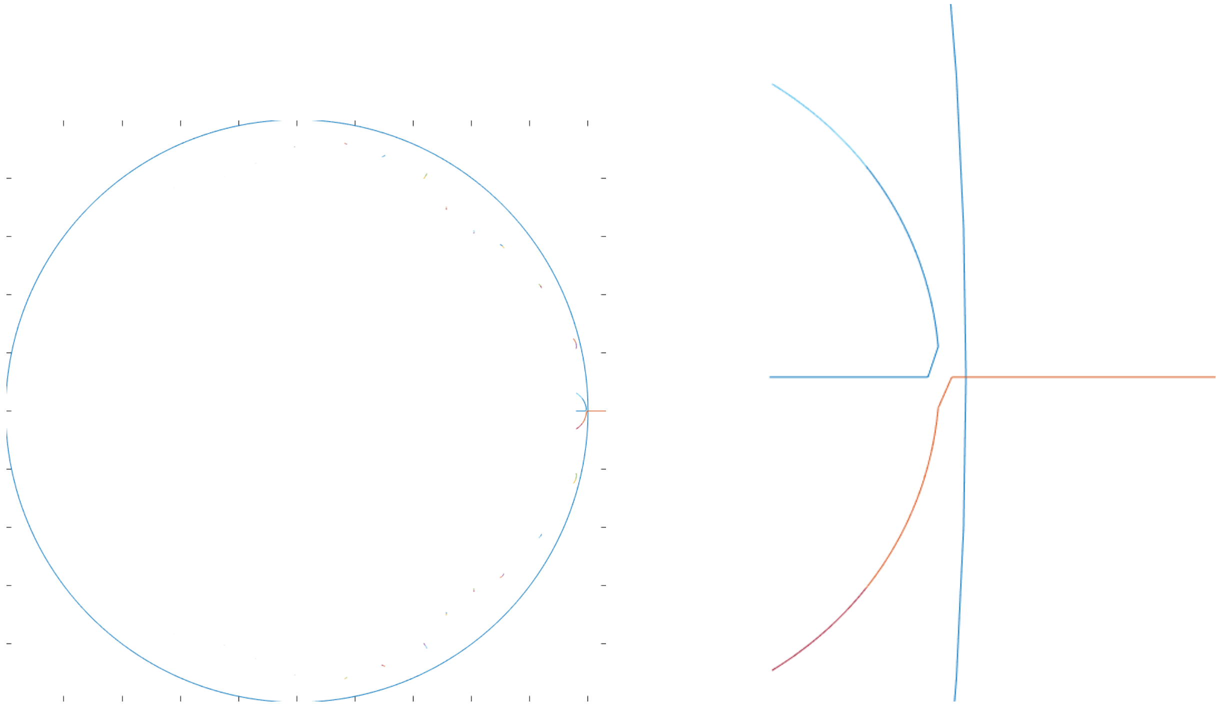


Figure 3.33: Left: the root locus of the transfer function of Eq. 3.10 parameterized by C_1C_2 ranging from 0 to 10^{-4}. The unit circle is included for reference. Right: Same as left, zoomed in on the rightmost dominant poles.

numerical values of the AR coefficients discussed in the previous sections. Therefore, the connections have a numerically much stronger influence on the dynamics of the system. As a result, and as will be discussed shortly, any estimation algorithm that jointly estimates the connectivities and the AR coefficients without balancing the influence of AR coefficients and the connectivities fails to return reasonable estimates. In summary, the connections should be numerically much smaller than AR coefficients. Furthermore, even small feedback values between the nodes can lead to instability.

To illustrate the ineffictiveness of standard vector autoregressive parameter estimation methods in estimating the connection values, consider an MVAR system with 4 time series where the AR coefficients are selected according to the parameters generating Fig. 3.30 . The connection strengths and delays are shown in Fig. 3.34.

Fig. 3.35 shows the EM-based maximum likelihood estimate of the connectivity matrix **A**

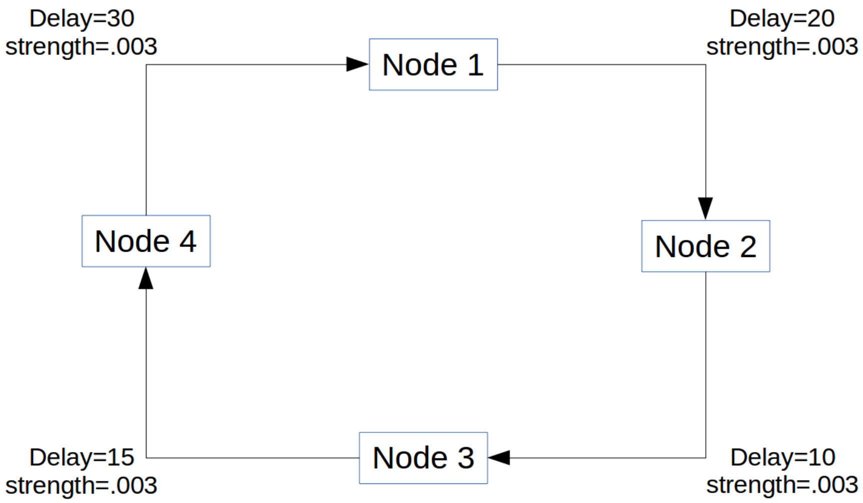


Figure 3.34: An MVAR system where the dependence of a node activity on its past is encoded by the AR coefficients discussed in Fig. 3.30. The connection delay and strengths are shown on the figure. The connection strengths of .003 place the system on the verge of instability.

partitioned into the self-connections, i.e. the AR coefficients relating the node activity to its own past, and the cross-node connections. It is observed that while the ML estimate of the self-connections are accurate, the cross-connections are dense rather than sparse, several orders of magnitude overestimated, and far from the ground truth. This behavior is persistent even if the self-connections are assumed to be known and only the cross-connections are estimated.

To impose the sparsity of the connections, assuming that the self-connections are known, the cross connections may be estimated using Lasso regression. Fig. 3.36 shows the estimated cross-connectivity vs. the ground truth. It is observed that the estimation accuracy improves significantly compared to Fig. 3.35 by applying the ℓ_1 regularization.

Unfortunately, assuming known values for the self-connections and then estimating the cross connections is suboptimal; The self-connection AR model is fitted to the data that already include the effect of possible cross-connections. If the AR estimation method overexploits the information in the waveform of interest, the cross-connectivity will have negligible descriptive power. In other words, the two stage method of first estimating the individual self-dynamics

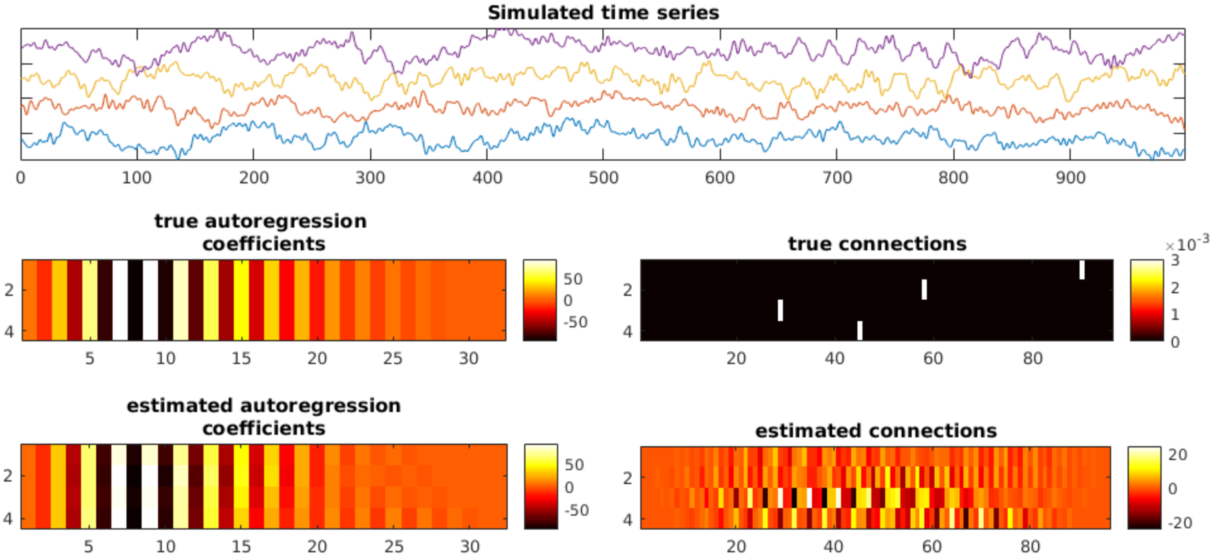


Figure 3.35: ML estimate of the connectivity matrix. Left: self-connections. Right: cross-connections. Row i of each colormap shows the dependency of time series i on its own past (left) and the history of other nodes (right).

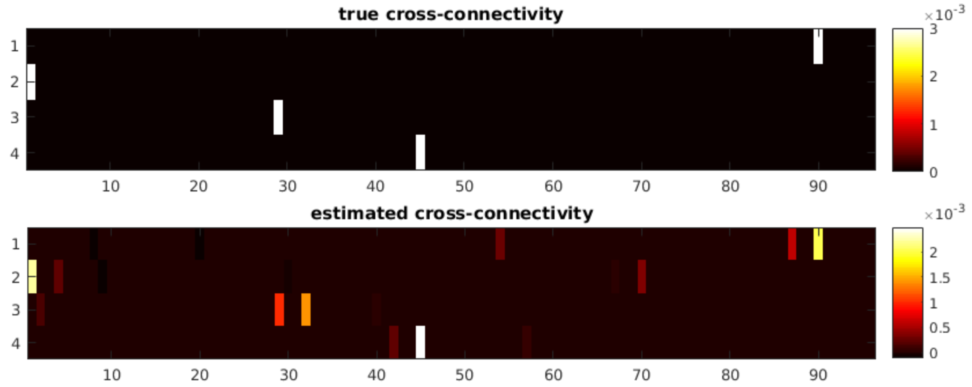


Figure 3.36: Lasso estimation of the cross-connections. The self-connections are assumed to be known and equal to the true values.

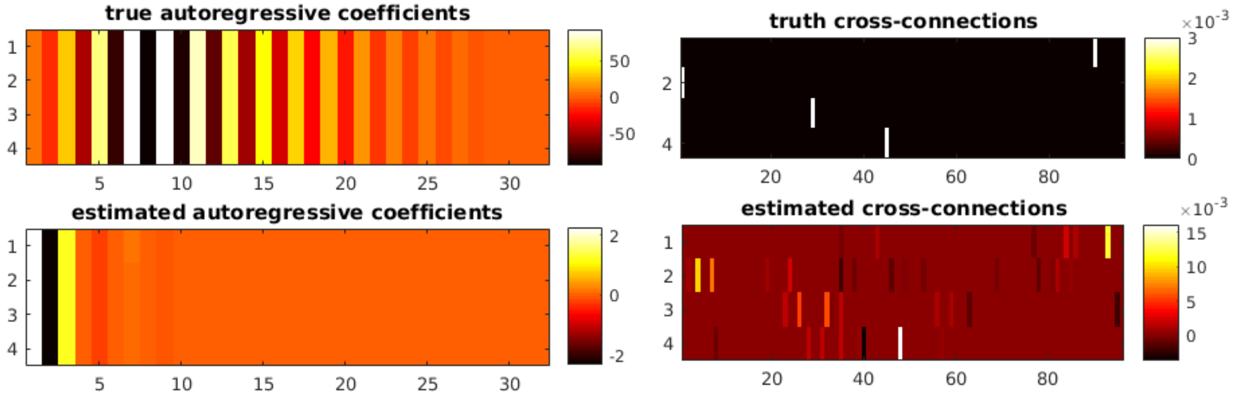


Figure 3.37: Lasso regression applied to the entire connectivity matrix.

(from the data that already includes the effects of connections) and then estimating the cross-connectivities is suboptimal to jointly estimating both the self- and cross-connections.

Finally, Fig. 3.37 shows the estimated self-connections and the cross-connections if Lasso regression is applied on the entire connectivity matrix rather than only on the cross-connections. While the cross-connections are estimated rather accurately, the estimates of the self-connections are not as rich as the ground truth. As a result, the dynamics of the waveforms generated based on the estimated connectivities do not mimic EEG dynamics. Furthermore, the estimated parameters lead to an unstable system.

A candidate solution is to use the generalized lasso [45, 109] to relax or weaken the ℓ_1 penalty on the self-connections. However, the generalized Lasso also fails to guarantee the stability of the resulting system. Without stability, the Kalman filter estimates used in the parameter estimation diverge.

Chapter 4

Realistic Neural Dynamics: Neural Mass Model

4.1 Introduction

Neural tissue generate oscillatory activity in many ways, driven either by mechanisms within individual neurons or by interactions between neurons. At the level of neural ensembles, synchronized activity of large numbers of neurons can give rise to macroscopic oscillations. Oscillatory activity in groups of neurons generally arises from feedback connections between the neurons. The interaction between neurons can give rise to oscillations at a different frequency than the firing frequency of individual neurons.

Neural oscillations are studied mathematically in the field of neurodynamics. To describe how neural activity evolves over time, the brain is modeled as a dynamical system governed by differential equations.

The mean field models on neural activity are based on the mean field theory, which studies the behavior of large and complex stochastic models by studying a simpler model. Such models consider a large number of small individual components that interact with each other. The effect of all the other individuals on any given individual is approximated by a single averaged effect.

The mean field models of neural activity can be divided into two classes: neural mass models (NMM) and neural field models (NFM). The main difference between these classes is that field models describe how a quantity characterizing neural activity (such as average depolarization of a neural population) evolves over both space and time as opposed to mass models, which characterize activity over time only, by assuming that all neurons in a population are located at (approximately) the same point. In this thesis, we focus on the neural mass model because the spatial granularity inherent in NMM matches the description of brain activity as the interaction of multiple regions-of-interest.

4.2 The Model

The neural mass model was first proposed by Freeman et. al. [40] based on the fact that neurons form populations and that the EEG is a reflection of ensemble dynamics arising from interconnected populations of pyramidal cells and interneurons. Their studies are based on experimental data and on computational models in which the dynamics of each neural ensemble are represented by a second order ordinary differential equation having a static nonlinearity identified as a sigmoid curve [32]. Similar ideas developed at the same time led to the development of a lumpedparameter population model able to explain the alpha rhythm of the EEG [24].

In its original form, the model represented a cluster of neurons containing three interacting subsets. The first subset was composed of the main cells (i.e. pyramidal cells in the hip-

	Excitatory	Pyramidal	Fast Inhibitory	Slow Inhibitory
Synaptic Gain	5.17 mV	5.17 mV	-57.1 mV	-4.45 mV
Synaptic Time Constant	1/75 sec	1/75 sec	1/60 sec	1/30 sec

Table 4.1: NMM parameters taken from [115]

pocampus or neocortex). It received a feedback from two other subsets composed of local interneurons, either excitatory or inhibitory. In order to explain the fast EEG rhythms that were not explained by the model, the model was later redesigned by Wendling et. al. [115] by adding a fourth subset to represent a second class of inhibitory interneurons with faster kinetics than those already included. The fast inhibitory interneurons terimnate near the soma and the slow ones in the dendrites.

From the a neuerodynamical perspective, each type of population (or subset) is characterized by the type of neurotrasmitters its neurons inject into the synapse. The neurotransmitter determines the temoral evolution of the contributed post synaptic potential.

Each neural population receives an average firing rate from other populations as well as from itself, which is converted to an average post-synaptic potential (PSP) (depending on type of neurotransmitter used by presynaptic population) by convolving the input firing rate with an impulse response. The convolution may also be represented by a second order differential equation. For each population type i (e for excitatory, p for pyramidal, f for fast inhibitory, and s for slow inhibitory), the impulse response mapping the firing rate to the PSP is given by:

$$h_i(t) = \frac{A_i}{\tau_i} t e^{-t/\tau_i} \tag{4.1}$$

where A is the synaptic gain measured in milivolts, τ is the synaptic time constant typically measured in miliseconds. Typical numbers for these parameters are reported in Table. 4.1.

The average PSPs then propagate to soma, are added together, and converted to an output average firing rate through a sigmoid function:

$$S(v) = \frac{2e_0}{1 + e^{r(v_0 - v)}} \tag{4.2}$$

where v is the PSP, e_0 is the maximum firing rate, v_0 is the voltage at which half max firing rate is obtained, and r is the slope of the sigmoid functions. Typical numbers for these parameters are $v_0 = 6$ mV, $e_0 = 2.5$ sec^{-1}, and $r = .56$ mV^{-1}.

The connections between two populations are modeled by synaptic connectivity C which is equal to the average number of synaptic connections that terminate on a neuron of the destination population and originate from the source population. To summarize, the dynamics of a single population is illustrated in Fig. 4.1. Importantly, similar to the previous work, it is assumed that the filtering operation performed at the synapse only depends on the type of the presynaptic population.

The independence of the filtering operation from the type of post-synaptic population can be used to lower the complexity of the model as follows. Since each population may terminate in multiple other populations, a naïve formulation requires adding a filtering block at every destination population for the connections originating from one source population. However, since the convolution operation (the filter $h(t)$) and multiplication (C_{ij} block) are interchangeable, all the filtering operations related to the connections originating from a single population may be combined as a single filter at the output of the population.

Fig. 4.2 illustrates the the model of a cortical column proposed by Wendling et. al. [115]. The connectivity parameters are given in Table . The external input driving a cortical column is assumed to excite the pyramidal population [124].

A well-studied appealing feature of the NMM is its capability to reproduce real-world brain waves. For instance, the authors of [115] show that the different types of activity produced by

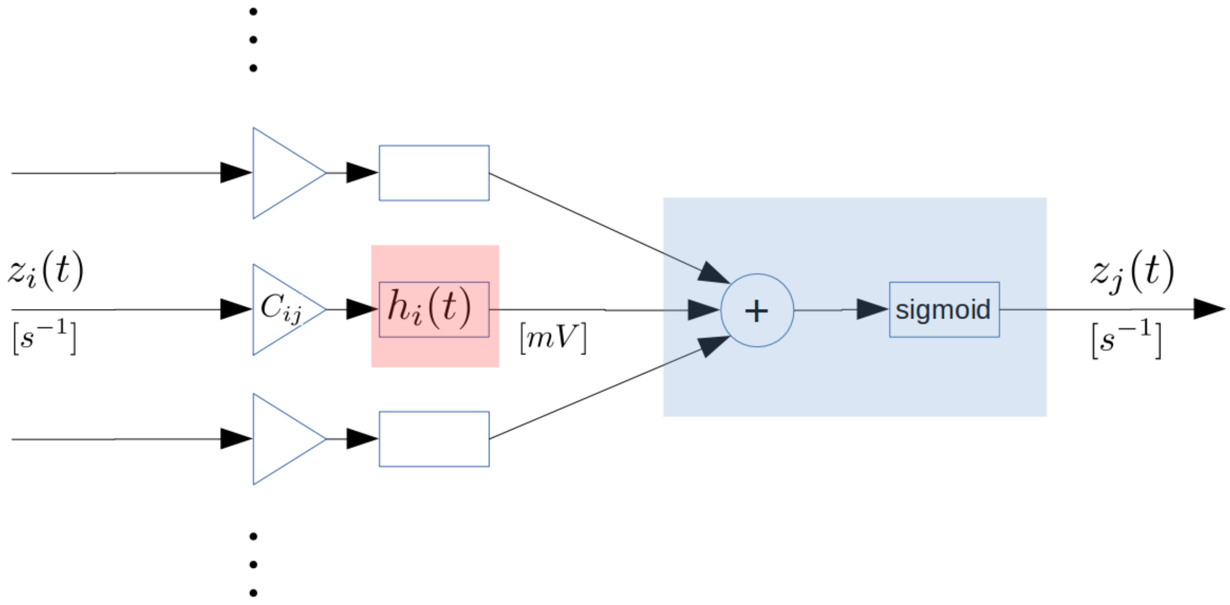


Figure 4.1: The dynamics and the interaction of populations in NMM. For population j, the inputs arrive as firing rates from other populations (only population i is indexed here). The firing rates are scaled according to the synaptic connectivity between the two populations (C_{ij}). The red rectangle highlights the processing done by the synapse connecting population i and j. The PSPs then propagate through the dendrites and add up at the soma of the destination population. The superposition of the PSPs is then converted to the output firing rate through the sigmoid compression. The blue region show the processing performed at the soma of the destination population.

	Value	Description
C_1	$C = 135$	Average number of synaptic contacts terminating at an excitatory neuron and originating from pyramidal population
C_2	$.8C$	Average number of synaptic contacts terminating at a pyramidal neuron and originating from excitatory population
C_3	$.25C$	Average number of synaptic contacts terminating at a slow inhibitory neuron and originating from pyramidal population
C_4	$.25C$	Average number of synaptic contacts terminating at a pyramidal neuron and originating from slow inhibitory population
C_5	$.3C$	Average number of synaptic contacts terminating at a fast inhibitory neuron and originating from pyramidal population
C_6	$.1C$	Average number of synaptic contacts terminating at a pyramidal neuron and originating from fast inhibitory population
C_7	$.8C$	Average number of synaptic contacts terminating at a fast inhibitory neuron and originating from slow inhibitory population

Table 4.2: The connectivity parameters of a cortical column [115]

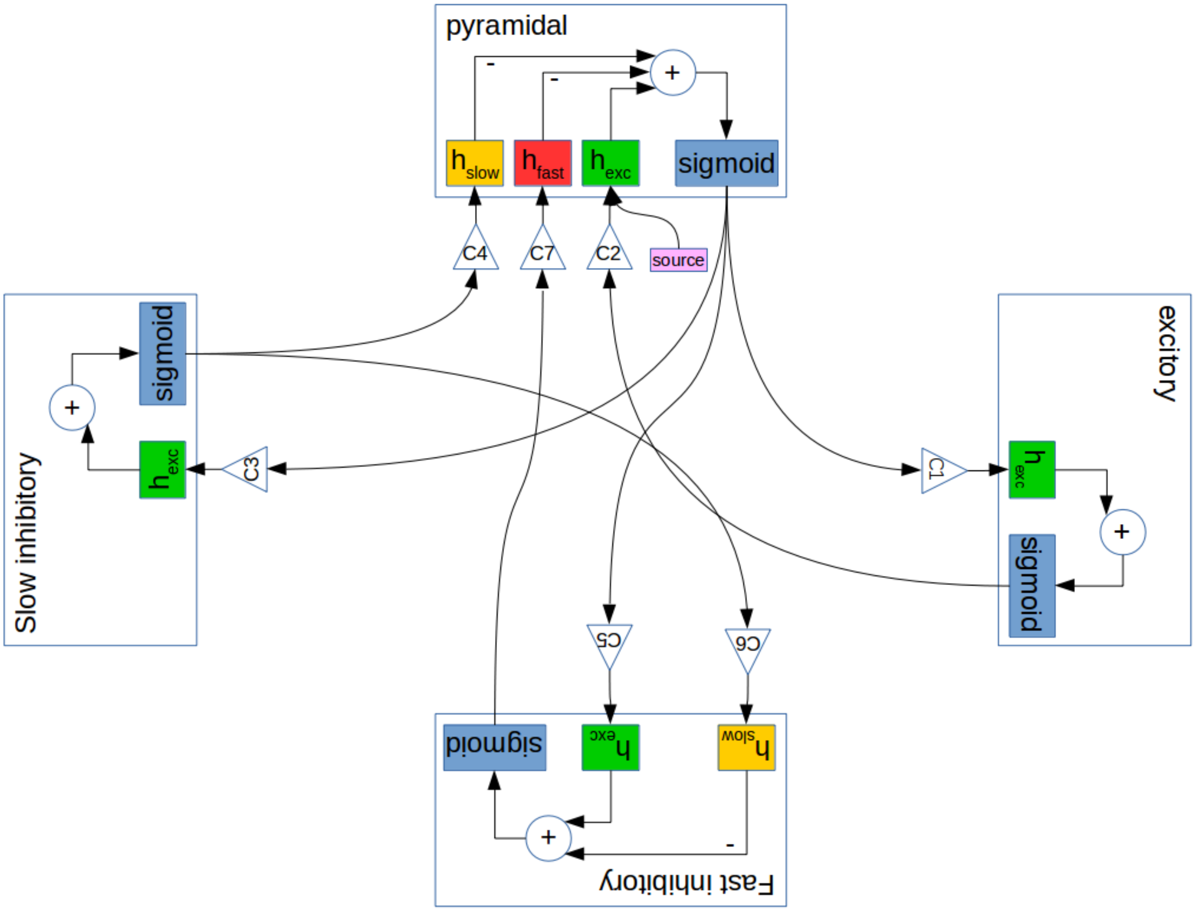


Figure 4.2: The NMM-based model of a cortical column as proposed in [115].

the model and match the types of real depthEEG signals recorded in human hippocampus. Specifically, the effect of synaotic responses A_{exc}, A_{slow}, and A_{fast} on the produces dynamics is studies. A reproduction of the results is shown in Fig. where $A_{exc} = 4$ mV and A_{slow} as well as A_{fast} are varied to produce different waveforms.

The model has been modified in later works by changing the connectivity structure, the involved populations, and the parameter values. For instance, Zavaglia et. al. [124] use slightly different values for the synaptic connectivites and add a self feedback loop on the fast inhibitory population to explain the gamma waves in EEG. Similar model may be found in [25], [124], and the references therein.

Finally, the delayed communication between two columns may be modeled using two approaches. In the first approach, the delay between source s and destination d is modeled by assuming that the firing rate arriving at d is $C_{sd}z_s(t - \tau_{sd})$, where $z_s(t)$ is the output firing rate of the source. In reality, however, the connection between the regions is realized by neurons, the operation of which may not be modeled by a simple delay. Therefore, in the second approach, these connecting neurons are modeled as populations following the NMM with time constants proportional to the delay of the connection.

4.3 Mathematical Description

The model described in this section is not the most general possibility. Specifically, it is assumed that the columns communicate through their pyramidal populations without any delays. The model may be readily generalized to more complicated structures.

The symbols used in the following discussion are summarized in Table 4.3. To model independent driving sources, we use the *external* populations which are only characterized by their output firing rate and do not receive any input. The regular dependent populations

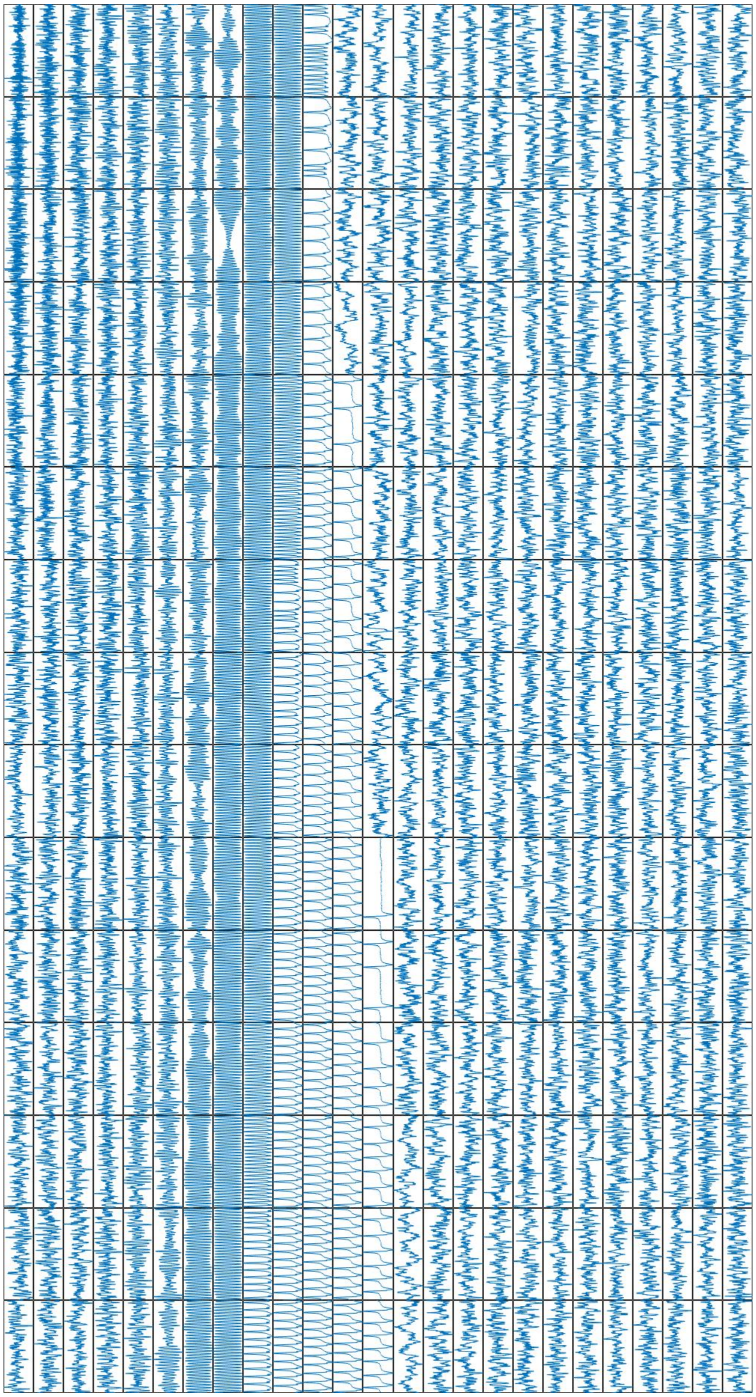


Figure 4.3: The NMM is capable of reproducing many EEG waveform types. A_{exc} is fixed at 4 mV. A_{fast} and A_{slow} increase from bottom to top and from left to right respectively.

Symbol	Definition
N_I	number of internal populations
N_E	number of external populations
N_P	total number of populations, $N_I + N_E$
$z_{n\rightarrow}(t)$	average firing rate out of population n
$v_{n\rightarrow}(t)$	the PSP contribution of $z_{n\rightarrow}(t)$ ($v_{n\rightarrow}(t) = h(t) \circledast z_{n\rightarrow}(t)$)
$w_n(t)$	additive noise on the PSP of internal population n
$W_{n\rightarrow m}$	average synaptic connections from population n to internal population m
$W_{\rightarrow n}$	$[W_{1\rightarrow n}, W_{2\rightarrow n}, \cdots, W_{N\rightarrow n}]^T$
$v(t)$	$[v_{1\rightarrow}(t), v_{2\rightarrow}(t), \cdots, v_{N\rightarrow}(t)]^T$
$v_{\rightarrow n}(t)$	average PSP delivered to internal population n, equal to $W_{\rightarrow n}^T v(t)$

Table 4.3: The symbols used in the mathematical description of NMM.

are also called *internal* populations.

In order to write the NMM in a state space form, first the filtering process is realized by a differential equation. Given that the Laplace transform of the filter $h(t) = \frac{A}{\tau} t e^{-t/\tau}$ is:

$$\mathcal{L}\{h(t)\} = \frac{A\omega}{s^2 + 2s\omega + \omega^2} \tag{4.3}$$

with $\omega = 1/\tau$, the corresponding differential equation is given by:

$$\ddot{v}_{n\rightarrow} + 2\omega\dot{v}_{n\rightarrow} + \omega^2 v_{n\rightarrow} = A\omega z_{n\rightarrow} \tag{4.4}$$

Each population n is then fully specified by the following set of equations:

$$z_{n\rightarrow}(t) = \begin{cases} S(v_{\rightarrow n}(t) + w_n(t)) & \text{if } n \text{ internal} \\ z_n^{\text{ext}}(t) & \text{if } n \text{ external} \end{cases} \tag{4.5}$$

$$\begin{bmatrix} \dot{\phi}_{n\rightarrow} \\ \dot{v}_{n\rightarrow} \end{bmatrix} = \begin{bmatrix} -2\omega_n & -\omega_n^2 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \phi_{n\rightarrow} \\ v_{n\rightarrow} \end{bmatrix} + \begin{bmatrix} A_n\omega_n \\ 0 \end{bmatrix} z_{n\rightarrow} \tag{4.6}$$

where $\phi = \dot{v}$. Defining $\phi(t)$ and $z(t)$ similar to $v(t)$ in Table 4.3, and with $A \triangleq \text{diag}[A_1 \cdots A_{N_P}]$ and $\omega \triangleq \text{diag}[\omega_1 \cdots \omega_{N_P}]$, the dynamics of the entire system is described by the following state space equation:

$$\begin{bmatrix} \dot{v}(t) \\ \\ \dot{\phi}(t) \end{bmatrix} = \begin{bmatrix} 0_{N_P} & I_{N_P} \\ \\ -\omega^2 & -2\omega \end{bmatrix} \begin{bmatrix} v(t) \\ \\ \phi(t) \end{bmatrix} + \begin{bmatrix} 0_{N_P} \\ \\ A\omega z(t) \end{bmatrix} \tag{4.7}$$

4.4 Simulating Normal EEG Activity

Each cortical column receives input from the external worlds. Therefore, with known internal mechanisms for each column, the behaviour and activity of the column depends on the external input. In order to simulate a network of cortical columns that generate EEG waveforms similar to real resting-state EEG signals, the influence of the input on the column activity should be studied.

To do this, let's consider a cortical column with parameters taken from Table 4.2. The pyramidal population of the column is driven by an external white noise source with a mean firing rate of 1 and a variance of 0.2. The connection strength is varied from 0 to 200 during the 5 second simulation.

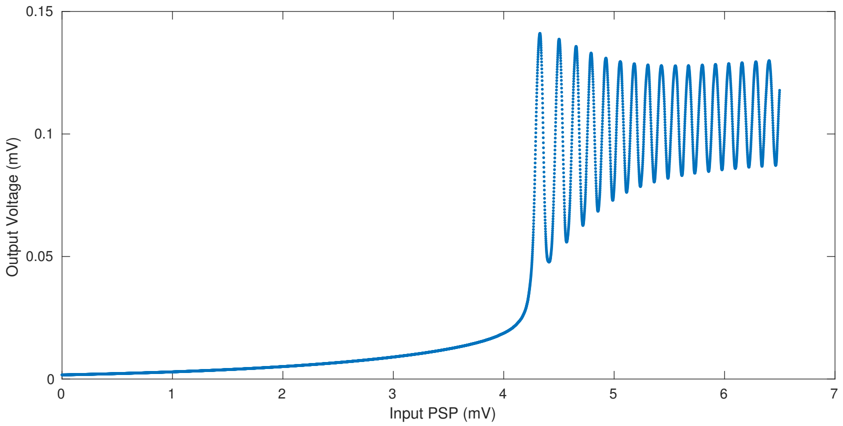


Figure 4.4: The relation between the input and the output of a cortical column.

Fig. 4.4 shows the output voltage of the pyraimdal population vs. its input (the voltage contribution of the external source) in milivolts. It is observed that once the input exceeds a threshold of 4 milivolts, the balancing feedback mechanism between the individual populations of the column breaks and the populations enter an unstable oscillating mode that is typical of abnormal (seizure) brain activity. Consequently, any synthetic network of cortical column used as the ground truth for the evaluation of the connectivity estimation algorithm should be constrained to generate cortical column inputs below 4 milivolts. This condition directly translates to a set of constraints over the strength of the connections between network elements.

In order to quantify this constraint, and without loss of generality, let's assume that each cortical column i has a base operating input of q_i milivolts provided by some internal or external independent source. Let q denote the vector $[q_1, \cdots, q_{N_c}]^T$, where N_C is the number of cortical columns. In order to perform a stationary point analysis and find a constraint on the connectivity such that the inputs remain under some V_{max} milivolts, the curve of Fig. 4.4 is approximated by a line $v_{out} \approx b + mv_{in}$, where a and m may be found by linear regression.

Let W temporarily encode the connectivity matrix between the cortical columns (pyramidal-to-pyramidal inter-column connections), and let W_i denote the vector of connections terminating in column i. Since the total input voltage on column i is given by $v_{in_i} = W_i^T v_{out} + q_i$ (where v_{out} is the vector of output potentials), the stationary point of the system is given by the solution of:

$$v_{out} = m(W^T v_{out} + q) + b \tag{4.8}$$

or

$$(I - mW^T)v_{out} = mq + b \tag{4.9}$$

To write an explicit condition on $v_{out} = (I - mW^T)^{-1}(mq + b)$, assume that mW^T is small enough such that the first order Taylor approximation $(I - mW^T)^{-1} \approx I + mW^T$ is relatively accurate. This condition may be satisfied by shrinking the W matrix until the largest eigenvalue of mW^T, denoted by $\lambda(mW^T)$, is sufficiently smaller than 1.

Finally, the condition on the stationary point is written as:

$$v_{out} \approx (I + mW^T)(mq + b) \leq V_{max} \tag{4.10}$$

or:

$$mW^T(m + b) \leq V_{max} - (mq + b) \tag{4.11}$$

which may be used to further shrink the connectivity matrix to ensure that the inputs into the columns remain in the stable region.

Our experiments show that ensuring $\lambda(mW^T) < .3$ is often sufficient for guaranteeing the stability of the system for denser connectivity matrices while not unnecessarily subduing the connection strengths. On the other hand, a sparse connection pattern typically requires further shrinking of the connectivity matrix by Eq. 4.11 as a highly sparse matrix with strong connections can have a zero λ but still lead to very high input potentials and thus an unstable system.

Chapter 5

Connectivity Estimation in Neural Mass Model

Similar to Chapter 3, the EM algorithm is used to estimate the connectivity parameters of the NMM. In the following sections, the E-step and the M-step of the EM algorithm are derived.

5.1 The E step

5.1.1 Inapplicability of the Standard Kalman Filter

As discussed in Section 4.4, the normal EEG activity requires that the total PSP on a population remains in a relatively linear operating area of the sigmoid function. Therefore, the standard Kalman filter is first applied to calculate the state expectations and state covariances that appear in the Q function of the EM algorithm.

For simplicity of the derivation, assume that the vector $v(t)$ in Eq. 4.7 is formed by first filling in the potential of the internal populations and then the potential of the external populations. Then, the system is described by:

$$\begin{bmatrix} \dot{v}(t) \\ \\ \\ \dot{\phi}(t) \end{bmatrix} = \begin{bmatrix} 0_{N\times N} & I_{N\times N} \\ \\ \\ -\omega^2 & -2\omega \end{bmatrix} \begin{bmatrix} v(t) \\ \\ \\ \phi(t) \end{bmatrix} + \begin{bmatrix} 0_{N\times 1} \\ \\ A\omega \begin{bmatrix} S(w(t) + W^T v(t)) \\ z^{\text{ext}}(t) \end{bmatrix} \end{bmatrix} \tag{5.1}$$

$$y(t) = GRW^T v(t) + \epsilon(t) \tag{5.2}$$

where y is $M \times 1$ the electrode measurements, G is the LFM, R is a row-selection matrix, W is the connectivity matrix, and ϵ is the measuremnt noise. The row selection matrix R encodes the fact that EEG recordings are primarily the result of the coherent activity of pyramidal populations. In the following derivations, the term GR is replaced by $\tilde{G}$ where necessary.

To apply the Kalman filter, this continuous-time model has to be rewritten in the following standard discrete-time state space form:

$$x[n+1] = Ax[n] + Bu[n] + Gw[n] \tag{5.3}$$

$$y[n] = Cx[n] + Du[n] + Hw[n] + v[n] \tag{5.4}$$

Denote the covariance matrix of w and v in Eq. 5.4 by Q and R, and let $E(w[n]v^T[n]) = N$. The noise terms in Eq. 5.4 are temporally white.

To do this, first the derivative is approximated by the finite difference. More accurate methods exist to convert a continuous-time state space system to the discrete time counterpart. However, these methods complicate the following derivation and provide little advantage over finite difference if the sampling rate is high enough. The finite-difference approximation results in:

$$\begin{bmatrix} v(t+1) \\ \\ \phi(t+1) \end{bmatrix} = \begin{bmatrix} v(t) \\ \\ \phi(t) \end{bmatrix} + dt \left(\begin{bmatrix} 0_{N\times N} & I_{N\times N} \\ \\ -\omega^2 & -2\omega \end{bmatrix} \begin{bmatrix} v(t) \\ \\ \phi(t) \end{bmatrix} + \begin{bmatrix} 0_{N\times 1} \\ \\ A\omega \begin{bmatrix} S(w(t) + W^T v(t)) \\ z^{\text{ext}}(t) \end{bmatrix} \end{bmatrix} \right) \tag{5.5}$$

Using the linear approximation $S(x) \approx a + mx$:

$$\begin{bmatrix} v(t+1) \\ \\ \phi(t+1) \end{bmatrix} = \begin{bmatrix} v(t) \\ \\ \phi(t) \end{bmatrix} + dt \left(\begin{bmatrix} 0_{N\times N} & I_{N\times N} \\ \\ -\omega^2 & -2\omega \end{bmatrix} \begin{bmatrix} v(t) \\ \\ \phi(t) \end{bmatrix} + \begin{bmatrix} 0_{N\times 1} \\ \\ A\omega \begin{bmatrix} a + mw(t) + mW^T v(t) \\ z^{\text{ext}}(t) \end{bmatrix} \end{bmatrix} \right) \tag{5.6}$$

where a is now a column vector with identical values.

If we break A and ω into internal and external parts as (each with dimensions N_I and N_E):

$$A = \begin{bmatrix} A_I & 0 \\ 0 & A_E \end{bmatrix}, \omega = \begin{bmatrix} \omega_I & 0 \\ 0 & \omega_E \end{bmatrix} \tag{5.7}$$

then:

$$\begin{bmatrix} v(t+1) \\ \\ \\ \phi(t+1) \end{bmatrix} = \begin{bmatrix} v(t) \\ \\ \\ \phi(t) \end{bmatrix} + dt \left(\begin{bmatrix} 0_{N\times N} & I_{N\times N} \\ \\ \\ -\omega^2 & -2\omega \end{bmatrix} \begin{bmatrix} v(t) \\ \\ \\ \phi(t) \end{bmatrix} + \begin{bmatrix} 0_{N\times 1} \\ \\ \begin{bmatrix} A_I\omega_I(a + mw(t) + mW^T v(t)) \\ \\ A_E\omega_E z^{\text{ext}}(t) \end{bmatrix} \end{bmatrix} \right) \tag{5.8}$$

or more explicitly in the state space form:

$$\begin{aligned} \begin{bmatrix} v(t+1) \\ \\ \\ \phi(t+1) \end{bmatrix} = \begin{bmatrix} v(t) \\ \\ \\ \phi(t) \end{bmatrix} + dt\Bigg(& \begin{bmatrix} 0_{N\times N} & I_{N\times N} \\ \\ \\ -\omega^2 & -2\omega \end{bmatrix} \begin{bmatrix} v(t) \\ \\ \\ \phi(t) \end{bmatrix} + \\ & \begin{bmatrix} & 0_{N\times 2N} & \\ A_I\omega_I m W^T & & 0_{N_I\times N} \\ & 0_{N_E\times 2N} & \end{bmatrix} \begin{bmatrix} v(t) \\ \\ \\ \phi(t) \end{bmatrix} + \\ & \begin{bmatrix} 0_{N\times N} & \\ A_I\omega_I m & 0_{N_I\times N_E} \\ 0_{N_E\times N_I} & A_E\omega_E \end{bmatrix} \begin{bmatrix} w(t) \\ z^{ext}(t) \end{bmatrix} + \begin{bmatrix} 0_{N\times 1} \\ A_I\omega_I a \\ 0_{N_E\times 1} \end{bmatrix} \Bigg) \end{aligned} \tag{5.9}$$

or:

$$\begin{bmatrix} v(t+1) \\ \\ \\ \phi(t+1) \end{bmatrix} = \left(I + dt \left\{ \begin{bmatrix} 0_{N\times N} & I_{N\times N} \\ \\ \\ -\omega^2 & -2\omega \end{bmatrix} + \begin{bmatrix} & 0_{N\times 2N} & \\ A_I\omega_I m W^T & & 0_{N_I\times N} \\ & 0_{N_E\times 2N} & \end{bmatrix} \right\} \right) \begin{bmatrix} v(t) \\ \\ \\ \phi(t) \end{bmatrix}$$
$$+ dt \begin{bmatrix} 0_{N\times 1} \\ A_I\omega_I a \\ 0_{N_E\times 1} \end{bmatrix} + dt \begin{bmatrix} 0_{N\times N} & \\ A_I\omega_I m & 0_{N_I\times N_E} \\ 0_{N_E\times N_I} & A_E\omega_E \end{bmatrix} \begin{bmatrix} w(t) \\ z^{ext}(t) \end{bmatrix} \tag{5.10}$$

In case the external excitation is composed of a deterministic part (e.g. mean value) and a random part as $z^{ext}(t) = Z^{ext}(t) + \delta z^{ext}(t)$, we can rewrite the last equation as:

$$\begin{bmatrix} v(t+1) \\ \\ \\ \phi(t+1) \end{bmatrix} = \left(I + dt \left\{ \begin{bmatrix} 0_{N\times N} & I_{N\times N} \\ \\ \\ -\omega^2 & -2\omega \end{bmatrix} + \begin{bmatrix} & 0_{N\times 2N} & \\ A_I\omega_I m W^T & & 0_{N_I\times N} \\ & 0_{N_E\times 2N} & \end{bmatrix} \right\} \right) \begin{bmatrix} v(t) \\ \\ \\ \phi(t) \end{bmatrix}$$
$$+ dt \begin{bmatrix} 0_{N\times 1} \\ A_I\omega_I a \\ A_E\omega_E Z^{ext} \end{bmatrix} + dt \begin{bmatrix} 0_{N\times N} & \\ A_I\omega_I m & 0_{N_I\times N_E} \\ 0_{N_E\times N_I} & A_E\omega_E \end{bmatrix} \begin{bmatrix} w(t) \\ \delta z^{ext}(t) \end{bmatrix} \tag{5.11}$$

which is in the same format as Eq. 5.4. The mapping of NMM parameters to the standard KF models is summarized in Table 5.1.

Fig. 5.1 illustrates the result of applying the standard Kalman filter to a synthetic system with 4 populations with each population driven by a white noise source. Clearly, the KF fails to provide good estimate of the hidden states. Although not presented here, a more granular Extended Kalman filter (EKF) was applied but did not exhibit any improvement.

A common reason behind the failure of Kalman filter is the unobservability of the system

Standard KF parameter	NMM counterpart
$x[n]$	$\begin{bmatrix} v(t) \\ \\ \\ \phi(t) \end{bmatrix}$
A	$\left(I + dt \left\{ \begin{bmatrix} 0_{N\times N} & I_{N\times N} \\ & \\ -\omega^2 & -2\omega \end{bmatrix} + \begin{bmatrix} & 0_{N\times 2N} & \\ A_I\omega_I m W^T & & 0_{N_I\times N} \\ & 0_{N_E\times 2N} & \end{bmatrix} \right\} \right)$
B	$dt \begin{bmatrix} 0_{N\times 1} \\ A_I\omega_I a \\ A_E\omega_E Z^{ext} \end{bmatrix}$
$u[n]$	1
G	$dt \begin{bmatrix} 0_{N\times N} & \\ A_I\omega_I m & 0_{N_I\times N_E} \\ 0_{N_E\times N_I} & A_E\omega_E \end{bmatrix}$
$w[n]$	$\begin{bmatrix} w(t) \\ z^{ext}(t) \end{bmatrix}$
C	$[GRW^T, 0_{M\times N}]$
D, H	$0_{M\times 1}$, $0_{M\times N}$
$v[n]$	$\epsilon(t)$
$E\{wv^T\}$	$0_{N\times M}$

Table 5.1: The relation between NMM parameters and the parameters of Eq. 5.4

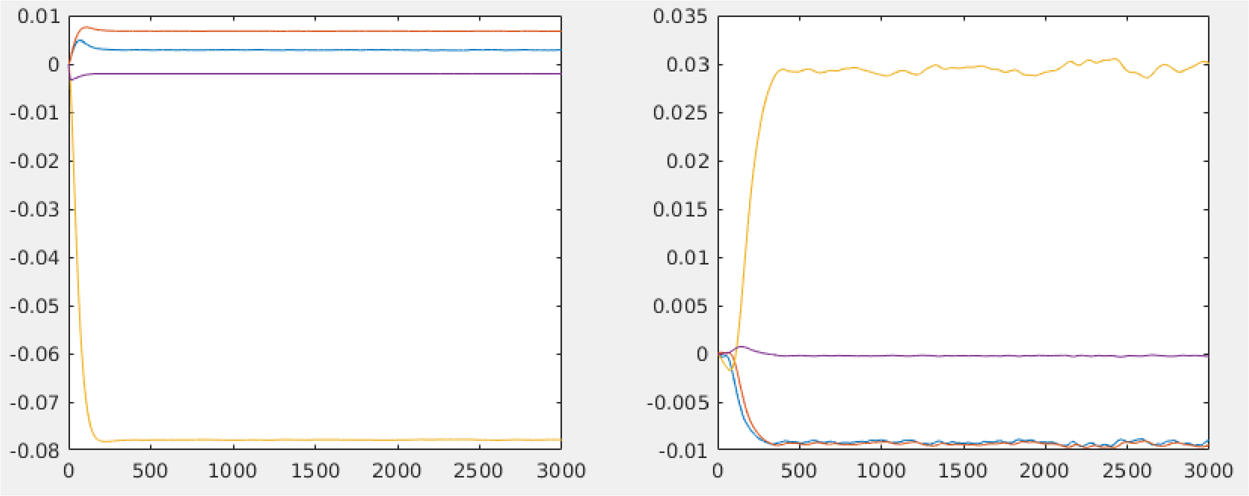


Figure 5.1: Left: the state vector $v(t)$ of a sample system with 4 populations. Right: the mean state estimated by the Kalman filter. Same colors encode the same time-series.

[49]. To check the observability of the linear state space system resulting from the NMM, an extensive set of synthetic systems were generated and it was observed that the number of non-observable states, as given by the difference between the rand of the state matrix (A in Eq. 5.4) and the rank of the observability matrix of the system, is always greater that zero. The unobservability implies that it is fundamentally impossible to draw meaningful conclusions about the unobservable states: while certain linear combinations of the states are observable, others are not.

To address this issue, Kalman decomposition is applied to the system to find the observable and unobservable subspaces. The state space is then written in terms of the observables only. Specifically, let U be the unitary decomposing matrix that partitions the generic state vector x into observable and unobservable parts:

$$z = Ux = \begin{bmatrix} U_o \\ U_u \end{bmatrix} x = \begin{bmatrix} z_o \\ z_u \end{bmatrix} \tag{5.12}$$

where subscripts o and u denote observable and unobservable, and z is the transformed state vector.

The standard state space representation is now tranformed as:

$$Ux[n+1] = UAU^T Ux[n] + UBu[n] + UGw[n] \tag{5.13}$$

$$y[n] = CU^T Ux[n] + Du[n] + Hw[n] + v[n] \tag{5.14}$$

or equivalently:

$$z[n+1] = (UAU^T)z[n] + (UB)u[n] + (UG)w[n] \tag{5.15}$$

$$y[n] = (CU^T)z[n] + Du[n] + Hw[n] + v[n] \tag{5.16}$$

Denoting the number of observable and unobservable states by n_o and n_u, and noting that UAU^T and CU^T will be of the form $[a_o, 0_{n_o \times n_u}; a_{uo}, a_u]$ and $[c_o, 0_{M \times n_u}]$ respectively, we can exclude the unobservables from the tranformed state space model:

$$z_o[n+1] = a_o z_o[n] + (UB)_o u[n] + (UG)_o w[n] \tag{5.17}$$

$$y[n] = c_o z_o[n] + Du[n] + Hw[n] + v[n] \tag{5.18}$$

Even after reducing the model, the output of the KF does not estimate the true states accurately as shown in the top row of Fig. 5.2. The root cause of the issue may be explained by comparing the operating point of the actual states and the estimated states. Ignoring the small variations around the operating points, the estimated states constantly overestimate or underestimate the true states. The operating points are found by solving for the stationary point of dynamical system in Eq. 5.1:

$$\omega^2 v = A\omega \begin{bmatrix} S(w + W^T v) \\ Z^{ext} \end{bmatrix} \tag{5.19}$$

where v denotes the operating point, and w and Z^{ext} denote the noise and external drive operating points respectively. Although not discussed here, the solution of this system of nonlinear equations is highly sensitive to any linearization of the sigmoid function. Therefore, the inaccuracy of the estimate states may be attributed to the use of a linear state estimator. As a sanity check, a linear NMM model is used to generate the bottom row of Fig. 5.2. It is observed that unlike the top row, the state estimate are accurate and close to the actual states. This suggests that a nonlinear state estimator must be used.

5.1.2 Unscented Kalman Filter

In order to use the unscented Kalman filter (UKF) for parameter estimation, the dynamical system of Eq. 5.1 should be rewritten in the following form:

$$x[k] = f(x[k-1], w[k-1], u_s[k-1]) \tag{5.20}$$

$$y[k] = h(x[k], v[k], u_m[k]) \tag{5.21}$$

where x is the state vector, w and v are the process and measurement noise, and u_s and u_m are exogeneous input the the state and measurement equation may depend on. Given the finite-difference approximation of Eq. 5.1:

$$\begin{bmatrix} v(t+1) \\ \\ \phi(t+1) \end{bmatrix} = \begin{bmatrix} v(t) \\ \\ \phi(t) \end{bmatrix} + dt \left(\begin{bmatrix} 0_{N\times N} & I_{N\times N} \\ \\ -\omega^2 & -2\omega \end{bmatrix} \begin{bmatrix} v(t) \\ \\ \phi(t) \end{bmatrix} + \begin{bmatrix} 0_{N\times 1} \\ \\ A\omega \begin{bmatrix} S(w(t) + W^T v(t)) \\ z^{\text{ext}}(t) \end{bmatrix} \end{bmatrix} \right) \tag{5.22}$$

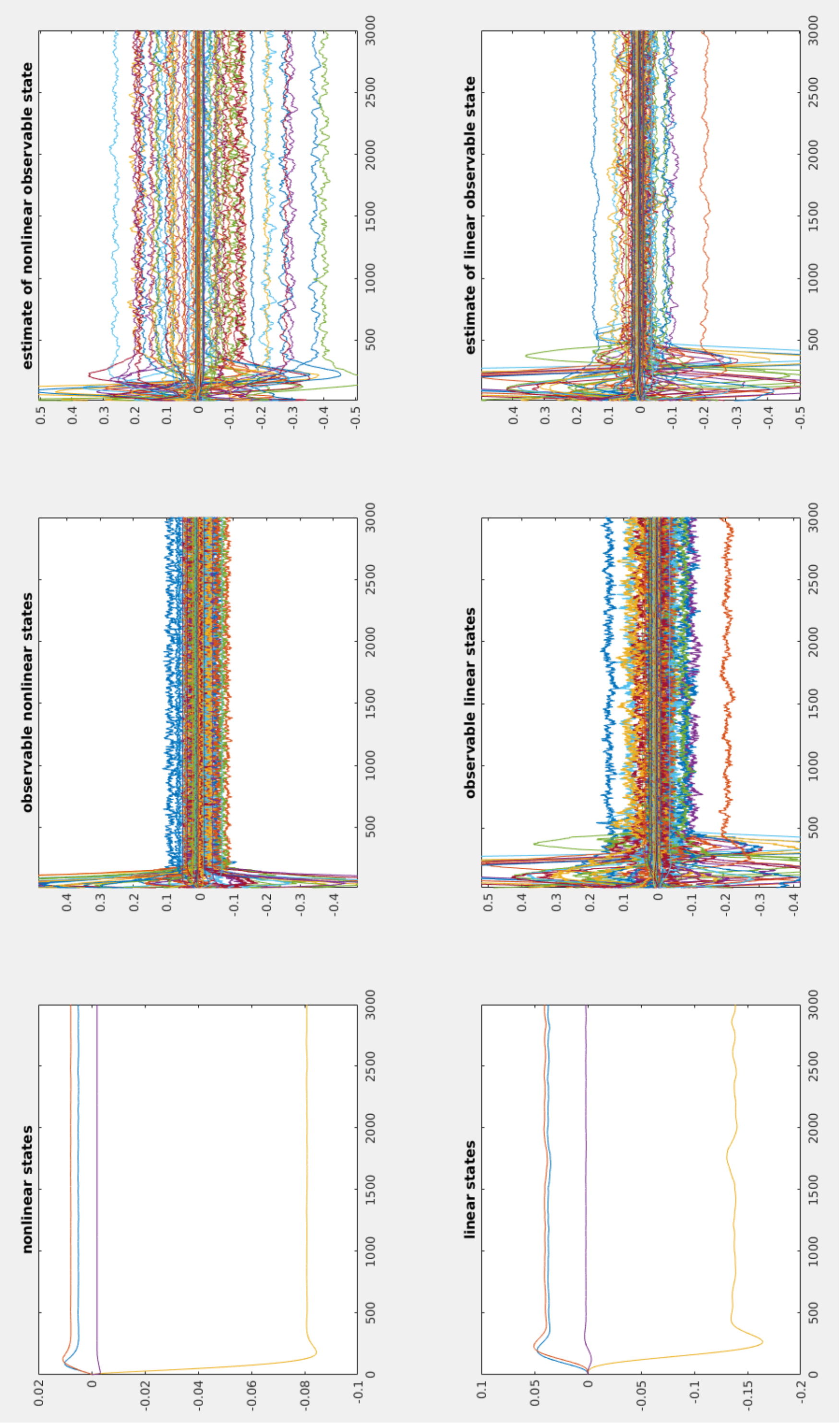


Figure 5.2: Top row: ground truth system is nonlinear. Top left: a subset of the true states. Top middle: the observable states. Top right: The observable states estimated by (E)KF. Bottom row: ground truth system is linear. Bottom left: a subset of the true states. Bottom middle: the observable states. Bottom right: The observable states estimated by KF.

the two functions f and h in Eq. 5.21 are given by:

$$f(x,w) = x + dt\left(\begin{bmatrix} 0_{N\times N} & I_{N\times N} \\ \\ \\ -\omega^2 & -2\omega \end{bmatrix} x + \begin{bmatrix} 0_{N\times 1} \\ \\ A\omega \begin{bmatrix} S(w_{1:N_I} + W^T x_{1:N_I}) \\ Z^{ext} + w_{(N_I+1):N} \end{bmatrix} \end{bmatrix}\right) \tag{5.23}$$

$$h(x,v) = GRW^T x_{1:N} + v \tag{5.24}$$

Fig. 5.3 shows the states estimated by UKF (red) vs. the actual states (blue) for a synthetic system with 10 columns each consisting of 4 different population types. Compared the the results of (E)KF, it is observed that the estimated states are accurate and follow the variations in the unknown state closely.

5.2 The M step

To formualate the M-step, the following steps should be completed:

1. Derivation of the joint Log-likelihood function for the NMM.
2. Calculating the derivative of the Q function.
3. Solving the optimization problem of the M-step (using coordinate descent)

5.2.1 Derivation of the Joint Log-Likelihood Function for the NMM

Let's denote the state vector $[v^T \phi^T]^T$ by x. The MAP parameter estimate requires maximizing the log-likelihood function $\mathcal{L}(Y|W) + \mathcal{L}(W)$ w.r.t the connectivity matrix W. Since

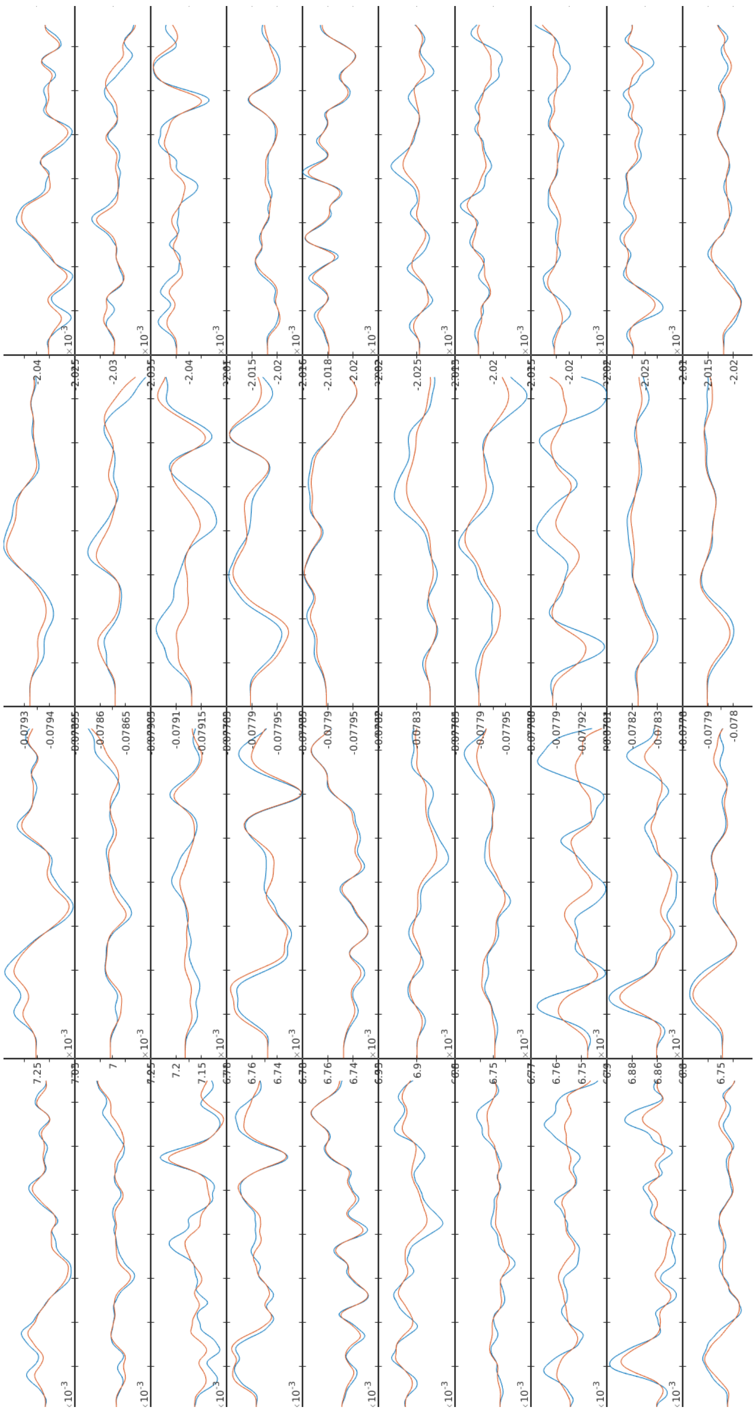


Figure 5.3: The states estimated by UKF (red) vs. the actual states. Each subplot belongs to one population. Each row belongs to one cortical column. Each column corresponds to one of the population types comprising the cortical column (pyramidal, excitatory, slow inhibitory, fast inhibitory).

$\mathcal{L}(Y|W)$ is complicated in W, we resort to expectation maximization which requires calculating the joint log likelihood function $\mathcal{L}(X, Y|W)$. We have:

$$\begin{aligned}\mathcal{L}(Y, X|W) =&\mathcal{L}(x(1)) + \mathcal{L}(y(1)|x(1)) + \mathcal{L}(x(2)|x(1)) + \mathcal{L}(y(2)|x(2)) + \cdots + \\ &\mathcal{L}(x(T)|x(T-1)) + \mathcal{L}(y(T)|x(T))\end{aligned} \tag{5.25}$$

To simplify the derivation, we may modify (augment) the beginning of the chain as $\mathcal{L}(x(1)) \rightarrow \mathcal{L}(x(1)|x(0)) + \mathcal{L}(x(0))$, and then drop the Ł$(x(0))$ as its contribution to the cost function is insignificant for large T:

$$\begin{aligned}\mathcal{L}(Y, X|W) \approx&\mathcal{L}(x(1)|x(0)) + \mathcal{L}(y(1)|x(1)) + \mathcal{L}(x(2)|x(1)) + \mathcal{L}(y(2)|x(2)) + \cdots + \\ &\mathcal{L}(x(T)|x(T-1)) + \mathcal{L}(y(T)|x(T))\end{aligned} \tag{5.26}$$

The $\mathcal{L}(y(t)|x(t))$ are readily calculated given the distribution of the measurement noise. Without loss of geenrality, assume that the measurement noise is temporally white and the covariance matrix is $\sigma_\epsilon^2\mathbf{I}$. Therefore:

$$\mathcal{L}(y(k)|x(k)) = \frac{-1}{2\sigma_\epsilon^2}(y(k) - GRW^Tv(k))^T(y(k) - GRW^Tv(k)) = \frac{-1}{2\sigma_\epsilon^2}\zeta^T(k)\zeta(k) \tag{5.27}$$

The $\mathcal{L}(x(k)|x(k-1))$ terms, however, are more complicated to calculate. This is because even if x starts as a Gaussian random vector, it passes through the nonlinear state equation and its distribution will not be Gaussian anymore. Therefore, writing a tractable analytical form for $\mathcal{L}(x(k)|x(k-1))$ is not possible. Unfortunately, this contradicts the tractability of the Q function which lies at the heart of EM utility. In other words, the EM is only useful if the Q function is tractable enough so the optimization problem can be solved with reasonable accuracy and complexity.

As shown in Section 4.4, the normal EEG activity requires the PSPs to fall within a linear domain of the sigmoid function. Therefore, it may be possible to linearize the sigmoid as $S(x) = a+mx$. The continuous time NMM is then discretized by finite-difference. Applying this approximation to $\mathcal{L}(x(t+1)|x(t))$ results in:

$$\frac{\phi(t+1)-\phi(t)}{dt} + \omega^2 v(t) + 2\omega\phi(t) - \begin{bmatrix} A_I\omega_I(a+mW^Tv(t)) \\ A_E\omega_E Z^{ext} \end{bmatrix} = \begin{bmatrix} A_I\omega_I mw(t)) \\ A_E\omega_E\delta z_{ext}(t) \end{bmatrix} \tag{5.28}$$

Both sides of which are vectors. However, only the top N_I elements include the connectivity matrix and enter the optimization problem. Therefore, the target equation is:

$$\frac{\phi_I(t+1)-\phi_I(t)}{dt} + \omega_I^2 v_I(t) + 2\omega_I\phi_I(t) - A_I\omega_I(a+mW^Tv(t)) = A_I\omega_I mw(t) \tag{5.29}$$

The right hand side is multivariate noise with covariance $\Sigma = A_I\omega_I m P_w m\omega_I^T A_I^T$, where P_w is the covariance of w. Therefore, denoting the left hand side by $\xi(t+1)$, we have:

$$\mathcal{L}(x(t+1)|x(t)) = -\frac{1}{2}\xi^T(t+1)\Sigma^{-1}\xi(t+1) \tag{5.30}$$

Therefore, the W-dependent portion of the joint likelihood function is:

$$\sum_{t=1}^{T} \frac{-1}{2\sigma_\epsilon^2}\zeta^T(t)\zeta(t) - \frac{1}{2}\xi^T(t)\Sigma^{-1}\xi(t) \tag{5.31}$$

We may normalize by the sample count and include the sparsity penalty to obtain:

$$\frac{1}{T}\left(\sum_{t=1}^{T} \frac{-1}{2\sigma_\epsilon^2}\zeta^T(t)\zeta(t) - \frac{1}{2}\xi^T(t)\Sigma^{-1}\xi(t)\right) - \lambda|W|_1 \tag{5.32}$$

5.2.2 Calculating the Derivative of the Q Function

The Q function is found by taking the derivative of the joint log-likelihood function. The derivative of the Q function is then used to solve the optimization problem. Since the order of expectation and derivative operations does not matter, the derivative may be calculated first and the expectation is then applied to the resulting expression. The latter approach is preferred here as it simplifies the derivation.

In general, some elements of the connectivity matrix may be know while other should be estimated from the data. For instance, the connections inside a cortical column may be fixed according to physiological studies, but the connection between cortical columns may be estimated from data. If gradient-based optimization methods such as gradient-descent are used, the full gradient vector w.r.t the unknown elements of W should be calculated. On the other hand, if element-wise optimization methods such as coordinate descent are used, the derivation does not have to distinguish between the known and unknown elements. In other words, a unified analytical solution is derived and is then only applied to the unknown elements. Although we use coordinate descent in this thesis, the full gradient vector is also derived as a groundwork for future work.

Partitioning the Connectivity Matrix into Known and Unknown Parts

In order to seperate the known and unknown parts of the connectivity matrix, the $W^T v$ terms is rewritten in terms of the vectorized version of W. For a generic matrix A and vector x, the product $A_{M\times N} x_{N\times 1}$ may be written as $(x^T \otimes I_M)\text{vec}(A)$. Then, if the index set of the unknown elements of $\text{vec}(A)$ is $\mathcal{I}$, and with $\bar{\mathcal{I}} = \{1 \cdots MN\} - \mathcal{I}$, we can write:

$$(x^T \otimes I_M)\text{vec}(A) = (x^T \otimes I_M)_{:,\mathcal{I}}\text{vec}(A)_{\mathcal{I}} + (x^T \otimes I_M)_{:,\bar{\mathcal{I}}}\text{vec}(A)_{\bar{\mathcal{I}}} \tag{5.33}$$

Applying this to $W^T v$ we may write $W^T v = (v^T \otimes I_{N_I})\text{vec}(W^T)$. We may also separate the known and unknown parts of $vec(W^T)$ and write:

$$W^T v = V_u W_u + V_k W_k \tag{5.34}$$

Using the chain rule and matrix calculus, the derivative w.r.t the unknown part is:

$$\begin{aligned}
\frac{\partial \zeta(t)}{\partial W_u} &= \frac{\partial}{\partial W_u}(y(t) - GRW^T v(t)) = \frac{\partial}{\partial W_u}(-GR[V_u(t)W_u + V_k(t)W_k]) \\
&= -GRV_u(t) && (5.35) \\
\frac{\partial \xi(t+1)}{\partial W_u} &= \frac{\partial}{\partial W_u}(-A_I\omega_I(a + mW^T v(t))) = \frac{\partial}{\partial W_u}(-A_I\omega_I m[V_u(t)W_u + V_k(t)W_k]) \\
&= -A_I\omega_I m V_u(t) && (5.36)
\end{aligned}$$

Now, we can write the derivative of the normalized joint log likelihood function as:

$$\frac{1}{T}\sum_{t=1}^{T}\left(-\zeta^T(t)\left[\frac{1}{\sigma_\epsilon^2}\right][-GRV_u(t)] - \xi^T(t)\left[\Sigma^{-1}\right][-A_I\omega_I m V_u(t-1)]\right) - \lambda\frac{\partial |W_u|_1}{\partial W_u} \tag{5.37}$$

or:

$$\frac{1}{T}\sum_{t=1}^{T}\Bigg(-\left(y(t) - GRV_u(t)W_u - GRV_k(t)W_k\right)^T \left[\frac{1}{\sigma_\epsilon^2}\right]\left[-GRV_u(t)\right] \tag{5.38}$$

$$- \Bigg\{ \frac{\phi_I(t) - \phi_I(t-1)}{dt} + \omega_I^2 v(t-1) + 2\omega_I \phi(t-1)$$

$$- A_I\omega_I(a + mW^T v(t-1))\Bigg\}^T \left[\Sigma^{-1}\right]\left[-A_I\omega_I m V_u(t-1)\right]\Bigg) \tag{5.39}$$

$$-\lambda\frac{\partial |W_u|_1}{\partial W_u} \tag{5.40}$$

Using the following definition:

$$\theta(t) \triangleq \frac{\phi_I(t) - \phi_I(t-1)}{dt} + \omega_I^2 v_I(t-1) + 2\omega_I\phi_I(t-1) - A_I\omega_I a \tag{5.41}$$

, the derivative is rewritten as:

$$\frac{1}{T}\sum_{t=1}^{T}\Bigg(-\left\{y(t) - GRV_u(t)W_u - GRV_k(t)W_k\right\}^T \left[\frac{1}{\sigma_\epsilon^2}\right]\left[-GRV_u(t)\right] \tag{5.42}$$

$$- \left\{\theta(t) - A_I\omega_I m V_u(t-1)W_u - A_I\omega_I m V_k(t-1)W_k\right\}^T \left[\Sigma^{-1}\right]$$

$$\left[-A_I\omega_I m V_u(t-1)\right]\Bigg) \tag{5.43}$$

$$-\lambda\frac{\partial |W_u|_1}{\partial W_u} \tag{5.44}$$

This expression may then be expanded and the expectation is calculated on the expansion. The resulting derivative, however, cannot be directly used for maximization. It may be used in a gradient descent approach. However, as mentiond before, the separation of W into known and unknown parts is not necessary if coordinate descent is used to solve the optimization problem.

Element-Wise Derivative: Groundwork for Coordinate Descent

Let $\mathcal{W}$ denote the vectorized version of W^T. Also, let $s(k)$ and $d(k)$ denote two functions of a linear index of $\mathcal{W}$ that map the linear index into the corresponding row and column index of W such that:

$$\mathcal{W}_k = W_{s(k),d(k)} \tag{5.45}$$

Note that $s(k)$ is indeed the index of the $\underline{s}$ource population and $d(k)$ is the $\underline{d}$estination population.

The derivative of the cost function, denoted by $\partial_{\mathcal{W}}$ is now written as:

$$\partial_{\mathcal{W}} \triangleq \frac{1}{T}\sum_{t=1}^{T-1}\Bigg(- \{y(t) - GRV(t)\mathcal{W}\}^T \left[\frac{1}{\sigma_\epsilon^2}\right] [-GRV(t)] \tag{5.46}$$

$$- \{\theta(t) - A_I\omega_I mV(t-1)\mathcal{W}\}^T \left[\Sigma^{-1}\right] [-A_I\omega_I mV(t-1)]\Bigg) \tag{5.47}$$

$$- \lambda\frac{\partial|\mathcal{W}|_1}{\partial\mathcal{W}} \tag{5.48}$$

where $V(t) \triangleq v^T(t)) \otimes I_{N_I}$, and the identity $W^T v(t) = V\mathcal{W}$ has been used. Note that $\partial_{\mathcal{W}}$ is a row vector.

The expansion of this equation generates four components, the expectation of which should be calculated.

$y^T(t)GRV(t)$: Starting from the definition of V, we have:

$$y^T(t)\tilde{G}V(t) = y^T(t)\tilde{G}(v^T(t) \otimes I_{N_I}) = y^T(t)(v^T(t) \otimes \tilde{G}) \tag{5.49}$$

Moving the expectation around v, we get:

$$y^T(t)[\overline{v(t)}^T \otimes \tilde{G}] \tag{5.50}$$

which expands as:

$$[\bar{v}_0(t)y^T(t)\tilde{G}|\bar{v}_1(t)y^T(t)\tilde{G}|\cdots|\bar{v}_{N-1}(t)y^T(t)\tilde{G}] \tag{5.51}$$

Thus, the k^{th} element will be ($k = 0 \cdots N \times N_I - 1$):

$$\bar{v}_{\lfloor \frac{k}{N_I} \rfloor}(t)y^T(t)\tilde{G}_{:,k\%N_I} \tag{5.52}$$

The term $\lfloor \frac{k}{N_I} \rfloor$ is the source index of the connection, and the modulus $k\%N_I$ is the destination index of the connection. Thus, if the derivative is reordered as a matrix to match the dimensions of W, the derivative of $y^T(t)GRV(t)$ w.r.t. the connection from source $s = 0 \cdots N-1$ to destination $d = 0 \cdots N_I - 1$ is:

$$\bar{v}_s(t)y^T(t)\tilde{G}_{:,d} \tag{5.53}$$

To interpret this in matrix form, the expression should be the element (s, d) of some matrix. Using the definition of matrix multiplication:

$$\bar{v}_s(t)y^T(t)\tilde{G}_{:,d} = [\bar{v}(t)[y^T(t)\tilde{G}]]_{s,d} \tag{5.54}$$

Therefore, the matrix representation of the derivative of $y^T(t)GRV(t)$ is given by $\bar{v}(t)y^T(t)\tilde{G}$.

$\mathcal{W}^T V^T(t) R^T G^T G R V(t)$: Using some Kronecker product identities, the definition of matrix multiplication, the definition of $\tilde{G}$, and the matrix reshaping explained above:

$$\begin{align}
\mathcal{W}^T V^T(t)\tilde{G}^T\tilde{G}V(t) &= \mathcal{W}^T \left(v^T(t)\otimes\tilde{G}\right)^T \left(v^T(t)\otimes\tilde{G}\right) \tag{5.55}\\
&= \mathcal{W}^T \left(v(t)\otimes\tilde{G}^T\right)\left(v^T(t)\otimes\tilde{G}\right) \tag{5.56}\\
&= \mathcal{W}^T \left(\left[v(t)v^T(t)\right]\otimes\left[\tilde{G}^T\tilde{G}\right]\right) \tag{5.57}
\end{align}$$

The k^{th} element of this row vector is now given by:

$$\begin{align}
\left[\mathcal{W}^T V^T(t)\tilde{G}^T\tilde{G}V(t)\right]_k &= \sum_\ell \mathcal{W}_\ell \left[\left[v(t)v^T(t)\right]\otimes\left[\tilde{G}^T\tilde{G}\right]\right]_{\ell,k} \tag{5.58}\\
&= \sum_\ell \mathcal{W}_\ell \left[v_{\lfloor\frac{\ell}{N_I}\rfloor}(t)v_{\lfloor\frac{k}{N_I}\rfloor}(t)[\tilde{G}^T\tilde{G}]_{\ell\%N_I,k\%N_I}\right] \tag{5.59}\\
&= \sum_\ell W_{s(\ell),d(\ell)}\left[v_{s(\ell)}(t)v_{s(k)}(t)\left[\tilde{G}^T\tilde{G}\right]_{d(\ell),d(k)}\right] \tag{5.60}\\
&= \sum_{s',d'} W_{s',d'}\left[v_{s'}(t)v_s(t)\left[\tilde{G}^T\tilde{G}\right]_{d',d}\right] \tag{5.61}\\
&= \sum_{s'}\left(\sum_{d'} W_{s',d'}\left[\tilde{G}^T\tilde{G}\right]_{d',d}\right)v_s(t)v_{s'}(t) \tag{5.62}\\
&= \sum_{s'}\left(W[\tilde{G}^T\tilde{G}]\right)_{s',d} v_s(t)v_{s'}(t) \tag{5.63}\\
&= \sum_{s'}[v_s(t)v_{s'}(t)]\left[W[\tilde{G}^T\tilde{G}]\right]_{s',d} \tag{5.64}\\
&= \left[(v(t)\otimes v^T(t))\left[W[\tilde{G}^T\tilde{G}]\right]\right]_{s,d} \tag{5.65}
\end{align}$$

which is the complete reshapred matrix form of the derivative of $\mathcal{W}^T V^T(t) R^T G^T G R V(t)$.

$\theta^T(t)\Sigma^{-1}A_I\omega_I m V(t-1)$: Using the definition $\Sigma^{-1} = A_I^{-T}\omega_I^{-T}m^{-1}P_w^{-1}m^{-1}\omega_I^{-1}A_I^{-1}$, the expression is equal to $\theta^T(t)\Theta V(t-1)$ with $\Theta \triangleq A_I^{-T}\omega_I^{-T}m^{-1}P_w^{-1}$. Therefore, the derivative is calculated as:

$$\begin{aligned}
\left[\theta^T(t)\Theta V(t-1)\right]_k &= \left[\theta^T(t)\left[v^T(t-1)\otimes\Theta\right]\right]_k = \sum_\ell \theta_\ell(t)\left[v^T(t-1)\otimes\Theta\right]_{\ell,k} \\
&= \sum_\ell \theta_\ell(t) v_{\lfloor \frac{k}{N_I} \rfloor}(t-1)\Theta_{\ell,k\%N_I} = \sum_\ell \theta_\ell(t) v_{s(k)}(t-1)\Theta_{l,d(k)} \\
&= \sum_\ell \left[\theta_\ell(t) v_{s(k)}(t-1)\right]\Theta_{\ell,d(k)} = \sum_\ell \left[v(t-1)\theta^T(t)\right]_{s,\ell}\Theta_{\ell,d(k)} \\
&= \left[v(t-1)\theta^T(t)\Theta\right]_{s,d} \qquad (5.66)
\end{aligned}$$

Then, we can plug in the definition of θ to get:

$$\begin{aligned}
&E\left\{v(t-1)\theta^T(t)\Theta\right\} = E\left\{v(t-1)\theta^T(t)\right\}\Theta \\
&= E\left\{v(t-1)\left[\frac{\phi_I^T(t)}{dt} - \frac{\phi_I^T(t-1)}{dt} + v_I^T(t-1)\omega_I^{2T} + 2\phi_I^T(t-1)\omega_I^T - a^T\omega_I^T A_I^T\right]\right\}\Theta \\
&= \frac{E\left\{v(t-1)\phi_I^T(t)\right\}}{dt}\Theta - \frac{E\left\{v(t-1)\phi_I^T(t-1)\right\}}{dt}\Theta + E\left\{v(t-1)v_I^T(t-1)\right\}\omega_I^{2T}\Theta \\
&\quad + 2E\left\{v(t-1)\phi_I^T(t-1)\right\}\omega_I^T\Theta - E\left\{v(t-1)\right\}a^T\omega_I^T A_I^T\Theta \\
&= \frac{E\left\{v(t-1)\phi_I^T(t)\right\}}{dt}\Theta - \frac{E\left\{v(t-1)\phi_I^T(t-1)\right\}}{dt}\Theta \\
&\quad + E\left\{v(t-1)v_I^T(t-1)\right\}A_I^{-1}\omega_I m^{-1}P_w^{-1} \\
&\quad + 2E\left\{v(t-1)\phi_I^T(t-1)\right\}A_I^{-1}m^{-1}P_w^{-1} - E\left\{v(t-1)\right\}a^T m^{-1}P_w^{-1} \\
&\triangleq \frac{\gamma_1(t)}{dt} - \frac{\gamma_2(t-1)}{dt} + \gamma_3(t-1) + 2\gamma_4(t-1) - \gamma_5(t-1) \qquad (5.67)
\end{aligned}$$

$\mathcal{W}^T V^T(t-1) m^T \omega_I^T A_I^T \Sigma^{-1} A_I \omega_I m V(t-1)$: Using the definition of V and some Kronecker product identities:

$$\begin{aligned}
\mathcal{W}^T V^T(t-1) P_w^{-1} V(t-1) &= \mathcal{W}^T \left(v^T(t-1) \otimes I_{N_I}\right)^T P_w^{-1} \left(v^T(t-1) \otimes I_{N_I}\right)^T \\
&= \mathcal{W}^T \left(v(t-1) \otimes I_{N_I}\right)^T P_w^{-1} \left(v^T(t-1) \otimes I_{N_I}\right)^T \\
&= \mathcal{W}^T \left(v(t-1) \otimes P_w^{-1}\right) \left(v^T(t-1) \otimes I_{N_I}\right) \\
&= \mathcal{W}^T \left(\left[v(t-1) v^T(t-1)\right] \otimes \left[P_w^{-1} I_{N_I}\right]\right) \\
&= \mathcal{W}^T \left(\left[v(t-1) v^T(t-1)\right] \otimes P_w^{-1}\right) \qquad (5.68)
\end{aligned}$$

The k^{th} element of this row vector and the corresponding element in the reshaped matrix is:

$$\begin{aligned}
\left[\mathcal{W}^T \left(\left[v(t-1) v^T(t-1)\right] \otimes P_w^{-1}\right)\right]_k &= \sum_{\ell} \mathcal{W}_\ell \left(v(t-1) v^T(t-1) \otimes P_w^{-1}\right)_{\ell,k} \\
&= \sum_{\ell} \mathcal{W}_\ell \left[v(t-1) v^T(t-1)\right]_{\lfloor \frac{\ell}{N} \rfloor, \lfloor \frac{k}{N} \rfloor} [P_w^{-1}]_{\ell \% N_I, k \% N_I} \\
&= \sum_{\ell} W_{s(\ell),d(\ell)} \left[v(t-1) v^T(t-1)\right]_{s(\ell),s(k)} [P_w^{-1}]_{d(\ell),d(k)} \\
&= \sum_{s',d'} W_{s',d'} \left[v(t-1) v^T(t-1)\right]_{s',s} [P_w^{-1}]_{d',d} \\
&= \sum_{s'} \left[v(t-1) v^T(t-1)\right]_{s',s} \sum_{d'} W_{s',d'} [P_w^{-1}]_{d',d} \\
&= \sum_{s'} \left[v(t-1) v^T(t-1)\right]_{s',s} \left(W P_w^{-1}\right)_{s',d} \\
&= \sum_{s'} \left[v(t-1) v^T(t-1)\right]_{s,s'} \left(W P_w^{-1}\right)_{s',d}
\end{aligned}$$

$$= \left(\left[v(t-1)v^T(t-1)\right] WP_w^{-1}\right)_{s,d} \tag{5.69}$$

Therefore, the matrix form of the derivative of $\mathcal{W}^T V^T(t-1) m^T \omega_I^T A_I^T \Sigma^{-1} A_I \omega_I m V(t-1)$ is given by:

$$v(t-1)v^T(t-1)WP_w^{-1} \tag{5.70}$$

Summary: Putting the componets together, the derivative of the cost function $\partial_{\mathcal{W}}$ is given by:

$$\begin{aligned}\partial_{\mathcal{W}} \; &= \tfrac{-1}{T}\textstyle\sum_{t=1}^{T} \; \Big(E\{\tfrac{-1}{\sigma_\epsilon^2} y^T(t) GRV(t)\} + E\{\tfrac{1}{\sigma_\epsilon^2}\mathcal{W}^T V^T(t) R^T G^T GRV(t)\} \\ &\quad + E\{-\theta^T(t)\Sigma^{-1}A_I\omega_I mV(t-1)\} \\ &\quad + E\{\mathcal{W}^T V^T(t-1) m^T \omega_I^T A_I^T \Sigma^{-1} A_I \omega_I mV(t-1)\} \Big) - \lambda \tfrac{\partial |\mathcal{W}|_1}{\partial \mathcal{W}}\end{aligned}$$

Note that this quantity is an $1 \times NN_I$ row vector. For a more intuitive interpretation in terms of the source and the destination of a connection, let $\mathcal{R}$ denote the reshaping operator that convert the input $1 \times NN_I$ row vector into an $N \times N_I$ matrix by filling out the rows first. With the following definitions:

$$\partial_W \triangleq \mathcal{R}(\partial_{\mathcal{W}}) \tag{5.71}$$

$$\langle x(t) \rangle \triangleq \frac{1}{T}\sum_{t=1}^{T} x(t) \tag{5.72}$$

Term	Definition	Element $(s,d) = \cdots$
$\alpha(t)$	$\bar{v}(t)y^T(t)\tilde{G}$	$y^T(t)\bar{v}_s(t)\tilde{G}_{:,d}$
$\beta(t)$	$\overline{v(t)v^T(t)}W\tilde{G}^T\tilde{G}$	$\sum_{s'=0}^{N-1} W_{s'\to}[\tilde{G}^T\tilde{G}]_{:,d}\overline{v_{s'}(t)v_s(t)}$
$\chi(t)$	$\overline{v(t-1)v^T(t-1)}WP_w^{-1}$	$\sum_{s'=0}^{N-1} W_{s'\to}\,[P_w^{-1}]_{:,d}\,\overline{v_{s'}(t-1)v_s(t-1)}$
$\gamma_1(t)$	$\overline{v(t-1)\phi_I^T(t)}\Theta$	$[\overline{v(t-1)\phi_I^T(t)}]_{s,:}\Theta_{:,d}$
$\gamma_2(t-1)$	$\overline{v(t-1)\phi_I^T(t-1)}\Theta$	$[\overline{v(t-1)\phi_I^T(t-1)}]_{s,:}\Theta_{:,d}$
$\gamma_3(t-1)$	$\overline{v(t-1)v_I^T(t-1)}A_I^{-1}\omega_I m^{-1}P_w^{-1}$	$[\overline{v(t-1)v_I^T(t-1)}]_{s,:}[A_I^{-1}\omega_I m^{-1}P_w^{-1}]_{:,d}$
$\gamma_4(t-1)$	$\overline{v(t-1)\phi_I^T(t-1)}A_I^{-1}m^{-1}P_w^{-1}$	$[\overline{v(t-1)\phi_I^T(t-1)}]_{s,:}[A_I^{-1}m^{-1}P_w^{-1}]_{:,d}$
$\gamma_5(t-1)$	$\overline{v(t-1)}a^T m^{-1}P_w^{-1}$	$[\overline{v(t-1)}]_{s,:}[a^T m^{-1}P_w^{-1}]_{:,d}$

Table 5.2: The definition of variables in Eq. 5.73. The overlines represent the expectation (calculated by UKF).

, the matrix-shaped derivative is given by:

$$\begin{aligned}
\partial_W = & \frac{1}{\sigma_\epsilon^2}\langle\alpha(t)\rangle - \frac{1}{\sigma_\epsilon^2}\langle\beta(t)\rangle \\
& + \frac{1}{dt}\langle\gamma_1(t)\rangle - \frac{1}{dt}\langle\gamma_2(t-1)\rangle + \langle\gamma_3(t-1)\rangle + 2\langle\gamma_4(t-1)\rangle - \langle\gamma_5(t-1)\rangle \\
& - \langle\chi(t-1)\rangle - \lambda\frac{\partial|W|_1}{\partial W}
\end{aligned} \tag{5.73}$$

with the definition of α, β, γ_i, and χ summarized in Table **??**.

Unfortunately, a standard UKF implementation does not return lagged expectations such as $\overline{v(t-1)\phi_I^T(t)}$ which appear in γ_1. To address this, we propose to augment the state vector of the UKF as:

$$x[k] = \begin{bmatrix} v[k] \\ \phi[k] \\ v[k-1] \\ \phi_{[}k-1] \end{bmatrix} \tag{5.74}$$

, for which the state transition equation is modified as in Eq. 5.75:

$$x_{4N\times 1}[k] = \begin{bmatrix} \begin{bmatrix} v[k-1] \\ \phi[k-1] \end{bmatrix} + dt\left(\begin{bmatrix} 0_N & I_N \\ -\omega^2 & -2\omega \end{bmatrix} \begin{bmatrix} v[k-1] \\ \phi[k-1] \end{bmatrix} + \begin{bmatrix} 0_N \\ A\omega \begin{bmatrix} S(w[k-1] + W^T v[k-1]) \\ z^{\text{ext}}[k-1] \end{bmatrix} \end{bmatrix} \right) \\ x_{1:2N}[k-1] \end{bmatrix}$$

$$= \begin{bmatrix} x_{1:2N}[k-1] + dt\left(\begin{bmatrix} 0_N & I_N \\ -\omega^2 & -2\omega \end{bmatrix} x_{1:2N}[k-1] + \begin{bmatrix} 0_{N\times 1} \\ A\omega \begin{bmatrix} S(w[k-1] + W^T x_{1:N}[k-1]) \\ z^{\text{ext}}[k-1] \end{bmatrix} \end{bmatrix} \right) \\ x_{1:2N}[k-1] \end{bmatrix} \tag{5.75}$$

5.2.3 Solving the Optimization Problem of the M-step Using Coordinate Descent

Given the derivative of the cost function w.r.t individual connectivity elements, we can now use coordinate descent to maximize the Q function iteratively. For $W_{s,d}$, the negated subdifferential will be linear in $W_{s,d}$ with an extra sign function (see Section 2.7):

$$-\partial_{W_{s,d}} = R_{s,d}W_{s,d} + S_{s,d} + \lambda \text{sign}(W_{s,d}) \tag{5.76}$$

The solution of zero-derivative is given by:

$$\text{sign}(S_{s,d}) \min\left(0, \frac{\lambda - |S_{s,d}|}{R_{s,d}}\right) \tag{5.77}$$

The only terms in the derivative that depend on $W_{s,d}$ are β and χ. Therefore:

$$\begin{aligned}-\partial_{W_{s,d}} = &-\frac{1}{\sigma_\epsilon^2}\langle\alpha_{s,d}(t)\rangle + \frac{1}{\sigma_\epsilon^2}\langle\beta_{s,d}(t)\rangle - \frac{1}{dt}\langle[\gamma_1(t)]_{s,d}\rangle + \frac{1}{dt}\langle[\gamma_2(t-1)]_{s,d}\rangle \\ &-\langle[\gamma_3(t-1)]_{s,d}\rangle - 2\langle[\gamma_4(t-1)]_{s,d}\rangle + \langle[\gamma_5(t-1)]_{s,d}\rangle \\ &+\langle\chi_{s,d}(t-1)\rangle + \lambda\text{sign}(W_{s,d})\end{aligned} \tag{5.78}$$

If we denote the independence from and dependence on $W_{s,d}$ by superscript $-sd$ and $+sd$

respectively, then:

$$\begin{aligned}
-\partial_{W_{s,d}} =& -\frac{1}{\sigma_\epsilon^2}\left\langle \alpha_{s,d}(t)\right\rangle + \frac{1}{\sigma_\epsilon^2}\left\langle \beta_{s,d}(t)\right\rangle^{+sd} + \frac{1}{\sigma_\epsilon^2}\left\langle \beta_{s,d}(t)\right\rangle^{-sd} - \frac{1}{dt}\left\langle [\gamma_1(t)]_{s,d}\right\rangle \\
&+ \frac{1}{dt}\left\langle [\gamma_2(t-1)]_{s,d}\right\rangle - \left\langle [\gamma_3(t-1)]_{s,d}\right\rangle - 2\left\langle [\gamma_4(t-1)]_{s,d}\right\rangle + \left\langle [\gamma_5(t-1)]_{s,d}\right\rangle \\
&+ \left\langle \chi_{s,d}(t-1)\right\rangle^{+sd} + \left\langle \chi_{s,d}(t-1)\right\rangle^{-sd} + \lambda \text{sign}(W_{s,d}) \\
=&\Bigg(-\frac{1}{\sigma_\epsilon^2}\left\langle \alpha_{s,d}(t)\right\rangle + \frac{1}{\sigma_\epsilon^2}\left\langle \beta_{s,d}(t)\right\rangle^{-sd} - \frac{1}{dt}\left\langle [\gamma_1(t)]_{s,d}\right\rangle + \frac{1}{dt}\left\langle [\gamma_2(t-1)]_{s,d}\right\rangle \\
&- \left\langle [\gamma_3(t-1)]_{s,d}\right\rangle - 2\left\langle [\gamma_4(t-1)]_{s,d}\right\rangle + \left\langle [\gamma_5(t-1)]_{s,d}\right\rangle + \left\langle \chi_{s,d}(t-1)\right\rangle^{-sd} \Bigg) + \\
&\left(\frac{1}{\sigma_\epsilon^2}\left\langle \beta_{s,d}(t)\right\rangle^{+sd} + \left\langle \chi_{s,d}(t-1)\right\rangle^{+sd} \right) + \lambda \text{sign}(W_{s,d}) \\
=& S_{s,d} + R_{s,d}W_{s,d} + \lambda \text{sign}(W_{s,d}) \qquad (5.79)
\end{aligned}$$

Starting from the definition of β and χ, and using the definition of matrix multiplication, we can decompose as:

$$\left\langle \chi_{s,d}(t-1)\right\rangle^{-sd} = \sum_{(s',d')\neq(s,d)} \left\langle \overline{v_s(t-1)v_{s'}(t-1)} \right\rangle W_{s',d'}[P_w^{-1}]_{d',d} \qquad (5.80)$$

$$\left\langle \chi_{s,d}(t-1)\right\rangle^{+sd} = \left\langle \overline{v_s(t-1)v_s(t-1)} \right\rangle W_{s,d}[P_w^{-1}]_{d,d} \qquad (5.81)$$

$$\left\langle \beta_{s,d}(t)\right\rangle^{-sd} = \sum_{(s',d')\neq(s,d)} \left\langle \overline{v_s(t)v_{s'}(t)} \right\rangle W_{s',d'}[\tilde{G}^T\tilde{G}]_{d',d} \qquad (5.82)$$

$$\left\langle \beta_{s,d}(t)\right\rangle^{+sd} = \left\langle \overline{v_s(t)v_s(t)} \right\rangle W_{s,d}[\tilde{G}^T\tilde{G}]_{d,d} \qquad (5.83)$$

The current values of the elements of the connectivity matrix other than $W_{s,d}$ are used to update the estimate of $W_{s,d}$. We may add and subtract the current value of the target element $W_{s,d}$ to enable a more efficient batch update of the elements by getting rid of the $\neq$ in the summations above. Note that we should distinguish between $W_{s,d}$ being a variable to optimize over, and $W_{s,d}^{(j)}$ being the current guess for $W_{s,d}$:

$$\begin{aligned}
\langle \chi_{s,d}(t-1)\rangle^{-sd} &= [\left\langle \overline{v(t-1)v^T(t-1)} \right\rangle^{(j)} W^{(j)} P_w^{-1}]_{s,d} \\
&- \left\langle \overline{v_s(t-1)v_s(t-1)} \right\rangle^{(j)} W_{s,d}^{(j)} [P_w^{-1}]_{d,d} \\
&= [\left\langle \overline{v(t-1)v^T(t-1)} \right\rangle^{(j)} W^{(j)} P_w^{-1}]_{s,d} \\
&- \left[Diag(\left\langle \overline{v(t-1)v^T(t-1)} \right\rangle^{(j)}) W^{(j)} Diag(P_w^{-1}) \right]_{s,d} \qquad (5.84) \\
\langle \beta_{s,d}(t)\rangle^{-sd} &= \left[\left\langle \overline{v(t)v^T(t)} \right\rangle^{(j)} W^{(j)} \tilde{G}^T \tilde{G} \right]_{s,d} \\
&- \left\langle \overline{v_s(t)v_s(t)} \right\rangle^{(j)} W_{s,d}^{(j)} [\tilde{G}^T \tilde{G}]_{d,d} \\
&= \left[\left\langle \overline{v(t)v^T(t)} \right\rangle^{(j)} W^{(j)} \tilde{G}^T \tilde{G} \right]_{s,d} \\
&- \left[Diag(\left\langle \overline{v(t)v^T(t)} \right\rangle^{(j)}) W^{(j)} Diag(\tilde{G}^T \tilde{G}) \right]_{s,d} \qquad (5.85)
\end{aligned}$$

where *Diag* returns a diagonal matrix consisting of the diagonal elements of the input.

In summary, the following steps are taken at iteration $j+1$ of the EM algorithm:

1) At iteration $j+1$, choose an (s,d) to update.

2) For the selected (s,d), and using the current value of parameters (from step j), calculate the following:

$$< \alpha_{s,d}^{(j)} >, < \beta_{s,d}^{(j)} >^{-sd}, < \chi_{s,d}^{(j)} >^{-sd}, < \gamma_{1\cdots5}^{(j)} > \qquad (5.86)$$

3) Calculate $S^{(j)}_{s,d}$:

$$
\begin{aligned}
S^{(j)}_{s,d} =& \Bigg(-\frac{1}{\sigma_\epsilon^2}\left\langle \alpha^{(j)}_{s,d}(t)\right\rangle + \frac{1}{\sigma_\epsilon^2}\left\langle \beta^{(j)}_{s,d}(t)\right\rangle^{-sd} - \frac{1}{dt}\left\langle [\gamma_1^{(j)}(t)]_{s,d}\right\rangle + \frac{1}{dt}\left\langle [\gamma_2^{(j)}(t-1)]_{s,d}\right\rangle \\
& - \left\langle [\gamma_3^{(j)}(t-1)]_{s,d}\right\rangle - 2\left\langle [\gamma_4^{(j)}(t-1)]_{s,d}\right\rangle + \left\langle [\gamma_5^{(j)}(t-1)]_{s,d}\right\rangle + \left\langle \chi^{(j)}_{s,d}(t-1)\right\rangle^{-sd} \Bigg) \\
=& -\frac{1}{\sigma_\epsilon^2}\left[\left\langle \bar{v}(t)y^T(t)\right\rangle^{(j)}\tilde{G}\right]_{s,d} \\
& + \frac{1}{\sigma_\epsilon^2}\left[\left\langle \overline{v(t)v^T(t)}\right\rangle^{(j)} W^{(j)}\tilde{G}^T\tilde{G} - Diag(\left\langle \overline{v(t)v^T(t)}\right\rangle^{(j)})W^{(j)}Diag(\tilde{G}^T\tilde{G})\right]_{s,d} \\
& - \frac{1}{dt}\left[\left\langle \overline{v(t-1)\phi_I^T(t)}\right\rangle^{(j)} A_I^{-1}\omega_I^{-1}m^{-1}P_w^{-1}\right]_{s,d} \\
& + \frac{1}{dt}\left[\left\langle \overline{v(t-1)\phi_I^T(t-1)}\right\rangle^{(j)} A_I^{-1}\omega_I^{-1}m^{-1}P_w^{-1}\right]_{s,d} \\
& - \left[\left\langle \overline{v(t-1)v_I^T(t-1)}\right\rangle^{(j)} A_I^{-1}\omega_I m^{-1}P_w^{-1}\right]_{s,d} \\
& - 2\left[\left\langle \overline{v(t-1)\phi_I^T(t-1)}\right\rangle^{(j)} A_I^{-1}m^{-1}P_w^{-1}\right]_{s,d} \\
& + \left[\left\langle \bar{v}(t-1)\right\rangle^{(j)} a^T m^{-1}P_w^{-1}\right]_{s,d} \\
& + \Bigg[\left\langle \overline{v(t-1)v^T(t-1)}\right\rangle^{(j)} W^{(j)}P_w^{-1} \\
& \qquad - Diag(\left\langle \overline{v(t-1)v^T(t-1)}\right\rangle^{(j)})W^{(j)}Diag(P_w^{-1})\Bigg]_{s,d}
\end{aligned}
$$

4) Calculate $R^{(j)}_{s,d}$ (derived from β^{+sd} and χ^{+sd}):

$$R^{(j)}_{s,d} = \frac{1}{\sigma^2_\epsilon} < \overline{v_s(t)v_s(t)} >^{(j)} [\tilde{G}^T\tilde{G}]_{d,d} + < \overline{v_s(t-1)v_s(t-1)} >^{(j)} [P_w^{-1}]_{d,d} \tag{5.87}$$

$$= \frac{1}{\sigma^2_\epsilon} \left[diag(< \overline{v(t)v^T(t)} >^{(j)}) diag^T(\tilde{G}^T\tilde{G}) \right]_{s,d} + \left[diag(< \overline{v(t-1)v^T(t-1)} >^{(j)}) diag^T(P_w^{-1}) \right]_{s,d} \tag{5.88}$$

where *diag* return a column vector consisting of the diagonal elements of the input matrix.

5) Update $W_{s,d}$ (other elements remain unchanged, but will change in the following iterations):

$$W_{s,d} \leftarrow \operatorname{sign}(S^{(j)}_{s,d}) \min \left(0, \frac{\lambda - |S^{(j)}_{s,d}|}{R^{(j)}_{s,d}} \right) \tag{5.89}$$

Complexity Reduction

In order to update multiple parameters in a single iteration (i.e. batch update), a subset of the matrices representing α, β, χ, and γ_i are selected using the matrix definitions in Table 5.2. Furthermore, since many multiplication operations overlap across the iterations due to the presence of matrix multiplications in $S^{(j)}$ and $R^{(j)}$, the complexity may be reduced by (partially or fully) storing the individual elements of the summations as well as the sum itself, and then update the sums incrementally.

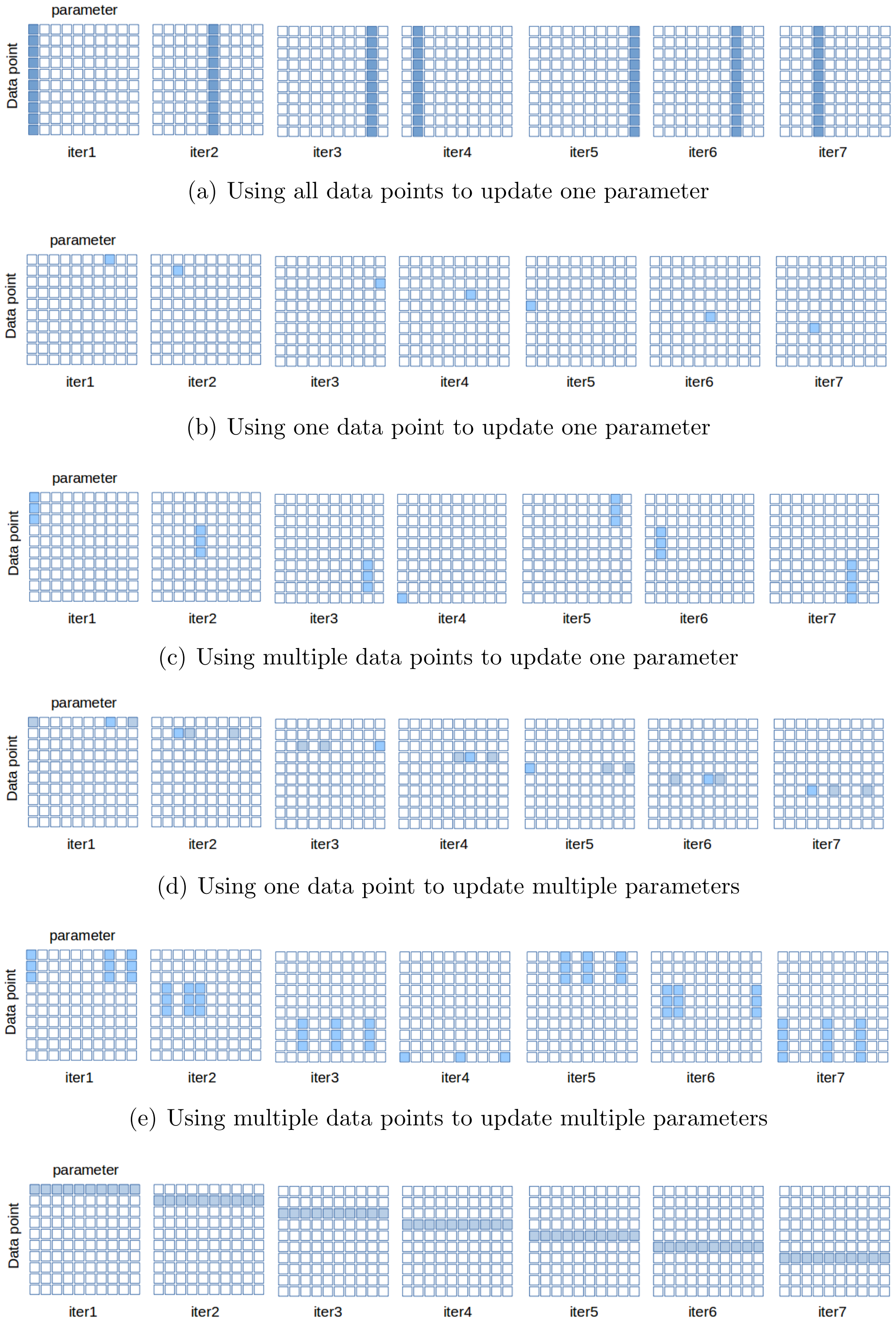
The concept of stochastic EM may be applied as well to reduce the computational complexity and enable an online version of the algorithm. With the current formulation, the parameters are fixed and the UKF is executed over the *entire dataset* to calculate the $\langle . \rangle$s. Then, the $\langle . \rangle$s are used to update a *single element*. Since the number of time samples may be very high, and since the EM is inherently iterative, the current updated value of a parameter might be overwritten in a later iteration, effectively wasting the significant resources used for the last update using the entire dataset.

To resolve this, one possibility is to start from the beginning of the dataset and stop as soon as the temporal averages converge. However, we will end up using a portion of the data over and over, and the algorithm is yet offline. In stochastic EM, the idea is to *use a fraction of the data to update a fraction of the coefficients.*

The two extremes are a) using all the data to update just one parameter, which is a waste of data and computation resource, and b) using a single data point to update all the parameters, which overemphasizes the voting power of the individual data sample and leads to inaccurate parameter values and EM divergence. Intermediate approaches are also possible as shown in Fig. 5.4. Rather than using har boundaries between the time samples used in each iteration, another method to update the $\langle . \rangle$s is to incorporate the most recent sample into the current $\langle . \rangle$ using a forgetting factor $0 \leq \kappa \leq 1$:

$$\langle \rangle_t = \kappa d_t + (1 - \kappa) \langle \rangle_{t-1} \tag{5.90}$$

The choice of the batch size (i.e. the degree of parallelism), and the number of data points consumed in one iteration, also referred to as consensus, affects the both the speed and quality of convergence. Specifically, a higher consensus means a higher complexity per iteration, but lower convergence jitter across iterations. Also, while a higher batch size can increase convergence speed, batch sizes larger than a threshold lead to the divergence of the EM. The trade-off is shown in Fig. 5.5.



(a) Using all data points to update one parameter

(b) Using one data point to update one parameter

(c) Using multiple data points to update one parameter

(d) Using one data point to update multiple parameters

(e) Using multiple data points to update multiple parameters

(f) Using one data point to update all parameters

Figure 5.4: Different combinations of batch processing and stochastic EM

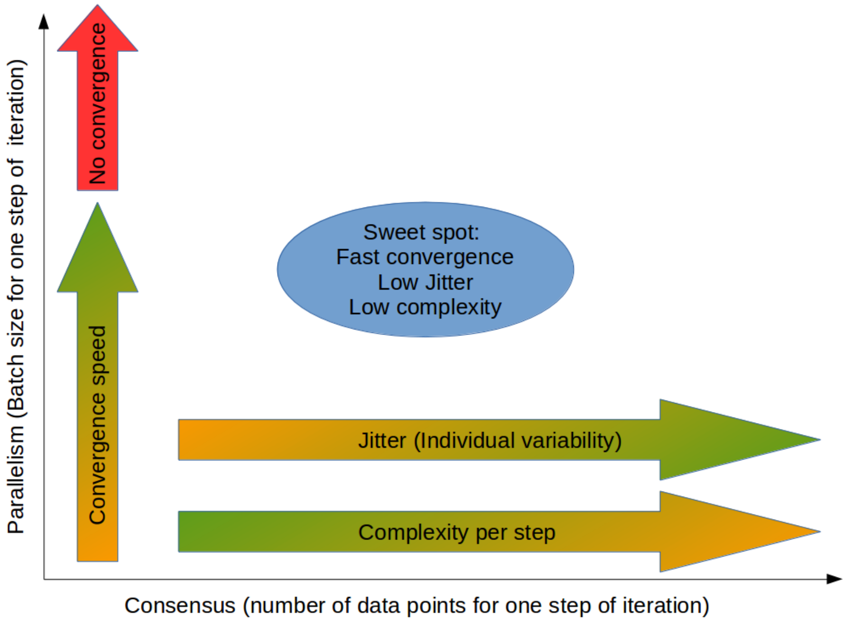

No convergence
Convergence speed
Sweet spot:
Fast convergence
Low Jitter
Low complexity
Jitter (Individual variability)
Complexity per step
Parallelism (Batch size for one step of iteration)
Consensus (number of data points for one step of iteration)

Figure 5.5: Choosing the right batch-size and consensus.

Chapter 6

Evaluation of NMM Sparse Connectivity Estimation

In this chapter, the NMM connectivity estimation algorithm proposed in Chapter 5 is evaluated on a small-dimensional problem with only one connection. The algorithm is found to be insensitive to the connection source. The cost function is further examined to find the root cause of the issue. We propose an approach based on the decomposition of the original cost function to address the source-insensitivity issue. The performance of the proposed algorithm and the possible shortcomings are to be investigated in future work.

6.1 The Source-Insensitivity Problem

In order to evaluate the method, the algorithm was tested on a simple configuration. Specifically, only one connections was included in the ground truth network of N_C cortical columns. Each column was internally driven by a white noise source that biases the input of the pyramidal population at 1 milivolts. The LFM was set equal to identity matrix to simplify debugging the results. The algorithm was tested on different problem dimensions (N_C) and with different ground-truth connection configuration.

Regardless of the problem dimension and connectivity configuration, the algorithm was observed to be accurate in determining the destination of the connection, but it was insensitive to the source of the connection. In other words, while different runs of the algorithm for the same system led to the same inferred connection destination, the inferred connection source varied with each simulation.

To find the root cause of this problem, consider a network of cortical columns with a connection of strength 300 from column 1 to column 2. The UKF is constructed based on an initial guess of very small connections. Fig.6.1 compares the true state (v) and the estimated state for a simulation of 660 miliseconds. A vertical magnification of the second row is provided in Fig. 6.2 .

First, it is observed that the state estimate for the cortical column 2 with underestimated incoming connectivity catches up with the true state after a transient time dictated by the time constants. Second, it is observed that while the existence of an incoming connection significantly affects the operating point of the destination column, it has little influence on the strength of the variations around the operating point.

6.2 Qualitative Justification of the Results

The source-insensitive behvaiour of the parameter estimation algorithm is qualitatively justified as follows. The Q function in Eq. 5.32, excluding the ℓ_1 penalty, is essentially a W-parameterized weighted combination of expected process and measurement innovations

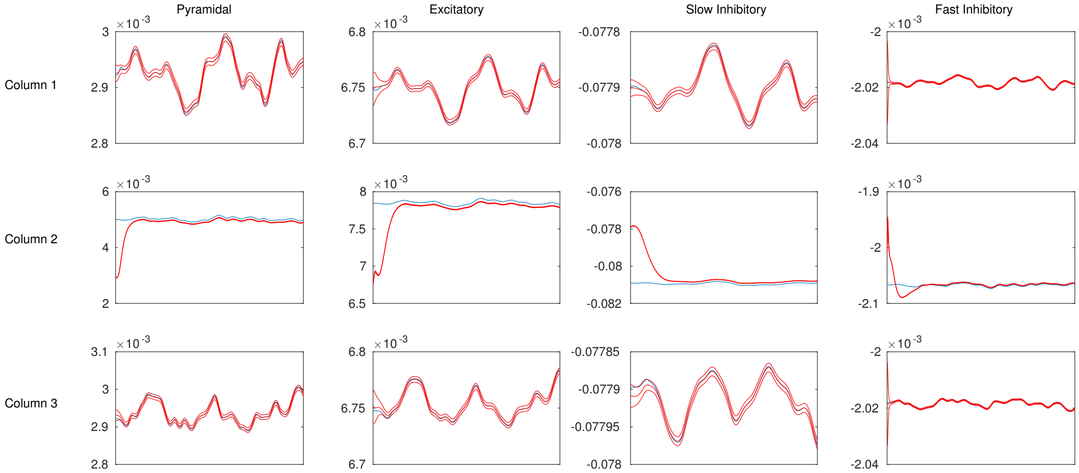


Figure 6.1: The true unknown state (blue) vs. the estimated state mean and confidence interval (red). There is a connection of strength 300 from cortical column 1 to column 2. The network is initially postulated to have weak connections.

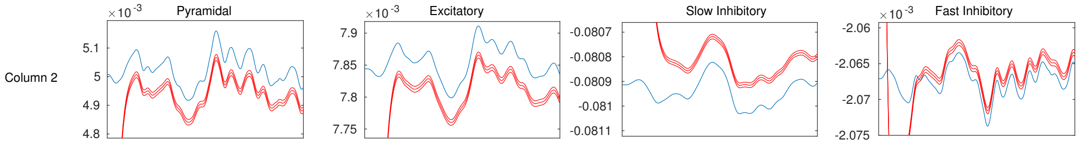


Figure 6.2: Vertical magnification of the second row in Fig. 6.1.

under current parameter values $W^{(j)}$:

$$Q^j(W) = \frac{-1}{2\sigma_\epsilon^2}\left\langle \overline{\zeta^T(t;W)\zeta(t;W)}^{(j)} \right\rangle_t - \frac{1}{2}\left\langle \overline{\xi^T(t)\Sigma^{-1}\xi(t)}^{(j)} \right\rangle_t - \lambda|W|_1 \tag{6.1}$$

which is merely a tractable lower bound on the actual complicated cost function (i.e. the parameter posterior). Based on Eq. 6.1 and the observations above, multiple scenarios may lead to incorrect parameter estimates.

First, similar to any other EM-based parameter estimation, the EM may be trapped in a local minimum which is far from the actual parameter. The more complex the actual cost function, the higher the probability of getting stuck in an undesired local extermum.

Second, the numerical range of $1/2\sigma_\epsilon^2$ and Σ^{-1} may be very different, leading to an unbalanced importance weighting of the process and measurement innovations.

Finally, and most importantly, the mapping from the space of operating points to the space of connectivity matrices is not unique. Roughly speaking, an optimal W is one that leads to small innovation powers $||\zeta||$ and $||\xi||$. As shown in Fig. 6.1, the large deviation between expected measurement and the actual measurement results in a $\langle E_j||\zeta(t;W^j)||\rangle$ that is significantly larger than $\langle E_j||\xi(t;W^j)||\rangle$. In this case, choosing a W that corrects the operating point of the state will significantly reduce $\langle E_j||\zeta(t;W^j)||\rangle$ and thus the cost function. Unfortunately, adding a connection from any of the other columns will satisfy this requirement with negligible difference in terms of cost function improvement. As discussed previously, the source of the connection may only distinctively affect the variations of the destination activity around the operating point but not the operating point itself. Since the variations are numerically much smaller than the operating point, the improvement in the overall cost function will be insensitive to the selected source. Furthermore, once a strong connection is established by coordinate descent, the subsequent updates will only introduce minor changes to other connections as the errors $\langle E_j||\xi(t;W^j)||\rangle$ and $\langle E_j||\zeta(t;W^j)||\rangle$ have already been diminished by the first established connection.

6.3 Quantitative Analysis and Potential Solutions

To quantify the root cause of the problem, note that with a diagonal P_w, both $||\zeta(t)||^2$ and $\xi^T(t)\Sigma^{-1}\xi(t)$ are essentially weighted summations over the individual elements of the square of the elements of the vectors ξ and ζ:

$$\zeta^T\zeta = \sum_{m=1}^{M}\zeta_m^2 \tag{6.2}$$

$$\xi^T\Sigma^{-1}\xi = \sum_{n_I=1}^{N_I}[\Sigma^{-1}]_{n_I,n_I}\xi_{n_I}^2 \tag{6.3}$$

where M is the measurement dimension, N_I is the number of internal populations (that may receive connections). Each individual scalar term ξ_n and ζ_m is itself a function of the connectivity parameters, where the function is a quadratic in elements of W augmented with an ℓ_1 penalty. Let's ignore the weights and focus on one of these individual terms.

Using the values at iteration j, and dropping the time dependence momentarily, we can emphasize the quadratic dependence on the connectivity paremters as follows:

$$\begin{aligned}
\zeta_m^{2^{(j)}} =& \overline{(y_m - [\tilde{G}W^Tv]_m)^2}^{(j)} \\
=& y_m^2 + \overline{[\tilde{G}W^Tv]_m^2}^{(j)} - \overline{2y_m[\tilde{G}W^Tv]_m}^{(j)} \\
=& y_m^2 + \overline{(\tilde{G}_{m,:}[W^Tv])^2}^{(j)} - 2y_m[\tilde{G}W^T\overline{v}^{(j)}]_m \\
=& y_m^2 + \overline{(\sum_{n_i=1}^{N_I}\tilde{G}_{m,n_i}\sum_{n=1}^{N}[W^T]_{n_i,n}v_n)^2}^{(j)} \quad - 2y_m(\sum_{n_i=1}^{N_I}\tilde{G}_{m,n_i}\sum_{n=1}^{N}[W^T]_{n_i,n}\overline{v_n}^{(j)}) \\
=& y_m^2 + \overline{(\sum_{n_i=1}^{N_I}\tilde{G}_{m,n_i}\sum_{n=1}^{N}W_{n,n_i}v_n)^2}^{(j)} \quad - 2y_m(\sum_{n_i=1}^{N_I}\tilde{G}_{m,n_i}\sum_{n=1}^{N}W_{n,n_i}\overline{v_n}^{(j)})
\end{aligned}$$

$$
\begin{aligned}
&=y_m^2 + \sum_{n,n_i}\sum_{n',n_i'} \left[\tilde{G}_{m,n_i}\tilde{G}_{m,n_i'}\overline{v_n v_{n'}}^{(j)}\right] W_{n,n_i} W_{n',n_i'} \\
&\quad - 2y_m \sum_{n,n_i} \left[\tilde{G}_{m,n_i}\tilde{G}_{m,n_i'}\overline{v_n}^{(j)}\right] W_{n,n_i}
\end{aligned} \tag{6.4}
$$

A similar equation may be derived for $\xi_{n_I}^2$. With the addition of temporal averaging, the last equation changes to:

$$
\begin{aligned}
\left\langle \zeta_m^{2(j)}(t) \right\rangle = \; \langle y_m^2(t)\rangle \; & + \sum_{n,n_i}\sum_{n',n_i'} \left[\tilde{G}_{m,n_i}\tilde{G}_{m,n_i'} \left\langle \overline{v_n(t) v_{n'}(t)}^{(j)} \right\rangle\right] W_{n,n_i} W_{n',n_i'} \\
& -2\sum_{n,n_i} \left[\tilde{G}_{m,n_i}\tilde{G}_{m,n_i'} \left\langle y_m(t)\overline{v_n(t)}^{(j)} \right\rangle\right] W_{n,n_i}
\end{aligned} \tag{6.5}
$$

Several important conclusions are drawn from the last equation:

6.3.1 Imbalanced Curvatures

Each m leads to a quadratic function in W with different *curvatures.* The curvature of a quadratic is defined as the trace of the eignevalue matrix of the Jacobian associated with the quadratic. If two qudratics with curvatures c_1 and c_2 are added together, and if $c_1 \gg c_2$, a gradient descent or coordinate descent on the resulting quadratic is far more in favor of the quadratic with larger curvature. In other words, any stepping approach will essentially ignore the small curvature quadratic as it seems flat compared to the large-curvature quadratic. Consequently, if a sum (over m) of quadratics is to be used as the final cost function, and if all individual cost functions carry information to improve the parameter estimate, the individual quadratics should be properly scaled so that gradient or coordinate descent converge to better parameter estimates. Alternatively, a single m (or n_i in case of process innovation) may be picked to form the cost function; the descent algorithm then taken a *small* step to ensure that the subsequent parameter updates can counteract any possible erroneous corrections.

6.3.2 Ill-conditioned Jacobian

The rank and the condition number of the Jacobian determine if the quadratic function has a well defined minimum, or if different linear combinations of the weights can yield similar cost function improvements. This is especially problematic for the example scenario discussed at the beginning of this section and directly result in the unidentifiability of the connection source. To see why, note that the Jacobian matrix is an $NN_I \times NN_I$ matrix as a vectorized version of W should be used. Let $s(.)$ and $d(.)$ denote the functions that map the linear index of $\text{vec}(W)$ to the 2D indices of W. The element (k, k') of the Jacobian is then given by:

$$\begin{aligned} J_{k,k'}(\left\langle \overline{\zeta_m^{2(j)}(t)}^{(j)} \right\rangle) =& \frac{\partial^2}{\partial W_{s(k),d(k)} \partial W_{s(k'),d(k')}} \left\langle \overline{\zeta_m^{2(j)}(t)}^{(j)} \right\rangle \\ =& \tilde{G}_{m,d(k)} \tilde{G}_{m,d(k')} \left\langle \overline{v_{s(k)}(t) v_{s(k')}(t)}^{(j)} \right\rangle \end{aligned} \tag{6.6}$$

Consider the element m of the measurement vector that measures the PSP of population n (as determined by the identity LFM). Based on Eq. 6.6, all elements (k, k') of the Jacobian matrix that satisfy $d(k) = n$ and $d(k') = n$ will be nonzero because of the term $G_{m,d(k)}\tilde{G}_{m,d(k')}$. Since the function $d(k)$ does not provide a one-to-one mapping from the input to the output, the resulting Jacobian matrix will consist of a grid of nonzero elements interspersed by zero elements. Therefore, the Jacobian will be rank deficient and infinitely many different combinations can lead to the same minimal cost. Although the ℓ_1 norm can help find a unique solution out of this infinite pool of solutions, there is no guarantee that the unique solution is associated with the ground truth because ℓ_1 by itself cannot incorporate the information available in the other individual cost functions (e.g. a cost function associated with a $\zeta_{m' \neq m}$, or even the cost function associate with an element of process innovation ξ.) In conclusion, while ill-conditioned individual cost functions can contribute to solving the optimization problem, they should not be the only source consulted by the descent algorithm. Other more informative cost functions, such as those sensitive to the activity waveform, should definitely be used by the descent algorithm.

6.3.3 Low Dispersion

The dispersion of a distribution is defined as the ratio of the variance to the mean. The low statistical dispersion of the estimated time series imposes another challenge on the optimization problem. As an example, consider the second order moment $E_j\{v_n(t)v_{n'}(t)\}$ that appears in the cost function associated with an element of ζ. In terms of the output of the UKF:

$$E_j\{v_n(t)v_{n'}(t)\} = Cov_j(v_n(t), v_{n'}(t)) + E_j\{v_n(t)\}E_j\{v_{n'}(t)\} \tag{6.7}$$

where $Cov(.,.)$ is the covariance of the two scalar inputs. As discussed before and shown with an Example in Fig. 6.1, the estimated covariances are numerically much smaller than the mean values (low dispersion). To see how it affects the optimization output, let's expand the individual cost function in Eq. 6.5 using Eq. 6.7:

$$\begin{aligned}\left\langle \zeta_m^{2^{(j)}}(t) \right\rangle = \quad \langle y_m^2(t)\rangle \quad & + \textstyle\sum_{n,n_i}\sum_{n',n'_i}\left[\tilde{G}_{m,n_i}\tilde{G}_{m,n'_i}\left\langle Cov_j(v_n(t), v_{n'}(t))\right\rangle\right] W_{n,n_i}W_{n',n'_i} \\ & + \textstyle\sum_{n,n_i}\sum_{n',n'_i}\left[\tilde{G}_{m,n_i}\tilde{G}_{m,n'_i}\left\langle E_j\{v_n(t)\}E_j\{v_{n'}(t)\}\right\rangle\right] W_{n,n_i}W_{n',n'_i} \\ & -2\textstyle\sum_{n,n_i}\left[\tilde{G}_{m,n_i}\tilde{G}_{m,n'_i}\left\langle y_m(t)E_j\{v_n(t)\}\right\rangle\right] W_{n,n_i}\end{aligned} \tag{6.8}$$

As was shown in Fig. 6.1, the covariance term $Cov_j(v_n(t), v_{n'}(t))$ is much smaller than the square mean term $E_j\{v_n(t)\}E_j\{v_{n'}(t)\}$. Each individual cost function such as $\left\langle \zeta_m^{2^{(j)}}(t) \right\rangle$ that involves second moments can be interpreted as the sum of two distinct quadratics; The quadratic hypersurface whose Jacobian depends on the larger elements $E_j\{v_n(t)\}E_j\{v_{n'}(t)\}$

will then have a higher curvature and will effectively mask the other smaller curvature quadratic. This is problematic as the Jacobian obtained from $E_j\{v_n(t)\}E_j\{v_{n'}(t)\}$ is ill-conditioned and the corresponding quadratic has no well-defined minimum. On ther other hand, the Jacobian obtain from $Cov_j(v_n(t), v_{n'}(t))$ is not ill-conditioned, and the corresponding quadratic carries variation-based information that may lead to higher quality parameter estimates.

To address this issue, the cost function should be properly split into a 'mean' and a 'variation' component. Let $v_{op}(t)$ and $\bar{y}_{op}(t)$ denote the operating points of the true voltages and the measurements. Also, let $\delta v(t)$ and $\delta y(t)$ denote the variations such that:

$$v(t) = v_{op}(t) + \delta v(t) \tag{6.9}$$

$$y(t) = y_{op}(t) + \delta y(t) \tag{6.10}$$

Then, for element m, the power of the measurement innovation is given by:

$$\begin{aligned}
(y_m(t) - \tilde{G}_{m,:}W^T v(t))^2 &= y_m^2(t) \\
&+ \textstyle\sum_{n,n_i} \sum_{n',n'_i} \left[\tilde{G}_{m,n_i}\tilde{G}_{m,n'_i} v_n(t) v_{n'}(t)\right] W_{n,n_i} W_{n',n'_i} \\
&-2 \textstyle\sum_{n,n_i} \left[\tilde{G}_{m,n_i}\tilde{G}_{m,n'_i} y_m(t) v_n(t)\right] W_{n,n_i} \\
&= y_{op_m}^2(t) \\
&+ \textstyle\sum_{n,n_i} \sum_{n',n'_i} \left[\tilde{G}_{m,n_i}\tilde{G}_{m,n'_i} v_{op_n}(t) v_{op_{n'}}(t)\right] W_{n,n_i} W_{n',n'_i} \\
&-2 \textstyle\sum_{n,n_i} \left[\tilde{G}_{m,n_i}\tilde{G}_{m,n'_i} y_{op_m}(t) v_{op_n}(t)\right] W_{n,n_i} \\
&+\delta y_m^2(t) \\
&+ \textstyle\sum_{n,n_i} \sum_{n',n'_i} \left[\tilde{G}_{m,n_i}\tilde{G}_{m,n'_i} \delta v_n(t) \delta v_{n'}(t)\right] W_{n,n_i} W_{n',n'_i} \\
&-2 \textstyle\sum_{n,n_i} \left[\tilde{G}_{m,n_i}\tilde{G}_{m,n'_i} \delta y_m(t) \delta v_n(t)\right] W_{n,n_i} \\
&+2 y_{op_m} \delta y_m \\
&+ \textstyle\sum_{n,n_i} \sum_{n',n'_i} \left[\tilde{G}_{m,n_i}\tilde{G}_{m,n'_i} \delta v_n(t) v_{op_{n'}}(t)\right] W_{n,n_i} W_{n',n'_i} \\
&+ \textstyle\sum_{n,n_i} \sum_{n',n'_i} \left[\tilde{G}_{m,n_i}\tilde{G}_{m,n'_i} v_{op_n}(t) \delta v_{n'}(t)\right] W_{n,n_i} W_{n',n'_i} \\
&-2 \textstyle\sum_{n,n_i} \left[\tilde{G}_{m,n_i}\tilde{G}_{m,n'_i} \delta y_m(t) v_{op_n}(t)\right] W_{n,n_i} \\
&-2 \textstyle\sum_{n,n_i} \left[\tilde{G}_{m,n_i}\tilde{G}_{m,n'_i} y_{op_m}(t) \delta v_n(t)\right] W_{n,n_i}
\end{aligned} \tag{6.11}$$

Unfortunately, the states are not observed directly and both the operating points and the variations should be replaced by their expected values under current parameter estimates. With the following approximations and substitutions:

$$\left\langle y_{op_m}^2 \right\rangle \approx \left\langle y_m^2(t) \right\rangle \tag{6.12}$$

$$\left\langle E_j\{v_{op_n}(t) v_{op_{n'}}(t)\} \right\rangle \to \langle E_j\{v_n(t)\} E_j\{v_{n'}(t)\} \rangle \tag{6.13}$$

$$\langle E_j\{y_{op_m}(t) v_{op_n}(t)\} \rangle \to \langle y_m(t) \rangle \langle E_j\{v_n(t)\} \rangle \tag{6.14}$$

$$\left\langle \delta y_m^2(t) \right\rangle = \left\langle (y(t) - \langle y_m(t) \rangle)^2 \right\rangle \tag{6.15}$$

$$\langle E_j\{\delta v_n(t) \delta v_{n'}(t)\} \rangle \to \langle Cov_j(v_n(t), v_{n'}(t)) \rangle \tag{6.16}$$

$$\langle E_j\{\delta y_m(t) \delta v_n(t)\} \rangle \to \langle (y_m(t) - \langle y_m(t) \rangle)(E_j\{v_n(t)\} - \langle E_j\{v_n(t)\} \rangle) \rangle \tag{6.17}$$

and noting that the expectation eliminates the last 5 lines of Eq. 6.11, the individual cost function $\left\langle \zeta_m^{2^{(j)}}(t) \right\rangle$ can now be decomposed as:

$$\left\langle \zeta_m^{2^{(j)}}(t) \right\rangle = \left\langle \zeta_{op_m}^{2^{(j)}}(t) \right\rangle + \left\langle \delta^2 \zeta_m^{2^{(j)}}(t) \right\rangle \tag{6.18}$$

with:

$$\begin{aligned} \left\langle \zeta_{op_m}^{2^{(j)}}(t) \right\rangle =& \left\langle y_m^2(t) \right\rangle \\ &+ \sum_{n,n_i} \sum_{n',n'_i} \left[\tilde{G}_{m,n_i} \tilde{G}_{m,n'_i} \left\langle E_j\{v_n(t)\} E_j\{v_{n'}(t)\} \right\rangle \right] W_{n,n_i} W_{n',n'_i} \\ &- 2 \sum_{n,n_i} \left[\tilde{G}_{m,n_i} \tilde{G}_{m,n'_i} \left\langle y_m(t) \right\rangle \left\langle E_j\{v_n(t)\} \right\rangle \right] W_{n,n_i} \end{aligned} \tag{6.19}$$

and:

$$\begin{aligned} \left\langle \delta^2 \zeta_m^{2^{(j)}}(t) \right\rangle =& \left\langle (y_m(t) - \langle y_m(t) \rangle)^2 \right\rangle \\ &+ \sum_{n,n_i} \sum_{n',n'_i} \left[\tilde{G}_{m,n_i} \tilde{G}_{m,n'_i} \left\langle Cov_j(v_n(t), v_{n'}(t)) \right\rangle \right] W_{n,n_i} W_{n',n'_i} \\ &- 2 \sum_{n,n_i} \left[\tilde{G}_{m,n_i} \tilde{G}_{m,n'_i} \left\langle (y_m(t) - \langle y_m(t) \rangle)(E_j\{v_n(t)\} - \langle E_j\{v_n(t)\} \rangle) \right\rangle \right] W_{n,n_i} \end{aligned} \tag{6.20}$$

Let $\langle\!\langle a(t), b(t) \rangle\!\rangle$ denote the temporal cross-correlation operator defined by

$$\langle\!\langle a(t), b(t) \rangle\!\rangle \triangleq \langle (a(t) - \langle a(t) \rangle)(b(t) - \langle b(t) \rangle) \rangle \tag{6.21}$$

Then:

$$\left\langle \delta^2 \zeta_m^{2^{(j)}}(t) \right\rangle = \langle\!\langle y_m(t), y_m(t) \rangle\!\rangle$$

$$
\begin{aligned}
&+ \sum_{n,n_i} \sum_{n',n'_i} \left[\tilde{G}_{m,n_i} \tilde{G}_{m,n'_i} \left\langle Cov_j(v_n(t), v_{n'}(t)) \right\rangle \right] W_{n,n_i} W_{n',n'_i} \\
&- 2 \sum_{n,n_i} \left[\tilde{G}_{m,n_i} \tilde{G}_{m,n'_i} \left\langle\!\left\langle y_m(t), E_j\{v_n(t)\} \right\rangle\!\right\rangle \right] W_{n,n_i} \qquad (6.22)
\end{aligned}
$$

The cost function $\left\langle \delta^2 \zeta_m^{2(j)}(t) \right\rangle$ factors out the operating points and thus prevents the large operating points from masking the information embedded in waveform correlations. Importantly, the Jacobian corresponding to $\left\langle \delta^2 \zeta_m^{2(j)}(t) \right\rangle$ is not ill-conditioned any may lead to a unique solution of the optimization problem.

Similar equations can be derived for the process innovation. Starting from the definition of ξ:

$$
\begin{aligned}
\xi_{n_I}(t) =& \frac{\phi_{n_I}(t) - \phi_{n_I}(t-1)}{dt} + \omega_{n_I}^2 v_{n_I}(t-1) + 2\omega_{n_I}\phi_{n_I}(t-1) \\
&- A_{n_I}\omega_{n_I}(a + [mW^T v(t-1)]_{n_I}) \qquad (6.23)
\end{aligned}
$$

where $n_I \in \{1 \cdots N_I\}$ denoted the index of an internal population. To reduce the clutter in the expansion of $\xi_{n_I}^2$, define vector c_{n_I} such that:

$$
\begin{aligned}
&\frac{\phi_{n_I}(t) - \phi_{n_I}(t-1)}{dt} + \omega_{n_I}^2 v_{n_I}(t-1) + 2\omega_{n_I}\phi_{n_I}(t-1) - A_{n_I}\omega_{n_I} a \\
&= A_{n_I}\omega_{n_I} m c_{n_I}^T \begin{bmatrix} 1 \\ x(t) \end{bmatrix} \\
&= A_{n_I}\omega_{n_I} m c_{n_I}^T \tilde{x}(t) \qquad (6.24)
\end{aligned}
$$

where $\tilde{x}(t)$ is the augmented state vector $x(t) = [1, v^T(t), \phi^T(t), v^T(t-1), \phi^T(t-1)]^T$. Then:

$$
\begin{aligned}
\frac{1}{(A_{n_I}\omega_{n_I} m)^2} \xi_{n_I}^2(t) =& (c_{n_I}^T \tilde{x}(t) - [W^T v(t-1)]_{n_I})^2 \\
=& (c_{n_I}^T \tilde{x}(t) - \sum_{n=1}^{N} W_{n,n_I} v_n(t-1))^2
\end{aligned}
$$

$$
\begin{aligned}
&= c_{n_I}^T \tilde{x}(t) \tilde{x}^T(t) c_{n_I} \\
&\quad + \sum_{n,n'} W_{n,n_I} W_{n',n_I} v_n(t-1) v_{n'}(t-1) \\
&\quad - 2 c_{n_I}^T \tilde{x}(t) \sum_{n=1}^{N} W_{n,n_I} v_n(t-1)
\end{aligned}
\tag{6.25}
$$

Similar to Eq. 6.10, $\tilde{x}$ is decomposed into $\tilde{x}_{op}$ and $\delta\tilde{x}$. For notational simplicity, let's denote the statistcal expectation can covariance $E_j\{x\}$ and $Cov_j(a,b)$ by $\overline{x}^{(j)}$ and $\overline{\overline{a,b}}^{(j)}$ respectively. Then, the same methods used for decomposing $\left\langle \zeta_m^{2^{(j)}}(t) \right\rangle$ as $\left\langle \zeta_{op_m}^{2^{(j)}}(t) \right\rangle + \left\langle \delta^2 \zeta_m^{2^{(j)}}(t) \right\rangle$ are applied to obtain the following decomposition for $\left\langle \overline{\xi_{n_I}^2(t)}^{(j)} \right\rangle$:

$$
\begin{aligned}
\frac{1}{(A_{n_I}\omega_{n_I}m)^2} \left\langle \overline{\xi^2_{op_{n_I}}(t)}^{(j)} \right\rangle =& c_{n_I}^T \left\langle \overline{\tilde{x}(t)}^{(j)} \right\rangle \left\langle \overline{\tilde{x}(t)}^{(j)} \right\rangle^T c_{n_I} \\
&+ \sum_{n,n'} W_{n,n_I} W_{n',n_I} \left\langle \overline{v_n(t-1)}^{(j)} \right\rangle \left\langle \overline{v_{n'}(t-1)}^{(j)} \right\rangle \\
&- 2 \sum_{n} W_{n,n_I} \left(\sum_{k=1}^{4N+1} [c_{n_I}]_k \left\langle \overline{\tilde{x}_k(t)}^{(j)} \right\rangle \left\langle \overline{v_n(t-1)}^{(j)} \right\rangle \right)
\end{aligned}
\tag{6.26}
$$

$$
\begin{aligned}
\frac{1}{(A_{n_I}\omega_{n_I}m)^2} \left\langle \overline{\delta\xi^2_{n_I}(t)}^{(j)} \right\rangle =& c_{n_I}^T \left\langle \overline{\overline{\tilde{x}(t), \tilde{x}(t)}}^{(j)} \right\rangle c_{n_I} \\
&+ \sum_{n,n'} W_{n,n_I} W_{n',n_I} \left\langle \overline{\overline{v_n(t-1), v_{n'}(t-1)}}^{(j)} \right\rangle \\
&- 2 \sum_{n} W_{n,n_I} \left(\sum_{k=1}^{4N+1} [c_{n_I}]_k \left\langle \overline{\overline{\tilde{x}_k(t), v_n(t-1)}}^{(j)} \right\rangle \right)
\end{aligned}
\tag{6.27}
$$

6.4 The Open Challenges

As was discussed in the previous section, the individual cost functions that contribute to the *overall cost* function are characterized by a) belonging to the process innovation or measurement innovation (ξ or ζ), b) the corresponding element of the innovation vector (m in ζ_m or n_I in ξ_{n_I}), and c) whether the cost is related to the operating point or the variations (e.g. $\delta^2\xi$ vs. ξ_{op}). Two of the problems, namely the low dispersion and imbalanced curvatures, originate from the improper linear combination of the individual cost functions. The remaining problem of ill-conditioned Jacobian is inherent to some of the individual cost functions and may only be mitigated using corrections from other well-conditioned individual cost functions.

The individual cost functions should be aggregated properly so the estimation algorithm functions properly and yields high quality estimates. In the following paragraphs, we discuss the details and the challenges of the two extremes of the spectrum of methods that may be used for the aggregation of the individual cost functions. It is assumed that stepping methods such as gradient descent or coordinate descent are used for solving the underlying optimization problem. The space of design choices should be systematically explored to find the methods that results in the most consistent and accurate parameter estimates.

Individual-Vote, Weak-Action

On one end of the spectrum, the estimation algorithm consults only one of the cost functions to determine the next step to take. The algorithm then takes a small step from the current parameter value toward the direction deemed optimal by the individual cost function. While the individual cost function might suggest a big jump to achieve a significant improvement for itself, it is important for the algorithm to only partially respect the individual votes so that all of the cost functions can fairly contribute. This procedure is similar to a democratic one-vote incremental-action improvement mechanism where the arbitrator repeatedly consults the individuals and takes a partial action towards the opinion of that single individual.

While this method can mitigate the low dispersion and the imbalanced curvature issue by circumventing the linear combination of the individual cost functions, two main challenges remain to be addressed. First, how does the algorithm respond to the opinion of individual cost functions? For instance, does the algorithm take equally-long steps toward the individual solutions, or does it assign different strides to different individuals? If different strides are used, how should the strides be assigned to get as close as possible to the true parameters?

Second, how should the algorithm incorporate sparsity? Does the algorithm fully delegates this responsibility to individual cost functions, or does it 'smartly' combine individual votes to find a sparse solution? If ℓ_1 regularization is used on individual cost functions, what regularization parameters (λ) should be used for different cost functions? The algorithm should probably avoid using the same λ value for all cost functions as different cost functions have different curvatures and using the same λ will have more significant effect on the more flat cost functions. Furthermore, even if the individual cost functions attempt to pull the estimated into a locally sparse solution, the iteration between different cost functions can act as a competition between the individuals and lead to a globally non-sparse solution, which is a direct result of the individuals not consulting each other before reporting an opinion.

All-Vote, Strong-Action

On the other end of the spectrum, the algorithm may combine all of the individual cost functions into a new cost function before making a decision that reflects all individual opinions. While this method simplifies the incorporation of sparsity by adding a single ℓ_1 regularization, it is not clear how the individual costs should be combined and how this decision affects the accuracy of the estimation result.

6.5 Future Work

6.5.1 Algorithm Modification

The challenges discussed in Section 6.4 should be addressed before the EM-based algorithm is applied to connectivity estimation from EEG signals. In particular, the number of cost functions that contribute to a single parameter update, the method of aggregation, and the incorporation of sparsity should be determined based on the resulting estimation accuracy on a comprehensive set of synthetic system configurations.

6.5.2 Non-Identity LFM

It is expected that realistic and ill-conditioned LFMs deteriorate the estimation performance compared to the identitiy LFM. A low-rank LFM may impose fundamental limits on the estimaiton accuracy caused by unidentifiability. The performance of the algorithm should be tested under such realistic LFMs.

6.5.3 Delayed Connections

The mathematical model of the network of neural masses was derived without including delays in the connections. Two methods exists for adding the connections to the model. First, similar to the augmented state space vector of Chapter 2, the state vector of NMM may be augmented with the state history, resulting in an expanded connectivity matrix that encodes discrete connections delays in addition to connection strengths. Another physiologically-based method also exists to model the connections. In this method, two populations are connected by a *connecting population*, where the connecting population follows the same differential equation with different parameters such as the time constant and the synaptic gain. The time constant of such a population may be used as a measure of connection delay between two populations. Although physiologically more meaningful, the latter approach requires re-designing the estimation algorithm in order to estimate synaptic gains and time constants from data.

6.5.4 Application to real EEG data

The NMM was adopted in this thesis to resolve the mismatch between the dynamics of real EEG data and the dynamics of synthetic data generated by linear MVAR models.

The capability of NMM in reproducing real EEG waveforms has been extensively studied in the literature. However, we observed that the parameter values listed by [115] cannot reproduce the spectrum of the available resting state EEG signals at frequencies below 4Hz. Based on the available literature on the very low frequency brain waves, the addition of a thalamo-cortical delayed connection between a cortical column ad a thalamical population might be a good candidate for reproducing the very low frequency features.

Finally, and most importantly, the estimation algorithm should be applied on real EEG data and the results should be validated against previous work. While the absence of ground truth information makes a true validation impossible, the (lack of) conformity of the results to certain physiologically verified facts may still be used to (in)validate the results.

Conclusion

In this work, the estimation of connectivity parameters in indirectly observed MVAR models was studied. The estimation algorithm used a maximum a posteriori criterion. The cost function was augmented by the ℓ_1 norm to promote sparse solutions. The resulting optimization problem was iteratively solved by coordinate descent. The closed-form solution to the single-coordinate problem rendered the algorithm lightweight and applicable to high dimensional data.

The estimation algorithm was evaluated on a comprehensive set of synthetic ground-truth configurations. The algorithm was numerically shown to outperform the previous work under moderate to high ground-truth sparsity. A complexity analysis of the algorithm, as well as complexity enhancements were provided.

The algorithm was then successfully applied to real temperature data to explore the predictive power of temperature time series in about 100 weather stations around the U.S. mainland. Not only the results were consistent with previous findings, but also did they suggest predictive powers for coastal stations; an observation that might account for the influence of the ocean on land temperature variations.

Applied to real EEG data, the algorithm failed to estimate meaningful connections. Several remedies based on the enrichment of the noise dynamics were tested, but failed to address the issue. The intrinsic limitations of the linear MVAR model in capturing EEG features were discussed. The Neural Mass Model was then adopted because of its flexibility in modeling EEG waveforms and because of its computational tractability compared to other realistic neurodynamic models. The estimation algorithm was re-derived to accommodate the nonlinearity of the model. The modified NMM connectivity estimation algorithm was then applied to synthetic data, and it was shown analytically that the MAP-based parameter estimate has fundamental limitations in identifying NMM connectivities.

Bibliography

[1] F. Abegaz and E. Wit. Sparse time series chain graphical models for reconstructing genetic networks. *Biostatistics*, 14(3):586–599, 2013.

[2] Z. A. Acar and S. Makeig. Neuroelectromagnetic forward head modeling toolbox. *Journal of Neuroscience Methods*, 190(2):258–270, 2010.

[3] S. Achard and E. Bullmore. Efficiency and cost of economical brain functional networks. *PLoS Computational Biology*, 3(2):e17, 2007.

[4] T. Acharya and A. K. Ray. *Image processing: principles and applications.* John Wiley & Sons, 2005.

[5] S. D. Babacan, R. Molina, and A. K. Katsaggelos. Bayesian compressive sensing using Laplace priors. *IEEE Transactions on Image Processing*, 19(1):53–63, 2010.

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

[7] L. Barnett, A. B. Barrett, and A. K. Seth. Granger causality and transfer entropy are equivalent for Gaussian variables. *Physical Review Letters*, 103(23):238701, 2009.

[8] H. E. Bell. Gershgorin's theorem and the zeros of polynomials. *The American Mathematical Monthly*, 72(3):292–295, 1965.

[9] M. K. Belmonte, G. Allen, A. Beckel-Mitchener, L. M. Boulanger, R. A. Carper, and S. J. Webb. Autism and abnormal development of brain connectivity. *The Journal of Neuroscience*, 24(42):9228–9231, 2004.

[10] A. Bolstad, B. D. Van Veen, and R. Nowak. Causal network inference via group sparse regularization. *IEEE transactions on signal processing*, 59(6):2628–2641, 2011.

[11] C. Boone, H. Bussey, and B. J. Andrews. Exploring genetic interactions and networks with yeast. *Nature Reviews Genetics*, 8(6):437, 2007.

[12] J. M. Borwein and Q. J. Zhu. A survey of subdifferential calculus with applications. *J. Nonlinear Analysis: Theory, Methods, and Applications*, 38:687–773, 1999.

[13] L. Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT*, pages 177–186. Springer, 2010.

[14] A. M. Bruckstein, D. L. Donoho, and M. Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Review*, 51(1):34–81, 2009.

[15] C. Büchel and K. Friston. Dynamic changes in effective connectivity characterized by variable parameter regression and Kalman filtering. *Human Brain Mapping*, 6(5-6):403–408, 1998.

[16] E. Bullmore and O. Sporns. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience*, 10(3):186, 2009.

[17] M. K. Carroll, G. A. Cecchi, I. Rish, R. Garg, and A. R. Rao. Prediction and interpretation of distributed neural activity with sparse models. *NeuroImage*, 44(1):112–122, 2009.

[18] B. Cassidy, C. Rae, and V. Solo. Brain activity: Connectivity, sparsity, and mutual information. *IEEE transactions on medical imaging*, 34(4):846–860, 2015.

[19] L. Cavalcante and R. J. Bessa. Solar power forecasting with sparse vector autoregression structures. In *2017 IEEE Manchester PowerTech*, pages 1–6, June 2017.

[20] M. Chávez, J. Martinerie, and M. Le Van Quyen. Statistical assessment of nonlinear causality: Application to epileptic EEG signals. *Journal of Neuroscience Methods*, 124(2):113–128, 2003.

[21] X. Chen, Z. J. Wang, and M. J. McKeown. A sparse unified structural equation modeling approach for brain connectivity analysis. *International Conference on Biomedical and Bioinformatics Engineering*, 2009.

[22] B. L. P. Cheung, B. A. Riedner, G. Tononi, and B. van Veen. Estimation of cortical connectivity from EEG using state-space models. *IEEE Transactions on Biomedical Engineering*, 57(9):2122–2134, 2010.

[23] E. Cuthill and J. McKee. Reducing the bandwidth of sparse symmetric matrices. In *Proceedings of the 1969 24th national conference*, pages 157–172. ACM, 1969.

[24] F. L. Da Silva, A. Hoeks, H. Smits, and L. Zetterberg. Model of brain rhythmic activity. *Kybernetik*, 15(1):27–37, 1974.

[25] O. David and K. J. Friston. A neural mass model for MEG/EEG: coupling and neuronal dynamics. *NeuroImage*, 20(3):1743–1755, 2003.

[26] R. A. Davis, P. Zang, and T. Zheng. Sparse vector autoregressive modeling. *arXiv preprint arXiv:1207.0520*, 2012.

[27] R. S. Desikan, F. Ségonne, B. Fischl, B. T. Quinn, B. C. Dickerson, D. Blacker, R. L. Buckner, A. M. Dale, R. P. Maguire, B. T. Hyman, et al. An automated labeling system for subdividing the human cerebral cortex on MRI scans into gyral based regions of interest. *Neuroimage*, 31(3):968–980, 2006.

[28] C. Destrieux, B. Fischl, A. Dale, and E. Halgren. Automatic parcellation of human cortical gyri and sulci using standard anatomical nomenclature. *Neuroimage*, 53(1):1–15, 2010.

[29] M. Ding, S. L. Bressler, W. Yang, and H. Liang. Short-window spectral analysis of cortical event-related potentials by adaptive multivariate autoregressive modeling: Data preprocessing, model validation, and variability assessment. *Biological Cybernetics*, 83(1):35–45, 2000.

[30] J. Dowell and P. Pinson. Very-short-term probabilistic wind power forecasts by sparse vector autoregression. *IEEE Transactions on Smart Grid*, 7(2):763–770, 2016.

[31] D. L. Duttweiler. Proportionate normalized least-mean-squares adaptation in echo cancelers. *IEEE Transactions on Speech and Audio Processing*, 8(5):508–518, Sep 2000.

[32] F. H. Eeckman and W. J. Freeman. Asymmetric sigmoid non-linearity in the rat olfactory system. *Brain Research*, 557(1-2):13–21, 1991.

[33] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, et al. Least angle regression. *The Annals of Statistics*, 32(2):407–499, 2004.

[34] J. Fan, J. Lv, and L. Qi. Sparse high-dimensional models in economics. *Annual Reviews on Econometrics*, 2011.

[35] J. Fan, J. Zhang, and K. Yu. Asset allocation and risk assessment with gross exposure constraints for vast portfolios. *arXiv preprint arXiv:0812.2604*, 2008.

[36] J. Fessler, A. O. Hero, et al. Space-alternating generalized expectation-maximization algorithm. *IEEE Transactions on Signal Processing*, 42(10):2664–2677, 1994.

[37] M. Feurer, J. T. Springenberg, and F. Hutter. Initializing Bayesian hyperparameter optimization via meta-learning. In *AAAI*, pages 1128–1135, 2015.

[38] M. Fiecas, H. Ombao, et al. The generalized shrinkage estimator for the analysis of functional connectivity of brain signals. *The Annals of Applied Statistics*, 5(2A):1102–1125, 2011.

[39] A. Fornito, A. Zalesky, and E. Bullmore. *Fundamentals of brain network analysis.* Academic Press, 2016.

[40] W. Freeman. Models of the dynamics of neural populations. *Electroencephalography and clinical neurophysiology. Supplement*, (34):9–18, 1978.

[41] W. A. Freiwald, P. Valdes, J. Bosch, R. Biscay, J. C. Jimenez, L. M. Rodriguez, V. Rodriguez, A. K. Kreiter, and W. Singer. Testing non-linearity and directedness of interactions between neural groups in the macaque inferotemporal cortex. *Journal of Neuroscience Methods*, 94(1):105–119, 1999.

[42] J. Friedman, T. Hastie, H. Höfling, R. Tibshirani, et al. Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1(2):302–332, 2007.

[43] A. Fujita, J. R. Sato, H. M. Garay-Malpartida, R. Yamaguchi, S. Miyano, M. C. Sogayar, and C. E. Ferreira. Modeling gene expression regulatory networks with the sparse vector autoregressive model. *BMC systems biology*, 1(1):39, 2007.

[44] A. Fusiello. A matter of notation: Several uses of the Kronecker product in 3D computer vision. *Pattern Recognition Letters*, 28(15):2127–2132, 2007.

[45] B. R. Gaines, J. Kim, and H. Zhou. Algorithms for fitting the constrained lasso. *Journal of Computational and Graphical Statistics*, (just-accepted), 2018.

[46] X. Gao, B. Shahbaba, N. Fortin, and H. Ombao. Evolutionary state-space model and its application to time-frequency analysis of local field potentials. *arXiv preprint arXiv:1610.07271*, 2016.

[47] T. S. Gardner, D. Di Bernardo, D. Lorenz, and J. J. Collins. Inferring genetic networks and identifying compound mode of action via expression profiling. *Science*, 301(5629):102–105, 2003.

[48] J. F. Geweke. Measures of conditional linear dependence and feedback between time series. *Journal of the American Statistical Association*, 79(388):907–915, 1984.

[49] M. Gopal. *Modern control system theory*. New Age International, 1993.

[50] C. W. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: Journal of the Econometric Society*, pages 424–438, 1969.

[51] P. J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

[52] P. Guillaume, J. Schoukens, and R. Pintelon. Sensitivity of roots to errors in the coefficient of polynomials obtained by frequency-domain estimation methods. *IEEE transactions on instrumentation and measurement*, 38(6):1050–1056, 1989.

[53] G. Guillot, R. Senoussi, and P. Monestiez. A positive definite estimator of the non stationary covariance of random fields. In *geoENV III - Geostatistics for Environmental Applications*, pages 333–344. Springer, 2001.

[54] L. Harrison, W. D. Penny, and K. Friston. Multivariate autoregressive modeling of fMRI time series. *NeuroImage*, 19(4):1477–1491, 2003.

[55] S. Haufe. *Towards EEG source connectivity analysis*. PhD thesis, Berlin Institute of Technology, 2011.

[56] S. Haufe, K.-R. Müller, G. Nolte, and N. Krämer. Sparse causal discovery in multivariate time series. In *Proceedings of the International Conference on Causality: Objectives and Assessment-Volume 6*, pages 97–106. JMLR. org, 2008.

[57] S. Haufe, V. Nikulin, and G. Nolte. Identifying brain effective connectivity patterns from EEG: Performance of Granger causality, DTF, PDC and PSI on simulated data. *BMC Neuroscience*, 12(Suppl 1):P141, 2011.

[58] S. Haufe, R. Tomioka, G. Nolte, K. R. Mller, and M. Kawanabe. Modeling sparse connectivity between underlying brain sources for eeg/meg. *IEEE Transactions on Biomedical Engineering*, 57(8):1954–1963, Aug 2010.

[59] Y. He, Z. J. Chen, and A. C. Evans. Small-world anatomical networks in the human brain revealed by cortical thickness from MRI. *Cerebral Cortex*, 17(10):2407–2419, 2007.

[60] C. Hendahewa and V. Pavlovic. Analysis of causality in stock market data. In *2012 11th International Conference on Machine Learning and Applications*, volume 1, pages 288–293, Dec 2012.

[61] M.-H. R. Ho, H. Ombao, and R. Shumway. A state-space approach to modelling brain dynamics. *Statistica Sinica*, 15:407–425, 2005.

[62] C. J. Honey, R. Kötter, M. Breakspear, and O. Sporns. Network structure of cerebral cortex shapes functional connectivity on multiple time scales. *Proceedings of the National Academy of Sciences*, 104(24):10240–10245, 2007.

[63] V. K. Jirsa and A. R. McIntosh. *Handbook of brain connectivity*, volume 1. Springer, 2007.

[64] M. Kaminski and K. J. Blinowska. A new method of the description of the information flow in the brain structures. *Biological Cybernetics*, 65(3):203–210, 1991.

[65] H. Kantz and T. Schreiber. *Nonlinear time series analysis*, volume 7. Cambridge university press, 2004.

[66] A. Korzeniewska, M. Mańczak, M. Kamiński, K. J. Blinowska, and S. Kasicki. Determination of information flow direction among brain structures by a modified directed transfer function (dDTF) method. *Journal of Neuroscience Methods*, 125(1):195–207, 2003.

[67] M. Lenz, M. Musso, Y. Linke, O. Tüscher, J. Timmer, C. Weiller, and B. Schelter. Joint EEG/fMRI state space model for the detection of directed interactions in human brains: a simulation study. *Physiological Measurement*, 32(11):1725, 2011.

[68] E. Levitan and G. T. Herman. A maximum a posteriori probability expectation maximization algorithm for image reconstruction in emission tomography. *IEEE Transactions on Medical Imaging*, 6(3):185–192, 1987.

[69] K. Li, L. Guo, J. Nie, G. Li, and T. Liu. Review of methods for functional brain connectivity detection using fMRI. *Computerized Medical Imaging and Graphics*, 33(2):131–139, 2009.

[70] X. Li, G. Marrelec, R. F. Hess, and H. Benali. A nonlinear identification method to study effective connectivity in functional MRI. *Medical Image Analysis*, 14(1):30–38, 2010.

[71] T. Limpiti, B. D. Van Veen, and R. T. Wakai. Cortical patch basis model for spatially extended neural activity. *IEEE Transactions on Biomedical Engineering*, 53(9):1740–1754, 2006.

[72] H.-J. Lin, P. A. Wolf, M. Kelly-Hayes, A. S. Beiser, C. S. Kase, E. J. Benjamin, and R. B. D'agostino. Stroke severity in atrial fibrillation: The Framingham study. *Stroke*, 27(10):1760–1764, 1996.

[73] Y. Liu and S. Aviyente. Information theoretic approach to quantify causal neural interactions from EEG. In *44th ASILOMAR Conference on Signals, Systems, and Computers*, pages 1380–1384. IEEE, 2010.

[74] H. Lütkepohl. *New introduction to multiple time series analysis*. Springer, Berlin, 3 edition, 2005.

[75] J. R. Magnus and H. Neudecker. The elimination matrix: Some lemmas and applications. *SIAM Journal on Algebraic Discrete Methods*, 1(4):422–449, 1980.

[76] J. R. Magnus and H. Neudecker. Matrix differential calculus with applications in statistics and econometrics. *Wiley Series in Probability and Mathematical Statistics*, 1988.

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

[78] H. Marko. The bidirectional communication theory–a generalization of information theory. *IEEE Transactions on communications*, 21(12):1345–1351, 1973.

[79] J. Massey. Causality, feedback and directed information. In *Proc. Int. Symp. Inf. Theory Applic.(ISITA-90)*, pages 303–305. Citeseer, 1990.

[80] G. McLachlan and T. Krishnan. *The EM algorithm and extensions*, volume 382. John Wiley & Sons, 2007.

[81] J. Mei and J. M. Moura. Signal processing on graphs: Causal modeling of unstructured data. *IEEE Trans. Signal Processing*, 65(8):2077–2092, 2017.

[82] J. Mei and J. M. F. Moura. Signal processing on graphs: Causal modeling of unstructured data. *IEEE Transactions on Signal Processing*, 65(8):2077–2092, April 2017.

[83] G. Michailidis and F. dAlché Buc. Autoregressive models for gene regulatory network inference: Sparsity, stability and causality issues. *Mathematical biosciences*, 246(2):326–334, 2013.

[84] E. Möller, B. Schack, M. Arnold, and H. Witte. Instantaneous multivariate EEG coherence analysis by means of adaptive high-dimensional autoregressive models. *Journal of Neuroscience Methods*, 105(2):143–158, 2001.

[85] K. P. Murphy. *Machine learning: A probabilistic perspective.* MIT press, 2012.

[86] A. Y. Mutlu and S. Aviyente. Inferring effective connectivity in the brain from EEG time series using dynamic Bayesian networks. In *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 4739–4742. IEEE, 2009.

[87] G. Nolte, A. Ziehe, V. V. Nikulin, A. Schlögl, N. Krämer, T. Brismar, and K.-R. Müller. Robustly estimating the flow direction of information in complex physical systems. *Physical Review Letters*, 100(23), 2008.

[88] W. Pan, Y. Yuan, J. Gonçalves, and G.-B. Stan. A sparse Bayesian approach to the identification of nonlinear state-space systems. *IEEE Transactions on Automatic Control*, 61(1):182–187, 2016.

[89] A. Pongrattanakul, P. Lertkultanon, and J. Songsiri. Sparse system identification for discovering brain connectivity from fMRI time series. In *2013 Proceedings of SICE Annual Conference (SICE)*, pages 949–954. IEEE, 2013.

[90] M. Pourahmadi. Covariance estimation: The GLM and regularization perspectives. *Statistical Science*, pages 369–387, 2011.

[91] N. Qian. On the momentum term in gradient descent learning algorithms. *Neural networks*, 12(1):145–151, 1999.

[92] J. C. Rajapakse and J. Zhou. Learning effective brain connectivity with dynamic Bayesian networks. *Neuroimage*, 37(3):749–760, 2007.

[93] M. E. Rodie, K. P. Forbes, and K. Muir. Advances in neuroimaging. In *Understanding Differences and Disorders of Sex Development (DSD)*, volume 27, pages 63–75. Karger Publishers, 2014.

[94] S. Roweis and Z. Ghahramani. A unifying review of linear Gaussian models. *Neural Computation*, 11(2):305–345, 1999.

[95] M. Rubinov and O. Sporns. Complex network measures of brain connectivity: Uses and interpretations. *Neuroimage*, 52(3):1059–1069, 2010.

[96] S. Sarkar and S. Chawla. Inferring the contiguity matrix for spatial autoregressive analysis with applications to house price prediction. *arXiv preprint arXiv:1607.01999*, 2016.

[97] J. R. Sato, E. A. Junior, D. Y. Takahashi, M. de Maria Felix, M. J. Brammer, and P. A. Morettin. A method to produce evolving functional connectivity maps during the course of an fMRI experiment using wavelet-based time-varying Granger causality. *Neuroimage*, 31(1):187–196, 2006.

[98] T. Schreiber. Measuring information transfer. *Physical Review Letters*, 85(2):461, 2000.

[99] P. Skudlarski, K. Jagannathan, K. Anderson, M. C. Stevens, V. D. Calhoun, B. A. Skudlarska, and G. Pearlson. Brain connectivity is not only lower but different in schizophrenia: A combined anatomical and functional approach. *Biological Psychiatry*, 68(1):61–69, 2010.

[100] V. A. Smith, J. Yu, T. V. Smulders, A. J. Hartemink, and E. D. Jarvis. Computational inference of neural information flow networks. *PLoS Computational Biology*, 2(11):161, 2006.

[101] S. N. Sotiropoulos, S. Jbabdi, J. Xu, J. L. Andersson, S. Moeller, E. J. Auerbach, M. F. Glasser, M. Hernandez, G. Sapiro, M. Jenkinson, et al. Advances in diffusion MRI acquisition and processing in the human connectome project. *Neuroimage*, 80:125–143, 2013.

[102] O. Sporns. Brain connectivity. *Scholarpedia*, 2(10), 2007. revision #91083.

[103] O. Sporns, G. Tononi, and G. Edelman. Theoretical neuroanatomy: Relating anatomical and functional connectivity in graphs and cortical connection matrices. *Cerebral Cortex*, 10(2):127–141, 2000.

[104] O. Sporns and J. D. Zwi. The small world of the cerebral cortex. *Neuroinformatics*, 2(2):145–162, 2004.

[105] J. H. Stock and M. W. Watson. Vector autoregressions. *Journal of Economic perspectives*, 15(4):101–115, 2001.

[106] K. Supekar, V. Menon, D. Rubin, M. Musen, M. D. Greicius, et al. Network analysis of intrinsic functional brain connectivity in Alzheimers disease. *PLoS Computational Biology*, 4(6), 2008.

[107] L. W. Swanson. *Brain architecture: Understanding the basic plan.* Oxford University Press, 2012.

[108] R. Tibshirani. Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[109] R. J. Tibshirani. *The solution path of the generalized lasso.* Stanford University, 2011.

[110] P. Tseng. Convergence of a block coordinate descent method for non-differentiable minimization. *Journal of Optimization Theory and Applications*, 109(3):475–494, 2001.

[111] P. Tseng and S. Yun. A coordinate gradient descent method for non-smooth separable minimization. *Mathematical Programming*, 117(1-2):387–423, 2009.

[112] P. A. Valdés-Sosa, J. M. Sánchez-Bornot, A. Lage-Castellanos, M. Vega-Hernández, J. Bosch-Bayard, L. Melie-García, and E. Canales-Rodríguez. Estimating brain functional connectivity with sparse multivariate autoregression. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 360(1457):969–981, 2005.

[113] D. Vidaurre, C. Bielza, and P. Larrañaga. A survey of $\ell 1$ regression. *International Statistical Review*, 81(3):361–387, 2013.

[114] S. Walter and H. Tiemeier. Variable selection: Current practice in epidemiological studies. *European Journal of Epidemiology*, 24(12):733, 2009.

[115] F. Wendling, F. Bartolomei, J. Bellanger, and P. Chauvel. Epileptic fast activity can be explained by a model of impaired gabaergic dendritic inhibition. *European Journal of Neuroscience*, 15(9):1499–1508, 2002.

[116] I. Wilms, S. Gelper, and C. Croux. Sparse vector autoregressive models with an application in marketing. 2015.

[117] D. Wu, J.-T. King, C.-H. Chuang, C.-T. Lin, and T.-P. Jung. Spatial filtering for EEG-based regression problems in brain-computer interface (BCI). *IEEE Transactions on Fuzzy Systems*, 2017.

[118] T. T. Wu and K. Lange. Coordinate descent algorithms for LASSO penalized regression. *The Annals of Applied Statistics*, pages 224–244, 2008.

[119] L. Xue, S. Ma, and H. Zou. Positive-definite $\ell 1$-penalized estimation of large covariance matrices. *Journal of the American Statistical Association*, 107(500):1480–1491, 2012.

[120] S. R. y Cajal. *Les nouvelles idées sur la structure du système nerveux chez l'homme et chez les vertébrés*. C. Reinwald, 1894.

[121] M. P. Young. The organization of neural systems in the primate cerebral cortex. *Proc. R. Soc. Lond. B*, 252(1333):13–18, 1993.

[122] G.-X. Yuan, K.-W. Chang, C.-J. Hsieh, and C.-J. Lin. A comparison of optimization methods and software for large-scale $\ell 1$-regularized linear classification. *The Journal of Machine Learning Research*, 11:3183–3234, 2010.

[123] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[124] M. Zavaglia, F. Cona, and M. Ursino. A neural mass model to simulate different rhythms in a cortical region. *Computational intelligence and neuroscience*, 2010:5, 2010.

[125] S. Zeemering. *Sparse estimation: Applications in atrial fibrillation.* PhD thesis, Maastricht University, 2015.

[126] X. Zheng and J. C. Rajapakse. Learning functional structure from fMRI images. *Neuroimage*, 31(4):1601–1613, 2006.