

Computational Intelligence & Neuroscience

Modeling and Analysis of Neural Spike Trains

Guest Editors: Wei Wu, Asohan Amarasingham, Zhe (Sage) Chen,
and Sung-Phil Kim





Modeling and Analysis of Neural Spike Trains

Computational Intelligence and Neuroscience

Modeling and Analysis of Neural Spike Trains

Guest Editors: Wei Wu, Asohan Amarasingham,
Zhe (Sage) Chen, and Sung-Phil Kim



Copyright © 2014 Hindawi Publishing Corporation. All rights reserved.

This is a special issue published in “Computational Intelligence and Neuroscience.” All articles are open access articles distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Editorial Board

Fabio Babiloni, Italy
Sylvain Baillet, Canada
Tonio Ball, Germany
Theodore W. Berger, USA
James M. Bower, USA
Steven L. Bressler, USA
Vince D. Calhoun, USA
Ke Chen, UK
Yuehui Chen, China
Michela Chiappalone, Italy
Seungjin Choi, Republic of Korea
Andrzej Cichocki, Japan
Sergio Cruces, Spain
Justin Dauwels, Singapore
Christian W. Dawson, UK

Thomas DeMarse, USA
Piotr Franaszczuk, USA
Samanwoy Ghosh-Dastidar, USA
Christoph Guger, Austria
David Hansel, France
Pasi A. Karjalainen, Finland
Robert Kozma, USA
Arvind Kumar, Germany
Yuanqing Li, China
Cheng-Jian Lin, Taiwan
Hongtao Lu, China
Kezhi Mao, Singapore
Sergio Martinoia, Italy
Ennio Mingolla, USA
Jennifer K. Mortimer, Canada

Klaus Obermayer, Germany
Karim G. Oweiss, USA
Saeid Sanei, UK
Christos N. Schizas, Cyprus
Jianwei Shuai, China
Thomas Shultz, Canada
Hiroshige Takeichi, Japan
Lefteri Tsoukalas, USA
Marc Van Hulle, Belgium
Pablo Varona, Spain
Meel Velliste, USA
Francois Benoit Vialatte, France
Liqing Zhang, China
Daoqiang Zhang, China

Contents

Modeling and Analysis of Neural Spike Trains, Wei Wu, Asohan Amarasingham, Zhe (Sage) Chen, and Sung-Phil Kim
Volume 2014, Article ID 161203, 2 pages

Sparse Data Analysis Strategy for Neural Spike Classification, Vincent Vigneron and Hsin Chen
Volume 2014, Article ID 757068, 18 pages

Spike Sorting by Joint Probabilistic Modeling of Neural Spike Trains and Waveforms, Brett A. Matthews and Mark A. Clements
Volume 2014, Article ID 643059, 12 pages

A Tensor-Product-Kernel Framework for Multiscale Neural Activity Decoding and Control, Lin Li, Austin J. Brockmeier, John S. Choi, Joseph T. Francis, Justin C. Sanchez, and José C. Príncipe
Volume 2014, Article ID 870160, 16 pages

Prediction of Human's Ability in Sound Localization Based on the Statistical Properties of Spike Trains along the Brainstem Auditory Pathway, Ram Krips and Miriam Furst
Volume 2014, Article ID 575716, 11 pages

Homogenous Chaotic Network Serving as a Rate/Population Code to Temporal Code Converter, Mikhail V. Kiselev
Volume 2014, Article ID 476580, 8 pages

An Overview of Bayesian Methods for Neural Spike Train Analysis, Zhe Chen
Volume 2013, Article ID 251905, 17 pages

Editorial

Modeling and Analysis of Neural Spike Trains

Wei Wu,¹ Asohan Amarasingham,² Zhe (Sage) Chen,³ and Sung-Phil Kim⁴

¹ Department of Statistics, Florida State University, Tallahassee, FL 32306, USA

² Department of Mathematics, The City College of New York, New York, NY 10031, USA

³ Department of Psychiatry and Department of Neuroscience and Physiology, New York University School of Medicine, New York, NY 10016, USA

⁴ School of Design and Human Engineering, Ulsan National Institute of Science and Technology, UNIST-gil 50, Ulsan 689-798, Republic of Korea

Correspondence should be addressed to Wei Wu; wwwu@stat.fsu.edu

Received 30 April 2014; Accepted 30 April 2014; Published 3 July 2014

Copyright © 2014 Wei Wu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

It is well known that time-dependent information is represented via sequences of stereotyped spike waveforms in the nervous system. Mathematical modeling and analysis of waveform sequences (or spike trains) have been one of the central problems in the field of computational neuroscience. This problem is significantly challenging since population neuronal activity is often stochastic, highly correlated, and nonstationary across time. A great deal of effort has been devoted to characterizing this activity by using state-of-the-art methodologies, such as artificial neural networks, signal processing methods, adaptive filtering theory, parametric and nonparametric probabilistic models, Bayesian inference, metric-based analysis, and information-theoretic methods. Advances in technology have enabled us to record larger-scale neuronal ensemble activity, and current research has devoted a lot to integrating and analyzing increasingly large-volume, high-dimensional, and fine-grain experimental data.

The main focus of this special issue is to provide an international forum for researchers to present the most recent developments and innovative ideas in the field. We aim to incorporate new contributions in theories, algorithms, and applications. A total of 17 submissions, which cover a broad spectrum of spike train modeling and analysis, were received for this special issue. Each submission was rigorously reviewed by external referees as well as the Guest Editor. To ensure the high quality of papers, we finally accepted 6 articles for this special issue. The following is a brief summary for each of these accepted articles.

The paper “*An overview of Bayesian methods for neural spike train analysis*” by Z. Chen presents a tutorial overview of Bayesian methods and their representative applications in neural spike train analysis, at both single neuron and population levels. On the theoretical side, the focus is on various approximate Bayesian inference techniques as applied to latent state and parameter estimation. On the application side, the topics include spike sorting, tuning curve estimation, neural encoding and decoding, deconvolution of spike trains from calcium imaging signals, and inference of neuronal functional connectivity and synchrony.

The paper “*A tensor-product-kernel framework for multiscale neural activity decoding and control*” by L. Li et al. proposes a tensor-product-kernel framework for integrating multiscale neural activity and applies it to an offline stimulus decoding and an open-loop control task in brain-machine interface. Choosing the kernels is equivalent to identifying a proper spatiotemporal scale among neural activity such as spike trains and local field potentials. This work provides a general framework to leverage heterogeneous neural activities recorded from neuroscience experiments.

The paper “*Homogenous chaotic network serving as a rate/population code to temporal code converter*” by M. V. Kiselev addresses an important relationship between rate coding and temporal coding in neuroscience and shows that conversation from the rate code to temporal code can be implemented by a homogeneous chaotic neural network. The paper demonstrates this approach using simulated leaky

integrate-and-fire neurons and spiking network in line with the selective polychromatic neuronal groups. It shows that the quality of conversation is dependent on a wide range of parameters, such as the stimulus diversity, intensity, background noise, and excitatory connection delays.

The paper “*Prediction of human’s ability in sound localization based on the statistical properties of spike trains along the brainstem auditory pathway*” by R. Krips and M. Furst demonstrates that aspects of the human ability to perform auditory localization are compatible with a theory of optimal estimation applied to responses of the auditory nerve and superior olivary complex. The nature of this fit is suggestive of the sources of information used to perform this computation.

The paper “*Spike sorting by joint probabilistic modeling of neural spike trains and waveforms*” by B. A. Matthews and M. A. Clements develops a novel probabilistic method for automatic neural spike sorting which uses stochastic point process models of neural spike trains and parameterized action potential waveforms. A novel likelihood model for observed firing times as the aggregation of hidden neural spike trains is derived as well as an iterative procedure for clustering the data and finding the parameters that maximize the likelihood.

The paper “*Sparse data analysis strategy for neural spike classification*” by V. Vigneron and H. Chen deals with the problem of identifying single neuronal units from multichannel extracellular recordings. This study proposes a family of metrics based on different levels of parsimony to separate spike waveform data into single unit waveforms. It demonstrates that the proposed method can provide an effective spike-sorting tool to visually analyze the spike data and to produce robust results for noisy, imbalanced, and highly correlated spike waveform data.

In summary, these six papers offer many examples of active research topics in neural spike train analysis. We thank all authors for submitting their papers to this special issue. We also thank all reviewers for providing their expertise and making valuable comments during the reviewing process. It is our hope that these papers would provide novel ideas and methodological development to the neural spike train modeling community, and the results would be useful for better understanding of neuronal mechanisms in the brain and providing more effective applications.

Acknowledgment

We are grateful to the editors for hosting this special issue and their significant support during the editorial process.

Wei Wu
Asohan Amarasingham
Zhe (Sage) Chen
Sung-Phil Kim

Research Article

Sparse Data Analysis Strategy for Neural Spike Classification

Vincent Vigneron¹ and Hsin Chen²

¹ IBISC-Lab, Université d'Évry Val d'Essonne, 40 rue du Pelvoux, 91020 Courcouronnes, France

² Department of Electrical Engineering, National Tsing Hua University, No. 101, Sec. 2, Kuang-Fu Road, Hsin-Chu 30013, Taiwan

Correspondence should be addressed to Vincent Vigneron; vvigne@iup.univ-evry.fr

Received 31 July 2013; Accepted 8 April 2014; Published 2 July 2014

Academic Editor: Sung Phil Kim

Copyright © 2014 V. Vigneron and H. Chen. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Many of the multichannel extracellular recordings of neural activity consist of attempting to sort spikes on the basis of shared characteristics with some feature detection techniques. Then spikes can be sorted into distinct clusters. There are in general two main statistical issues: firstly, spike sorting can result in well-sorted units, but by with no means one can be sure that one is dealing with single units due to the number of neurons adjacent to the recording electrode. Secondly, the waveform dimensionality is reduced in a small subset of discriminating features. This shortening dimension effort was introduced as an aid to visualization and manual clustering, but also to reduce the computational complexity in automatic classification. We introduce a metric based on common neighbourhood to introduce sparsity in the dataset and separate data into more homogeneous subgroups. The approach is particularly well suited for clustering when the individual clusters are elongated (that is nonspherical). In addition it does need not to select the number of clusters, it is very efficient to visualize clusters in a dataset, it is robust to noise, it can handle imbalanced data, and it is fully automatic and deterministic.

1. Introduction

Neurophysiologists assume that the brain encodes information in the firing rate of neurons, that is, the number of “spikes” over a temporal interval. While many powerful imaging techniques have been used in neuroscience, extracellular recording remains the only choice that provides resolution of neuron activity in the brain. However, multiple extracellular recordings are useful only when the spikes generated by different neurons can be correctly sorted.

Lewicki [1] reviewed numerous methods that have been proposed to classify spikes. The usual assumptions for spike sorting are (1) that all spikes generated by a specific neuron are characterized by a similar waveform, (2) that this waveform is unique, and (3) that this waveform is conserved for each neuron during a stationary recording [2]. Analysis of neural recordings requires first detecting action potentials, *spikes*, from noise, which is achieved with thresholding discrimination by manual or semiautomatic classification methods. The second process is spikes sorting and produces a number of “spike trains” corresponding to the temporal sequence of real signals [3–5].

Among different methods used for spike sorting, template matching is one of the most popular procedures. The usual practice to produce templates is to use a “supervisor,” that is, an experienced and knowledgeable operator, to preliminarily classify the waveforms following a selection of templates corresponding to distinct neurons. Few methods have dealt with unsupervised template creation. Atiya [3] for instance used the Isodata clustering algorithm to estimate typical spike shapes and then compared all possible combinations of templates to find the combination with the highest likelihood. Letelier and Weber [6] applied Bayesian probability theory to quantify the probability of both the form and the number of spike shapes. Zouridakis and Tam [7] proposed a procedure based on fuzzy *k*-means clustering algorithms to create reliable spike templates. Some authors [8–10] used independent component analysis (ICA) for distinguishing the spikes according to their sources; the independence assumption of the firing neurons helps to identify spikes from the same source. In [11] the occurrence time information of spikes and features related to the shape simultaneously is applied to estimate the interspike interval for each neuron and sort the spikes using a Monte Carlo algorithm. Pouzat et al.

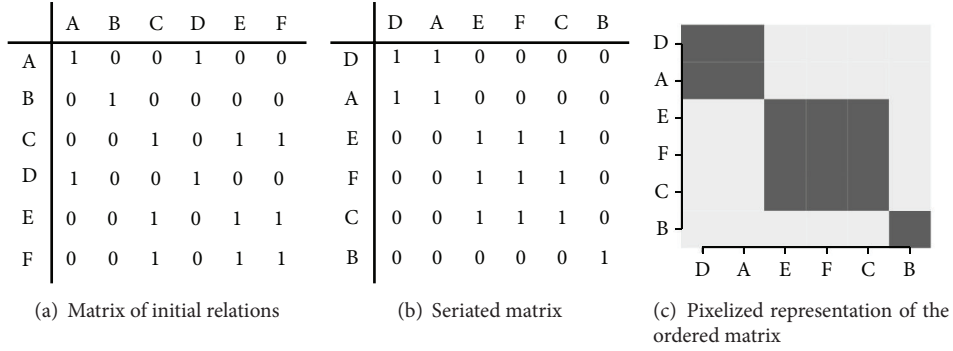


FIGURE 1: Effects of an algorithm of arrangement on a data set.

[12] used an empirical characterization of the recording noise to optimize the action potentials clustering and for assessing the quality of each cluster. Zhang et al. [13] reconstructed the spike templates according to the clustering results from principal component analysis (PCA) and subtractive clustering techniques. Probabilistic methods have been proposed [14, 15] and have focused on the modeling of each group in specific subspaces of low dimensionality.

Several approaches are associated with a visualization objective such as factorial methods [16, 17]. The latter methods can be *global* when they are based on the proximity between various groups such as graph methods or *local* when they evaluate the proximity between the individuals, like the hierarchical methods, they can also combine the local and global relations, as in the case of *seriation*.

A taxonomy of the methods was proposed by Carroll and Arabie [18] which associates a particular mode of seriation with each type of table.

Seriation aims to display and to reveal natural clusters and their dependencies in a dataset only by reordering rows and columns so that the adjacent rows and, respectively, columns are the most similar. This situation is illustrated by Figure 1 where, starting from a table of relations presented by Figure 1(a), the lines and the columns permuted to form a partition in which the similar elements were gathered together, thus forming groups (Figure 1(b)), and, in order to better appreciate the presence of the diagonal structure per block, this ordered matrix is pixelized (Figure 1(c)). Such an approach could be connected with a local technique of ordered clustering in so far as information is brought on one hand about the local relations between individuals because of an order in the data and on the other hand about the total structure of the data. Seriation has other advantages outlined by several authors such as Arabie et al. [19], like the no need of prior knowledge on the number of clusters and direct visualization of the structure on the table of values.

These advantages might disappear when the data are noisy or *imbalanced* or when groups of data are superimposed. The presence of noisy data prevents a clear visualization of the various blocks and distinguishing the clusters becomes a difficult task. Our approach is based on symmetric binary matrices of similarities (or dissimilarities) linked to common neighborhood. Such matrices indicate similarities

between pairs of observations and can be computed by different measures depending on the nature of the dataset such as Euclidean distances or more generally p -norm, correlation coefficients [20], or divergences [21] for example. A criterion derived from the problems of data compression selects the most compact ordered matrix—in the form of diagonal blocks—in order to obtain the most informative visualization off the intrinsic data structure. In some situations, too great a parsimony generates the ousting of underrepresented data forming very small clusters. To mitigate this nondetection, we propose a multiscale approach combining various levels of sparsity of the data.

This paper is organized in the following way: in Section 2, seriation is presented according to two different points of view, one as a mathematical optimization problem to be solved and the other on its algorithmic bases. Section 3 details our original approach as well as a multiscale algorithm of the proposed arrangement, called Parsimonious Block-Clustering. Experiments on simulations and benchmark data are presented in Section 4.

2. Method

2.1. The Optimization Problem. Seriation seeks an order in the data that reveals the locality/proximity between adjacent lines or columns to thus reveal a structure. This order is obtained by successive permutations of lines and columns which makes it possible to tackle seriation optimization problem (The number of possible combinations of permutations of lines and of columns is $n!p!$ for a rectangular $n \times p$ table or $n!$ in the case of a symmetrical matrix of dissimilarity.) through two different angles: one being to determine all the best possible permutations, the other being related to the complexity of the solution (np -complete problem).

The seriation approach can be applied to any type of matrices but we focus in this work on dissimilarity matrices. Let us consider a set of N samples (x_1, \dots, x_N) described by a symmetrical matrix of dissimilarity $D = (d_{ij})_{i,j \in (1, \dots, N)}$ of size $N \times N$ where each element d_{ij} gives a “measure” of dissimilarity between the pair of observations (x_i, x_j) . Let Ψ define a permutation function which orders the elements of matrix D , according to a given criterion C . The objective of

the seriation is thus to find the optimal permutation Ψ^* which optimizes the arrangement criterion C , such that

$$\Psi^* = \arg \max_{\Psi} \mathcal{C}(\Psi(D)). \quad (1)$$

These criteria are based on a measure of similarity $s(\cdot)$ between the successive elements of the matrix D and maximize $\max \sum_{i=1}^{n-1} s(i, i+1)$.

This measure of similarity is declined in a different way according to the authors as one can observe in Table 1. McCormick et al. [22] and Arabie and Hubert [23], for example, seek to maximize a *measure of effectiveness* (cf. \mathcal{C}_6 criterion in Table 1) based on the sum of the scalar products in lines and columns of the data matrix; this measure was generalized thereafter by Climer and Zhang [20]. Other authors, such as Hubert et al. [24] or Chen [25], based their optimization on the divergence measure between the matrix of dissimilarity and an anti-Robinson structure seeking to gather the values of the smallest dissimilarities around the diagonal (cf. \mathcal{C}_4 criterion). On the other hand, some authors such as Caraux and Pinloche [26] (cf. \mathcal{C}_1 and \mathcal{C}_2 criteria) or Brusco and Steinley [27] (cf. \mathcal{C}_3 criterion) rather seek to place the smallest dissimilarities out of the diagonal (Robinson structure). Lastly, in the framework of data compression, Johnson et al. [28] proposed to minimize a criterion based on the number of sequences of consecutive elements (on a line) different from 0 (cf. \mathcal{C}_5 criterion). Many authors proposed new criteria of arrangement like Niermann [29] who seeks to compare each observation with its adjacent neighbors through vicinity criteria (cf. \mathcal{C}_7 criterion) or Batagelj [30] or Doreian et al. [31] who propose criteria of structural equivalence or Dhillon et al. [32] who use mutual information and an entropy-based criterion.

These recent approaches require a *prior* knowledge of the number of clusters formed by the individuals and the variables whose determination is not trivial.

2.2. A Family of Embedded Binary Matrices. To deal with the problem of imbalanced datasets, noisy data, overlapping clusters, and outliers, we propose a new algorithm based on a family of embedded binary matrices which stands for different degrees of sparsity of the data. The binary matrices are ordered according to an algorithm named Parsimonious Block-Clustering (PB-clus). This algorithm makes it possible to select the level of parsimony to produce the optimal compact block structure.

In our approach, the degree of vicinity is defined as a “threshold value” equal to the number of common neighbors between pairs of observations after which pairs of observations are eliminated. The larger the number of common neighbours imposed is, the more parsimonious the matrix will be (filled with zeros). Hence, the degree of parsimony is associated with the degree of common vicinity. Let us consider a data matrix X with elements in \mathbb{R}^p and $X^d = (x_{ij}^d)$, $i, j \in \{1, \dots, n\}$ the dissimilarity matrix associated to X , the choice of the distance function depending on the type of data: it can be an Euclidean distance between individuals i and j (and more generally p -norm), a correlation, or any other function characterizing the concept of proximity between

pairs of observations (see Table 1). Let $A = (a_{ij})$, $i, j \in \{1, \dots, n\}$, and the (0,1)-matrix with elements

$$a_{ij} = \begin{cases} 1 & \text{if } x_{ij}^d \leq \epsilon \\ 0 & \text{if } x_{ij}^d > \epsilon, \end{cases} \quad (2)$$

where ϵ is the threshold characterizing the proximity of the pairs of observations. Its value can be given arbitrarily; we propose to fix it at the first quartile of the distribution of the distances between pairs of observations. In addition, the matrix of similarity is symmetrical; that is, $a_{ij} = a_{ji}$. Let the Gram matrix $B = A^T A$ where each element b_{ij} is the number of neighbors of the two data i and j . This matrix corresponds to a matrix of common vicinity.

Definition 1. A binary matrix $B_{\lambda_m} = (b_{ij}^{\lambda_m})$, $i, j \in \{1, \dots, n\}$, parsimonious with a degree λ_m (with $m \in \{1, \dots, M\}$) is characterized by

$$b_{ij}^{\lambda_m} = \begin{cases} 1 & \text{if } b_{ij} \geq \epsilon \\ 0 & \text{if } b_{ij} < \epsilon, \end{cases} \quad (3)$$

where b_{ij} represent the elements of the Gram matrix B defined previously. The set $(B_{\lambda_m}, \dots, B_{\lambda_M})$ forms a family of binary matrices whose level of parsimony is related to the number of common neighbors.

Taking into consideration this definition, the greater λ_m the fewer the number of pairs of observations which satisfy this condition. The associated matrix will contain a greater number of zeros and will thus be more parsimonious. The sequence $(\lambda_m)_{m \in \{1, \dots, M\}}$ such that $\lambda_1 < \dots < \lambda_M$ makes it possible to establish an order relation \subset between the M elements of the set $B_{\lambda_m} \text{ } m \in \{1, \dots, M\}$:

$$B_{\lambda_M} \subset B_{\lambda_{M-1}} \subset \dots \subset B_{\lambda_1}, \quad (4)$$

in which the most parsimonious matrix is contained in all the other matrices of its family. One of the advantages of such a matrix is the cancellation of the extreme values and of the noise when the level of parsimony increases, which facilitates the arrangement of the matrix as well as the appearance of adiaagonal block structure. In relation to this family of matrices, a question remains: how to obtain the “best” level of parsimony, that is, the one which will make it possible to obtain a comprehensive visualization of the data structure?

The ordered matrix $B_{\lambda_m, \text{ord}}^* = (b_{ij, \text{ord}}^{\lambda_m})_{\text{ord}}$, $i, j \in \{1, \dots, n\}$, $m \in I$ with the set $I \in \{1, \dots, M\}$ contained in a set of ordered matrices, verifies that

$$B_{\lambda_m, \text{ord}}^* = \arg \min_{m \in I} \mathcal{C}_{\lambda_m} = \arg \min_{m \in I} \sum_{i=1}^{n-2} \sum_{j=i+1}^{n-1} \frac{|b_{ij, \text{ord}}^{\lambda_m} - b_{i(j+1), \text{ord}}^{\lambda_m}|}{|b_{ij}^{\lambda_m} - b_{i(j+1)}^{\lambda_m}|}. \quad (5)$$

This criterion is based on the idea that the fewer the alternations between the 0 and the 1 on the lines of the matrix considered, the more compact a structure this matrix

TABLE 1: Criteria of arrangement used within the framework of the clustering one-mode.

Type	Criterion to optimize depending on the dissimilarity matrix $D = \{d_{ij}\}_{i,j \in \{1, \dots, n\}}$
Structural criteria	$C_1 = \sum_{i=1}^n \sum_{j=1}^n d_{ij} i - j ^2$
	$C_2 = \sum_{i=1}^n \sum_{j=1}^n (d_{ij} - \alpha i - j ^2)$
	$C_3 = \sum_{i=1}^{n-2} \sum_{j=i+1}^{n-1} \sum_{k=j+1}^n (d_{ij} - \alpha i - j ^2)$
	$C_4 = \sum_{1 \leq i < j < k \leq n} f(d_{ik}, d_{ij}) + \sum_{1 \leq i < j < k \leq n} f(d_{kj}, d_{ij}) \quad \text{with}$
	$f(x, y) = \text{sign}(x - y)$
	$f(x, y) = x - y \text{sign}(x - y)$
	$f(x, y) = \mathbb{I}_{x > y}$
	$f(x, y) = x - y \mathbb{I}_{x > y}$
	$C_5 = \sum_{j=i+1}^{n-1} \sum_{k=j+1}^n d_{ij} - d_{i,j+1} $
	$C_6 = \sum_{i,j=1}^n d_{ij} (d_{i,j-1} + d_{i,j+1} + d_{i+1,j} + d_{i-1,j})$
Similarity criteria	$C_7 = \sum_{i,j=1}^n f_{ij} \quad \text{with}$
	$f_{i,j} = \sum_{k=\max(1,i-1)}^{\min(n,i+1)} \sum_{\ell=\max(1,j-1)}^{\min(n,j+1)} (d_{ij} - d_{k\ell})^2$
	$f_{i,j} = \sum_{k=\max(1,i-1)}^{\min(n,i+1)} (d_{ij} - d_{kj})^2 + \sum_{\ell=\max(1,j-1)}^{\min(n,j+1)} (d_{ij} - d_{i\ell})^2$

will have. Indeed, in Table 2, if one considers the quantity $\sum_{i=1}^n \sum_{j=1}^{n-1} |b_{ij, \text{ord}}^{\lambda_m} - b_{i(j+1), \text{ord}}^{\lambda_m}|$ accounting for the number of changes between the 0 and the 1 of an ordered matrix of degree λ_m and the quantity $|b_{ij, \text{ord}}^{\lambda_m} - b_{i(j+1), \text{ord}}^{\lambda_m}|$ associated with the nonarranged matrix of the same degree, it is notable that the number of changes between the 0 and the 1 stays smaller in the case of the ordered matrices. As the degree of parsimony increases, the number of alternations between the 0 and the 1 falls: in the example, the numerator $\sum_{i=1}^n \sum_{j=1}^{n-1} |b_{ij, \text{ord}}^{\lambda_m} - b_{i(j+1), \text{ord}}^{\lambda_m}|$ is equal to 9 for a level $\lambda = 1$ and to 3 when the degree of parsimony is 3. In order for the selection criteria not to be biased in favour of an infinite sparsity, \mathcal{E}_{λ_m} is standardized by the number of alternations between the 0 and the 1 of the nonordered binary matrix associated with the same degree of parsimony. Thus, according to the example of Table 2, the level of parsimony retained is $\lambda \geq 2$.

Let us note that, at this level, a structure with two groups is selected and a piece of data that can be regarded as extreme data is excluded. This criterion derives from the concept of *run* used in *data compression* [28, 33], characterizing the biggest sequences of 1 on a line in a Boolean matrix. The chosen criterion \mathcal{E}_{λ_m} is related to the full number of changes present in the nonordered binary matrix of the same degree of parsimony so that it is not skewed in favour of an infinite parsimony or conversely, of too low a parsimony.

2.3. The Pb-Clus Geometry-Based Criterion. There are a plethora of criteria for the task of seriation [34] but the reordering algorithm that we proposed is based on the inner product because of its geometric interpretation. Since our work is based on symmetric matrices, the Tanimoto's norm (is also based on the dot product but adapted for binary data.) defined by $x_i^T x_j / (x_i^T x_i + x_j^T x_j - x_j^T x_i)$ can be used for binary matrices B^{λ_m} of parsimony degrees λ_m ($\forall m \in \{1, \dots, M\}$) defined in Section 2.2.

The permutation function Ψ which seeks to optimize the sum of the consecutive scalars can be written as

$$\Psi^* = \arg \max_{\Psi} \sum_{i=1}^{n-1} \frac{b_{\Psi(i)}^{\lambda_m T} b_{\Psi(i+1)}^{\lambda_m}}{\|b_{\Psi(i)}^{\lambda_m}\| + \|b_{\Psi(i+1)}^{\lambda_m}\| - b_{\Psi(i)}^{\lambda_m T} b_{\Psi(i+1)}^{\lambda_m}}. \quad (6)$$

This criterion is based on the principle of *connected components*: when several observations share the same neighborhood then these observations will belong to the same cluster or to the nearest clusters. The algorithm is based on a branch and bound method meaning that an exhaustive search is made in various subsets that are determined by the geometric properties of the dot product: the algorithm first searches the independent vectors which the separated clusters produce, then considers the connected component of each of these vectors and finally, and reorders the correlated vectors in each group. These steps can be done for a binary neighborhood matrix B_{λ} with level λ in the following way.

TABLE 2: An example of calculation of the C_{λ_m} criterion calculated from the matrix of the introductory example cf Figure 1(a).

Matrix of common neighbors	$B = \begin{matrix} 2 & 0 & 0 & 3 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 3 & 3 \\ 2 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 & 3 & 3 \\ 0 & 0 & 3 & 0 & 3 & 3 \end{matrix}$																							
	Parsimony level						$\lambda \geq 1$						$\lambda \geq 2$						$\lambda \geq 3$					
	Binary matrices of common neighbors						$B = \begin{matrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{matrix}$						$B = \begin{matrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{matrix}$						$B = \begin{matrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{matrix}$					
$\sum_{i=1}^6 \sum_{j=1}^5 b_{i,j}^{\lambda_m} - b_{i,j+1}^{\lambda_m} $						17						15						9						
(calculus per line)						(3 + 2 + 3 + 3 + 3 + 3)						(3 + 0 + 3 + 3 + 3 + 3)						(0 + 0 + 3 + 0 + 3 + 3)						
Sorted binary matrices						$B_1^{\text{sor}} = \begin{matrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{matrix}$						$B_2^{\text{sor}} = \begin{matrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$						$B_3^{\text{sor}} = \begin{matrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$						
$\sum_{i=1}^6 \sum_{j=1}^5 (b_{i,j}^{\lambda_m})_{\text{sort}} - (b_{i,j+1}^{\lambda_m})_{\text{sort}} $						9						85						3						
(calculus per line)						(1 + 1 + 2 + 2 + 2 + 1)						(1 + 1 + 2 + 2 + 2 + 0)						(1 + 1 + 11 + 0 + 0 + 0)						
Criterion C_{λ_m}						$C_{\lambda_m \geq 1} = 9/17 = 1.89$						$C_{\lambda_m \geq 2} = 8/15 = 1.88$						$C_{\lambda_m \geq 3} = 3/9 = 3.03$						

- (1) Compute a matrix of dot products (inner products or Tanimoto's product) for each pair of columns of X_λ without considering the columns full of zeros.
- (2) Select a column and find its connected components. Then find an orthogonal vector of the previous column and extract its connected components. This procedure is performed until there are no more vectors. In this way, several independent submatrices are built.
- (3) In each submatrix, place the most correlated vector alongside the first column and keep on doing this process until the submatrix is reordered.
- (4) Gather the rearranged submatrices and apply this order to B_λ .

The most informative visualization in terms of block-matrix is derived from the concept of *run* in compression approaches which characterizes a maximal sequences of nonzero entries in a row of a Boolean matrix [33]. It is intuitive that the fewer changes between series of ones and zeros are on each row the better the reordered matrix is. Since the sizes of the binary neighborhood matrices are different, this quantity is normalized by the minimum between the number of zeros or the number of ones of each rows so that

$$\mathcal{E}_\lambda = \sum_{i=1}^{n_\lambda} \frac{\text{card}_i(0, 1) + \text{card}_i(1, 0)}{\min(\text{card}_i(0, 0), \text{card}_i(1, 1))}, \quad (7)$$

where n_λ is the number of nonzero columns of the reordered matrix B_λ .

The algorithm enables us to find all the connected components of a cluster and to display relationships between clusters. This algorithm is straight forward deterministic algorithm, meaning that for a current move, the previous permutations are not challenged. Such an approach does not pretend to be optimal compared with the other approaches proposed in the literature but remains efficient and very fast even for large datasets and performs well when the data are noisy.

Since the proposed algorithm is a forward procedure (see Table 1), the final rearrangement obtained depends strongly on the first column selected in each submatrix. To deal with this problem, we propose to select a central observation for each submatrix to initialize the algorithm. The initialization is based on the idea that if we find a central observation in each cluster, then all connected components can be gathered. So, the first column is selected according to the number of strong correlated vectors which has to be maximum.

Lastly, Pb-Clus has a higher cost of calculation than the other methods of seriation since the arrangement is carried out not on only one matrix but on M matrices relative to different degrees of parsimony. In the case of a matrix of size $n \times n$ with K groups of same size n/K , there are at most $K(n/K)!$ calculations. As the degree of parsimony increases, the matrix is filled with columns (lines) of zeros, which decreases the number of elements to be arranged, and

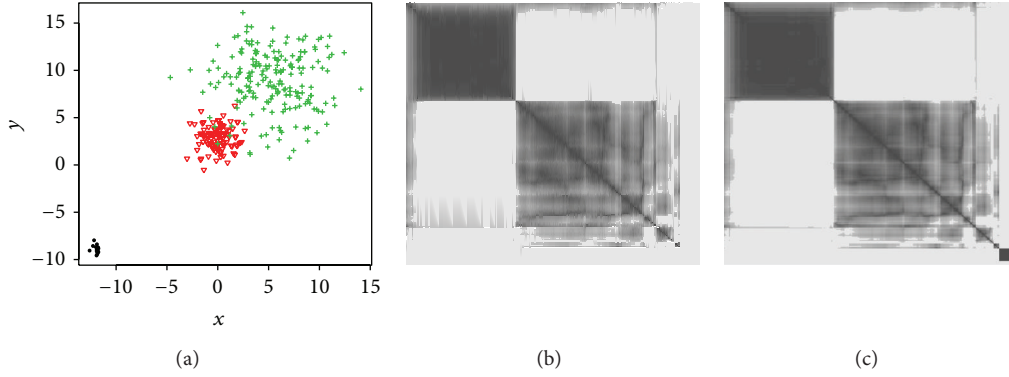


FIGURE 2: (a) Projection of the data in their space (b) visualizations of the central matrix ordered by Pb-Clus with a sparsity level of 16 common neighbors and (c) 8 common neighbors.

consequently the computing time. The calculation cost would remain significantly lower than $M \cdot K(n/K)!$.

3. Experiments on Simulated Data

3.1. Case of Non-Separated Clusters. For this experiment, the data are simulated from three different 2-dimensional Gaussian mixtures with large variances and two clusters are superposed as illustrated in Figure 2(a). The first cluster is formed of 5% of the data (15 observations) while the two others account for 32% and 63% of the data, respectively (i.e., 100 and 200 observations). The central partition linked to this situation is represented in Figure 2(b) with a sparsity threshold of 16 common neighbors.

For this level of parsimony, more than 6% of the data were excluded which results in the removal of the smallest cluster. For a level of 8 common neighbors, it is possible to recover the third cluster.

Even if the central visualization Figure 2(c) is a bit less clear than previously, it is still informative and three different clusters can be seen. Moreover, the superposition of two clusters can be identified since in the central visualization, the two relative squares are inscribed in a bigger square which means correlations or proximities between these two groups. Lastly, among the seriated data, 98% have been correctly classified.

3.2. Influence of the Level of Superimposition of Clusters. In this second experiment, we seek to evaluate the influence of the level of covering of clusters in the search for a data structure. With this intention, we simulated 3 Gaussian distributions in a 2-dimensional space so that their respective averages check: $m_1 = (x, y)^T$, $m_2 = (x, -y)^T$, $m_3 = (0, -y)^T$ with $y \in [0, 0.3]$, and $y \in [0, 0.225]$. Consequently, the relative position of the averages varies and this variation determines the level of superposition of the groups. Thus, when $x = 0$ and $y = 0$, the 3 groups are mixed and that corresponds to a superposition of 100%. In the opposite case of separate groups where the covering rate is zero, the averages of the clusters check: $m_1 = (0.3, 0.225)^T$, $m_2 = (0.3, -0.225)^T$, $m_3 = (0, -0.225)^T$. Table 3 presents the

evolution of the sparsity level and its associated ousting rate, according to the covering of the groups.

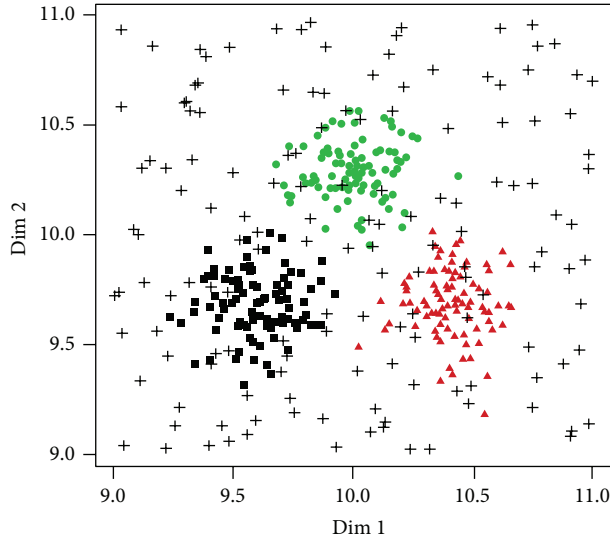
First of all, one notices that the greater the superposition of the clusters is the more the \mathcal{C}_λ criterion selects a parsimonious representation of the data. Indeed, when the visible data structure becomes less marked, this effect is balanced by a greater sparsity in the data with a bigger common vicinity. In the same way, as the data structure becomes increasingly complex, the rate of classification related to the subsets of seriated data decreases as well as the quality of visualization. In our example, beyond a rate of covering of the data of 40%, the rate of classification becomes weak (<60%) since the algorithm Pb-Clus no longer detects a structure in the data and this, whatever the level of parsimony imposed.

3.3. Case of Noisy Data. In this experiment, 30% of the data are replaced by a uniform noise in a hypercube $[-1, 1]^4$ and the rest of the data are distributed from a mixture of three closed four dimensional Gaussian distributions as illustrated in Figure 3(a). Figure 4(c) depicts the central visualization which brings out a natural structure of three clusters in the dataset even if the data are noisy.

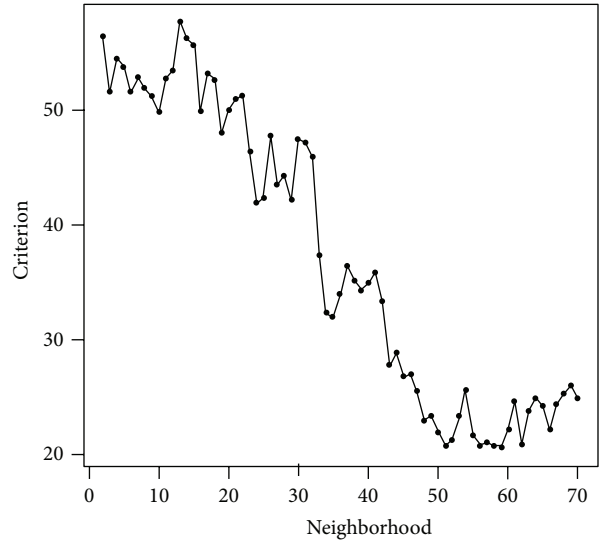
Figure 3(b) presents the evolution of the compactness criterion \mathcal{S}_λ according to the various degrees of parsimony, namely, the number of common neighbors. The central partition (Figure 3(c)) selected is the one for which the \mathcal{C}_λ criterion is minimal. This corresponds to a common vicinity of 59. This sparsity results in the ousting of 16% of the data and only 84% of the initial data make it possible to obtain a block diagonal representation; the subsets of excluded data are entirely made of noisy data. The rate of correct classification among the seriated data amounts to 99%, which implies that these subsets of seriated data are a structural visualization of the 3 clusters. In order to evaluate the performance of our approach, three methods of seriation based on distance matrices were applied: hierarchical clustering (HC) for the seriation (Figure 4(a)), the approach of Chen based on an anti-Robinson structure [25] (Figure 4(b)), and another method of anti-Robinson seriation by simulated annealing [35] (Figure 4(c)).

TABLE 3: Influence of the degree of covering of the clusters on the structure detection.

Degree of covering (en%)	0	6.7	13.3	20.0	26.6	33.3	40.0	46.6	53.30	73.3	100
x	0.30	0.28	0.26	0.24	0.22	0.10	0.18	0.16	0.14	0.08	0
y	0.225	0.21	0.195	0.18	0.165	0.15	0.135	0.120	0.09	0.06	0
Degree of parsimony	5	6	9	35	33	34	35	35	35	35	34
% of evinced values	0.00	0.00	0.00	0.26	0.23	0.34	0.37	0.35	0.39	0.35	0.43
Value of C_λ	1.95	2.01	2.42	2.64	2.90	2.82	3.34	3.29	3.32	3.54	3.65
Classification rate	0.99	0.99	0.99	0.99	0.95	0.90	0.86	0.60	0.49	0.44	0.39



(a) Visualizations of the Gaussian mixture disturbed in the data space

(b) Evolution of the \mathcal{E}_λ criterion according to the number of common neighbors

(c) The central matrix ordered by Pb-Clus with a sparsity threshold of 67 common neighbors

FIGURE 3: Seriation in the case of noisy data.

Among the methods of seriation used, we notice that only the central partition provided by Pb-Clus brings a clear visualization of the three clusters. The representation of this structure in three distinct groups is possible thanks to the family of parsimonious binary matrices. Indeed, the higher the degree of parsimony in the matrices, the greater the decrease in the quantity of noisy data taken into account.

3.4. Influence of the Noise Level. This second experiment aims to demonstrate the behavior of Pb-Clus in the case

of very noisy data. For this purpose, we simulated three 2-dimensional Gaussian distributions of 50 observations each with the following means $m_1 = (-0.4, -0.3)^T$, $m_2 = (-0.4, -0.3)^T$, and $m_3 = (0, 0.3)^T$, respectively, and matrix of variance-covariance $S = \text{diag}(0.1, 0.1)$. These groups are voluntarily separated in order to be able to evaluate the sensitivity of the algorithm to the noise. The noisy data were generated according to a uniform law on the support $[-1, 1]^2$. To evaluate the impact of the noise on visualization, we varied the quantity of noise from 10% to 200% of the number of data

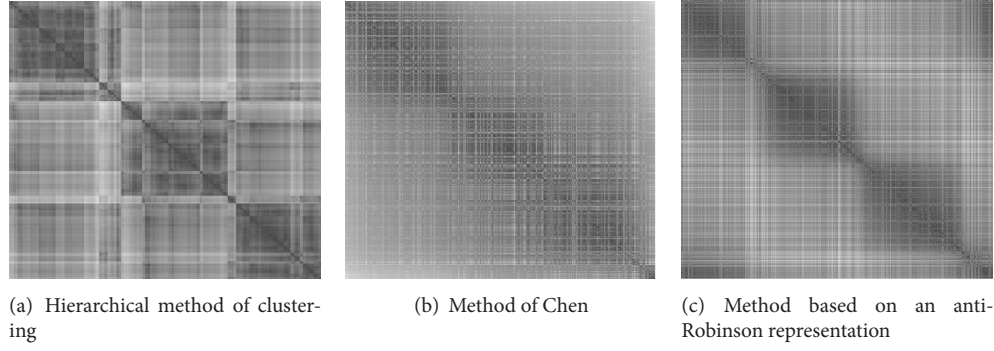


FIGURE 4: Visualizations of the pixelized distance matrix seriated.

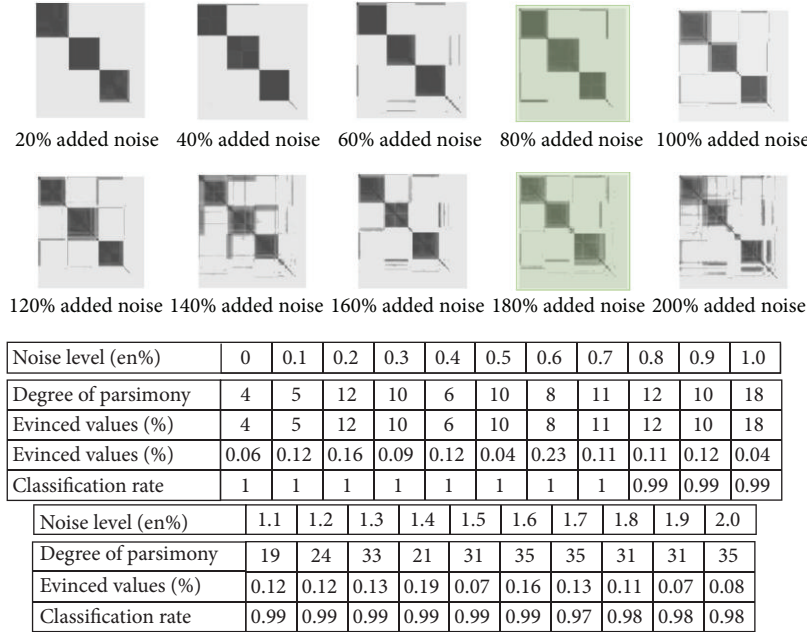


FIGURE 5: Visualization of the data structures according to various levels of noise added to the initial data.

in the initial sample. Figure 5 presents how the visualization of the data evolves with additional noise.

One notes that the group visualization degrades little with the additional noise. Indeed, in Figure 5, the structure is degraded only when the disturbed data represent more than half of the whole data.

3.5. Comparison on Classical Datasets. In this section, we compare the performance of PB Clus in terms of visualization firstly with two other methods of seriation, one using hierarchical classification (HC) and the other using a criterion of divergence related to an anti-Robinson structure described in Hahsler et al. [36] and, secondly, with an unsupervised classification method based on the Euclidean distances, the k -means. The 5 chosen datasets are detailed below.

- (i) Fisher's irises database collects 3 different species of iris in the Gaspé peninsula: *setosa*, *virginica* and the *versicolor*. Each species is represented by 50 flowers

which are described by 4 morphometric characteristics based on the width and the length of their sepals and their petals. This database is extremely popular in the statistical community because of difficulty of distinguishing the *virginica* and the *versicolor*.

- (ii) The *ruspini* data come from work of Ruspini [37] on clustering: they are made of 75 points in 2 dimensions and divided into 4 homogeneous and balanced classes.
- (iii) The *townships* data are binary data reporting the presence or the absence of 9 descriptive characteristics of 16 cities, such as the presence or the absence of universities, agricultural cooperatives, and railroads. There is no information on the number of groups structuring the data.
- (iv) *Old Faithful geyser data* evaluate the time between two eruptions of geysers of the national park of

TABLE 4: Comparison of 3 methods of seriation, PB-Clus, HC, and Chen approach according to Moore and Neumann criteria on the data benchmarks.

Method	PB-Clus seriation		HC seriation		Chen seriation	
Criterion	Moore	Neumann	Moore	Neumann	Moore	Neumann
Dataset						
Iris	1 371.2	471.1	31 728.8	10 893.1	19 357.8	7 304.0
Townships	244.5	91.8	1 109.9	441.5	849.0	342.0
Ruspini	1 290.1	442.2	8 724.9	3 036.4	6 503.7	2 277.1
Faithful	2 634.1	889.4	34 045.5	11 503.5	23 390.0	9 894.2
Geysers	2 514.9	850.4	68 205.3	2 302.1	12 866.8	4 501.4

Yellowstone of Wyoming (USA) and their duration. They are characterized by 272 observations [38].

- (v) The *geysers* data represent a full version of the preceding data that were collected by Azzalini and Bowman [39]. These relate to the 299 eruptions which were studied (same types of measurements as previously) between 1st and 15th August, 1985.

The quality of the visualization is calculated from two criteria proposed by Niermann [29] and presented in Section 2; the partition obtained will be evaluated by cross-validation with the true label when available or with the labels estimated by the k -means. As the latter supposes a prior knowledge of the number of groups of the mixture, we use the number of clusters detected by Pb-Clus in order to obtain comparable partitions.

The right-hand column of Figure 6 represents the consecutive dot products of elements i and $i + 1$ ordered out of the 5 previous databases. These curves of consecutive dot product give an evaluation of the proximity between two adjacent observations and points of rupture for the passage of one cluster to another, which makes it possible to select the number of clusters in the mixture and to obtain a partition of the data. In Figure 6 the left-hand column of represents the central visualization of the parsimonious matrix ordered with the algorithm PB-Clus. In the case of the Fisher's irises, the observation of its central matrix of degree of common vicinity 8 shows a total structure of two clusters.

One finds here the particular structure of the irises in which the *versicolor* and the *virginica* are not very distinct species. In addition, this partition in 3 groups is confirmed by the 2 break points present on the curves of its consecutive dot products. These 2 graphs demonstrate the performance of our parsimonious approach for the visualization of the data, especially as the methods of clustering which select one optimal model with 3 iris classes are rare (cf. mixture models of Raftery and Dean [40]). In the case of the *Ruspini* and the *Old Faithful* data, ruptures on the curve of the consecutive dot products are clear and large which show the total disconnection of the clusters between them. The same conclusion is visible on their ordered central matrix of degree 5 for the *Ruspini* data and of degree 2 for the *Faithful* data.

On the contrary, the *Geysers* and the *Townships* data present small breaking points. In the case of *Geysers* data, they are explained by the proximity of the clusters. Then, in

the case of the *Townships* data, the curve of the consecutive scalars shows that the first city is, certainly, connected to the 7 following cities but less strongly than these 7 cities between each other. The central visualization of the parsimonious ordered matrix of degree 2 with Pb-Clus brings a better comprehension of the relationships between the cities. Indeed, it is noticed that the first data is strongly correlated with two distinct blocks of cities. This is confirmed by an analysis of Hahsler et al. who showed the existence of a structure with 3 groups: urban cities, country towns, and transition cities. This first evaluation based on our visual perception is supplemented by the measure of quality based on seriation criteria evaluating the vicinity in the ordered matrix. Table 4 evaluates the performances of 3 methods of seriation, the best method being the one whose criterion is minimum. It is noticed that the 2 criteria of Niermann are minimum for a parsimonious approach for all the databases.

Lastly, Table 5 presents the tables of cross-classification with the true label for the irises of Fisher and with the labels obtained by k -means in the case of the data *Ruspini*, *Townships*, *Geysers*, and *Faithful*. Let us note that in the case of the *irises* and *Geysers* data, we threshold the scalars in order to obtain a label for each data. Concerning the Fisher irises, the correct classification rate of PB-Clus is 89.0%, slightly weaker than that obtained by the k -means (90.6%). This difference in rate is related to the data located at the intersection of the *virginica* and the *versicolor* and with initialization of our algorithm. For the other data files, one observes that the partitions obtained by Pb-Clus and the k -means agree almost perfectly, the rates of classification bordering 98%.

4. Experimental Methods

In this section, we approach the task of classifying spike waveforms using PB-Clus.

4.1. Animal Training and Behavioral Tasks. The detection of neural spike activity is a technical challenge that is a prerequisite for studying many types of brain function (for more details see Vigneron et al. [41]).

The study, approved by the Institutional Animal Care and Use Committee at the National Chiao Tung University, was conducted according to the standards established in the Guide for the Care and Use of Laboratory Animals. Four male rats weighing 250–300 g (BioLASCO Taiwan Corp., Ltd.)

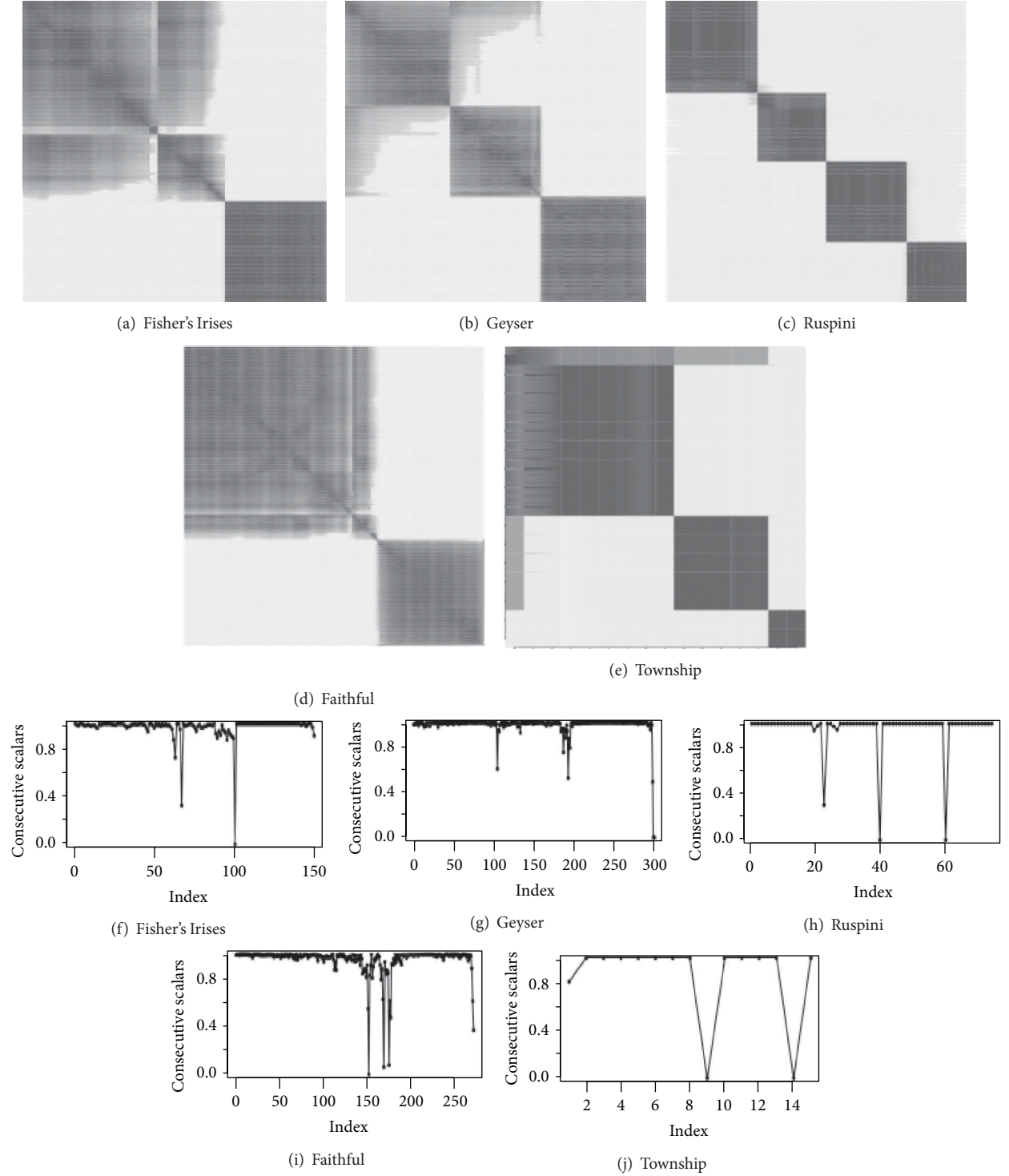


FIGURE 6: (a)–(e) rearranged matrices obtained with the PB-Clust (f)–(j) consecutive scalars resulting from the rearranged matrices.

were individually housed applying a 12 h light/dark cycle, with access to food and water *ad libitum*.

Dataset was collected from the motor cortex of awake animals performing a simple reward task. In this task, male rats (BioLACO Taiwan Co.,Ltd) were trained to press a lever to initiate a trial in return for a water reward. The

animals were water restricted 8-hours/day during training and recording session but food was always provided to the animal *ad lib* every day.

4.2. Chronic Animal Preparation and Neural Ensemble Recording. The animals were anesthetized with pentobarbital

TABLE 5: Tables of cross-validation of the data benchmarks.

(a)												
Known classes	Fisher Iris				Known classes	Township						
	Clusters PB-Clus			Clusters k -means		Known classes	Clusters PB-Clus					
	1	2	3				1	2	3	4		
Setosa	50	0	0	Setosa	50	0	0	Urban cities	8	0	0	0
Versicolor	0	50	0	Versicolor	0	49	1	Transitions	0	4	0	0
Virginica	0	17	33	Virginica	0	13	37	Country towns	0	0	2	0
								Unclassified	0	1	0	1
Classification rate = 0.88				Classification rate = 0.90				Classification rate = 0.94				
(b)												
Clusters k -means	Ruspini				Clusters k -means	Faithful		Clusters k -means	Geysers			
	Clusters PB-Clus					Clusters PB- Clus	Clusters PB-Clus					
	1	2	3	4					1	2	1	2
Group 1	50	0	0	0	Group 1	168	4	Group 1	88	2	7	
Group 2	0	35	0	0	Group 2	0	100	Group 2	0	105	0	
Group 3	0	0	15	0				Group 3	0	0	97	
Group 4	0	0	0	20								
Classification rate = 1.00				Classification rate = 0.98				Classification rate = 0.97				

(50 mg/kg i.p.) and placed on a standard stereotaxic apparatus (Model 9000, David Kopf, USA). The dura was retracted carefully before the electrode array was implanted. The pairs of 8 microwire electrode arrays (no.15140/13848, 50 m in diameter; California Fine Wire Co., USA) were implanted into the layer V of the primary motor cortex (M1). The area related to forelimb movement is located anterior 2–4 mm and lateral 2–4 mm from bregma. After implantation, the exposed brain should be sealed with dental acrylic and a recovery time of a week is needed.

During the recording sessions, the animal was free to move within the behavior task box (30 cm × 30 cm × 60 cm), where rats only pressed the lever via the right forelimb, and then they received 1-mL water reward as shown in Figure 7. A multichannel Acquisition Processor (MAP, Plexon Inc., USA) was used to record neural signals. The recorded neural signals were transmitted from the head-stage to an amplifier, through a band-pass filter (spike preamp filter: 450–5 kHz; gain: 15,000–20,000), and sampled at 40 kHz per channel. Simultaneously, the animal's behavior was recorded by the video tracking system (CinePlex, Plexon Inc., USA) and examined to ensure that it was consistent for all trials included in a given analysis.

4.3. Preprocessing. Neural activity was collected from 400–700 ms before to 200–300 ms after lever release for each trail. Action potentials (spikes) crossing set thresholds were detected and sorted and the firing rate for each neuron was computed in 33 ms time bins. Since the signals are collected with 10 nanometers invasive probes, the noise effects are limited.

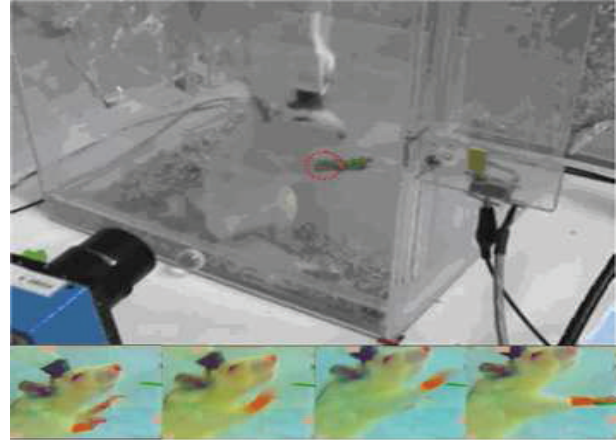


FIGURE 7: The experimental setup (top). Light-color (red virtual ring) was belted up the right forelimb to be recognized the trajectory by video tracking system. The sequence images captured the rat performing the lever press tasks in return for a reward of water drinking (bottom).

The experiment was made on 16 channels which collected EEG signals from microprobes which are implanted in the layer V of the M1 region of a rat.

4.4. Manual Scatterplot Classification. A method for classification is by plotting a selection of 2 or 3 spike features in a scatter diagram. This results in a 2- or 3-D graph with separate groups. The groups can only be assigned when there is enough spacing between the groups. Elliptic shaped areas are drawn around the groups isolating the classes.

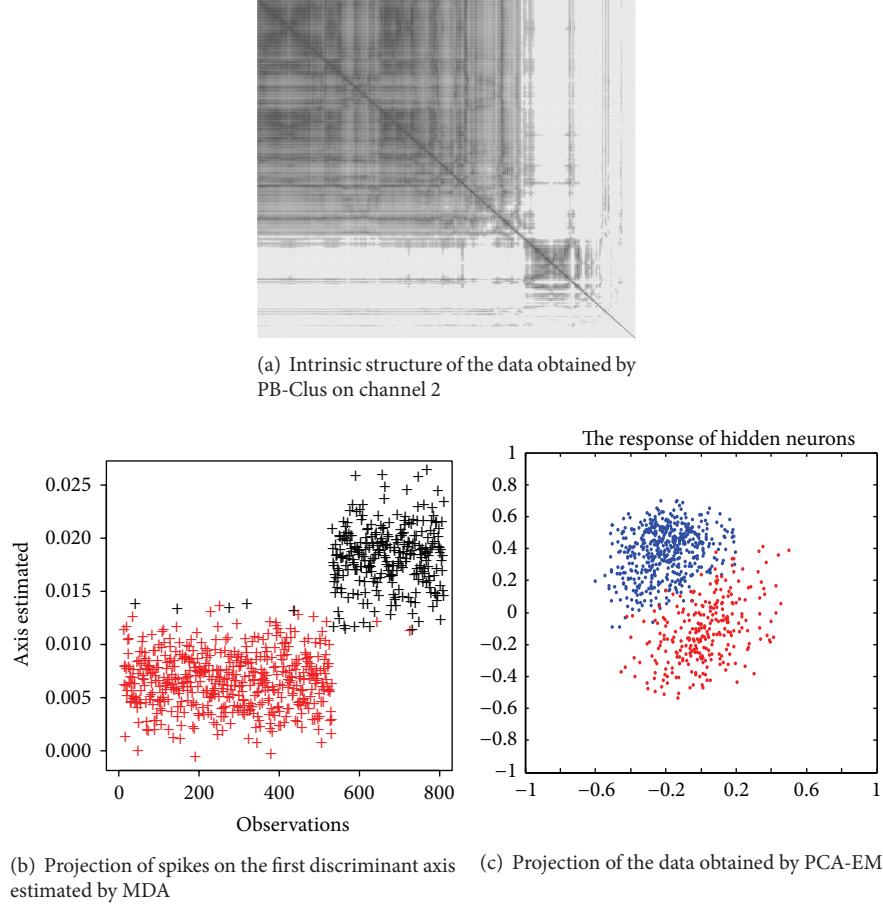


FIGURE 8: Results obtained by PB-Clus and 2 unsupervised approaches MDA and PCA-EM on channel 2.

TABLE 6: Number and type of spikes recorded in the 16 channels.

Channel	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Number of spikes	—	799	60	405	727	489	300	229	475	224	533	538	21	1833	1491	421
Types of spikes	—	2	2	2	1	2	4	1	1	2	2	2	1	2	1	4

4.5. Spike Waveforms Classification. To both reduce the size of these patterns and to cluster the spike mixture in a finite number of classes, we use two different tools: a seriation approach (PB-Clus) and a subspace clustering approach [42], named MDA (Mixture Discriminant analysis). Statistical discriminant analysis methods such as MDA aims to find both a parsimonious and discriminative fit for the data in order to ease the clustering and the visualization of the clustered data in a Gaussian mixture model context. MDA, developed by Hastie and Tibshirani [43], is a generalization of LDA (Linear Discriminant Analysis) in which each class is modeled by a mixture of Gaussians (see [44, chp. 4] for more details). This modelization gives more flexibility in the classification rule than LDA and allows MDA to take into account heterogeneity in a class. Breiman et al. [45], MacLachlan and Basford [46] have actually contributed and tested this generative approach on many fields. This latent subspace orientation is chosen such as it best discriminates

the groups. The quality of the partition obtained by both approaches will be measured by the Fisher index which is defined by the ratio between the within (S_w) and the between (S_B) scatter matrices:

$$F_{\text{index}} = \frac{S_w}{S_B} = \frac{\sum_{k=1}^K \sum_{i \in C_k} (x_i - m_k)(x_i - m_k)^t}{\sum_{k=1}^K n_k (m_k - \bar{x})(m_k - \bar{x})^t}, \quad (8)$$

where $m_k = (1/n_k) \sum_{i \in C_k} x_i$ is the empirical mean of the observed column vector x_i in the class k and $\bar{x} = (1/n) \sum_{k=1}^K n_k m_k$ is the mean column vector of the observations. Besides, both methods will be compared with a traditional approach of clustering which first reduces the dimension by principal component analysis (PCA) and then clusters the data in the projected space and refers in this paper to PCA-EM. Clustering accuracy will be computed between the partition obtained by both approaches and that obtained by a k -means approach.

4.6. Results for Some Prominent Channels. This first study aims to satisfy the existence of 4 classes of spikes. For this experiment, the clustering task was made channel by channel and, in each channel, we consider all the different events which correspond to movements of the rat. Finally, for each event, many spikes were recorded. Each normalized spike waveform is a time series that are of 32 dimensions.

Table 6 presents the number of spikes recorded in each of 16 channels and also the a priori number of kinds of spikes found by the preprocessing task. Besides, in the preprocessing task, as PCA components are computed so that different types of spikes are separated, we are going to first consider the projection on the 2 first components of PCA on each channel.

Figure 9 stand for the projection of the spikes of all the events of a selection of channels on the two first components. Whereas Table 6 describes the number of supposed types of spikes and given the preprocessing task, we expect to visualize on Figure 9 the intrinsic structure of the dataset where the number of separated clusters corresponds to those obtained in Table 6. However, it is difficult to visualize in Figure 9 a partition of several clusters in the data for each channel, whereas different clusters for channels 2, 7, and 16 can be observed in Figures 9(a), 9(e), and 9(k); such distinctions cannot be generalized since, on the other channels, it is not possible to visualize a group structure in the projected data. Without the label information of the preprocessing task, nothing enables us to suppose the true existence of different clusters. Furthermore it can be observed in Figure 9(b)—which stands for the projection of data of channel 3 plotted with the labeled spikes elaborated by the preclassification task—that the manual labels give no sense to a partition of the 2 groups of the data.

Consequently, from now, the proposed labels will not be taken into account and the main purpose of this work is to check the relevance of the preprocessing task. This study focuses on channels 2 and 7 whose datasets appear structured.

4.6.1. On Channel 2. The possible existence of two types of spikes in the axes of PCA in Figure 9(a) is satisfied by both the seriation and the subspace clustering approaches. In Figure 8(a) which represents the rearranged observations obtained by the Algorithm 1, one can observe 2 different blocks, one for each types of spikes in the data. In Figure 8(b) which stands for the projection of the data in the discriminative axes estimated by MDA algorithm, it can be observed that the clusters appear to be well separated compared with those obtained in the PCA axis. Figure 8(c) illustrates the projection of the data in the discriminative axes estimated by algorithm, it can be observed that the clusters appear to be well separated compared with those obtained in the PCA axis. Figure 8(c) which stands for the response of the supervised classification by PCA-EM approach has a similar representation of the data as those obtained by their projection in the 2 first principal components of PCA illustrated in Figure 9(a). In addition, Table 7 represents the Fisher index which has been computed for the different approaches previously presented. For the PB-Clus partition, the Fisher index is lower than those obtained by PCA-EM and

TABLE 7: Fisher index computed in channel 2 for PCA-EM, PB-Clus, and MDA.

Methods	PCA-EM	PB-Clus	MDA
F_{index}	864.5	210.1	588.7

MDA. It can be explained by the fact that PB-Clus introduces sparsity in the data, which produces smaller clusters that are more compact than those produced by PCA. Besides, the Fisher index for the MDA approach is equal to 588.7, which is equivalent to the result obtained by the PCA-EM classification ($F_{\text{index}} = 585.0$) and lower than the PCA's one ($F_{\text{index}} = 864.5$). Finally, to check the validity of the partition obtained by both methods, a cross-validation on the k -means results obtained by the work of [25] has been made. The contingency table and the clustering accuracy are presented in Table 8 and for each approach, it can be noted that 99% of the labeled data match with the PCA-EM labels. Consequently, it seems that, in channel 2, there are 2 different kinds of spikes and their respective shape obtained by both PB-Clus and MDA approaches is detailed in Figure 10.

4.6.2. On Channel 7. According to Figure 9(e), it can be observed on the first two components of PCA that there are at least 3 different groups of spikes. This remark is satisfied by the seriation approach, since Figure 11(a) which represents the intrinsic structure obtained by PB-Clus stresses 3 different kinds of spikes. In the same way, 3 components have been selected by using the Bayesian information criterion (BIC) for the mixture model in the case of PCA-EM, whereas both the preprocessing task and MDA, with the computation of BIC, have found 4 types of spikes. Figure 11(b) represents the projection of the clustered data on the 3 discriminant axes estimated by PCA-EM. In addition, since the k -means approach is based on the results of the preprocessing task, the prediction of the class membership of this dataset is made amongst 4 classes as can be seen in Figure 11(c).

Since the number of clusters varies between the different methods, data have been modeled by mixture models with 3 and then 4 components for both PCA-EM and MDA approaches, in order to be able to compare all the approaches. In Table 9 the Fisher index has been computed for the different cases. As expected, this criterion is much lower in the case of PB-Clus since it includes parsimony in the data whereas the ones obtained for MDA or PCA-EM remain high for a mixture of 3 components. Finally, the contingency table and the clustering accuracy are presented in Table 10. It can be observed that for the first case, PB-Clus detects the types 1, 2, and 4 of spikes whereas the 3rd type of spike is mixed with the first one. Furthermore, the classification rate reaches 91% on the spikes retained by PB-Clus when 40% of the data are ousted because of a high level of sparsity. In the second case, the partition obtained by MDA is comparable to these obtained by the PCA-EM classification except for type 1 which is mainly spread on type 3.

Finally, Figures 12(a) and 12(b) show the different spikes clustered by the PB-Clus and MDA algorithms. The difference

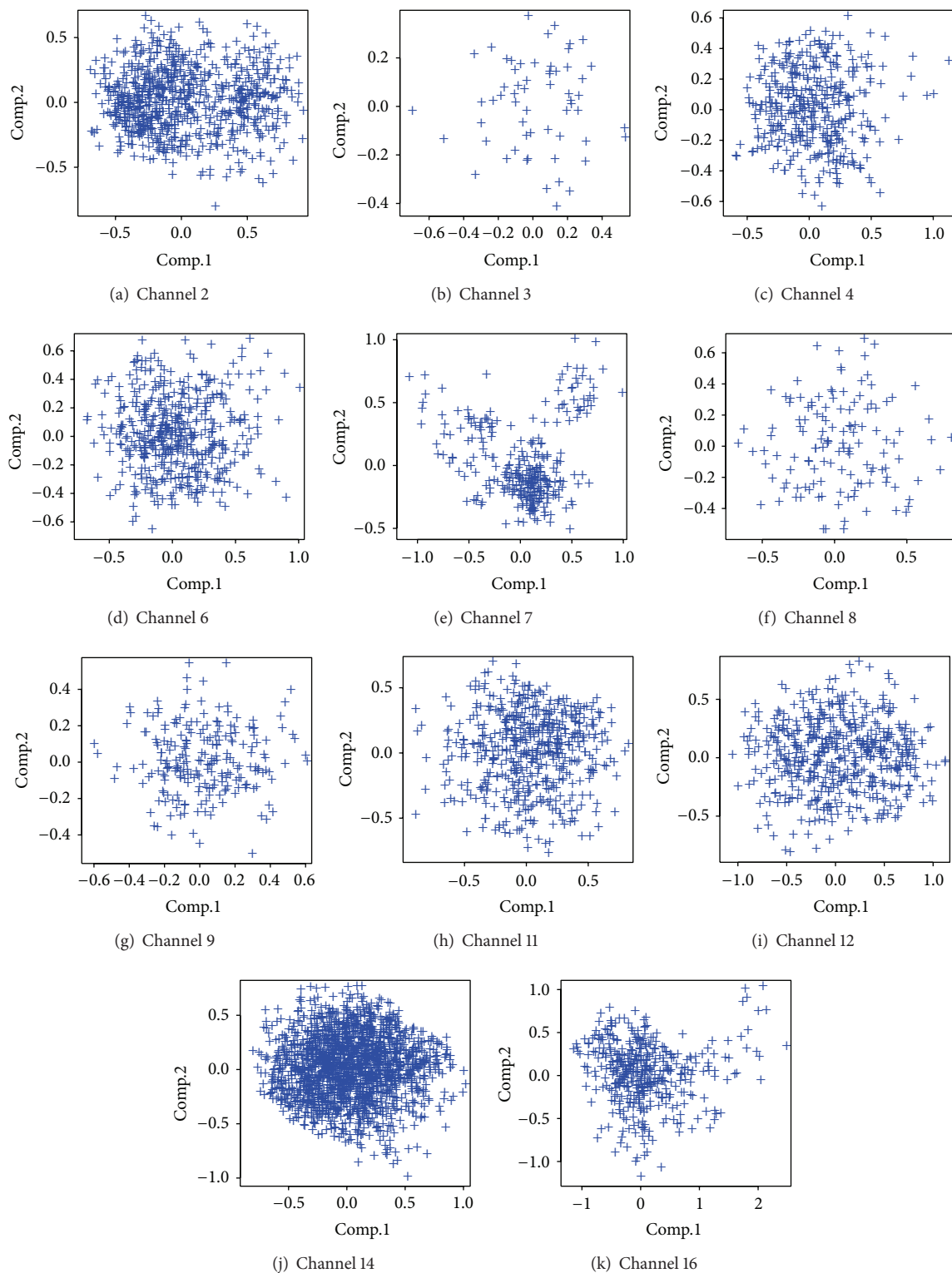


FIGURE 9: Projection of spikes of each channel on the 2 first components of PCA.

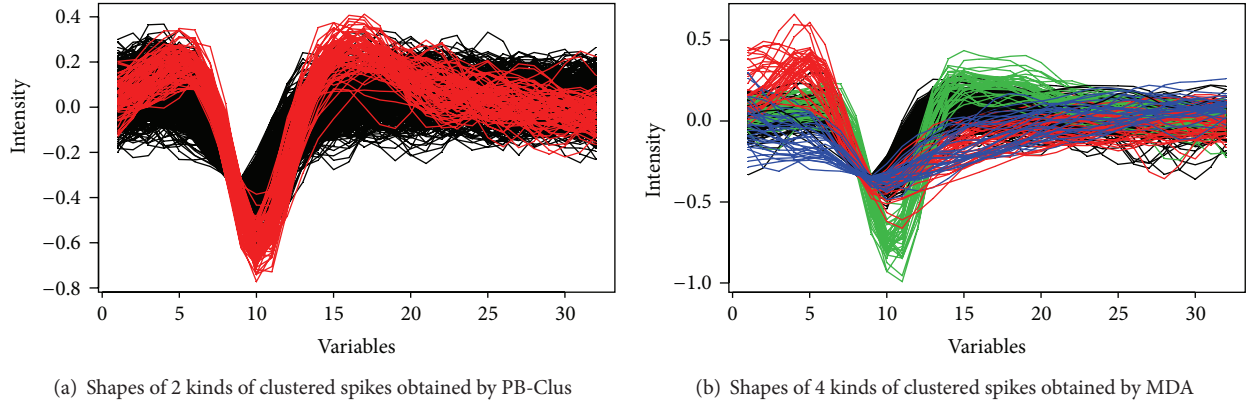


FIGURE 10: Clustered spikes obtained by PB-Clus in channel 2.

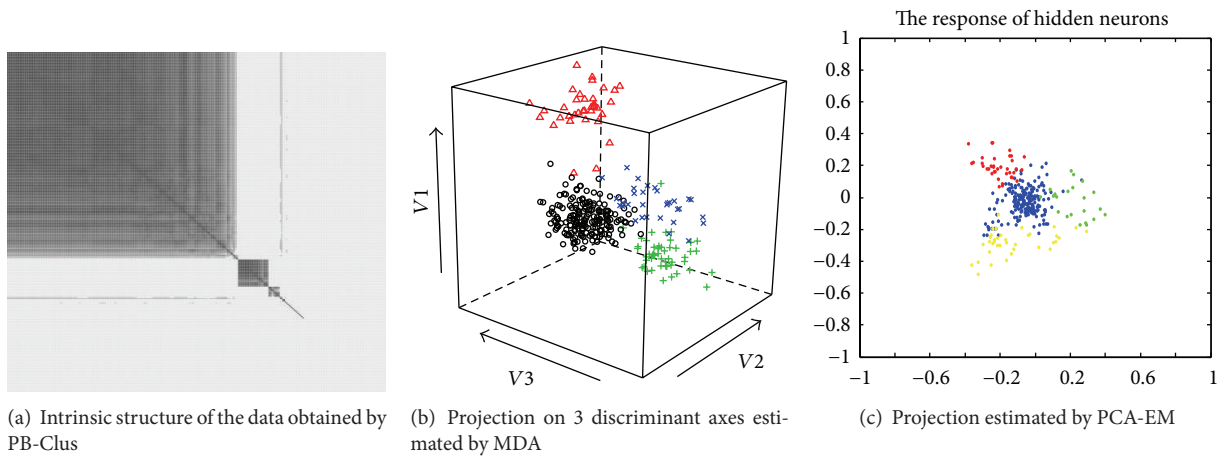


FIGURE 11: Results obtained by unsupervised approach PB-Clus and 2 supervised approaches MDA and PCA-EM on channel 7.

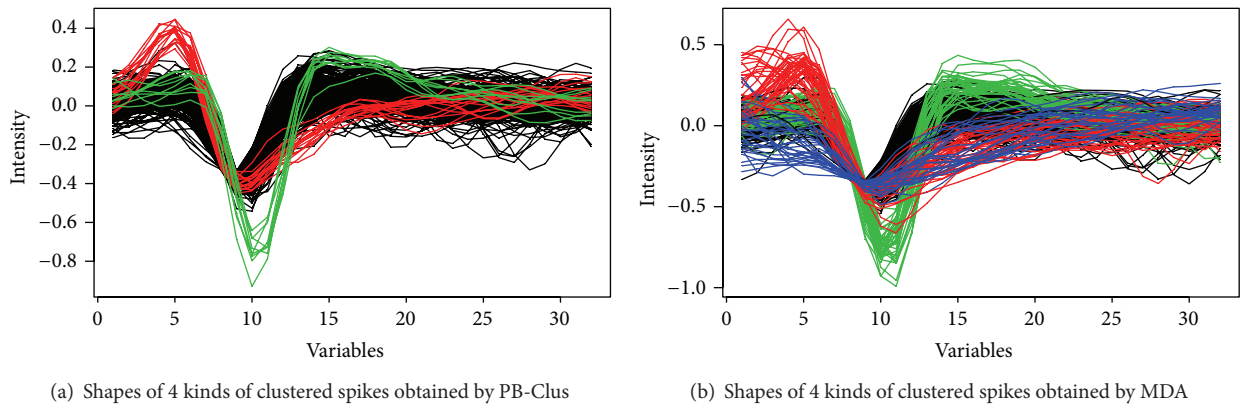


FIGURE 12: Clustered spikes obtained by PB-Clus and MDA for the channel 7.

between the two approaches is clearly seen on the 3rd type of spikes (blue in Figure 12) which is detected by MDA whereas it is not by PB-Clus. This could be explained by the weak dissimilarity between the shape of the 1st and the 3rd type of spikes (resp., black and blue in Figure 12) which is not taken into account by the measure of similarity, the

euclidean distance, used in PB-Clus. Different measures of similarity have been tried on PB-Clus such as Spearman correlation or maximum distances but have not brought any more information or any improvement for the visualization.

To conclude, given these results, the existence of 4 different types of clusters does not seem really relevant

Require: seq := vector of expected number of neighbors
Require: liste \mathcal{C} := vector of criterion values

- (1) **for all** $i \in \text{seq}$ **do**
- (2) Compute the binary matrix B_λ of common neighbors for $\lambda = \text{seq}(i)$
- (3) Compute B_λ the scalar products matrix of B_λ
- (4) $\lambda \leftarrow \text{sort}(S_\lambda)$ (sort the individuals)
- (5) $C_\lambda \leftarrow \text{crit}(B_\lambda, S_\lambda)$, compute the criterion
- (6) liste $\mathcal{C} \leftarrow \text{merge}(\text{liste}\mathcal{C}, C_\lambda)$
- (7) **end for**
- (8) $\lambda^* \leftarrow \arg \min_\lambda (\text{liste}\mathcal{C})$
- (9) $B_{\lambda^*} \leftarrow \min(\text{liste}\mathcal{C})$
- (10) $O_{\lambda^*} \leftarrow \text{ordre}(S_{\lambda^*})$

The PB-Clus algorithm returns the minimum of the criterion B_{λ^*} , the related number of common neighbors and the optimal sorting matrix O_{λ^*} .

- (1) **procedure** SORT(S_λ)
- (2) $V := \text{colinear}(i.\text{obs}, S_\lambda)$
- (3) liste := list of individuals whose common neighbors (cn) is non-zero ($cn = 0$)
- (4) $i.\text{max} :=$ the individual for which cn is maximum
- (5) $i.\text{perm} := 0$ (list containing the ranking value of the individuals)
- (6) **repeat**
- (7) $i.\text{perm} \leftarrow \text{merge}(i.\text{perm}, \text{liste}[V.\text{col}])$
- (8) **if** length($V.\text{cor}$) ≥ 1 **then**
- (9) $S_j \leftarrow S_j[c(V.\text{cor}, V.\text{ind}), c(V.\text{cor}, V.\text{ind})]$
- (10) liste $\leftarrow \text{liste}[c(V.\text{cor}, V.\text{ind})]$
- (11) **if** length($V.\text{cor}$) > 1 **then**
- (12) $V \leftarrow \text{colinear}(1, B_\lambda)$
- (13) **end if**
- (14) **else**
- (15) $S_j \leftarrow S_j[V.\text{ind}, V.\text{ind}]$
- (16) liste = liste[$V.\text{ind}$]
- (17) **if** length($V.\text{cor}$) > 1 **then**
- (18) $i.\text{max} = \arg \max_j cn(S_j)$
- (19) $B_\lambda = B_\lambda[V.\text{ind}, V.\text{ind}]$
- (20) $V = \text{colinear}(i.\text{max}, B_\lambda)$
- (21) **end if**
- (22) **end if**
- (23) **until** length(liste) > 1

- (1) **procedure** COLINEAR(i, S) returns 3 different lists
- (2) $s_i := i$ th line of S
- (3) $V.\text{col} :=$ list of individuals that are colinear with s_i
- (4) $V.\text{cor} :=$ list of individuals that are correlated with s_i
- (5) $V.\text{ind} :=$ list of individuals that are independent of s_i

ALGORITHM 1: PB-Clus algorithm.

TABLE 8: Contingency tables for PB-Clus and MDA partitions with k -means classification in channel 2.

PB-Clus			MDA		
k -means classes	Clusters		k -means classes	Clusters	
	1	2		1	2
Type 1	214	0	Type 1	515	2
Type 2	0	79	Type 2	5	277
Classification rate = 100%			Classification rate = 99.12%		

since some types of spikes, in particular types 3 and 4, are often mixed with the first type in both PB-Clus and MDA

TABLE 9: Fisher index computed in channel 7 for PCA-EM, PB-Clus, and MDA.

Methods	PCA-EM	PB-Clus	MDA
F_{index} for 3 clusters	614.1	287.9	483.6
F_{index} for 4 clusters	392.2	—	324.1

approaches. Consequently, either the preprocessing task is biased since the different types of spikes do not really exist or the 32 dimensions of the studied spikes are not sufficient to discriminate the 4 different types of spikes.

TABLE 10: Contingency tables for PB-Clus and MDA partitions with k -means classification for the channel 7.

PB-Clus				MDA				
k -means classes	Clusters			k -means classes	Clusters			
	1	2	3		1	2	3	4
Type 1	183	11	0	Type 1	181	0	17	6
Type 2	0	0	7	Type 2	1	36	0	0
Type 3	5	0	0	Type 3	0	0	33	4
Type 4	1	5	1	Type 4	0	0	0	22
Classification rate = 91%				Classification rate = 90.1%				

5. Conclusion

Controlled numerical experiments using spike and noise data extracted from neural recordings indicate significant improvements in detection and classification accuracy compared with amplitude and linear template-based spike sorting techniques. Algorithm 1 makes it possible to visualize subsets of spike data and their dependencies. With this intention, we proposed a family of embedded parsimonious matrices of different levels of parsimony whose level is directly determined by the number of common neighbors between pairs of observations. This is an effective tool for the analysis of data, which offers better results visually than the traditional clustering methods, in particular when the data are noisy or imbalanced or when the groups are superposed.

Moreover, this parsimonious approach facilitates the interpretation of the data and offers a quality of partitioning comparable with the k -means method with the advantage of not posing any assumption about the number of clusters. In addition, choosing a level of parsimony in the data corresponds to seeking explicative subsets of a structure. This new point of view can be connected with an approach by levels of density, commonly called *level sets*, which was initially approached by Hartigan [47] and then by Nolan [48]. A comparison of these two approaches and the search for a theoretical bond are part of our research tasks in progress.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This project was supported in part by funding from the Hubert Curien program of the Foreign French Minister and from the Taiwan NSC. The neural activity recordings were kindly provided by the Neuroengineering lab. of the National Chiao-Tung University.

References

- [1] M. S. Lewicki, "A review of methods for spike sorting: the detection and classification of neural action potentials," *Network: Computation in Neural Systems*, vol. 9, no. 4, pp. R53–R78, 1998.
- [2] D. A. Willming and B. C. Wheeler, "Real-time multichannel neural spike recognition with DSPs," *IEEE Engineering in Medicine and Biology Magazine*, vol. 9, no. 1, pp. 37–39, 1990.
- [3] A. F. Atiya, "Recognition of multiunit neural signals," *IEEE Transactions on Biomedical Engineering*, vol. 39, no. 7, pp. 723–729, 1992.
- [4] M. S. Fe, P. P. Mitra, and D. Kleinfeld, "Automatic sorting of multiple unit neuronal signals in the presence of anisotropic 473 and non-gaussian variability," *Journal of Neuroscience Methods*, vol. 9, pp. 175–188, 1968.
- [5] E. H. D'Hollander and G. A. Orban, "Efficient approximation and on-line classification by an unsupervised learning system," *IEEE Transactions on Biomedical Engineering*, vol. 26, pp. 279–284, 1979.
- [6] J. C. Letelier and P. P. Weber, "Spike sorting based on discrete wavelet transform coefficients," *Journal of Neuroscience Methods*, vol. 101, no. 2, pp. 93–106, 2000.
- [7] G. Zouridakis and D. C. Tam, "Identification of reliable spike templates in multi-unit extracellular recordings using fuzzy clustering," *Computer Methods and Programs in Biomedicine*, vol. 61, no. 2, pp. 91–98, 2000.
- [8] V. Vigneron, "Signal subspace separation based on the divergence measure of a set of wavelets coefficients," in *Independent Component Analysis and Blind Signal Separation*, vol. 5441, pp. 171–177, Springer, 2009.
- [9] S. Takahashi, Y. Anzai, and Y. Sakurai, "Automatic sorting for multi-neuronal activity recorded with tetrodes in the presence of overlapping spikes," *Journal of Neurophysiology*, vol. 89, no. 4, pp. 2245–2258, 2003.
- [10] A. M. Mamlouk, H. Sharp, K. M. L. Menne, U. G. Hofmann, and T. Martinetz, "Unsupervised spike sorting with ICA and its evaluation using GENESIS simulations," *Neurocomputing*, vol. 65–66, pp. 275–282, 2005.
- [11] C. Pouzat, M. Delescluse, P. Viot, and J. Diebolt, "Improved spike-sorting by modeling firing statistics and burst-dependent spike amplitude attenuation: a Markov chain Monte Carlo approach," *Journal of Neurophysiology*, vol. 91, no. 6, pp. 2910–2928, 2004.
- [12] C. Pouzat, O. Mazon, and G. Laurent, "Using noise signature to optimize spike-sorting and to assess neuronal classification quality," *Journal of Neuroscience Methods*, vol. 122, no. 1, pp. 43–57, 2002.
- [13] P. M. Zhang, J. Y. Wu, Y. Zhou, P. J. Liang, and J. Q. Yuan, "Spike sorting based on automatic template reconstruction with a partial solution to the overlapping problem," *Journal of Neuroscience Methods*, vol. 135, no. 1–2, pp. 55–65, 2004.
- [14] J. P. Stitt, R. P. Gaumond, J. L. Frazier, and F. E. Hanson, "An artificial neural network for neural spike classification,"

- in *Proceedings of the IEEE 1997 23rd Northeast Bioengineering Conference*, pp. 15–16, May 1997.
- [15] P. Barthó, H. Hirase, L. Monconduit, M. Zugaro, K. D. Harris, and G. Buzsáki, “Characterization of neocortical principal cells and interneurons by network interactions and extracellular features,” *Journal of Neurophysiology*, vol. 92, no. 1, pp. 600–608, 2004.
 - [16] K. Pearson, “On lines and planes of closest fit to systems of points in space,” *Philosophical Magazine Series*, vol. 2, no. 6, pp. 157–175, 1901.
 - [17] I. T. Jolliffe, *Principal Component Analysis*, Springer, 1986.
 - [18] D. J. Carroll and P. Arabie, “Multidimensional scaling,” *Annual Review of Psychology*, vol. 31, pp. 607–649, 1980.
 - [19] P. Arabie, L. J. Hubert, and G. de Soete, “An overview of combinatorial data analysis,” in *Clustering and Classification*, pp. 5–63, World Scientific, 1996.
 - [20] S. Climer and W. Zhang, “Rearrangement clustering: pitfalls, remedies, and applications,” *Journal of Machine Learning Research*, vol. 7, pp. 919–943, 2006.
 - [21] T. Villman and S. Haase, “Mathematical aspects of divergence based vector quantization using fréchet-derivatives,” Tech. Rep., University of applied sciences Mittweida, 2009.
 - [22] W. T. McCormick Jr., P. J. Schweitzer, and T. W. White, “Problem decomposition and data reorganization by a clustering technique,” *Operations Research*, vol. 20, no. 5, pp. 993–1009, 1972.
 - [23] P. Arabie and L. J. Hubert, “Bond energy algorithm revisited,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 20, no. 1, pp. 268–274, 1990.
 - [24] L. Hubert, P. Arabie, and J. Meulman, *Combinatorial Data Analysis: Optimization by Dynamic Programming*, Society for industrial and Applied Mathematics, 2001.
 - [25] C. H. Chen, “Generalized association plots: information visualization via iteratively generated correlation matrices,” *Statistica Sinica*, vol. 12, no. 1, pp. 7–29, 2002.
 - [26] G. Caraux and S. Pinloche, “PermutMatrix: a graphical environment to arrange gene expression profiles in optimal linear order,” *Bioinformatics*, vol. 21, no. 7, pp. 1280–1281, 2005.
 - [27] M. Brusco and D. Steinley, “Inducing a blockmodel structure of two-mode binary data using seriation procedures,” *Journal of Mathematical Psychology*, vol. 50, no. 5, pp. 468–477, 2006.
 - [28] D. Johnson, S. Krishnan, and J. Chhugani, “Compressing large boolean matrices using reordering techniques,” in *Proceedings of the 13th International Conference on Very Large Data Bases (VLDB '04)*, vol. 30, pp. 13–23, 2004.
 - [29] S. Niermann, “Optimizing the ordering of tables with evolutionary computation,” *American Statistician*, vol. 59, no. 1, pp. 41–46, 2005.
 - [30] V. Batagelj, “Notes on blockmodeling,” *Social Networks*, vol. 7, pp. 143–155, 1997.
 - [31] P. Doreian, V. Batagelj, and A. Ferligoj, “Generalized blockmodeling of two-mode network data,” *Social Networks*, vol. 26, no. 1, pp. 29–53, 2004.
 - [32] I. S. Dhillon, S. Mallela, and D. S. Modha, “Information-theoretic co-clustering,” in *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '03)*, pp. 89–98, August 2003.
 - [33] T. Apaydin, A. Ş. Tosun, and H. Ferhatosmanoglu, “Analysis of basic data reordering techniques,” in *Scientific and Statistical Database Management*, vol. 5069, pp. 517–524, 2008.
 - [34] I. van Mechelen, H. H. Bock, and P. de Boeck, “Two-mode clustering methods: a structured overview,” *Statistical Methods in Medical Research*, vol. 13, no. 5, pp. 363–394, 2004.
 - [35] M. J. Brusco, H. F. Köhn, and S. Stahl, “Heuristic implementation of dynamic programming for matrix permutation problems in combinatorial data analysis,” *Psychometrika*, vol. 73, no. 3, pp. 503–522, 2008.
 - [36] M. Hahsler, K. Hornik, and C. Buchta, “Getting things in order: an introduction to the R package seriation,” Tech. Rep. 58, 2009.
 - [37] E. H. Ruspini, “Numerical methods for fuzzy clustering,” *Information Sciences*, vol. 2, no. 3, pp. 319–350, 1970.
 - [38] W. Härdle, *Smoothing Techniques with Implementation*, Springer, New York, NY, USA, 1991.
 - [39] A. Azzalini and A. W. Bowman, “A look at some data on the old faithful geyser,” *Applied Statistics*, vol. 39, no. 3, pp. 357–365, 1990.
 - [40] A. E. Raftery and N. Dean, “Variable selection for model-based clustering,” *Journal of the American Statistical Association*, vol. 101, no. 473, pp. 168–178, 2006.
 - [41] V. Vigneron, Y. T. Chen, H. Y. Chen, and Y. Y. Chen, “Decomposition of eeg signals for multichannel neural activity analysis in animal experiments,” in *Independent Component Analysis and Blind Signal Separation*, vol. 6365, pp. 474–481, Springer, 2010.
 - [42] V. Vigneron, C. Brunet, and T. Willman, “Une famille de matrices sparses pour une modélisation multi-échelle par blocs,” in *Revue des Nouvelles Technologies de l'Information*, pp. 123–147, RNTI, Hermann, Mo, USA, 2011.
 - [43] T. Hastie and R. Tibshirani, “Discriminant analysis by gaussian mixtures,” Tech. Rep., AT & T Bell laboratories, Murray Hill, NJ, USA, 1994.
 - [44] C. Bishop, *Pattern Recognition and Machine Learning*, Springer, New York, NY, USA, 2006.
 - [45] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*, Wadsworth International Group, Belmont, Calif, USA, 1984.
 - [46] G. MacLachlan and K. Basford, *Mixtures Models: Inference and Applications to Clustering*, Marcel Dekker, 1988.
 - [47] J. A. Hartigan, *Clustering Algorithms*, John Wiley, New York, NY, USA, 1975.
 - [48] D. Nolan, “The excess-mass ellipsoid,” *Journal of Multivariate Analysis*, vol. 39, no. 2, pp. 348–371, 1991.

Research Article

Spike Sorting by Joint Probabilistic Modeling of Neural Spike Trains and Waveforms

Brett A. Matthews and Mark A. Clements

School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA

Correspondence should be addressed to Brett A. Matthews; matthews@ge.com

Received 16 September 2013; Revised 20 December 2013; Accepted 7 February 2014; Published 16 April 2014

Academic Editor: Wei Wu

Copyright © 2014 B. A. Matthews and M. A. Clements. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper details a novel probabilistic method for automatic neural spike sorting which uses stochastic point process models of neural spike trains and parameterized action potential waveforms. A novel likelihood model for observed firing times as the aggregation of hidden neural spike trains is derived, as well as an iterative procedure for clustering the data and finding the parameters that maximize the likelihood. The method is executed and evaluated on both a fully labeled semiartificial dataset and a partially labeled real dataset of extracellular electric traces from rat hippocampus. In conditions of relatively high difficulty (i.e., with additive noise and with similar action potential waveform shapes for distinct neurons) the method achieves significant improvements in clustering performance over a baseline waveform-only Gaussian mixture model (GMM) clustering on the semiartificial set (1.98% reduction in error rate) and outperforms both the GMM and a state-of-the-art method on the real dataset (5.04% reduction in false positive + false negative errors). Finally, an empirical study of two free parameters for our method is performed on the semiartificial dataset.

1. Introduction

Trace signals from extracellular electrodes implanted in a population of neurons are extremely valuable for studying the behavior of neurons and are used as the primary input source for motor and communication prostheses based on brain-computer interfaces [1, 2]. Given a trace of extracellular electric potential signals, neural spike sorting is the task of detecting neuronal action potential events in the form of “spikes” in the extracellular signal and identifying which neurons or neuronal clusters produced each spike in the trace. Many approaches to the spike sorting problem are based on discriminating between neuronal units by extracting features from spike waveforms and forming clusters in the feature space. This includes manual spike sorting approaches, where an expert visually identifies clusters of spike waveforms, and automatic approaches, which are typically based on rigorous signal processing and statistical modeling for clustering or classification. Many studies and practical implementations confirm the effectiveness of waveform-based spike sorting approaches; thorough reviews are given in [3, 4]. While approaches that use only the waveform shape have been successful, these methods are particularly susceptible to

some very common sources of errors in intracortical BCIs. Distinct neurons having similar waveform shapes, changes in time-domain, and parameterized waveform shapes due to movement of the electrodes and excessive background noise all have a direct negative impact on the performance of waveform-only spike sorting methods.

An ensemble of neural spike trains, one for each neuron or neuronal cluster in the vicinity of the electrode, is typically the most important result of the spike sorting operation [5]. Estimates of the firing rates of neural spike trains, as well as stochastic point process models of the spike trains themselves, are used to make inferences about populations of neurons [6–8] and for decoding in intracortical BCIs [9–13]. While modeling and analysis of spike trains is usually associated with neural decoding, a number of recent studies have advocated incorporating neural spike trains (and other temporal information) into the process of spike sorting. A complete maximum-likelihood framework based on a variant of hidden Markov models (HMMs) is described in [14] to model neuronal bursting behavior. The sparse HMM framework proposed in [14] models counts of neural firing events in equally spaced time frames. In another study,

extracellular traces were divided into short-duration time segments to model the nonstationarity of neuronal action potential waveform features [15]; Viterbi decoding was then used to find the optimal clustering across time segments. Several studies have explicitly incorporated models of interspike interval (ISI) durations into spike sorting [16, 17]. Stationary models of spike amplitudes and ISI durations are used in [16], while HMMs are incorporated in [17] to model the time-varying firing behavior of each neuron. Finally, a joint model of waveforms and firing rates, as modulated by a function of a known covariate quantity, was introduced in [18]. In [16, 17], Markov chain Monte Carlo distribution sampling was used for model inference.

In this paper, we detail a novel, probabilistic framework for neural spike sorting, based on stationary, stochastic ISI models of neural spike trains. A preliminary version of this work was presented previously in conference proceedings [19, 20]. Our method for spike sorting introduces a joint probabilistic model of parameterized spike waveforms and the occurrence times of an ensemble of neural spike trains. We derive a novel likelihood model and a procedure for clustering and parameter estimation and then evaluate the model on both fully labeled semiartificial neural data and partially labeled real data.

In Section 2 we describe our methods in detail. In Section 2.1, we define neural spike sorting as a latent-variable problem with three variables: the set of observed waveforms, the observed firing times, and hidden neuronal cluster labels. We give likelihood expressions for the three variables, and formulate the solution as maximizing their joint likelihood. We model the set of observed firing times as the aggregation of an ensemble of neural spike trains, one for each neuronal cluster, using a stationary, stochastic interspike interval model for each spike train. We then express the likelihood of the observed firing times as the joint likelihood of K neural spike trains. In Section 2.2, with the likelihood model defined, we derive a novel iterative method for clustering the data by finding the model parameters that maximize the joint likelihood of the firing times, waveforms, and hidden labels. For computational efficiency, the clustering method depends on iterating forward in time and keeping a short history of recent firing times at each stage. Section 2.3 gives detail about our choices for the probability distributions used in the likelihood model, and Section 2.4 explains the meaning and use of the procedure's two hyperparameters.

Finally, in Section 3, we evaluate our method on publicly available datasets. We first use realistic, semiartificial data (i.e., real spike waveforms, with synthesized firing times), with a complete set of labels, to evaluate the performance of our spike sorting procedure versus a state-of-the-art method. We then evaluate our procedure on a completely real dataset, consisting of extracellular traces, and an intracellular trace for one neuron.

2. Methods

The task of neuronal action potential identification, or spike sorting, can be seen as a latent variable problem where

the set of detected firing times and corresponding action potential waveforms are observed in an extracellular electric trace, and the identity of the underlying neurons is a hidden variable. In the remainder of this section, we describe a new approach to the spike sorting problem, where we model the set of observed, threshold-crossing neuronal firing times as the aggregation of multiple hidden point processes, one for each neuron. We use an iterative procedure to estimate the maximum likelihood sequence of states based on the set of observed action potential waveforms and firing times.

2.1. Likelihood Model. Let the vector $\mathbf{z} = \{z_i\}_{i=1}^N$ be the time occurrences of N observed, threshold-crossing events corresponding to firings of a population of K cortical neuronal clusters in the vicinity of the electrode. Let $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$ be the set of corresponding parameterized action potential waveforms, where each vector \mathbf{x}_i has dimension D , and let \mathbf{c} be an N -length, discrete-valued vector containing the set of unknown neuronal labels corresponding to each observed event.

We define the posterior probability $P(\mathbf{c} | \mathbf{X}, \mathbf{z})$ as follows:

$$P(\mathbf{c} | \mathbf{X}, \mathbf{z}) = \frac{P(\mathbf{X}, \mathbf{z}, \mathbf{c})}{P(\mathbf{X}, \mathbf{z})} \propto P(\mathbf{X}, \mathbf{z}, \mathbf{c}), \quad (1)$$

where we note that the term $P(\mathbf{X}, \mathbf{z})$ in (1) does not vary with respect to \mathbf{c} .

The optimal sequence $\hat{\mathbf{c}}$ thus satisfies

$$\hat{\mathbf{c}} = \arg \max_{\mathbf{c}} P(\mathbf{X}, \mathbf{z}, \mathbf{c}). \quad (2)$$

The graphical model in Figure 1 illustrates the assumptions about the statistical dependencies between the observed variables \mathbf{X} and \mathbf{z} and the latent variable \mathbf{c} that we will use in our modeling framework. The figure illustrates that the observed variables in \mathbf{X} and \mathbf{z} depend on the hidden labels in \mathbf{c} , but not on each other. On this basis, we express the likelihood $P(\mathbf{X}, \mathbf{z}, \mathbf{c})$ as follows:

$$P(\mathbf{X}, \mathbf{z}, \mathbf{c}) = P(\mathbf{X}, \mathbf{z} | \mathbf{c}) P(\mathbf{c}) \quad (3)$$

$$= P(\mathbf{X} | \mathbf{c}) P(\mathbf{z} | \mathbf{c}) P(\mathbf{c}), \quad (4)$$

where the terms $P(\mathbf{X} | \mathbf{c})$ and $P(\mathbf{z} | \mathbf{c})$ express the likelihood of the observed set of extracted neuronal waveforms and their corresponding occurrence times, respectively, given a sequence of neuronal labels \mathbf{c} , and $P(\mathbf{c})$ is the likelihood of the sequence itself.

In all experiments, we model the parameterized action potential waveform for each neuronal cluster as a single, multivariate Gaussian with parameters $\theta = \{\mu, \Sigma\}$, such that the waveform likelihood for cluster j is given by $p(\mathbf{x}; \theta_j) = \mathcal{N}(\mathbf{x}; \mu_j, \Sigma_j)$ and the likelihood for the complete set of waveforms is given by

$$p(\mathbf{X} | \mathbf{c}) = \prod_{i=1}^N p(\mathbf{x}_i; \theta_{c_i}). \quad (5)$$

We characterize the temporal behavior of a population of K neuronal clusters by modeling the set of neuronal firing

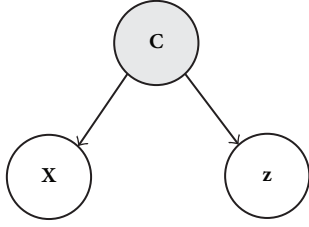


FIGURE 1: Statistical dependencies for parameterized waveforms \mathbf{X} , occurrence times \mathbf{z} , and labels \mathbf{c} .

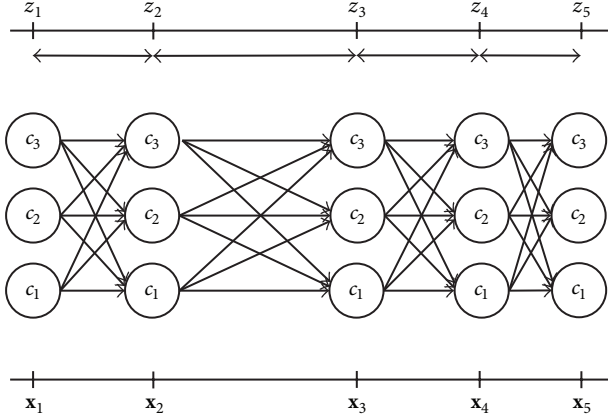


FIGURE 2: Lattice structure for clustering and parameter estimation.

times \mathbf{z} as the aggregation of K independent point processes $(\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_K)$, where each $\mathbf{t}_k = \{z_j\}_{j \in c_k}$ is the subset of \mathbf{z} corresponding to firings of the k th neuronal cluster. It is convenient to model the likelihood $P(\mathbf{t}_k)$ based on the distribution of interspike interval durations. Let $f_k(\tau; \phi_{\text{isi}})$ be a probability density function with parameter set ϕ_{isi} , characterizing the distribution of the continuous, univariate time period $\tau = t_{k,i} - t_{k,i-1}$ between two consecutive firings of neuronal cluster k occurring at times $t_{k,i}$ and $t_{k,i-1}$. Assuming that interspike interval durations are independent and identically distributed, the likelihood $P(\mathbf{t}_k)$ can be expressed as

$$P(\mathbf{t}_k) = w_k(t_{k,1}; \phi_{\text{init}}) \prod_{i=2}^{N_k} f_k(t_{k,i} - t_{k,i-1}; \phi_{\text{isi}}) g_k(t_{k,N_k}; \psi), \quad (6)$$

where N_k is the number of neuronal firings in \mathbf{t}_k , $w_k(t; \phi_{\text{init}})$ is the distribution of the first firing time $t_{k,1}$, and $g_k(t_{k,N_k}) = \int_T^\infty f_k(x - t_{k,N_k}; \phi_k) dx$ is the distribution of the last firing time t_{k,N_k} , where T is the total time length of the dataset [21]. We model the likelihood $P(\mathbf{z} | \mathbf{c})$ of the complete set of firing times in terms of the joint occurrence of all class-conditional firing times; that is, $P(\mathbf{z} | \mathbf{c}) = P(\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_K)$. Since we have assumed that these K point processes are independent, we say

$$P(\mathbf{z} | \mathbf{c}) = P(\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_K) = \prod_{k=1}^K P(\mathbf{t}_k). \quad (7)$$

The last term in (4), $P(\mathbf{c})$, is the likelihood of the set of neuronal firing labels. It is important to note that since we

have assumed that each neuronal cluster fires independently, all temporal modeling is expressed in terms of firing times and interspike intervals. Thus, unlike a hidden Markov model, we do not apply any explicit statistical modeling to the sequence of labels, and the likelihood $P(\mathbf{c})$ is simply given by

$$P(\mathbf{c}) = \prod_{i=1}^N P(c_i). \quad (8)$$

2.2. Clustering and Parameter Estimation. We can represent the dynamic relationship between \mathbf{X} , \mathbf{c} , and \mathbf{z} with the lattice structure depicted in Figure 2. Figure 2 depicts a dataset consisting of $K = 3$ neuronal clusters, with $N = 5$ observed firing times in \mathbf{z} and corresponding action potential waveforms in \mathbf{X} . The lattice structure is similar in appearance to the commonly used HMM trellis but has some important differences. Particularly, since we do not explicitly model transitions between states and all temporal modeling is based on $P(\mathbf{z} | \mathbf{c})$, uneven horizontal spacing is used to illustrate observed interarrival durations in \mathbf{z} .

For the spike sorting task, we seek to exploit both the action potential waveform shape in \mathbf{X} and the temporal information in \mathbf{z} . To find the maximum likelihood sequence $\hat{\mathbf{c}}$ as defined in (2) we use an approximate, iterative procedure to find the best path through the state space depicted in Figure 2. The procedure is initialized with a clustering based on the set of action potential waveforms only.

Given a set of parameters $\lambda = \{\theta, \phi_{\text{init}}, \phi_{\text{isi}}\}$, we determine the maximum likelihood state sequence $\hat{\mathbf{c}}$ by deriving a recursive expression for the joint likelihood $P(\mathbf{X}, \mathbf{z}, \mathbf{c})$. Let the notation $P(\{\mathbf{x}_i\}_{i=1}^n, \{\mathbf{z}_i\}_{i=1}^n, \{\mathbf{c}_i\}_{i=1}^n)$ indicate the joint likelihood of the first n data points, such that the likelihood of the full set of N data points is given by $P(\mathbf{X}, \mathbf{z}, \mathbf{c}) = P(\{\mathbf{x}_i\}_{i=1}^N, \{\mathbf{z}_i\}_{i=1}^N, \{\mathbf{c}_i\}_{i=1}^N)$. We decompose $P(\{\mathbf{x}_i\}_{i=1}^n, \{\mathbf{z}_i\}_{i=1}^n, \{\mathbf{c}_i\}_{i=1}^n)$ as follows:

$$\begin{aligned} P(\{\mathbf{x}_i\}_{i=1}^n, \{\mathbf{z}_i\}_{i=1}^n, \{\mathbf{c}_i\}_{i=1}^n) &= P(\mathbf{x}_n, \mathbf{z}_n, \mathbf{c}_n, \{\mathbf{x}_i\}_{i=1}^{n-1}, \{\mathbf{z}_i\}_{i=1}^{n-1}, \{\mathbf{c}_i\}_{i=1}^{n-1}) \\ &= P(\mathbf{x}_n, \mathbf{z}_n, \mathbf{c}_n | \{\mathbf{x}_i\}_{i=1}^{n-1}, \{\mathbf{z}_i\}_{i=1}^{n-1}, \{\mathbf{c}_i\}_{i=1}^{n-1}) \\ &\quad \cdot P(\{\mathbf{x}_i\}_{i=1}^{n-1}, \{\mathbf{z}_i\}_{i=1}^{n-1}, \{\mathbf{c}_i\}_{i=1}^{n-1}) \\ &= P(\mathbf{x}_n | \mathbf{c}_n, \{\mathbf{x}_i\}_{i=1}^{n-1}, \{\mathbf{z}_i\}_{i=1}^{n-1}, \{\mathbf{c}_i\}_{i=1}^{n-1}) \\ &\quad \cdot P(\mathbf{z}_n | \mathbf{c}_n, \{\mathbf{x}_i\}_{i=1}^{n-1}, \{\mathbf{z}_i\}_{i=1}^{n-1}, \{\mathbf{c}_i\}_{i=1}^{n-1}) \\ &\quad \cdot P(\mathbf{c}_n | \{\mathbf{x}_i\}_{i=1}^{n-1}, \{\mathbf{z}_i\}_{i=1}^{n-1}, \{\mathbf{c}_i\}_{i=1}^{n-1}) \\ &\quad \cdot P(\{\mathbf{x}_i\}_{i=1}^{n-1}, \{\mathbf{z}_i\}_{i=1}^{n-1}, \{\mathbf{c}_i\}_{i=1}^{n-1}), \end{aligned} \quad (9)$$

where we have used the statistical dependency assumptions given in (4) and (5) and illustrated in Figure 1. Assuming that

action potential waveforms in \mathbf{x}_n and hidden labels in c_n do not depend on any previous samples, we obtain

$$\begin{aligned} P(\{\mathbf{x}_i\}_{i=1}^n, \{z_i\}_{i=1}^n, \{c_i\}_{i=1}^{n-1}, c_n = j) \\ = P(\mathbf{x}_n | c_n = j) \cdot P(z_n | c_n = j, \zeta_j) \cdot P(c_n = j) \quad (10) \\ \cdot P(\{\mathbf{x}_i\}_{i=1}^{n-1}, \{z_i\}_{i=1}^{n-1}, \{c_i\}_{i=1}^{n-1}). \end{aligned}$$

Note in (10) that we have expressed the likelihood of an n -length label sequence ending in state j and that we have introduced a new variable ζ_j . Given a label sequence ending in state j , the likelihood $P(z_n | c_n = j, \zeta_j)$ depends on $\zeta_j < z_n$, which we define as the most recent, previous occurrence time of state j . To find ζ_j , some bookkeeping is necessary. Specifically, at each iteration we retain the L highest likelihood label sequences or paths through the lattice structure illustrated in Figure 2 (the number of paths, L , is determined empirically). Each path contains only the most recent spikes occurring within a history time window, starting at time $z_n - \tau_{\text{win}}$ and ending at z_n . The length τ_{win} of the history window is constant and is determined empirically. The likelihood in (10) is computed for the best L paths retained from the previous iteration. For a given path, the duration $z_n - \zeta_j$ is modeled with an interarrival distribution f_j as expressed in the second term of (6). If no previous occurrences of state j are found in a given path, we say that $\zeta_j = -\infty$ and the distribution w_j (expressed in the first term in (6)) for the first firing time is used instead, with the window length τ_{win} as its argument. This is expressed in (11):

$$\begin{aligned} p(z_n | c_n = j, \zeta_j) \\ = \begin{cases} w_j(\tau_{\text{win}} | \phi_{\text{init}}), & \zeta_j = -\infty \\ f_j(z_n - \zeta_j | \phi_{\text{isi}}), & \text{otherwise.} \end{cases} \quad (11) \end{aligned}$$

Iterative Procedure. Though our spike sorting method uses both spike waveforms and firing times, we must initialize the procedure using spike waveforms only. We model the waveforms in \mathbf{X} as a Gaussian mixture model (GMM) and find the maximum likelihood waveform parameters using the expectation-maximization (EM) algorithm to produce an initial clustering. Based on the initial clustering, we estimate parameters $\phi_{\text{isi},j}$ for the interarrival distribution f_j and $\phi_{\text{init},j}$ for the first-firing distribution w_j , for each neuron j . For the first firing and interarrival distributions w_j and f_j , we use the exponential and lognormal probability density functions, respectively. We then assign each data point \mathbf{x}_i to the maximum a posteriori GMM component to produce a clustering and estimate parameters $\lambda = \{\theta, \phi_{\text{isi}}, \phi_{\text{init}}\}$ based on the clustering. The 3-step procedure is then as follows.

- (1) Decode with parameters λ and produce a segmentation.
- (2) Estimate parameters λ_{new} based on the segmentation.
- (3) Iterate until convergence.

2.3. Probability Distributions. A breakdown of the probability distributions and their parameters used in all experiments is given in Table 1. We model parameterized action potential waveforms for each cluster as single, multivariate Gaussians and model interarrival durations with the conditional distribution expressed in (11). For w_j , the distribution of the “first firing” after a long duration, we use a simple Poisson distribution $w_j(k; \beta t) = (\beta t)^k e^{-\beta t} / k!$ with duration parameter β and event count $k = 1$. For the ISI distribution f_j , we use a log-normal density with parameters μ and σ^2 . The log-normal density has been shown to have a superior empirical fit to neuronal ISI durations having a necessary minimum refractory period [16, 17].

2.4. Parameters L and τ . In addition to the distribution parameters $\lambda = [\theta, \phi_{\text{init}}, \phi_{\text{isi}}]$, our procedure has two free parameters L and τ , which are determined empirically; these are the number of paths and the history window length, respectively. The number of paths L is typically chosen according to a trade-off of accuracy against speed and memory usage. We choose the window length τ such that the “ $\zeta_j = -\infty$ ” condition in (11) occurs rarely. τ is chosen to be larger than an interarrival duration $t_{k,i} - t_{k,i-1}$ for any neuron k with high probability. To estimate τ we fit a lognormal distribution to each neuronal cluster based on an initial waveform-only clustering of the data and choose τ_k to cover 99% of the area under the lognormal ISI curve for neuron k . The history window τ is then simply $\tau = \max_k \tau_k$. The general expression for the log-normal density function for a variable t with parameters μ and σ^2 is given by

$$t \sim \text{Log Norm}(t; \mu, \sigma^2) = \frac{1}{t\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{\log t - \mu}{\sigma}\right)^2\right]. \quad (12)$$

Given a set of univariate, Gaussian-distributed data $x \sim \mathcal{N}(x; \mu, \sigma^2)$, if χ is the logarithm of x then, by definition, χ is log-normal distributed, such that $\chi = \log(x) \sim \text{Log Norm}(\chi; \mu, \sigma^2)$. The log-normal parameters μ and σ^2 are then the mean and variance of $\exp(\chi)$, respectively. The log-normal distribution is supported on the range $[0, \infty)$ and has been used successfully to model neuronal interspike interval durations [16, 17].

3. Experimental Results

Given a real, continuous extracellular trace, it is typically impracticable to obtain a complete set of ground truth labels since it cannot be directly observed which neuron caused each action potential spike in the trace. This makes evaluation for spike sorting difficult in most nontrivial cases. Synthetic extracellular traces, which are often partially composed of real data, provide fully labeled datasets useful for development and evaluation of spike sorting methods. When fully authentic data are desired, however, it is possible to collect data using both an extracellular electrode and a carefully placed intracellular electrode in one neuronal cell to obtain a partial ground truth labeling. Spikes on an intracellular

TABLE 1: Breakdown of parameter set $\lambda = [\theta, \phi_{\text{init}}, \phi_{\text{isi}}]$ and probability distributions for joint waveform and firing rate spike sorting.

Waveform	Gaussian	$\theta_j = \{\mu_j, \Sigma_j\}$
First firing $[w_j]$	Poisson	$\phi_{\text{init},j} = \{\beta_j\}$
ISI $[f_j]$	Log-normal	$\phi_{\text{isi},j} = \{\mu_j, \sigma_j^2\}$

electrode identify the firing times of one neuron with near certainty. In Sections 3.1 and 3.2, we apply our spike sorting method to two publicly available sets of cortical extracellular traces to demonstrate its performance. One of these datasets is real and partially labeled, and the other is semiartificial and fully labeled. Finally, in Section 3.3, we perform an empirical study of the parameters L and τ using the semiartificial, fully labeled dataset.

3.1. WaveClus Semiartificial Dataset. We evaluate our spike sorting methods with labeled data collected, in part, from the publicly available WaveClus artificial dataset [22]. We use randomly selected action potential waveforms from “Example 1” and “Example 2” subsets, hereafter referred to as “Easy1” and “Difficult1,” respectively. Each data subset consists of $K = 3$ neuronal clusters with characteristic action potential waveform shapes drawn from a library of templates. The 3 characteristic waveforms in the “Difficult1” set are similar to each other in shape and are generally more difficult to separate than in the “Easy1” set. All of the WaveClus datasets contain realistic additive background noise at varying power levels. For our spike sorting experiments, we added additional Gaussian noise to the baseline data at various SNR levels. We use principal components analysis (PCA) for dimensionality reduction in all experiments. Scatter plots of the first 2 principal components are given in Figure 3 for both data subsets under various noise conditions.

All subsets of the WaveClus dataset contain 3 neuronal clusters with artificial firing times having identical firing rate statistics. For our experiments, we generated firing times according to a Monte Carlo sampling of 3 independent log-normal distributions, resulting in a dataset of 2483 firing times 24 seconds in length. A minimum 3-millisecond interval duration was enforced to model the refractory period for all clusters.

Table 2 gives the simulation parameters, μ and σ^2 , we used to generate interspike interval durations, along with the mean, in milliseconds, of the generated data. The parameters listed in Table 2 were determined by computing the sample mean and variance of putative log interarrival times taken from another publicly available dataset. (These data were collected in the Laboratory of Dario Ringach at UCLA and downloaded from the CRCNS website.) Plots of interspike interval histograms are given in Figure 4.

Results. To evaluate the accuracy of our method for spike sorting, we compute the classification accuracy of the best match between the set of true clusters and the set of putative clusters identified by our procedure. Quantitative performance results for the WaveClus dataset are given for the baseline GMM

TABLE 2: Simulation parameters for interspike interval data.

Cluster	Parameters		Mean ISI (ms)
	μ	σ^2	
1	1.5814	2.4203	24.5342
2	2.1610	1.9380	30.0564
3	1.9651	2.7068	33.9112

TABLE 3: Classification error rates for the WaveClus semiartificial dataset.

Dataset	SNR	GMM	WaveClus method	Proposed
Easy1	Clean	0.93%	0.00%	0.97%
	10 dB	0.72%	0.00%	0.77%
	5 dB	0.89%	0.00%	0.85%
	0 dB	0.81%	0.21%	0.85%
	−5 dB	1.25%	0.52%	0.97%
	−10 dB	8.18%	3.47%	6.20%
Difficult1	Clean	3.18%	0.45%	2.58%
	10 dB	4.83%	0.94%	3.46%
	5 dB	6.89%	1.36%	5.32%
	0 dB	19.77%	20.0%	37.21%

procedure and for our joint waveform and firing rate method in Table 3. We compare our proposed approach to a GMM baseline clustering (i.e., the waveform-only initialization) and to the state-of-the-art superparamagnetic clustering or “WaveClus” method [22]. Gaussian noise was added to both the “Easy1” and “Difficult1” datasets at various SNR levels with the original WaveClus dataset labeled “Clean” in the table.

Overall, we find that our method, which extends the waveform-only baseline by incorporating a hidden point process model for each neuron, reduces the error rate in the presence of noise. The error rate for the “Easy1” dataset at higher SNR levels ($\text{SNR} > 0$ dB) is not significantly changed from the baseline initialization by our joint waveform and firing rate method with respect to the initial clustering, which was already quite low (less than 1.0% error). The absolute difference in error rate for the baseline GMM clustering and our proposed method is less than 0.05% (i.e., 1 out of 2483 spikes in the dataset) at higher SNR levels. Since this dataset was particularly easy to classify, we added noise at −5 dB and −10 dB SNR. In the presence of high noise (−10 dB SNR), we reduce the error rate from 8.18% to 6.20% by incorporating temporal information.

A similar trend is seen with the “Difficult1” dataset, but at higher SNR levels. At 10 dB SNR and 5 dB SNR, we reduce the error rate with respect to the baseline by 1.37% and 1.57%, respectively. However, when SNR is reduced to 0 dB for the “Difficult1” dataset, we see a significant increase in the error rate. The WaveClus method, however, performs significantly better on this dataset overall.

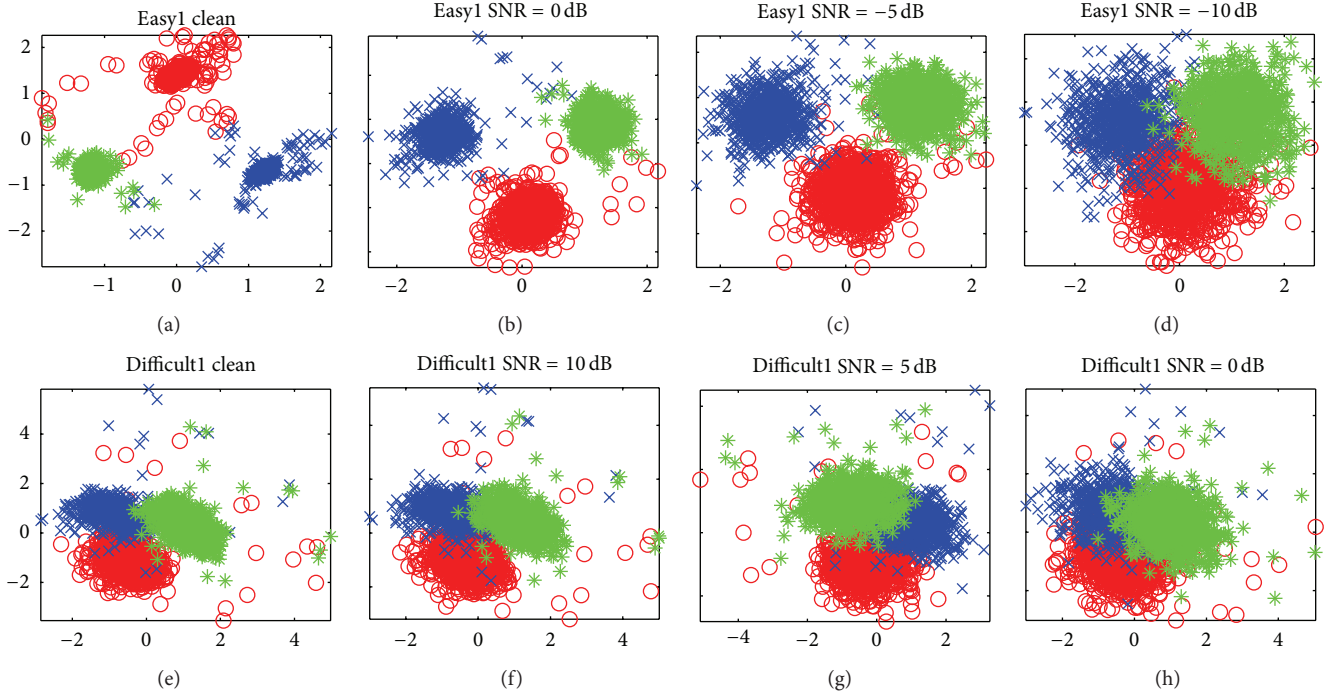


FIGURE 3: First 2 PCA coefficients of action potential wave forms plus noise at various SNR levels for the “Easy1” and “Difficult1” semi-artificial data sets.

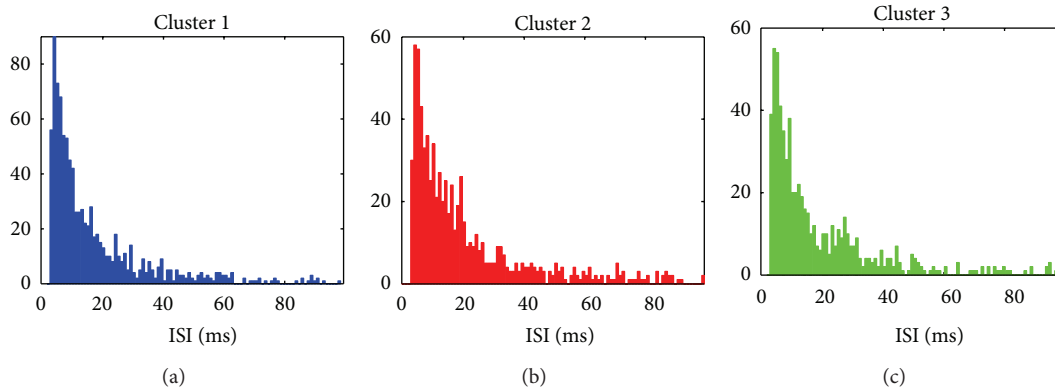


FIGURE 4: Inter-spike interval histograms for both the “Easy1” and “Difficult1” data sets. Spike firing times were generated according to a log-normal ISI distribution with parameters listed in Table 2, and a minimum refractory period of 3 msec.

3.2. Continuous Extracellular Traces. To evaluate our methods on real, continuous data, we use a publicly available dataset of cortical electrical traces taken from hippocampus of anesthetized rats, hereafter referred to as “HC1” [23]. The HC1 dataset consists of traces of extracellular (EC) electric potentials, as well as intracellular (IC) traces for 1 of K neurons in the vicinity of the EC electrodes. We use two subsets of the HC1 dataset, each 4 minutes in length, to evaluate our spike sorting procedure. Both the EC and IC electric potential signals for Datasets 1 and 2 were recorded at a sample rate of 20 kHz. We use a high-pass filter to eliminate waveform drift for the extracellular signals. A plot of a 1.79 s segment of simultaneously recorded EC and IC signals

from Dataset 1 is given in Figure 5. Three peaks in the lower panel of Figure 5 indicate firing times of the “IC neuron” and correspond to 3 of the peaks in the EC signal in the upper panel.

Methodology. We detect neuronal action potentials as “spikes” in the extracellular signal exceeding a threshold of 5σ , where σ is an estimate of the standard deviation as defined in [22]. In all spike sorting experiments, we extract observed action potential events as 4 ms waveforms centered at the peak point in each extracellular spike. To locate spikes on the intracellular channel in each data subset, we take the first backward difference of the IC signal and apply a peak-picking

TABLE 4: Classification error rates (FP + FN) for the HC1 dataset.

Dataset	SNR	GMM	WaveClus Method		Proposed
			Wavelets	PCA	
Dataset 1	Clean	10.64%	32.02%	17.98%	5.60%
	15 dB	9.73%	24.5%	16.79%	4.77%
	10 dB	9.73%	31.84%	21.47%	4.95%
	5 dB	11.28%	31.93%	32.02%	6.51%
	0 dB	10.19%	32.11%	31.56%	9.27%
	-5 dB	20.09%	32.11%	32.02%	15.23%
	-10 dB	31.93%	67.98%	31.93%	32.11%
Dataset 2	Clean	2.09%	8.49%	6.56%	1.86%
	15 dB	1.96%	7.56%	6.20%	1.76%
	10 dB	1.99%	7.59%	6.27%	1.89%
	5 dB	2.29%	19.62%	11.04%	2.06%
	0 dB	3.51%	19.42%	14.29%	3.45%
	-5 dB	6.99%	19.65%	19.52%	5.87%
	-10 dB	32.55%	19.59%	19.56%	30.53%

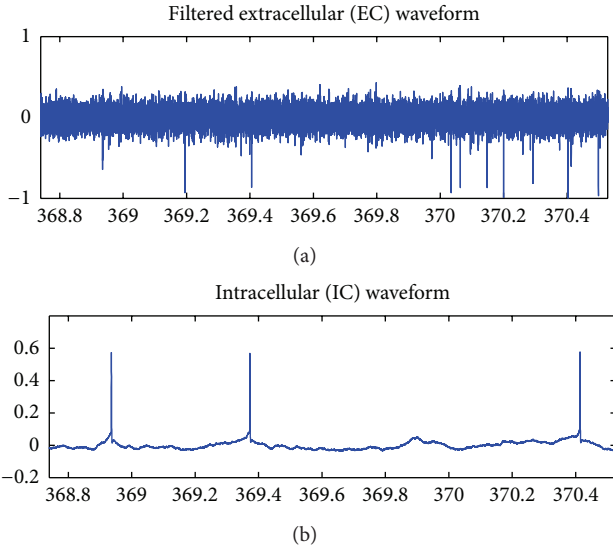


FIGURE 5: Simultaneous extracellular (EC) and intracellular (IC) electric potential traces taken from Dataset 1 of completely real data set HC1. True firing times of the “IC neuron” are known with near certainty.

algorithm to it. EC action potentials occurring within 1 ms of an IC spike are labeled as belonging to the IC neuron. In Dataset 1, we detected 1090 total extracellular firings and 396 intracellular firings. In Dataset 2, we detect 3017 EC firings and 1100 IC firings. In each dataset, there are $K = 3$ neuronal clusters.

For EC waveforms, we use principal components analysis (PCA) for dimensionality reduction. We keep the first 3 principal components as features for \mathbf{X} , the matrix of observed action potential waveforms. For our spike sorting experiments, we add Gaussian noise to the waveform at various SNR levels before applying PCA. Scatter plots of the first two principal components at various SNR levels are given

in Figure 6; extracellular waveform features corresponding to firings of the IC neuron are distinguished with black “X” markers.

Evaluation. Given only a partial labeling of the data, we can evaluate the performance of a spike sorting result in terms of false positive (FP) and false negative (FN) errors for the labeled IC neuron. When a spike corresponding to the IC neuron is misclassified, a FN error is counted; inversely, when a spike is erroneously classified as belonging to the IC neuron, a FP error is counted. The error rate is defined as the sum of the FN and FP counts, divided by the number of EC firings.

Results. Quantitative performance results, in terms of the total (FP + FN) error rate, are given in Table 4. For the WaveClus method, we use both a wavelet-based parameterization (the default choice for the WaveClus software package) of action potential waveforms and PCA features for a more direct comparison with the other results.

Our proposed joint waveform and firing rate approach performs the best in all cases except at the lowest SNR level for both data sets 1 and 2. The WaveClus method results in very high FN error counts, but low FP counts (only the total error rate, FP + FN, is shown in Table 4). We see a larger improvement over the GMM baseline (nearly 5% at some levels) for Dataset 1, which is ostensibly the more difficult set of the two, as evidenced by higher overall error rates for all classifiers.

3.3. Empirical Study of Parameters. Our clustering approach involves retaining a large number of paths, L , and a time history window of length τ of recent firings. In all of our previously reported results, we have used a fixed value of $L = 10000$ paths chosen on the basis of computational and memory constraints. The value of τ was determined to cover 99% of the area of the estimated interspike interval probability density curve, as described in Section 2.4. In this section, we perform an empirical study on the impact of our

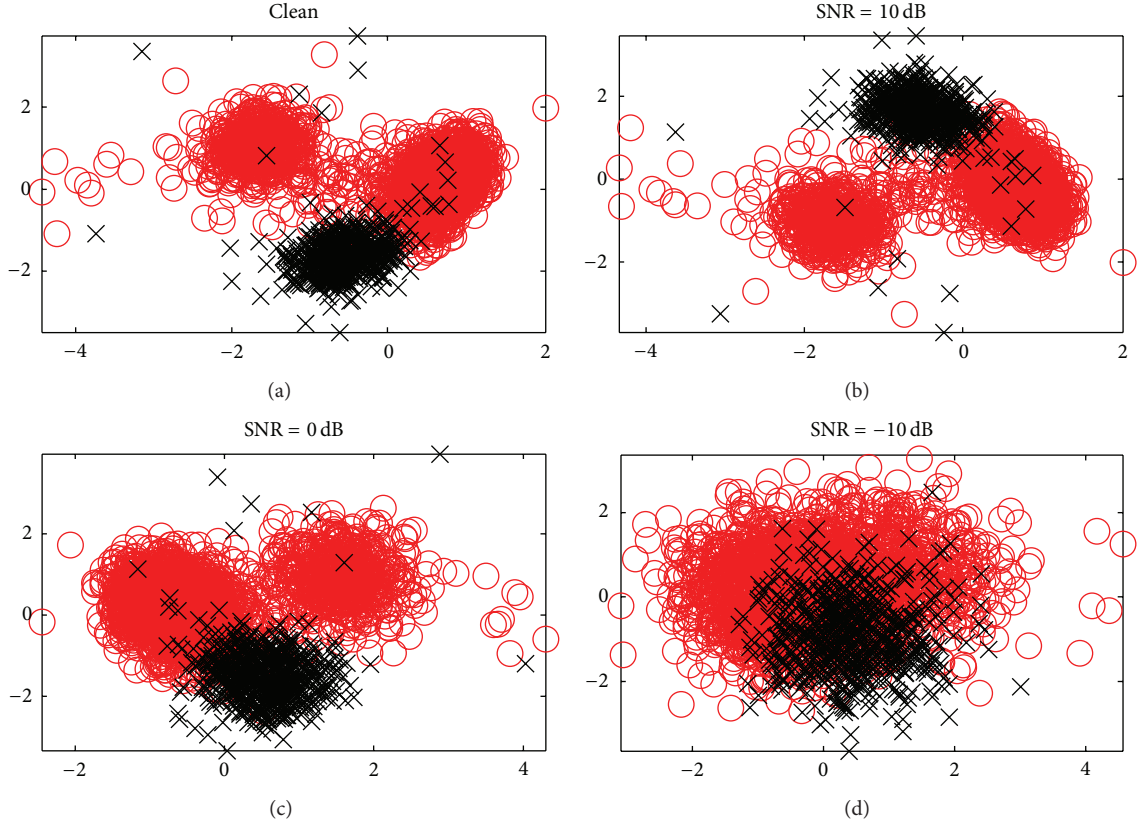


FIGURE 6: PCA waveform features plus noise for Dataset 2 of HCl. Features for the “IC neuron” are shown with black “X” markers.

two free parameters L and τ on spike sorting performance on the WaveClus dataset. We first study the impact of increasing L on classification accuracy with τ determined as described in Section 2.4. Then, with a fixed value of L (we chose $L = 1000$) we study the effect of τ on the accuracy over a reasonable range. To evaluate, we simply compute the classification accuracy for the best match between the set of true clusters and the putative result.

In Figure 7, we plot the classification error rate for our spike sorting method with the value of τ determined empirically for values of L ranging from 100 to 10000 paths on a logarithmically scaled ordinate axis. For the Easy1 dataset and at higher SNR levels, the performance is largely unaffected by the number of paths L . For Difficult1 dataset, the value of L has a much more significant impact on the outcome. The impact is more pronounced for lower SNR levels, reducing the error rate for Difficult1, 5 dB SNR case, from 7.5% to 5.3% across the extremes of the range. The results in Figure 7 illustrate a trade-off of accuracy against computation and memory requirements, both of which increase with L , and suggest that except in difficult, high-noise conditions, the number of paths L can be effectively reduced with minimal impact on performance.

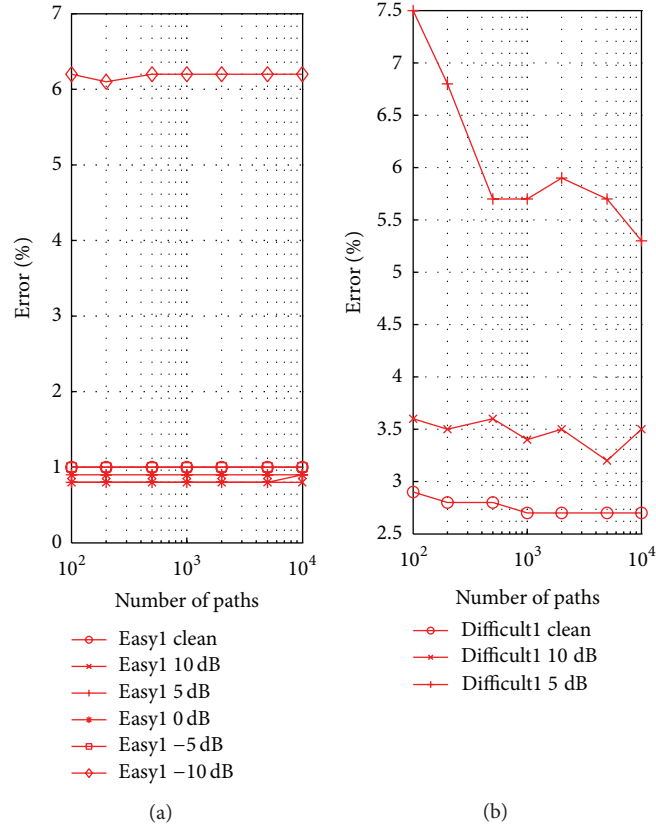
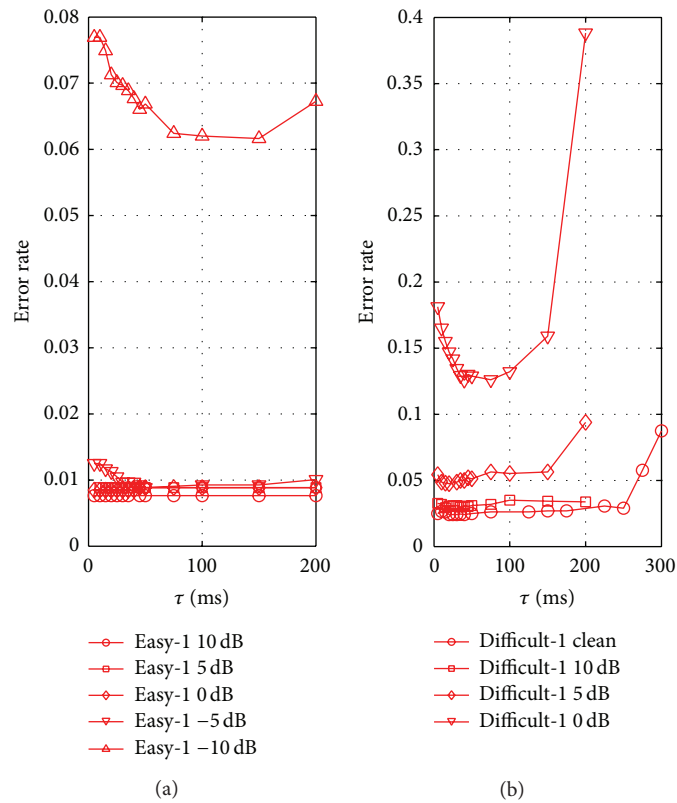
To evaluate the impact of the history window length τ , we apply our spike sorting procedure to the WaveClus data over a range of values, this time holding the number of paths L fixed. Choosing a value of $L = 1000$, we implement our procedure

for values of τ ranging up to 300 ms. Plots of the error rates obtained on the WaveClus dataset are given in Figure 8. As with L , the accuracy is less sensitive to the value of τ in easier, low noise conditions.

4. Discussion

Our probabilistic method for spike sorting is motivated by the idea that both the spike waveforms and their corresponding firing times constitute observed data useful for making inferences about the underlying hidden process of which neurons produced them. In doing so, we incorporate relevant data largely unused by many traditional spike sorting approaches. We combine a single Gaussian model of spike waveforms and a first-order renewal process model of firing times for each neuron into a joint probabilistic model of several neurons in the vicinity of an electrode. The observed firing times are modeled as the aggregation of K independent point processes.

Our method is designed to improve accuracy over waveform-only spike sorting methods, especially under conditions to which these methods are particularly sensitive, such as the presence of high noise and having similar wave shapes for distinct neurons. Our method consists of a joint probabilistic model of multivariate single Gaussians for the first two principal components of neural spike waveforms

FIGURE 7: Error rate versus L , the number of paths.FIGURE 8: Error rate versus the window length τ .

and log-normal and Poisson ISI models for the neural spike trains. To study the impact of incorporating neural spike train models we compared the performance of our method to a waveform-only clustering of multivariate Gaussians using the expectation-maximization GMM (EM-GMM) algorithm and the state-of-the-art WaveClus algorithm. First, we evaluated all methods on a modified version of the semiartificial WaveClus dataset, for which all neuronal labels and firing times are known with certainty. While the state-of-the-art WaveClus method outperforms our method on this dataset, we achieved significant reductions in the error rate versus the EM-GMM waveform-only model in two general cases: (1) “Easy” waveforms (i.e., distinct wave shapes for distinct neurons) in the presence of high additive background noise and (2) “Difficult” waveforms with relatively low noise. In both of these cases, there is significant overlap in the waveform feature space, but the firing times, which are determined through detecting threshold-crossing spikes, are not significantly impacted. Under these conditions, our model uses all of the information and converges on a better result after an iterative procedure.

In the second dataset, composed of completely real data, our method consistently achieves better performance than both the baseline waveform-only EM method and the WaveClus method. It should be restated here that error rates reported on this dataset reflect only one out of three neurons and are only a partial measure of total classification or clustering performance. We achieve considerable reductions in the error rate versus the baseline on Dataset 1 at all noise levels and on Dataset 2 in relatively high noise (i.e., -5 dB SNR). As with the “Easy” data set from the semiartificial WaveClus data, when the baseline EM-GMM method already achieves a low error rate, our proposed method achieves only a small improvement. With Dataset 1 and the high noise or “Difficult” configurations of the WaveClus set, the baseline error is already relatively high, and our model reduces it significantly (as much as 1.98% for the WaveClus data and 5.04% for Dataset 1). However, in extremely high noise (-10 dB for both real datasets and 0 dB for the “Difficult” WaveClus dataset) our method obtains high error rates (greater than 30% error rate). The likely cause for the failure in extremely high noise is that our method is sensitive to its initial clustering (recall that the waveform-only EM-GMM method is actually used to provide the initial clustering for our proposed procedure) and our iterative parameter estimation procedure described in Section 2.2 is unable to converge. Therefore, our spike sorting procedure, which uses models of neural spike trains, achieves the most impactful reductions in error rate on problems of intermediate difficulty.

Contributions of this work include modeling the likelihood of the observed point process of firing times as the joint likelihood of an ensemble of spike trains; a mathematical derivation of the joint likelihood of waveforms, firing times, and labels in terms of previous data points; and the derivation of an iterative clustering and parameter estimation procedure using a piecewise ISI likelihood expression. Our procedure for clustering and parameter estimation operates by alternately maximizing the data likelihood and estimating new

parameters based on the result. While the basic idea is similar to expectation-maximization or Viterbi-based parameter estimation in HMMs, our procedure is suboptimal with respect to the data likelihood. Since our approach depends on retaining a large number of the highest likelihood paths, performance, then, depends on the available computational resources. For this reason, we studied the impact of the number of stored paths on spike sorting performance and found that, except in the most difficult, high-noise conditions, we could reduce the number of paths by a factor of 10 without a significant loss in accuracy.

It should be noted that some issues important to the spike sorting problem, and to clustering problems in general, have not been directly addressed here. Using datasets each containing the same number of neurons, we have not addressed determining the number of neurons automatically. Since we initialize our procedure with a waveform-only GMM clustering, determining the number of clusters using the Aikake information criterion (AIC), the Bayesian information criterion (BIC), or several other methods requires only a simple extension. However, the iterative clustering procedure can potentially prune out clusters dynamically. A measured approach to pruning out clusters (including initializing with a high number of clusters) is a topic for future study. Overlapping spikes, changes in waveform shape due to electrode drift, and waveform attenuation, especially due to bursting activity, are also important topics for future research. Though our findings suggest that our algorithm, which incorporates firing information, might be particularly robust to electrode drift and waveform attenuation, since it improves over the baseline with difficult waveform shapes and high noise, this should be verified experimentally in future research.

4.1. Comparisons to Related Work. In this paper, we have proposed and demonstrated a novel spike sorting method motivated by the idea that both observed spike waveforms and observed information about the timing of spike events are potentially useful for the spike sorting task. A number of other recent studies have introduced spike sorting methods based on this general idea and, in this section, we contrast these approaches and their merits to ours.

In [14], Sahani introduced a statistical model of neuronal action potentials, particularly well suited for “bursts” of neural firings, that is, sequences of very rapidly occurring action potentials. The method for spike sorting proposed in [14], termed Sparse Hidden Markov Models (SHMMs), is a special case of the well-known hidden Markov model (HMM), but with the restriction that the majority of outputs are expected to be in a null state corresponding to nonfirings. The SHMM, as applied to modeling neural firings in [14] involves partitioning the neural firing activity into relatively long (0.5 ms) equal length time bins. Transitions between states are then restricted in the HMM transition matrix in such a way as to model neural bursting behavior. The SHMM has multiple nonnull states to model changes in the amplitude expected during bursting behavior and multiple null states which effectively keep track of how much time

has passed since the last firing. In this study, the author recommends using simpler models for nonbursting neural cells. Sahani’s approach is different from ours in several important ways. Instead of explicitly modeling transitions between states using a discrete set of transition probabilities as with an HMM, we incorporate temporal information using a likelihood model of the continuous-valued interval between firings. This means that we model the precise timing of firing events, rather than grouping them into short time bins. Furthermore, our approach models the joint likelihood of an ensemble of point processes, which differs significantly from both the SHMM approach used in [14] and the general HMM approach as well.

Bar-Hillel et al. introduced a nonstationary method for spike sorting designed to account for the apparent nonstationary nature of spike waveform data [15]. The method, referred to in the study as a “chain of Gaussian mixtures,” first segments neural firing data into short frames, which are either equal in length or contain an equal number of neural firings. The authors fit a separate GMM to the data in each frame (thus assuming stationarity within the frame), compute a set of transition probabilities between frames, and finally solve for the best maximum a posteriori clustering across the whole dataset using a variant of the Viterbi algorithm. This study directly incorporates temporal information into the spike sorting operation and is particularly well suited to account for waveform drift, where the amplitude and shape of the spike waveform are affected by movements in the electrode over time. While this method incorporates temporal information into spike sorting, it does so not by modeling either time occurrences of neural firings or the neural firing rate. Rather, the method is essentially a nonstationary model of spike waveforms, in which waveform parameters change over time. Similar to HMMs, the approach in [15] explicitly models transitions between time frames, which is very different from our approach. Furthermore, although our approach is explicitly designed to model temporal information in the extracellular signal, it is actually a *stationary* probabilistic model. As discussed in Section 2.3, all of the parameters $\lambda = \{\theta, \phi_{\text{isi}}, \phi_{\text{init}}\}$ for the joint model belong individually to stationary distributions. The ISI and “first firing” distributions, which account for temporal information in the data, are both stationary models of point processes.

Ventura introduced a method in [18], in which neural firing rates are incorporated into the spike sorting operation. The method consists of a probabilistic model of spike waveforms and an analytic model $\lambda_i(c)$ of the firing rate for neuron i as modulated by covariate information in the value c , such as an applied stimulus or a tuning curve or some other known experimental condition that impacts neural firing rates. The firing rate model $\lambda_i(c)$ in Ventura’s method, which can be either parametric or nonparametric, is used to determine the probability of each possible combination of the K neurons in the vicinity of the electrode firing in short, equal length time bins. Modeling all possible combinations of spikes makes the method particularly well suited to accounting for overlapping spike waveforms. Since firing activity is incorporated in the form of $\lambda_i(c)$, given in units of spike per msec, the method implicitly models neural firing as an inhomogeneous Poisson

process. The paper introduces an expectation-maximization procedure for estimating all of the parameters of the model. In experiments with simulated data, Ventura showed that a parametric model of firing rates $\lambda_i(c)$ could be used when the form of the firing with respect to the covariate quantity is known and that a nonparametric model can be used in more general cases. Ventura’s approach is based on simultaneously performing spike sorting and tuning in an integrated procedure and, as such, it depends directly on modeling some known experimental condition or covariate quantity; this is in contrast with our proposed approach which does not depend on modeling or observing any covariate information. Instead, we assume only a general analytic form of the observed firing times as point processes. Also, an important part of our approach is explicitly modeling the distribution of the observed firing times. We used the lognormal distribution in this paper for its facility in modeling the necessary minimum refractory period, but our method is not limited to this or any other distribution assumptions.

A probabilistic model of spike peak amplitudes and firings was presented by Pouzat et al. in [16]. In that study, the authors advocate modeling the temporal behavior of spike timings using a probability distribution for the interspike interval duration. Also, since it is known that spike amplitude depends on the elapsed time since the last firing, the paper models variation in spike peak amplitude based on ISI durations as well. Much of the development in this study focuses on the use of Markov chain Monte Carlo (MCMC) methods, which allow considerable flexibility in the choice of probability distributions for the data. The MCMC method consists of simulating the posterior density of the model parameters and sampling from that distribution for parameter estimation. The authors use a lognormal distribution for neural firing ISIs and evaluate their model on simulated data. In another study [17], the authors extend the MCMC methodology for spike sorting, this time using an HMM. In [17], *sequences* of ISI duration observations for each neuron were modeled with a 3-state HMM having the lognormal distribution as the emission probability density function for each state. The HMM method was applied to a real dataset, exhibiting bursty neural firings recorded from Purkinje cells in rats, achieving high accuracy. While these two studies use continuous-valued ISI distributions for temporal modeling, they use MCMC sampling for parameter estimation, in lieu of an analytical development for the model. While the MCMC methodology generally allows considerable freedom in constructing compound probabilistic models, methods developed analytically are inherently simpler and more transparent. A key contribution of our approach is the mathematical development of a recursive likelihood model of the data, including a piecewise ISI term, and an iterative procedure for clustering and parameter estimation based on that model.

4.2. Summary. We have developed a model of observed, threshold-crossing neuronal firing times as the aggregation of K point processes and incorporated it into a joint waveform-

and ISI-based framework. We developed an analytic expression of the joint likelihood of the observed and hidden data and formulated a recursive expression for the likelihood at any time n in terms of previous data. The ISI likelihood at time n is a piecewise expression that depends on whether previous spikes occur in a time history window. We developed an iterative procedure for clustering the data and estimating parameters based on finding the best path through a lattice structure. Our method outperformed the baseline, waveform-only GMM in noisy and otherwise difficult signal conditions on a semiartificial data. On a completely real dataset, our proposed approach outperformed both the baseline and a state-of-the-art method. We showed that we can obtain improvements in accuracy and computational efficiency by tuning our model's 2 hyperparameters. Our future work includes developing more sophisticated methods of pruning the search space for the best path and developing a more rigorous clustering method.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] F. H. Guenther, J. S. Brumberg, E. Joseph Wright et al., "A wireless brain-machine interface for real-time speech synthesis," *PLoS ONE*, vol. 4, no. 12, Article ID e8218, 2009.
- [2] L. R. Hochberg, M. D. Serruya, G. M. Fries et al., "Neuronal ensemble control of prosthetic devices by a human with tetraplegia," *Nature*, vol. 442, no. 7099, pp. 164–171, 2006.
- [3] M. S. Lewicki, "A review of methods for spike sorting: the detection and classification of neural action potentials," *Network*, vol. 9, no. 4, pp. R53–78, 1998.
- [4] E. M. Schmidt, "Computer separation of multi-unit neuroelectric data: a review," *Journal of Neuroscience Methods*, vol. 12, no. 2, pp. 95–111, 1984.
- [5] E. N. Brown, R. E. Kass, and P. P. Mitra, "Multiple neural spike train data analysis: state-of-the-art and future challenges," *Nature Neuroscience*, vol. 7, no. 5, pp. 456–461, 2004.
- [6] R. Barbieri, M. C. Quirk, L. M. Frank, M. A. Wilson, and E. N. Brown, "Construction and analysis of non-Poisson stimulus-response models of neural spiking activity," *Journal of Neuroscience Methods*, vol. 105, no. 1, pp. 25–37, 2001.
- [7] E. N. Brown, R. Barbieri, V. Ventura, R. E. Kass, and L. M. Frank, "The time-rescaling theorem and its application to neural spike train data analysis," *Neural Computation*, vol. 14, no. 2, pp. 325–346, 2002.
- [8] I. Park and J. C. Principe, "Quantification of inter-trial non-stationarity in spike trains from periodically stimulated neural cultures," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '10)*, pp. 5442–5445, IEEE, March 2010.
- [9] F. Wood, P. Prabhat, J. P. Donoghue, and M. J. Black, "Inferring attentional state and kinematics from motor cortical firing rates," in *Proceedings of the 27th Annual International Conference of the Engineering in Medicine and Biology Society*, pp. 149–152, September 2005.
- [10] S.-P. Kim, J. D. Simeral, L. R. Hochberg, J. P. Donoghue, G. M. Fries, and M. J. Black, "Multi-state decoding of point-and-click control signals from motor cortical activity in a human with tetraplegia," in *Proceedings of the 3rd International IEEE EMBS Conference on Neural Engineering*, pp. 486–489, May 2007.
- [11] W. Wu, M. J. Black, Y. Gao et al., "Neural decoding of cursor motion using a kalman filter," in *Advances in Neural Information Processing Systems 15*, S. Becker, S. Thrun, and K. Obermayer, Eds., pp. 133–140, The MIT Press, Cambridge, Mass, USA, 2003.
- [12] Y. Gao, M. J. Black, E. Bienenstock, S. Shoham, and J. Donoghue, "Probabilistic inference of hand motion from neural activity in motor cortex," in *Advances in Neural Information Processing Systems*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds., vol. 14, pp. 221–228, The MIT Press, Cambridge, Mass, USA, 2002.
- [13] W. Wu, Y. Gao, E. Bienenstock, J. P. Donoghue, and M. J. Black, "Bayesian population decoding of motor cortical activity using a Kalman filter," *Neural Computation*, vol. 18, no. 1, pp. 80–118, 2006.
- [14] M. Sahani, *Latent variable models for neural data analysis [Ph.D. thesis]*, California Institute of Technology, 1999.
- [15] A. Bar-Hillel, A. Spiro, and E. Stark, "Spike sorting: bayesian clustering of non-stationary data," *Journal of Neuroscience Methods*, vol. 157, no. 2, pp. 303–316, 2006.
- [16] C. Pouzat, M. Delescluse, P. Viot, and J. Diebolt, "Improved spike-sorting by modeling firing statistics and burst-dependent spike amplitude attenuation: a Markov chain Monte Carlo approach," *Journal of Neurophysiology*, vol. 91, no. 6, pp. 2910–2928, 2004.
- [17] M. Delescluse and C. Pouzat, "Efficient spike-sorting of multi-state neurons using inter-spike intervals information," *Journal of Neuroscience Methods*, vol. 150, no. 1, pp. 16–29, 2006.
- [18] V. Ventura, "Automatic spike sorting using tuning information," *Neural Computation*, vol. 21, no. 9, pp. 2466–2501, 2009.
- [19] B. Matthews and M. Clements, "Joint modeling of observed inter-arrival times and waveform data with multiple hidden states for neural spike-sorting," in *Proceedings of the 36th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '11)*, pp. 637–640, May 2011.
- [20] B. Matthews and M. Clements, "Joint waveform and firing rate Spike-Sorting for continuous extracellular traces," in *Proceedings of the 45th Asilomar Conference on Signals, Systems and Computers*, November 2011.
- [21] K. Kang and S.-I. Amari, "Discrimination with spike times and ISI distributions," *Neural Computation*, vol. 20, no. 6, pp. 1411–1426, 2008.
- [22] R. Q. Quiroga, Z. Nadasdy, and Y. Ben-Shaul, "Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering," *Neural Computation*, vol. 16, no. 8, pp. 1661–1687, 2004.
- [23] D. A. Henze, Z. Borhegyi, J. Csicsvari, A. Mamiya, K. D. Harris, and G. Buzsáki, "Intracellular features predicted by extracellular recordings in the hippocampus in vivo," *Journal of Neurophysiology*, vol. 84, no. 1, pp. 390–400, 2000.

Research Article

A Tensor-Product-Kernel Framework for Multiscale Neural Activity Decoding and Control

**Lin Li,¹ Austin J. Brockmeier,² John S. Choi,³ Joseph T. Francis,⁴
Justin C. Sanchez,⁵ and José C. Principe²**

¹ Philips Research North America, Briarcliff Manor, NY 10510, USA

² Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611, USA

³ Joint Program in Biomedical Engineering, NYU Polytechnic School of Engineering and SUNY Downstate, Brooklyn, NY 11203, USA

⁴ Department of Physiology and Pharmacology, State University of New York Downstate Medical Center, Joint Program in Biomedical Engineering, NYU Polytechnic School of Engineering and SUNY Downstate, Robert F. Furchgott Center for Neural & Behavioral Science, Brooklyn, NY 11203, USA

⁵ Department of Biomedical Engineering, Department of Neuroscience, Miami Project to Cure Paralysis, University of Miami, Coral Gables, FL 33146, USA

Correspondence should be addressed to Lin Li; lin-li@philips.com

Received 22 August 2013; Revised 28 January 2014; Accepted 11 February 2014; Published 14 April 2014

Academic Editor: Zhe (Sage) Chen

Copyright © 2014 Lin Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Brain machine interfaces (BMIs) have attracted intense attention as a promising technology for directly interfacing computers or prostheses with the brain's motor and sensory areas, thereby bypassing the body. The availability of multiscale neural recordings including spike trains and local field potentials (LFPs) brings potential opportunities to enhance computational modeling by enriching the characterization of the neural system state. However, heterogeneity on data type (spike timing versus continuous amplitude signals) and spatiotemporal scale complicates the model integration of multiscale neural activity. In this paper, we propose a tensor-product-kernel-based framework to integrate the multiscale activity and exploit the complementary information available in multiscale neural activity. This provides a common mathematical framework for incorporating signals from different domains. The approach is applied to the problem of neural decoding and control. For neural decoding, the framework is able to identify the nonlinear functional relationship between the multiscale neural responses and the stimuli using general purpose kernel adaptive filtering. In a sensory stimulation experiment, the tensor-product-kernel decoder outperforms decoders that use only a single neural data type. In addition, an adaptive inverse controller for delivering electrical microstimulation patterns that utilizes the tensor-product kernel achieves promising results in emulating the responses to natural stimulation.

1. Introduction

Brain machine interfaces (BMIs) provide new means to communicate with the brain by directly accessing, interpreting, and even controlling neural states. They have attracted attention as a promising technology to aid the disabled (i.e., spinal cord injury, movement disability, stroke, hearing loss, and blindness) [1–6]. When designing neural prosthetics and brain machine interfaces (BMIs), the fundamental steps involve quantifying the information contained in neural

activity, modeling the neural system, decoding the intention of movement or stimulation, and controlling the spatiotemporal neural activity pattern to emulate natural stimulation. Furthermore, the complexity and distributed dynamic nature of the neural system pose challenges for the modeling tasks.

The development of recording technology enables access to brain activity from multiple functional levels, including the activity of individual neurons (spike trains), local field potentials (LFPs), electrocorticogram (ECoG), and electroencephalogram (EEG), collectively forming a multiscale

characterization of brain state. Simultaneous recording of multiple types of signals could facilitate enhanced neural system modeling. Although there are underlying relationships among these brain activities, it is unknown how to leverage the heterogeneous set of signals to improve the identification of the neural-response-stimulus mappings. The challenge is in defining a framework that can incorporate these heterogeneous signal formats coming from multiple spatiotemporal scales. In our work, we mainly address integrating spike trains and LFPs for multiscale neural decoding.

Spike trains and LFPs encode complementary information of stimuli and behaviors [7, 8]. In most recordings, spike trains are obtained by detecting transient events on a signal that is conditioned using a high-pass filter with the cutoff frequency set at about 300–500 Hz, while LFPs are obtained by using a low-pass filter with 300 Hz cutoff frequency [9]. Spike trains represent single- or multiunit neural activity with a fine temporal resolution. However, their stochastic properties induce considerable variability, especially when the stimulation amplitude is small; that is, the same stimuli rarely elicit the same firing patterns in repeated trials. In addition, a functional unit of the brain contains thousands of neurons. Only the activity of a small subset of neurons can be recorded and, of these, only a subset may modulate with respect to the stimuli or condition of interest.

In contrast, LFPs reflect the average synaptic input to a region near the electrode [10], which limits specificity but provides robustness for characterizing the modulation induced by stimuli. Furthermore, LFPs naturally provide population-level measure of neural activity. Therefore, an appropriate aggregation of LFPs and spike trains enables enhanced accuracy and robustness of neural decoding models. For example, the decoder can coordinate LFPs or spike patterns to tag particularly salient events or extract different stimulus features characterized by multisource signals. However, heterogeneity between LFPs and spike trains complicates their integration into the same model. The information in a spike train is coded in a set of ordered spike timings [11, 12], while an LFP is a continuous amplitude time series. Moreover, the time scale of LFPs is significantly longer than spike trains. Whereas recent work has compared the decoding accuracy of LFPs and spikes [13], only a small number of simple models have been developed to relate both activities [14]. However, the complete relationship between LFPs and spike trains is still a subject of controversy [15–17], which hinders principled modeling approaches.

To address these modeling issues, this paper proposes a signal processing framework based on tensor-product kernels to enable decoding and even controlling multiscale neural activities. The tensor-product kernel uses multiple heterogeneous signals and implicitly defines a kernel space constructed by the tensor product of individual kernels designed for each signal type [18]. The tensor-product kernel uses the joint features of spike trains and LFPs. This enables kernel-based machine learning methodologies to leverage multiscale neural activity to uncover the mapping from the neural system states and the corresponding stimuli.

The kernel least mean square (KLMS) algorithm is used to estimate the dynamic nonlinear mapping from the two

types of neural responses to the stimuli. The KLMS algorithm exploits the fact that the linear signal processing in reproducing kernel Hilbert spaces (RKHS) corresponds to nonlinear processing in the input space and is used in the adaptive inverse control scheme [19] designed for controlling neural systems. Utilizing the tensor-product kernel, we naturally extend this scheme to multiscale neural activity. Since the nonlinear control is achieved via linear processing in the RKHS, it bypasses the local minimum issues normally encountered in nonlinear control.

The validation of the effectiveness of the proposed tensor-product-kernel framework is done in a somatosensory stimulation study. Somatosensory feedback remains underdeveloped in BMI, which is important for motor and sensory integration during movement execution, such as proprioceptive and tactile feedback about limb state during interaction with external objects [20, 21]. A number of early experiments have shown that spatiotemporally patterned microstimulation delivered to somatosensory cortex can be used to guide the direction of reaching movements [22–24]. In order to effectively apply the artificial sensory feedback in BMI, it is essential to find out how to use microstimulation to replicate the target spatiotemporal patterns in the somatosensory cortex, where neural decoding and control are the critical technologies to achieve this goal. In this paper, our framework is applied to leverage multiscale neural activities to decode both natural sensory stimulation and microstimulation. Its decoding accuracy is compared with decoders that use a single type of neural activity (LFPs or spike trains).

In the neural system control scenario, this tensor-product-kernel methodology can also improve the controller performance. Controlling the neural activity via stimulation has raised the prospect of generating specific neural activity patterns in downstream areas, even mimicking natural neural responses, which is central both for our basic understanding of neural information processing and for engineering “neural prosthetic” devices that can interact with the brain directly [25]. From a control theory perspective, the neural circuit is treated as the “plant,” where the applied microstimulation is the control signal and the plant output is the elicited neural response represented by spike trains and LFPs. Most conventional control schema cannot be directly applied to spike trains because there is no algebraic structure in the space of spike trains. Therefore, most existing neural control approaches have been applied to binned spike trains or LFPs [25–31]. Here, we will utilize the kernel-based adaptive inverse controller for spike trains proposed in our previous work [19] as an input-output (system identification) based control scheme. This methodology can directly be extended to the tensor-product kernel to leverage the availability of multiscale neural signals (e.g., spike trains and LFPs) and improves the robustness and accuracy of the stimulation optimization by exploiting the complementary information of the heterogeneous neural signals recorded from multiple sources.

The adaptive inverse control framework controls patterned electrical microstimulation in order to drive neural

responses to mimic the spatiotemporal neural activity patterns induced by tactile stimulation. This framework creates new opportunities to improve the ability to control neural states to emulate the natural stimuli by leveraging the complementary information from multiscale neural activities. This better interprets the neural system internal states and thus enhances the robustness and accuracy of the optimal microstimulation pattern estimation.

The rest of the paper is organized as follows. Section 2 introduces kernels for spike trains and LFPs and the tensor-product kernel that combines them. The kernel-based decoding model and the adaptive inverse control scheme that exploit kernel-based neural decoding technology to enable control in RKHS are introduced in Sections 3 and 4, respectively. Section 5 discusses the somatosensory stimulation emulation scenario and illustrates the test results by applying tensor-product kernel to leverage multiscale neural activity for decoding and controlling tasks. Section 6 concludes this paper.

2. Tensor-Product Kernel for Multiscale Heterogeneous Neural Activity

The mathematics of many signal processing and pattern recognition algorithms is based on evaluating the similarity of pairs of exemplars. For vectors or functions, the inner product defined on Hilbert spaces is a linear operator and a measure of similarity. However, not all data types exist in Hilbert spaces. Kernel functions are bivariate, symmetric functions that implicitly embed samples in a Hilbert space. Consequently, if a kernel on a data type can be defined, then algorithms defined in terms of inner products can be used. This has enabled various kernel algorithms [18, 32–34].

To begin, we define the general framework for the various kernel functions used here, keeping in mind that the input corresponds to assorted neural data types. Let the domain of a single neural response dimension, that is, a single LFP channel or one spiking unit, be denoted by \mathcal{X} and consider a kernel $\kappa : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$. If κ is positive definite, then there is an implicit mapping $\phi : \mathcal{X} \rightarrow \mathcal{H}$ that maps any two sample points, say $x, x' \in \mathcal{X}$, to corresponding elements in the Hilbert space $\phi(x), \phi(x') \in \mathcal{H}$ such that $\kappa(x, x') = \langle \phi(x), \phi(x') \rangle$ is the inner product of these elements in the Hilbert space. As an inner product, the kernel evaluation $\kappa(x, x')$ quantifies the similarity between x and x' .

A useful property is that this inner product induces a distance metric,

$$\begin{aligned} d(x, x') &= \|\phi(x) - \phi(x')\|_{\mathcal{H}} \\ &= \sqrt{\langle \phi(x) - \phi(x'), \phi(x) - \phi(x') \rangle} \\ &= \sqrt{\kappa(x, x) + \kappa(x', x') - 2\kappa(x, x')}. \end{aligned} \quad (1)$$

For normalized and shift-invariant kernels where, for all $x, \kappa(x, x) = 1$, the distance is inversely proportional to the kernel evaluation $d(x, x') = \sqrt{2 - 2\kappa(x, x')}$.

To utilize two or more dimensions of the neural response, a kernel that operates on the joint space is required. There are two basic approaches to construct multidimensional kernels from kernels defined on the individual variables: direct sum and tensor-product kernels. In terms of kernel evaluations, they consist of taking either the sum or the product of the individual kernel evaluations. In both cases, the resulting kernel is positive definite as long as the individual kernels are positive definite [18, 35].

Let \mathcal{X}_i denote the neural response domain of the i th dimension and consider a positive-definite kernel $\kappa_i : \mathcal{X}_i \times \mathcal{X}_i \rightarrow \mathbb{R}$ and corresponding mapping $\phi_i : \mathcal{X}_i \rightarrow \mathcal{H}_i$ for this dimension. Again, the similarity between a pair of samples x and x' on the i th dimension is $\kappa_i(x_{(i)}, x'_{(i)}) = \langle \phi_i(x_{(i)}), \phi_i(x'_{(i)}) \rangle$.

For the sum kernel, the joint similarity over a set of dimensions \mathcal{J} is

$$\kappa_{\Sigma}(\mathbf{x}, \mathbf{x}') = \sum_{i \in \mathcal{J}} \kappa_i(x_{(i)}, x'_{(i)}). \quad (2)$$

This measure of similarity is an average similarity across all dimensions. When the sum is over a large number of dimensions, the contributions of individual dimensions are diluted. This is useful for multiunit spike trains or multichannel LFPs when the individual dimensions are highly variable, which if used individually would result in a poor decoding performance on a single trial basis.

For the tensor-product kernel, the joint similarity over two dimensions i and j is computed by taking the product between the kernel evaluations $\kappa_{[i,j]}([x_{(i)}, x_{(j)}], [x'_{(i)}, x'_{(j)}]) = \kappa_i(x_{(i)}, x'_{(i)}) \cdot \kappa_j(x_{(j)}, x'_{(j)})$. The new kernel $\kappa_{[i,j]}$ corresponds to a mapping function that is the tensor product between the individual mapping functions $\phi_{[i,j]} = \phi_i \otimes \phi_j$ where $\phi_{[i,j]}(x_{(i,j)}) \in \mathcal{H}_{[i,j]}$. This is the tensor-product Hilbert space. The product can be taken over a set of dimensions \mathcal{J} and the result is a positive-definite kernel over the joint space: $\kappa_{\Pi}(\mathbf{x}, \mathbf{x}') = \prod_{i \in \mathcal{J}} \kappa_i(x_{(i)}, x'_{(i)})$.

The tensor-product kernel corresponds to a stricter measure of similarity than the sum kernel. Due to the product, if for one dimension $\kappa_i(x_{(i)}, x'_{(i)}) \approx 0$ then $\kappa_{\Pi}(\mathbf{x}, \mathbf{x}') \approx 0$. The tensor-product kernel requires the joint similarity; that is, for samples to be considered similar in the joint space they must be close in all the individual spaces. If even one dimension is dissimilar the product will appear dissimilar. If some of the dimensions are highly variable, then they will have a deleterious effect on the joint similarity measure. On the other hand, the tensor product is a more precise measure of similarity that will be used later to combine multiscale neural activity.

More generally, an explicit weight can be used to adjust the influence of the individual dimensions on the joint kernel. Any convex combinations of kernels are positive definite, and learning the weights of this combination is known as multiple kernel learning [36–38]. In certain cases of the constituent kernels, namely, that they are infinitely divisible [39], a weighted product kernel can also be applied [40]. However, the optimization of these weightings is not explored in the current work.

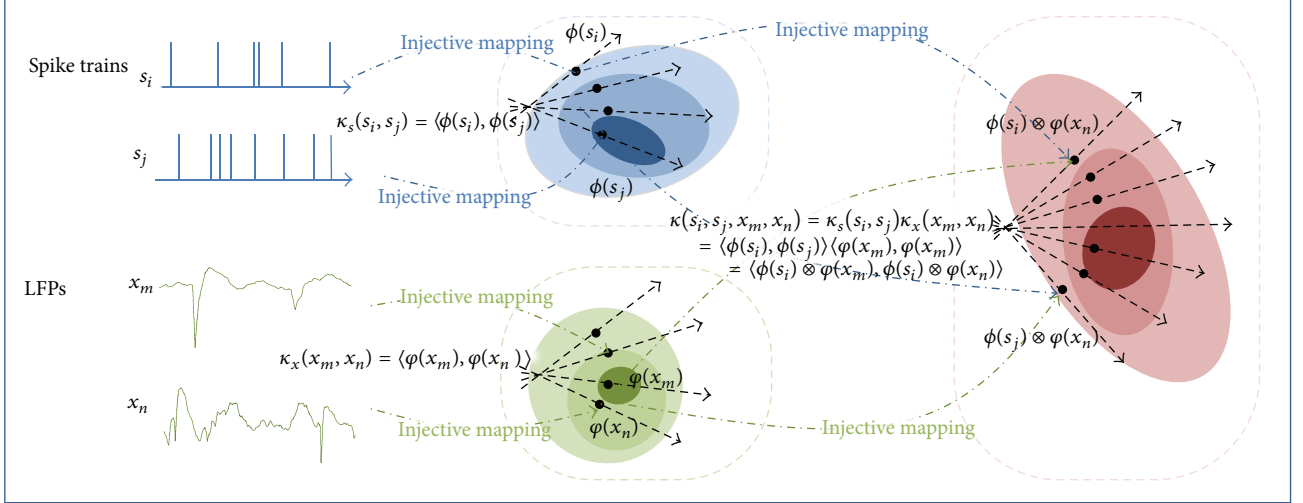


FIGURE 1: Schematic representation of the construction of the RKHS defined by the tensor-product kernel from the individual spike and LFP kernel spaces, along with the mapping from the original data. Specifically, s_i denotes a window of multiunit spike trains; x_n denotes a window of multichannel LFPs; $\kappa_s(\cdot, \cdot)$ denotes the spike train kernel with implicit mapping function $\phi(\cdot)$; and $\kappa_x(\cdot, \cdot)$ denotes the LFP kernel with implicit mapping function $\phi(\cdot)$.

In general, a joint kernel space, constructed via either direct sum or tensor product, allows the homogeneous processing of heterogeneous signal types all within the framework of RKHSs. We use direct sum kernels to combine the different dimensions of multiunit spike trains or multichannel LFPs. For the spike trains, using the sum kernel across the different units enables an “average” population similarity over the space of spike trains where averages cannot be computed. Then a tensor-product kernel combines the two kernels: one for the multiunit spike trains and one for the multichannel LFPs; see Figure 1 for an illustration. The kernels for spike trains and LFPs can be selected and specified individually according to their specific properties.

In conclusion, composite kernels are very different from those commonly used in kernel-based machine learning, for example, for the support vector machine. In fact, here a pair of windowed spike trains and windowed LFPs is mapped into a feature function in the joint RKHS. Different spike train and LFP pairs are mapped to different locations in this RKHS, as shown in Figure 1. Due to its versatility, Schoenberg kernels defined for both the spike timing space and LFPs are employed in this paper and discussed below.

2.1. Kernel for Spike Trains. Unlike conventional amplitude data, there is no natural algebraic structure in the space of spike trains. The binning process, which easily transforms the point processes into discrete amplitude time series, is widely used in spike train analysis and allows the application of conventional amplitude-based kernels to spike trains [41] at the expense of losing the temporal resolution of the neural responses. This means that any temporal information in spikes within and between bins is disregarded, which is especially alarming when spike timing precision can be in the millisecond range. Although the bin size can be set small enough to preserve the fine time resolution, it will sparsify

the signal representation, increase the artifact variability, and cause high-dimensionality in the model, which requires voluminous data for proper training.

According to the literature, it is appropriate to consider a spike train to be a realization of a point process, which describes the temporal distribution of the spikes. Generally speaking, a point process p_i can be completely characterized by its conditional intensity function $\lambda(t | H_t^i)$, where $t \in \tau = [0, T]$ denotes the time coordinate and H_t^i is the history of the process up to t . A recent research area [42, 43] is to define an injective mapping from spike trains to RKHS based on the kernel between the conditional intensity functions of two point processes [42]. Among the cross-intensity (CI) kernels, the Schoenberg kernel is defined as

$$\begin{aligned} \kappa(\lambda(t | H_t^i), \lambda(t | H_t^j)) &= \exp\left(-\frac{\|\lambda(t | H_t^i) - \lambda(t | H_t^j)\|^2}{\sigma^2}\right) \\ &= \exp\left(-\frac{\int_{\tau} (\lambda(t | H_t^i) - \lambda(t | H_t^j))^2 dt}{\sigma^2}\right), \end{aligned} \quad (3)$$

where σ is the kernel size. The Schoenberg kernel is selected in this work because of its modeling accuracy and robustness to free parameter settings [44]. The Schoenberg kernel is a Gaussian-like kernel defined on intensity functions that is strictly positive definite and sensitive to the nonlinear coupling of two intensity functions [42]. Different spike trains will then be mapped to different locations in the RKHS. Compared to kernels designed on binned spike trains (e.g., spikernel [41]), the main advantage of the Schoenberg kernel is that the precision in the spike event location is better preserved and the limitations of the sparseness and

high-dimensionality for model building are also avoided, resulting in enhanced robustness and reduced computational complexity, especially when the application requires fine time resolution [44].

In order to be applicable, the methodology must lead to a simple estimation of the quantities of interest (e.g., the kernel) from experimental data. A practical choice used in our work estimates the conditional intensity function using a kernel smoothing approach [42, 45], which allows estimating the intensity function from a single realization and nonparametrically and injectively maps a windowed spike train into a continuous function. The estimated intensity function is obtained by simply convolving $s(t)$ with the smoothing kernel $g(t)$, yielding

$$\hat{\lambda}(t) = \sum_{m=1}^M g(t - t_m), \quad \{t_m \in \mathcal{T} : m = 1, \dots, M\}, \quad (4)$$

where the smoothing function $g(t)$ integrates to 1. Here $\hat{\lambda}(t)$ can be interpreted as an estimation of the intensity function. The rectangular and exponential functions [42, 46] are both popular smoothing kernels, which guarantee injective mappings from the spike train to the estimated intensity function. In order to decrease the kernel computation complexity, the rectangular function $g(t) = (1/\mathcal{T})(U(t) - U(t - \mathcal{T}))$ ($\mathcal{T} \gg$ the interspike interval) is used in our work, where $U(t)$ is a Heaviside function. This rectangular function approximates the cumulative density function of spikes counts in the window T and compromises the locality of the spike trains; that is, the mapping places more emphasis on the early spikes than the later ones. However, our experiments show that this compromise only causes a minimal impact on the kernel-based regression performance.

Let $s_i^n(t)$ denote the spike train for the i th sample of the n th spiking unit. The multiunit spike kernel is taken as the unweighted sum over the kernels on the individual units

$$\kappa_s(\mathbf{s}_i(t), \mathbf{s}_j(t)) = \sum_n \kappa_s(s_i^n(t), s_j^n(t)). \quad (5)$$

2.2. Kernels for LFPs. In contrast with spike trains, LFPs exhibit less spatial and temporal selectivity [15]. In the time domain, LFP features can be obtained by sliding a window on the signal, which describes the temporal LFP structure. The length of the window is selected based on the duration of neural responses to certain stimuli; the extent of the duration can be assessed by its autocorrelation function, as we will discuss in Section 5.3.1. In the frequency domain, the spectral power and phase in different frequency bands are also known to be informative features for decoding, but here we concentrate only on the time-domain decompositions. In the time domain, we can simply treat single channel LFPs as a time series and apply the standard Schoenberg kernel to the sequence of signal samples in time. The Schoenberg

kernel, defined in continuous spaces, maps the correlation time structure of the LFP $x(t)$ into a function in RKHS,

$$\begin{aligned} \kappa_x(x_i(t), x_j(t)) &= \exp\left(-\frac{\|x_i(t) - x_j(t)\|^2}{\sigma_x^2}\right) \\ &= \exp\left(-\frac{\int_{\mathcal{T}_x} (x_i(t) - x_j(t))^2 dt}{\sigma_x^2}\right) \\ &= \exp\left(-\left(\int_{\mathcal{T}_x} x_i(t) x_i(t) + x_j(t) x_j(t) - 2x_i(t) x_j(t) dt\right) \times (\sigma_x^2)^{-1}\right), \end{aligned} \quad (6)$$

where $\mathcal{T}_x = [0, T_x]$.

Let $x_i^n(t)$ denote the LFP waveform for the i th sample of the n th channel. Similar to the multiunit spike kernel, the multichannel LFP kernel is defined by the direct sum kernel

$$\kappa_x(\mathbf{x}_i(t), \mathbf{x}_j(t)) = \sum_n \kappa_x(x_i^n(t), x_j^n(t)). \quad (7)$$

2.3. Discrete Time Sampling. Assuming a sampling rate with period τ , let $\mathbf{x}_i = [x_i^1, x_{i+1}^1, \dots, x_{i-1+T/\tau}^1, x_i^2, \dots, x_{i-1+T/\tau}^N]$ denote the i th multichannel LFP vector obtained by sliding the T -length window with step τ . Let $\mathbf{s}_i = \{t_m - (i-1)\tau, t_m \in [(i-1)\tau, (i-1)\tau + T] : m = 1, \dots, M\}$ denote the corresponding i th window of the multiunit spike timing sequence. The time scale, both in terms of the window length and sampling rate, of the analysis for LFPs and spikes is very important and needs to be defined by the characteristics of each signal. The tensor-product kernel allows the time scales of the analysis for LFPs and spike trains to be specified individually; that is, the window length T and sampling rate for spike trains and LFPs could be different. The suitable time scale can be estimated through autocorrelation coefficients of the signal as will be explained below.

3. Adaptive Neural Decoding Model

For neural decoding applications, a regression model with multiscale neural activities as the input is built to reconstruct a stimulus. The appeal of kernel-based filters is the usage of the linear structure of RKHS to implement well-established linear adaptive algorithms and to obtain a nonlinear filter in the input space that leads to universal approximation capability without the problem of local minima. There are several candidate kernel-based regression methods [32], such as support vector regression (SVR) [33], kernel recursive least squares (KRLS), and kernel least mean square (KLMS) [34]. The KLMS algorithm is preferred here because it is an online methodology of low computation complexity.

Input: $\{x_n, d_n\}, n = 1, 2, \dots, N$
Initialization: initialize the weight vector $\Omega(1)$: codebook (set of centers) $\mathcal{C}(0) = \{\}$ and coefficient vector $a(0) = []$
Computation:
For $n = 1, 2, \dots, N$
 (1) compute the output

$$y_n = \langle \Omega(n), \phi(x_n) \rangle = \sum_{j=1}^{\text{size}(\mathcal{C}(n-1))} a_j(n-1) \kappa(x_n, \mathcal{C}_j(n-1))$$

 (2) compute the error, $e_n = d_n - y_n$
 (3) compute the minimum distance in RKHS between x_n and each $x_i \in \mathcal{C}(n-1)$,

$$d(x_n) = \min_j (2 - 2\kappa(x_n, \mathcal{C}_j(n-1)))$$

 (4) if $d(x_n) \leq \varepsilon$, then keep the codebook unchanged: $\mathcal{C}(n) = \mathcal{C}(n-1)$, and update the coefficient of the center closest to x_n :

$$a_k(n) = a_k(n-1) + \eta e(n), \text{ where } k = \underset{j}{\text{argmin}} \sqrt{2 - 2\kappa(x_n, \mathcal{C}_j(n-1))}$$

 (5) otherwise, store the new center: $\mathcal{C}(n) = \{\mathcal{C}(n-1), x_n\}$, $a(n) = [a(n-1), \eta e_n]$
 (6) $\Omega(n+1) = \sum_{j=1}^{\text{size}(\mathcal{C}(n))} a_j(n) \phi(\mathcal{C}_j(n))$
end

ALGORITHM 1: Quantized kernel least mean square (QKLMS) algorithm.

The quantized kernel least mean square (Q-KLMS) is selected in our work to decrease the filter growth. Algorithm 1 shows the pseudocode for the Q-KLMS algorithm with a simple online vector quantization (VQ) method, where the quantization is performed based on the distance between the new input and each existing center. In this work, this distance between a center and the input is defined by their distance in RKHS, which for a shift-invariant normalized kernel for all $x\kappa(x, x) = 1$ is $\|\phi(x_n) - \phi(x_i)\|_2^2 = 2 - 2\kappa(x_n, x_i)$. If the smallest distance is less than a prespecified quantization size ε , the new coefficient ηe_n adjusts the weight of the closest center; otherwise a new center is added. Compared to other techniques [47–50] that have been proposed to curb the growth of the networks, the simple online VQ method is not optimal but is very efficient. Since, in our work, the algorithm must be applied several times to the same data for convergence after the first iteration over the data, we choose $\varepsilon = 0$, which merges the repeated centers and enjoys the same performance as KLMS.

We use the Q-KLMS framework with the multiunit spike kernels, the multichannel LFP kernels, and the tensor-product kernel using the joint samples of LFPs and spike trains. This is quite unlike previous work in adaptive filtering that almost exclusively uses the Gaussian kernel with real-valued time series.

4. Adaptive Inverse Control of the Spatiotemporal Patterns of Neural Activity

As the name indicates, the basic idea of adaptive inverse control is to learn an inverse model of the plant as the controller in Figure 2(a), such that the cascade of the controller and the plant will perform like a unitary transfer function, that is, a perfect wire with some delay. The target plant output is used as the controller's command input. The controller parameters are updated to minimize the dissimilarity between the target

output and the plant's output during the control process, which enables the controller to track the plant variation and cancel system noises. The filtered- ε LMS adaptive inverse control diagram [51] shown in Figure 2(a) represents the filtered- ε approach to find $\widehat{C}(z)$. If the ideal inverse controller $C(z)$ is the actual inverse controller, the mean square of the overall system error ϵ_k would be minimized. The objective is to make $\widehat{C}(z)$ as close as possible to the ideal $C(z)$. The difference between the outputs of $C(z)$ and $\widehat{C}(z)$, both driven by the command input, is therefore an error signal ϵ' . Since the target stimulation is unknown, instead of ϵ' , a filtered error ϵ , obtained by filtering the overall system error ϵ_k through the inverse plant model $\widehat{P}^{-1}(z)$, is used for adaptation in place of ϵ' .

If the plant has a long response time, a modeling delay is advantageous to capture the early stages of the response, which is determined by the sliding window length that is used to obtain the inverse controller input. There is no performance penalty from the delay Δ as long as the input to $\widehat{C}(z)$ undergoes the same delay. The parameters of the inverse model $\widehat{P}^{-1}(z)$ are initially modeled offline and updated during the whole system operation, which allows $\widehat{P}^{-1}(z)$ to incrementally identify the inverse system and thus make ϵ approach ϵ' . Moreover, the adaptation enables $\widehat{P}^{-1}(z)$ to track changes in the plant. Thus, minimizing the filter error obtained from $\widehat{P}^{-1}(z)$ makes the controller follow the system variation.

4.1. Relation to Neural Decoding. In this control scheme, there are only two models, $\widehat{C}(z)$ and $\widehat{P}^{-1}(z)$, adjusted during the control process, which share the same input (neural activity) and output types (continuous stimulation waveforms). Therefore, both models perform like neural decoders and can be implemented using the Q-KLMS method we introduced in the previous section. Since all the mathematical models

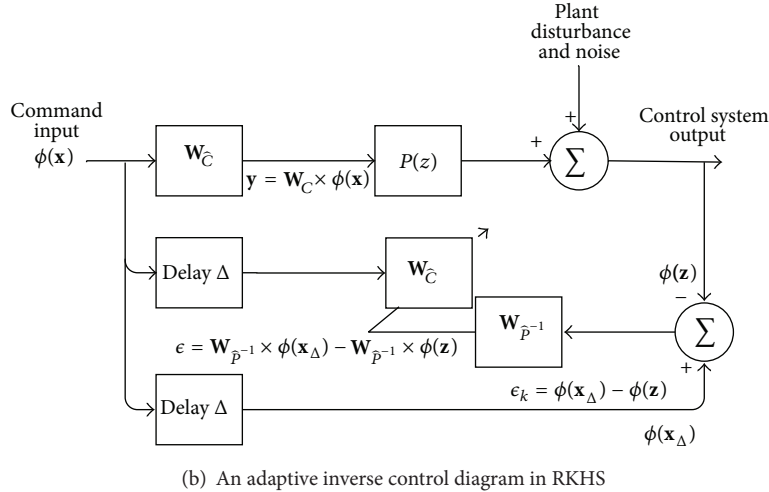
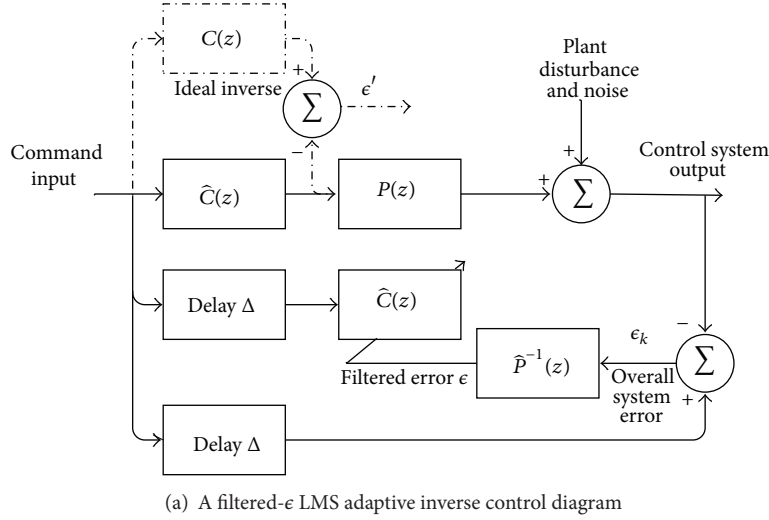


FIGURE 2: Adaptive inverse control diagram.

in this control scheme are kernel-based models, the whole control scheme can be mapped into an RKHS space, which holds several advantages as follows. (1) No restriction is imposed on the signal type. As long as a kernel can map the plant activity to RKHS, the plant can be controlled with this scheme. (2) Both the plant inverse, $\hat{P}^{-1}(z)$, and the controller $\hat{C}(z)$ have linear structure in RKHS, which avoids the danger of converging to local minima.

Specifically, the controller $\hat{C}(z)$ and the plant inverse $\hat{P}^{-1}(z)$ are separately modeled with the tensor-product kernel that we described in Section 2, and the model coefficients are updated with Q-KLMS. This structure is shown in Figure 2(b). The model coefficients $\mathbf{W}_{\hat{C}}$ and $\mathbf{W}_{\hat{P}^{-1}}$ represent the weight matrix of $\hat{C}(z)$ and $\hat{P}^{-1}(z)$ obtained by Q-KLMS, respectively. As this is a multiple-input multiple-output model, $\mathbf{W}_{\hat{C}}$ and $\mathbf{W}_{\hat{P}^{-1}}$ are the concatenation of the filter weights for each stimulation channel.

The variables \mathbf{x} , \mathbf{y} , and \mathbf{z} denote the concatenation of the windowed target spike trains and LFPs as the command input of the controller, the estimated stimulation, and the

plant output, respectively. \mathbf{x}_{Δ} is delayed target signal, which is aligned with the plant output \mathbf{z} . $\phi(\cdot)$ represents the mapping function from input space to the RKHS associated with the tensor-product kernel.

The overall system error is defined as $\epsilon_k = \phi(\mathbf{x}_{\Delta}) - \phi(\mathbf{z})$, which means that the controller's parameter adaptation seeks to minimize the distance in the RKHS between the target spike train/LFP and the output of the plant inverse $\hat{P}^{-1}(z)$. In this way, since the inverse model $\hat{P}^{-1}(z)$ has a linear structure in RKHS, the filtered error for stimulation channel $j \in 1, \dots, M$ is

$$\epsilon(j) = \mathbf{W}_j^{\hat{P}^{-1}} \phi(\mathbf{x}_{\Delta}) - \mathbf{W}_j^{\hat{P}^{-1}} \phi(\mathbf{z}). \quad (8)$$

The controller model $\hat{C}(z)$ has a single input \mathbf{x} , corresponding to the concatenation of the spike trains and LFPs, and has an M -channel output \mathbf{y} , corresponding to the microstimulation. Q-KLMS is used to model $\hat{C}(z)$ with N input samples. The target spike trains are repeated among different trials, which means that the repeated samples will

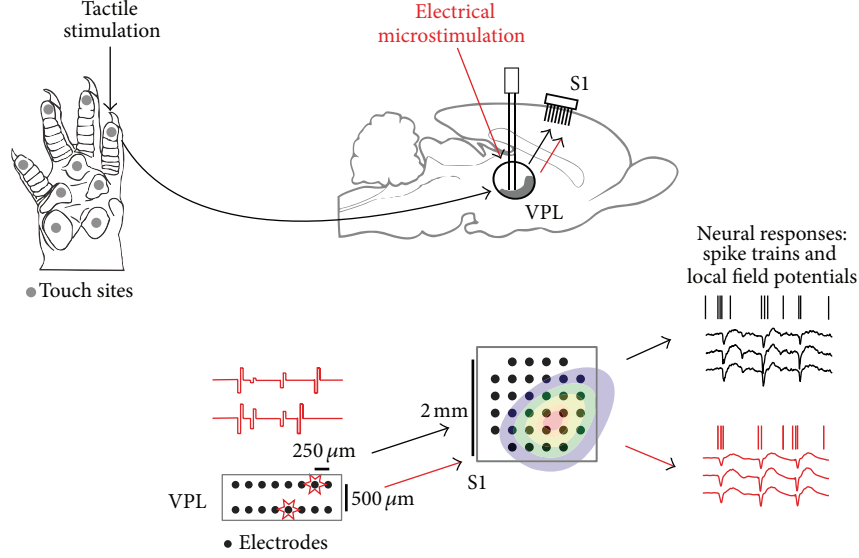


FIGURE 3: Neural elements in tactile stimulation experiments. To the left is the rat's hand with representative cutaneous receptive fields. When the tactor touches a particular "receptive field" on the hand, VPL thalamus receives this information and relays it to S1 cortex. To emulate "natural touch" with microstimulation, the optimized spatiotemporal microstimulus patterns are injected into the same receptive field on VPL thalamus through a microarray so that the target neural activity pattern can be replicated in somatosensory regions (S1) to convey the natural touch sensation to the animal.

be merged on the same kernel center of the first pass through the data by the quantization and thus the network size of the inverse controller is fixed (N centers). Only the coefficient matrix \mathbf{a} is updated with the filtered error ϵ during the whole control operation. The output of $\hat{C}(z)$ can be calculated by

$$\begin{aligned} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{bmatrix} &= \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{12} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & & \vdots \\ a_{M1} & a_{M2} & \cdots & a_{MN} \end{bmatrix} \begin{bmatrix} \phi(\hat{\mathbf{c}}_1)' \\ \phi(\hat{\mathbf{c}}_2)' \\ \vdots \\ \phi(\hat{\mathbf{c}}_N)' \end{bmatrix} \phi(\mathbf{x}) \\ &= \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{12} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & & \vdots \\ a_{M1} & a_{M2} & \cdots & a_{MN} \end{bmatrix} \begin{bmatrix} \kappa(\hat{\mathbf{c}}_1, \mathbf{x}) \\ \kappa(\hat{\mathbf{c}}_2, \mathbf{x}) \\ \vdots \\ \kappa(\hat{\mathbf{c}}_N, \mathbf{x}) \end{bmatrix}, \end{aligned} \quad (9)$$

where $\hat{\mathbf{c}}_n$ is the n th center and a_{mn} is the coefficient assigned to the n th kernel center for the m channel of the output.

5. Sensory Stimulation Experiment

5.1. Experimental Motivation and Framework. We applied these methods to the problem of converting touch information to electrical stimulation in neural prostheses. Somatosensory information originating in the peripheral nervous system ascends through the ventral posterior lateral (VPL) nucleus of the thalamus on its way to the primary

somatosensory cortex (S1). Since most cutaneous and proprioceptive information is relayed through this nucleus, we expect that a suitably designed electrode array could be used to selectively stimulate a local group of VPL neurons so as to convey similar information to cortex. Electrophysiological experiments [52] suggest that the rostral portion of the rat VPL nucleus carries a large amount of proprioceptive information, while the medial and caudal portions code mostly for cutaneous stimuli. Since the body maps for both VPL thalamus and S1 are known and fairly consistent, it is possible to implant electrode arrays in somatotopically overlapping areas of both regions.

We applied the proposed control method to generate multichannel electrical stimulation in VPL so as to evoke a naturalistic neural trajectory in S1. Figure 3 shows a schematic depiction of our experiment, which was conducted in rats. After implanting arrays in both VPL and S1, the responses to natural stimulation, delivered by a motorized tactor, were recorded in S1. Then, we applied randomly patterned microstimulation in the VPL while recording the responses in S1. Using these responses, we then trained our controller to output the microstimulation patterns that would most accurately reproduce the neural responses to natural touch in S1. To solve this control problem, we first investigated how reliably the two types of stimulation, natural touch and electrical microstimulation, can be decoded.

5.2. Data Collection. All animal procedures were approved by the SUNY Downstate Medical Center IACUC and conformed to National Institutes of Health guidelines. A single female Long-Evans rat (Hilltop, Scottsdale, PA) was implanted with two microarrays while under anesthesia.

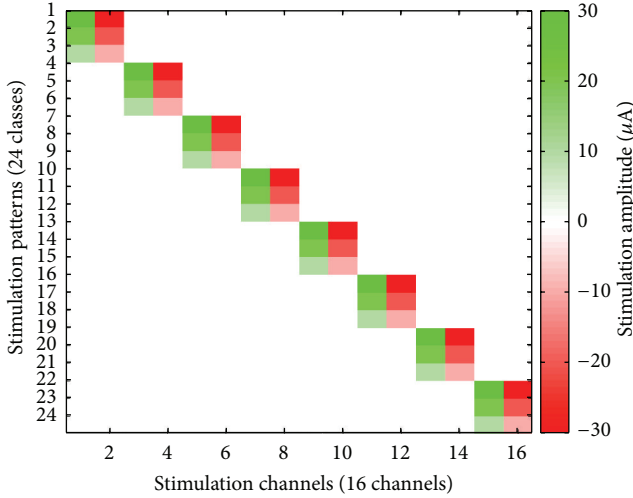


FIGURE 4: Bipolar microstimulation patterns applied in sensory stimulation experiment.

After induction using isoflurane, urethane was used to maintain anesthetic depth. The array in VPL was a 2×8 grid of 70% platinum 30% iridium $75 \mu\text{m}$ diameter microelectrodes (MicroProbes Inc.), with $500 \mu\text{m}$ between the rows and $250 \mu\text{m}$ interelectrode spacing within the rows. The microelectrodes had a 25:1 taper on the distal 5 mm with a tip diameter of $3 \mu\text{m}$. The approximate geometric surface area of the conducting tips was $1250 \mu\text{m}^2$. The shank lengths were custom designed to fit the contour of the rat VPL [52]. Both rows were identical and the shaft lengths for each row, from medial to lateral, were (8, 8, 8, 8, 8, 7.8, 7.6, 7.4) mm. The long axis of the VPL array was oriented along the rat's mediolateral axis.

The cortical electrode array (Blackrock Microsystems) was a 32-channel Utah array. The electrodes are arranged in a 6×6 grid excluding the 4 corners, and each electrode is 1.5 mm long. A single craniotomy that exposed the cortical insertions sites for both arrays was made, and, after several probing insertions with a single microelectrode (FHC) in an area 1 mm surrounding the stereotaxic coordinates for the digit region of S1 (4.0 mm lateral and 0.5 mm anterior to bregma) [53, 54], the Utah array was inserted using a pneumatic piston. The electrodes cover somatosensory areas of the S1 cortex and the VPL nucleus of the thalamus [52]. Neural recordings were made using a multichannel acquisition system (Tucker Davis).

Spike and field potential data were collected while the rat was maintained under anesthesia. The electrode voltages were preamplified with a gain of 1000, filtered with cutoffs at 0.7 Hz and 8.8 kHz, and digitized at 25 kHz. LFPs are further filtered from 1 to 300 Hz using a 3rd-order Butterworth filter. Spike sorting is achieved using k -means clustering of the first 2 principal components of the detected waveforms.

The experiment involves delivering microstimulation to VPL and tactile stimulation to the rat's fingers in separate sections. Microstimulation is administered on adjacent pairs

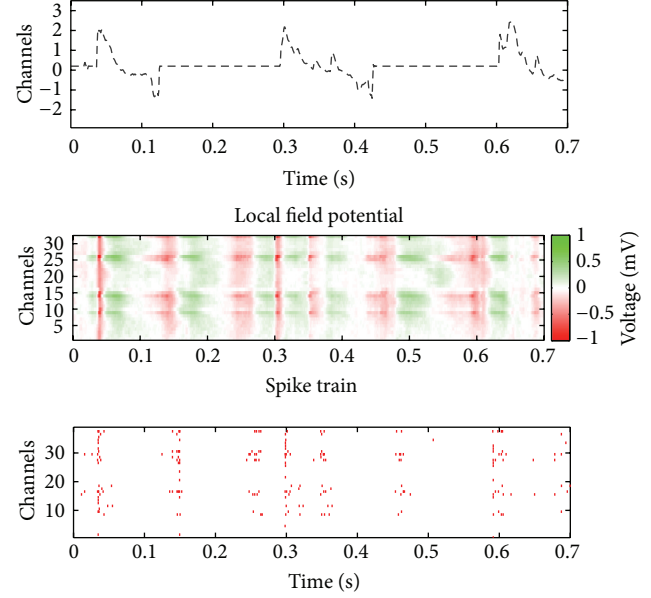


FIGURE 5: Rat neural response elicited by tactile stimulation. The upper plot shows the normalized derivative of tactile force. The remaining two plots show the corresponding LFPs and spike trains stimulated by tactile stimulation.

(bipolar configurations) of the thalamic array. The stimulation waveforms are single symmetric biphasic rectangular current pulses; each rectangular pulse is $200 \mu\text{s}$ long and has an amplitude of either $10 \mu\text{A}$, $20 \mu\text{A}$, or $30 \mu\text{A}$. Interstimulus intervals are exponentially distributed with mean interval of 100 ms. Stimulus isolation used a custom built switching headstage. The bipolar microstimulation pulses are delivered in the thalamus. There are 24 patterns of microstimulation: 8 different sites and 3 different amplitude levels for each site, as shown in Figure 4. Each pattern is delivered 125 times.

The experimental procedure also involves delivering 30–40 short 100 ms tactile touches to the rat's fingers (repeated for digit pads 1–4) using a hand-held probe. The rat remained anesthetized for the recording duration. The applied force is measured using a lever attached to the probe that pressed against a thin-film resistive force sensor (Trossen Robotics) when the probe tip contacted the rat's body. The resistive changes were converted to voltage using a bridge circuit and were filtered and digitized in the same way as described above. The digitized waveforms were filtered with a passband between 1 and 60 Hz using a 3rd-order Butterworth filter. The first derivative of this signal is used as the desired stimulation signal, which is shown in Figure 5.

5.3. Decoding Results. We now present the decoding results for the tactile stimulus waveform and microstimulation using Q-KLMS operating on the tensor-product kernel. The performance using the multiscale neural activity, both spike trains and LFPs, is compared with the decoder using single-type neural activity. This illustrates the effectiveness of the tensor-product-kernel-based framework to exploit the complementarity information from multiscale neural activities.

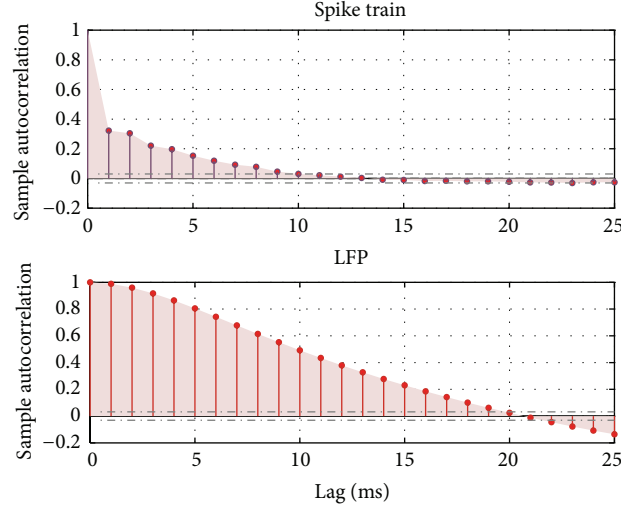


FIGURE 6: Autocorrelation of LFPs and spike trains for window size estimation.

5.3.1. Time Scale Estimation. The tensor-product kernel allows the time scales of the analysis for LFPs and spike trains to be specified individually, based on their own properties. In order to find reasonable time scales, we estimate the autocorrelation coefficients of LFPs and spike trains, which indicates the response duration induced by the stimulation. For this purpose, spike trains are binned with bin size of 1 ms. The LFPs are also resampled with sampling rate 1000 Hz. The autocorrelation coefficients of each signal average over channels are calculated by

$$\hat{\rho}_h = \frac{\sum_{t=h+1}^T (y_t - \bar{y})(y_{t-h} - \bar{y})}{\sum_{t=1}^T (y_t - \bar{y})^2}. \quad (10)$$

The 95% confidence bounds of the hypothesis that the autocorrelation coefficient is effectively zero are approximately estimated by $\pm 2SE\rho$, where

$$SE\rho = \sqrt{\frac{(1 + 2 \sum_{i=1}^{h-1} \rho_i^2)}{N}}. \quad (11)$$

The average confidence bounds for LFPs and spike trains are $[-0.032 \ 0.032]$ and $[-0.031 \ 0.031]$, respectively. The autocorrelation coefficients of LFPs fall into the confidence interval after 20 ms, while the autocorrelation coefficients of spike trains die out after 9 ms, as shown in Figure 6. Therefore, the decoder inputs are obtained by sliding the window with a size of $T_s = 9$ ms for spike trains and $T_x = 20$ ms for LFPs. In addition, the time discretization for the stimuli is 5 ms.

The learning rates for each decoder are determined by the best cross-validation results after scanning the parameters. The kernel sizes σ_s and σ_x are determined by the average distance in RKHS of each pair of training samples. The normalized mean square error (NMSE) between the estimated stimulus (\mathbf{y}) and the desired stimulus (\mathbf{d}) is utilized as an accuracy criterion.

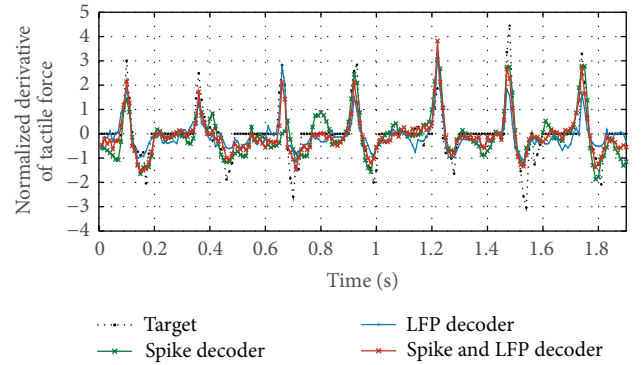


FIGURE 7: Qualitative comparison of decoding performance of the first tactile stimulation trial among LFP decoder, spike decoder, and spike and LFP decoder.

TABLE 1: Comparison among neural decoders.

Property	Input		
	LFP and spike	LFP	Spike
NMSE (mean/STD)	0.48/0.05	0.55/0.03	0.63/0.11

5.3.2. Results for Decoding Tactile Stimulation. NMSEs of tactile stimulation are obtained across 8 trial data sets. For each trial, we use 20 s data to train the decoders and compute an independent test error on the remaining 2.5 s data. The results are shown in Table 1, where we can observe that the LFP and spike decoder significantly outperformed both the LFP decoder and the spike decoder with P value < 0.05 .

In order to illustrate the details of the decoding performance, a portion of the test results of the first trial are shown in Figure 7. It is observed that the output of the spike decoder fluctuates and misses some pulses (e.g., around 0.65 s) due to the sparsity and variability of spike trains. In contrast, the

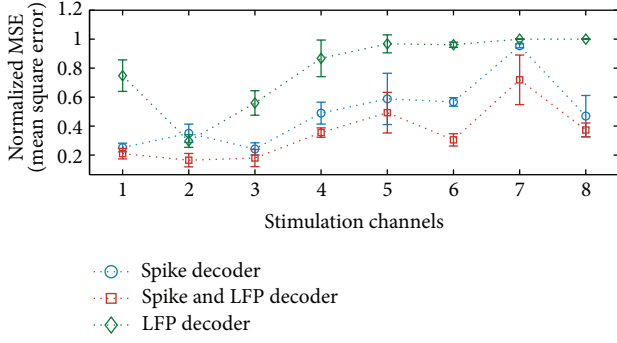


FIGURE 8: Performance comparison of the microstimulation reconstruction performance among spike decoder, LFP decoder, and spike and LFP decoder in terms of NMSE for each microstimulation channel.

output estimated by LFP decoder is smooth and more robust than the spike decoder, but the decoded signal undershoots the maximum force deflections. The LFP and spike decoder performed better than the LFP decoder by reincorporating the precise pulse timing information from spike trains.

5.3.3. Results for Decoding Electrical Microstimulation. We also implemented a decoder to reconstruct the microstimulation pattern. First, we mapped the 8 different stimulation configurations to 8 channels. We dismissed the shape of each stimulus, since the time scale of the stimulus width is only $200 \mu s$. The desired stimulation pattern of each channel is represented by a sparse time series of the stimulation amplitude.

NMSEs are obtained with ten subsequence decoding results. We used 120 s data to train the decoders and compute an independent test error on the remaining 20 s data. The spike and LFP decoder also outperformed both the LFP decoder and the spike decoder. The comparison of results is shown in Figure 8, which indicates that spike and LFP decoder is able to obtain the best performance amongst the stimulation channels, especially for channels 2, 4, 6, and 7. It is observed that stimulations on channels 4, 5, 6, 7, and 8 cannot be decoded from the LFP decoder at all, since the fine time information is averaged out in LFPs. For the spike trains decoder, the stimulation channels are not well discriminated. However, the combination of spike trains and LFPs enriched the stimulation information, which contributes to better discrimination of stimulation patterns among channels and also enables the model to capture the precise stimulation timing.

5.4. Open Loop Adaptive Inverse Control Results. The challenge of implementing a somatosensory prosthesis is to precisely control the neural response in order to mimic the neural response induced by natural stimulation. As discussed, the kernel-based adaptive inverse control diagram with tensor-product kernel is applied to address this problem. The adaptive inverse control model is based on a decoder which maps the neural activity in S1 to the microstimulation

delivered in VPL. We proceed to show how the adaptive inverse control model can emulate the neural response to “natural touch” using optimized microstimulation.

In the same recording, open loop adaptive inverse control via optimized thalamic (VPL) microstimulations is implemented. First, the inverse controller $C(z)$ is trained with 300 s of the data generated by recording the response to randomly patterned thalamic microstimulation. Then, the first 60 s of the neural response recorded during tactile stimulation at each touch site is used as the target pattern and control input. When this entire neural response sequence is fed offline to the controller, it generates a corresponding sequence of multichannel microstimulation amplitudes.

However, the generated microstimulation sequence needs further processing to meet the restrictions of bipolar microstimulation, before it applied to VPL. The restrictions and processing are the following.

- (1) The minimal interval between two stimuli 10 ms is suggested by the experimental setting. The mean shift algorithm [55] is used to locate the local maxima of a subsequence of stimuli (10 ms) for each single channel. The maximum amplitude and corresponding timing are used to set the amplitude and time of stimuli.
- (2) At any given time point, only a single pulse across all channels can be stimulated. Therefore, at each time point, only the maximum value across channels is selected for stimulation. The values at other channels are set to zero.
- (3) The maximum/minimum stimulation amplitude is set in the range $[8 \mu A - 30 \mu A]$, which has been suggested as the effective and safe amplitude range in previous experiments.

After this processing, the generated multichannel microstimulation sequence (60 s in duration) is ready to be applied to the microstimulator immediately following computation.

The neural response to the microstimulation is recorded and compared with the target natural response. Ideally, these two neural response sequences should be time-locked and be very similar. In particular, the portions of the controlled response in windows corresponding to a natural touch should match. As this corresponds to a *virtual touch* delivered by the optimized microstimulation, we define the term virtual touch to refer to the sequence of microstimulation patterns—the output of the controller—corresponding to a particular target natural touch.

Portions of the neural response for both natural and virtual touches are shown in Figure 9. The responses are aligned to the same time scale, even though they were not recorded concurrently. It is clear that the multiunit neural responses recorded during the controlled microstimulation share similar spatiotemporal patterns as those in the target set. Each virtual touch is achieved by a sequence of microstimulation pulses that evokes synchronized bursting across the neurons. In addition, microstimulation pulses are delivered in between touch times to mimic population spiking that is not associated with the touch timing.

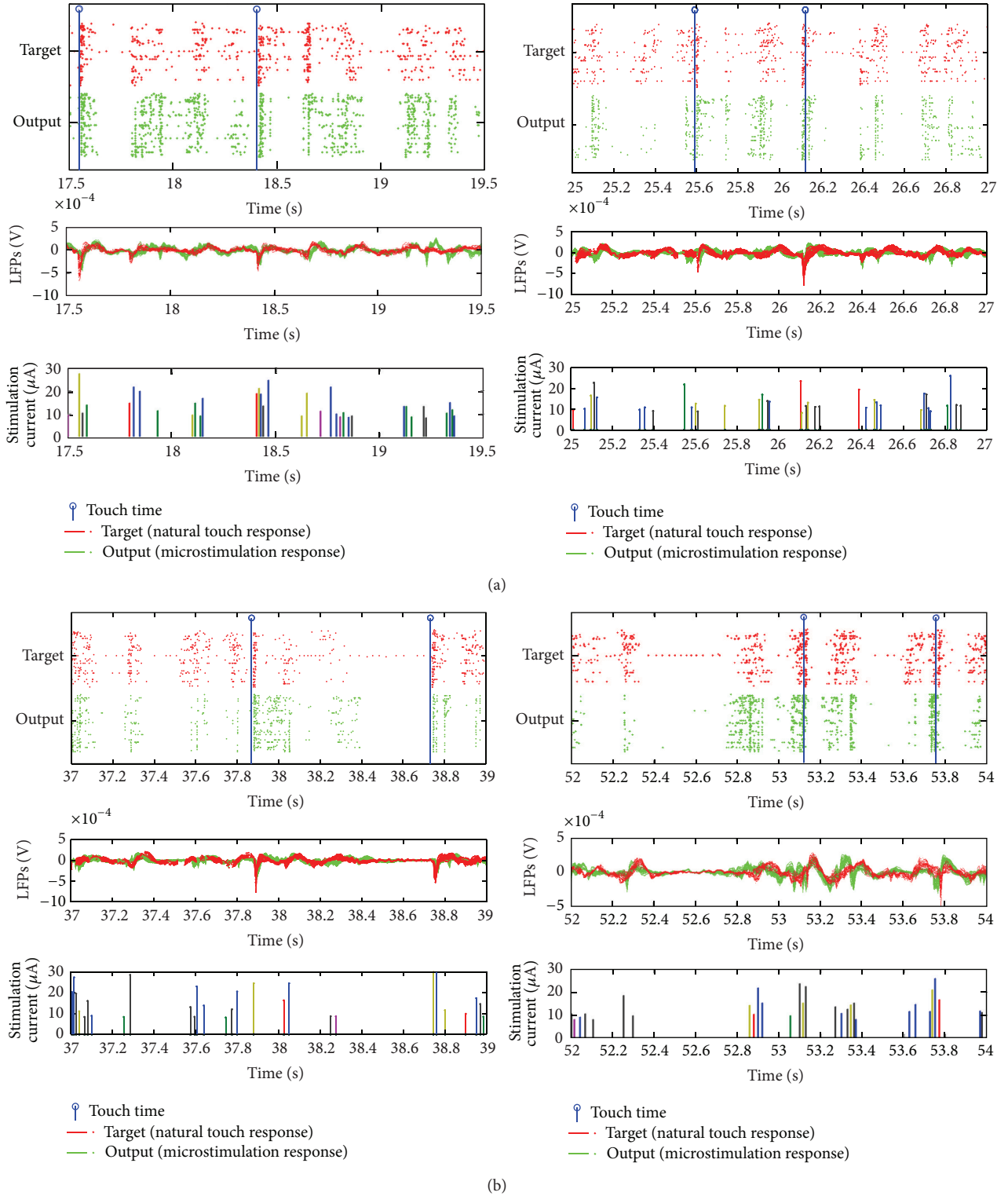


FIGURE 9: Neural responses to natural and virtual touches for touch on digit 1 (d1), along with the microstimulation corresponding to the virtual touches. Each of the four subfigures corresponds to a different segment of the continuous recording. In each subfigure, the timing of the touches, spatiotemporal pattern of spike trains and LFPs are shown in the top two panels; the bottom panel shows the spatiotemporal pattern of microstimulation, where different colors represent different microstimulation channels. The neural responses are time-locked, but not concurrently recorded, as the entire natural touch response is given as input to the controller which generates the optimized microstimulation patterns. When the optimized microstimulation is applied in the VPL, it generates S1 neural responses that qualitatively match the natural, that is, the target, response.

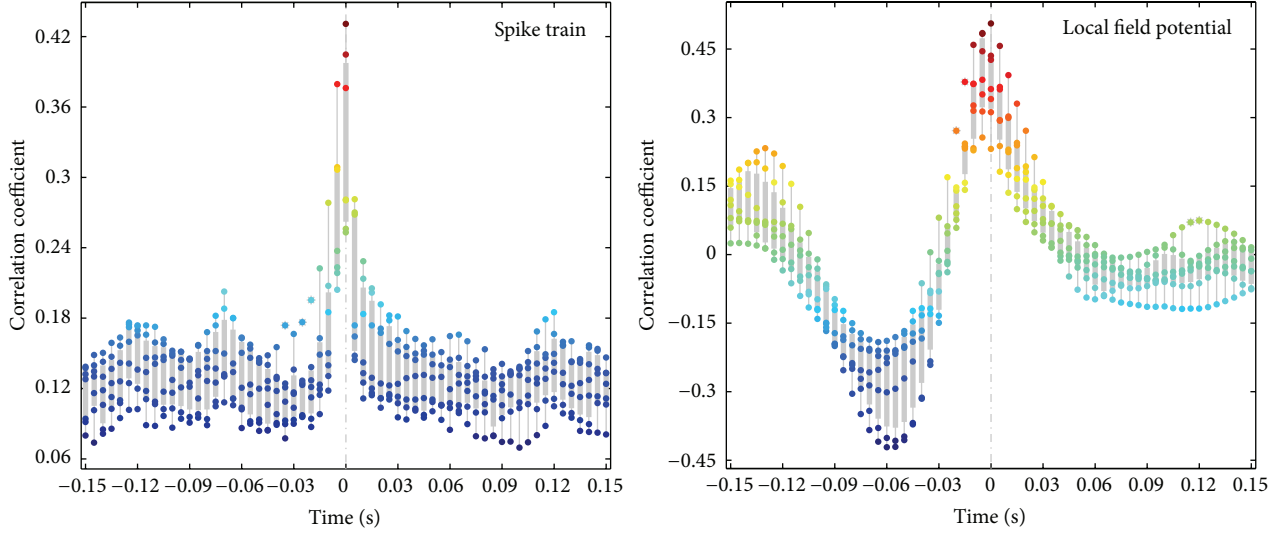


FIGURE 10: Correlation coefficients between the controlled neural system output and the corresponding target neural response stimulated by actual touch. Boxplot of correlation coefficients represents the results of 6 test trials. Each trial is corresponding to a particular touch site (digits: d1, d2, d4, p3, p1, and mp).

To evaluate performance, we concentrate on the following two aspects of virtual touches.

(i) *Touch Timing*. Whether the neural response to virtual touch is capable of capturing the timing information of the actual target touch is studied.

(ii) *Touch Site*. Whether the actual target touch site information can be discriminately represented by neural activity controlled by the microcirculation is studied.

For *touch timing*, we estimate the correlation coefficients (CC) over time between virtual touch responses and the corresponding target natural touch responses. To simplify the spike train correlation estimation, we bin the data using 5 ms bins. The correlation coefficients of both spike trains and LFPs are calculated. Figure 10 shows the boxplot plot of the correlation coefficients (CC) over 6 test trials, each of which corresponds to a particular natural touch site, a forepaw digit or pad (d1, d2, d4, p3, p1, or mp). It is observed that the maximum correlation coefficient is at lag zero for each trial, meaning that the virtual touch response is correctly time-locked. For each *touch site*, we estimate the similarity between the natural touch response and virtual touch response in the following two cases.

(i) *Matched Virtual*. Pairs consist of a virtual touch trial and a natural touch trial corresponding to the same touch site.

(ii) *Unmatched Virtual*. Pairs consist of a virtual touch trial and a natural touch trial corresponding to different touch sites.

We extract all the neural responses in the 300 ms window after touch onset and calculate the correlation coefficients between natural touch responses and virtual touch response across each pair of trials. The one-tailed Kolmogorov-Smirnov test (KS) is implemented to test the alternative

TABLE 2: Average and standard deviation of the correlation coefficient (CC) between natural touch spike train responses and virtual touch spike train responses (matched or unmatched). The P value is for the one-sided KS test between the matched and unmatched CC distributions.

Touch site	CC		P value
	Matched virtual	Unmatched virtual	
d1	0.42 ± 0.06	0.35 ± 0.06	0.00
d2	0.40 ± 0.05	0.37 ± 0.06	0.01
d4	0.40 ± 0.05	0.37 ± 0.05	0.02
p3	0.38 ± 0.05	0.37 ± 0.06	0.11
p2	0.40 ± 0.07	0.36 ± 0.05	0.00
mp	0.41 ± 0.07	0.37 ± 0.06	0.00

hypothesis that the distribution of the correlation coefficients for the *matched virtual* case is higher than the distribution for the *unmatched virtual* case (the null hypothesis is that the distributions are the same). The correlation coefficients and P value of KS test for spike trains and LFPs are shown in Tables 2 and 3. The similarity between natural touch responses and virtual touch responses in the *unmatched virtual* case is found to be significantly lower than the *matched virtual* case for most touch sites (P value < 0.05) except for touch site p3. Without psychophysical testing, it is unclear how effective the microstimulations are in producing true sensory sensations. Nonetheless, these are promising results to show the effectiveness of a controller utilizing the multiscale neural decoding methodology.

6. Conclusions

This work proposes a novel tensor-product-kernel-based machine learning framework, which provides a way to

TABLE 3: Average and standard deviation of the correlation coefficient (CC) between natural touch LFP responses and virtual touch LFP responses (matched or unmatched). The P value is for the one-sided KS test between the matched and unmatched CC distributions.

Touch site	CC		P value
	Matched virtual	Unmatched virtual	
d1	0.42 ± 0.20	0.28 ± 0.23	0.00
d2	0.46 ± 0.13	0.28 ± 0.22	0.00
d4	0.41 ± 0.19	0.26 ± 0.21	0.00
p3	0.38 ± 0.18	0.29 ± 0.22	0.07
p2	0.33 ± 0.19	0.26 ± 0.23	0.20
mp	0.34 ± 0.17	0.25 ± 0.21	0.00

decode stimulation information from the spatiotemporal patterns of multiscale neural activity (e.g., spike trains and LFPs). It has been hypothesized that spike trains and LFPs contain complementary information that can enhance neural data decoding. However, a systematic approach to combine, in a single signal processing framework, these two distinct neural responses has remained elusive. The combination of positive definite kernels, which can be defined in both the spike train space and the LFP space, seems to be a very productive approach to achieve our goals. We have basically used two types of combination kernels to achieve the multiscale combination: sum kernels to “average” across different spike channels, as well as across LFP channels, which combine evidence for the neural event in each modality, and product kernels across the spike and LFP modalities to emphasize events that are represented in both multiscale modalities. The results show that this approach enhances the accuracy and robustness of neural decoding and control. However, this paper should be interpreted as a first step of a long process to optimize the joint information contained in spike trains and LFPs. The first question is to understand why this combination of sum and product kernels works. Our analyses show that the sum kernel (particularly for the spike trains) brings stability to the neural events because it decreases the variability of the spike responses to stimuli. On the other hand, the product kernel requires that the neural event presents at both scales to be useful for decoding, which improves specificity. If we look carefully at Figure 6, we can understand the effect of decoding with the product kernel. Notice that the correlation times of spikes and LFPs are very different (LFPs have a much longer correlation time). Moreover, composite kernel definition can be naturally configured to different brain areas and even neuronal types with distinctive firing patterns. Each pattern will lead to different correlation profiles, which will immediately tune the properties of the kernels across brain areas and neural populations. If only LFPs are used, we can expect that the response time of the decoder will be very long and miss some events. The product kernel in fact limits the duration of the LFP kernel to that of the spike kernel and brings stability to the spike kernel. This explains exactly the decoding results. Therefore, results show that the proposed tensor-product-kernel framework can effectively integrate the information

from spikes and LFPs into the same model and enhance the neural decoding robustness and accuracy.

Furthermore, we applied the tensor-product-kernel framework in a more complex BMI scenario: how to emulate “natural touch” with microstimulation. Our preliminary results show that the kernel-based adaptive inverse control scheme employing tensor-product-kernel framework also achieves better optimization of the microstimulation than spikes and LFPs alone (results not shown). This result can be expected because the inverse controller is basically a decoder. However, we have to realize that not all the tasks of interest reduce to neural decoding, and we do not even know if neural control can be further improved by a different kernel design. This is where further research is necessary to optimize the joint kernels. For instance, we can weight both the channel information and the multiscale information to maximize the task performance using metric learning [40].

Overall, this tensor-product-kernel-based framework proposed in this work provides a general and practical framework to leverage heterogeneous neural activities in decoding and control scenario, which is not limited to spike trains and LFPs applications.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors would like to thank Ryan Burt for proofreading the paper. This work was supported in part by the US NSF Partnerships for Innovation Program 0650161 and DARPA Project N66001-10-C-2008.

References

- [1] J. K. Chapin, K. A. Moxon, R. S. Markowitz, and M. A. L. Nicolelis, “Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex,” *Nature Neuroscience*, vol. 2, no. 7, pp. 664–670, 1999.
- [2] M. Velliste, S. Perel, M. C. Spalding, A. S. Whitford, and A. B. Schwartz, “Cortical control of a prosthetic arm for self-feeding,” *Nature*, vol. 453, no. 7198, pp. 1098–1101, 2008.
- [3] D. M. Taylor, S. I. H. Tillery, and A. B. Schwartz, “Direct cortical control of 3D neuroprosthetic devices,” *Science*, vol. 296, no. 5574, pp. 1829–1832, 2002.
- [4] M. D. Serruya, N. G. Hatsopoulos, L. Paninski, M. R. Fellows, and J. P. Donoghue, “Instant neural control of a movement signal,” *Nature*, vol. 416, no. 6877, pp. 141–142, 2002.
- [5] J. M. Carmena, M. A. Lebedev, R. E. Crist et al., “Learning to control a brain-machine interface for reaching and grasping by primates,” *PLoS Biology*, vol. 1, no. 2, pp. 193–208, 2003.
- [6] J. DiGiovanna, B. Mahmoudi, J. Fortes, J. C. Principe, and J. C. Sanchez, “Coadaptive brain-machine interface via reinforcement learning,” *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 1, pp. 54–64, 2009.

- [7] A. Belitski, S. Panzeri, C. Magri, N. K. Logothetis, and C. Kayser, "Sensory information in local field potentials and spikes from visual and auditory cortices: time scales and frequency bands," *Journal of Computational Neuroscience*, vol. 29, no. 3, pp. 533–545, 2010.
- [8] J. R. Huxter, T. J. Senior, K. Allen, and J. Csicsvari, "Theta phase-specific codes for two-dimensional position, trajectory and heading in the hippocampus," *Nature Neuroscience*, vol. 11, no. 5, pp. 587–594, 2008.
- [9] M. J. Rasch, A. Gretton, Y. Murayama, W. Maass, and N. K. Logothetis, "Inferring spike trains from local field potentials," *Journal of Neurophysiology*, vol. 99, no. 3, pp. 1461–1476, 2008.
- [10] G. Buzsáki and A. Draguhn, "Neuronal oscillation in cortical networks," *Science*, vol. 304, no. 5679, pp. 1926–1929, 2004.
- [11] D. L. Snyder and M. I. Miller, *Random Point Processes in Time and Space*, Springer, 1991.
- [12] R. E. Kass, V. Ventura, and E. N. Brown, "Statistical issues in the analysis of neuronal data," *Journal of Neurophysiology*, vol. 94, no. 1, pp. 8–25, 2005.
- [13] R. D. Flint, E. W. Lindberg, L. R. Jordan, L. E. Miller, and M. W. Slutzky, "Accurate decoding of reaching movements from field potentials in the absence of spikes," *Journal of Neural Engineering*, vol. 9, no. 4, Article ID 046006, 2012.
- [14] R. C. Kelly, M. A. Smith, R. E. Kass, and T. S. Lee, "Local field potentials indicate network state and account for neuronal response variability," *Journal of Computational Neuroscience*, vol. 29, no. 3, pp. 567–579, 2010.
- [15] J. Liu and W. T. Newsome, "Local field potential in cortical area MT: stimulus tuning and behavioral correlations," *Journal of Neuroscience*, vol. 26, no. 30, pp. 7779–7790, 2006.
- [16] P. Berens, G. Keliris, A. Ecker, N. Logothetis, and A. Tolias, "Feature selectivity of the gamma-band of the local field potential in primate primary visual cortex," *Frontiers in Neuroscience*, vol. 2, Article ID 199207, 2008.
- [17] D. Xing, C.-I. Yeh, and R. M. Shapley, "Spatial spread of the local field potential and its laminar variation in visual cortex," *Journal of Neuroscience*, vol. 29, no. 37, pp. 11540–11549, 2009.
- [18] B. Schölkopf and A. J. Smola, *Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, Ser. Adaptive Computation and Machine Learning*, MIT Press, 2002.
- [19] L. Li, I. M. Park, A. Brockmeier et al., "Adaptive inverse control of neural spatiotemporal spike patterns with a reproducing kernel Hilbert space (RKHS) framework," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 21, no. 4, pp. 532–543, 2013.
- [20] S. Monaco, G. Króliczak, D. J. Quinlan et al., "Contribution of visual and proprioceptive information to the precision of reaching movements," *Experimental Brain Research*, vol. 202, no. 1, pp. 15–32, 2010.
- [21] R. S. Johansson and G. Westling, "Roles of glabrous skin receptors and sensorimotor memory in automatic control of precision grip when lifting rougher or more slippery objects," *Experimental Brain Research*, vol. 56, no. 3, pp. 550–564, 1984.
- [22] J. E. O'Doherty, M. A. Lebedev, P. J. Ifft et al., "Active tactile exploration using a brain-machine-brain interface," *Nature*, vol. 479, no. 7372, pp. 228–231, 2011.
- [23] J. E. O'Doherty, M. A. Lebedev, T. L. Hanson, N. A. Fitzsimmons, and M. A. Nicolelis, "A brain-machine interface instructed by direct intracortical microstimulation," *Frontiers in Integrative Neuroscience*, vol. 3, 2009.
- [24] N. A. Fitzsimmons, W. Drake, T. L. Hanson, M. A. Lebedev, and M. A. L. Nicolelis, "Primate reaching cued by multichannel spatiotemporal cortical microstimulation," *The Journal of Neuroscience*, vol. 27, no. 21, pp. 5593–5602, 2007.
- [25] X.-J. Feng, B. Greenwald, H. Rabitz, E. Shea-Brown, and R. Kosut, "Toward closed-loop optimization of deep brain stimulation for Parkinson's disease: concepts and lessons from a computational model," *Journal of Neural Engineering*, vol. 4, no. 2, pp. L14–L21, 2007.
- [26] J. Liu, H. K. Khalil, and K. G. Oweiss, "Model-based analysis and control of a network of basal ganglia spiking neurons in the normal and Parkinsonian states," *Journal of Neural Engineering*, vol. 8, no. 4, Article ID 045002, 2011.
- [27] A. J. Brockmeier, J. S. Choi, M. M. Distasio, J. T. Francis, and J. C. Principe, "Optimizing microstimulation using a reinforcement learning framework," in *Proceedings of the 33rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS '11)*, pp. 1069–1072, September 2011.
- [28] A. Brockmeier, J. Choi, M. Emigh, L. Li, J. Francis, and J. Principe, "Subspace matching thalamic microstimulation to tactile evoked potentials in rat somatosensory cortex," in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC '12)*, pp. 2957–2960, 2012.
- [29] Y. Ahmadian, A. M. Packer, R. Yuste, and L. Paninski, "Designing optimal stimuli to control neuronal spike timing," *Journal of Neurophysiology*, vol. 106, no. 2, pp. 1038–1053, 2011.
- [30] J. Moehlis, E. Shea-Brown, and H. Rabitz, "Optimal inputs for phase models of spiking neurons," *Journal of Computational and Nonlinear Dynamics*, vol. 1, no. 4, pp. 358–367, 2006.
- [31] D. Brugger, S. Butovas, M. Bogdan, and C. Schwarz, "Real-time adaptive microstimulation increases reliability of electrically evoked cortical potentials," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 5, pp. 1483–1491, 2011.
- [32] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, Cambridge University Press, 2000.
- [33] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, 2004.
- [34] W. Liu, J. C. Principe, and S. Haykin, *Kernel Adaptive Filtering*, Edited by S. Haykin, John Wiley & Sons, 2010.
- [35] B. Schölkopf and A. J. Smola, *Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press, 2001.
- [36] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semidefinite programming," *The Journal of Machine Learning Research*, vol. 5, pp. 27–72, 2004.
- [37] C. Cortes, M. Mohri, and A. Rostamizadeh, "Algorithms for learning kernels based on centered alignment," *Journal of Machine Learning Research*, vol. 13, pp. 795–828, 2012.
- [38] M. Yamada, W. Jitkrittum, L. Sigal, E. P. Xing, and M. Sugiyama, "High-dimensional feature selection by feature-wise kernelized lasso," *Neural Computation*, vol. 26, no. 1, pp. 185–207, 2013.
- [39] R. A. Horn, "On infinitely divisible matrices, kernels, and functions," *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, vol. 8, no. 3, pp. 219–230, 1967.
- [40] A. J. Brockmeier, J. S. Choi, E. G. Kriminger, J. T. Francis, and J. C. Principe, "Neural decoding with kernel-based metric learning," *Neural Computation*, vol. 26, no. 6, 2014.

- [41] L. Shpigelman, Y. Singer, R. Paz, and E. Vaadia, *Spikernels: Embedding Spiking Neurons in Inner Product Spaces*, vol. 15 of *Advances in Neural Information Processing Systems*, MIT Press, 2003.
- [42] A. R. C. Paiva, I. Park, and J. C. Príncipe, “A reproducing kernel Hilbert space framework for spike train signal processing,” *Neural Computation*, vol. 21, no. 2, pp. 424–449, 2009.
- [43] I. M. Park, S. Seth, M. Rao, and J. C. Principe, “Strictly positive definite spike train kernels for point process divergences,” *Neural Computation*, vol. 24, no. 8, pp. 2223–2250, 2012.
- [44] L. Li, *kernel based machine learning framework for neural decoding [Ph.D. dissertation]*, University of Florida, 2012.
- [45] H. Ramlau Hansen, “Smoothing counting process intensities by means of kernel functions,” *The Annals of Statistics*, vol. 11, no. 2, pp. 453–466, 1983.
- [46] P. Dayan and L. F. Abbott, *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*, MIT Press, 2001.
- [47] J. Platt, “A resource-allocating network for function interpolation,” *Neural Computation*, vol. 3, no. 2, pp. 213–225, 1991.
- [48] L. Csató and M. Opper, “Sparse on-line gaussian processes,” *Neural Computation*, vol. 14, no. 3, pp. 641–668, 2002.
- [49] Y. Engel, S. Mannor, and R. Meir, “The kernel recursive least-squares algorithm,” *IEEE Transactions on Signal Processing*, vol. 52, no. 8, pp. 2275–2285, 2004.
- [50] W. Liu, I. Park, and J. C. Príncipe, “An information theoretic approach of designing sparse kernel adaptive filters,” *IEEE Transactions on Neural Networks*, vol. 20, no. 12, pp. 1950–1961, 2009.
- [51] B. Widrow and E. Walach, *Adaptive Inverse Control*, Edited by E. Cliff, Prentice-Hall, 1995.
- [52] J. T. Francis, S. Xu, and J. K. Chapin, “Proprioceptive and cutaneous representations in the rat ventral posterolateral thalamus,” *Journal of Neurophysiology*, vol. 99, no. 5, pp. 2291–2304, 2008.
- [53] G. Paxinos and C. Watson, *The Rat Brain in Stereotaxic Coordinates*, Academic Press, 6th edition, 2006.
- [54] G. Foffani, J. K. Chapin, and K. A. Moxon, “Computational role of large receptive fields in the primary somatosensory cortex,” *Journal of Neurophysiology*, vol. 100, no. 1, pp. 268–280, 2008.
- [55] K. Fukunaga and L. D. Hostetler, “The estimation of the gradient of a density function, with application in pattern recognition,” *IEEE Transactions on Information Theory*, vol. 21, no. 1, pp. 32–40, 1975.

Research Article

Prediction of Human's Ability in Sound Localization Based on the Statistical Properties of Spike Trains along the Brainstem Auditory Pathway

Ram Krips and Miriam Furst

The School of Electrical Engineering, Faculty of Engineering, Tel Aviv University, 69978 Tel Aviv, Israel

Correspondence should be addressed to Miriam Furst; mira@eng.tau.ac.il

Received 15 September 2013; Revised 6 February 2014; Accepted 2 March 2014; Published 31 March 2014

Academic Editor: Asohan Amarasingham

Copyright © 2014 R. Krips and M. Furst. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The minimum audible angle test which is commonly used for evaluating human localization ability depends on interaural time delay, interaural level differences, and spectral information about the acoustic stimulus. These physical properties are estimated at different stages along the brainstem auditory pathway. The interaural time delay is ambiguous at certain frequencies, thus confusion arises as to the source of these frequencies. It is assumed that in a typical minimum audible angle experiment, the brain acts as an unbiased optimal estimator and thus the human performance can be obtained by deriving optimal lower bounds. Two types of lower bounds are tested: the Cramer-Rao and the Barankin. The Cramer-Rao bound only takes into account the approximation of the true direction of the stimulus; the Barankin bound considers other possible directions that arise from the ambiguous phase information. These lower bounds are derived at the output of the auditory nerve and of the superior olivary complex where binaural cues are estimated. An agreement between human experimental data was obtained only when the superior olivary complex was considered and the Barankin lower bound was used. This result suggests that sound localization is estimated by the auditory nuclei using ambiguous binaural information.

1. Introduction

Adrian's classic research on neural activity [1] presented three essential observations which are as relevant today as they were when he first introduced them: (1) as individual neurons produce action potential which propagate through the brain, the information of the neural activity is encoded by spiking events; (2) the rate of the spikes is dependent upon the external stimuli that drives the neural cell; and (3) there is an adaptation mechanism that adjusts the cell response; that is, the neural activity is reduced for constant stimuli. Any model that purports to characterize a neural activity must take into account these basic principles.

In this study we refer to auditory systems in which irregular neuronal activity was demonstrated during *in vivo* recordings [2]. *In vivo* observations have also shown that a specific neuron might respond with a single spike or several spikes to a given stimuli as shown in [2]. Kiang's

[2] observation is not in agreement with that of Adrian [1], who suggested that the stimuli information is coded by the average rate of the neural response. This contradiction raises the possibility that the timing of the spikes relative to the stimulus should be considered as well.

The origin of the stochastic activity of neurons is poorly understood. This activity results in both intrinsic noise sources that generate stochastic behavior on the level of the neuronal dynamics and extrinsic sources that arise from network effects and synaptic transmission [3]. Another source of noise that is specific to neurons arises from the finite number of ion channels in a neuronal membrane patch [4, 5].

There are a number of different ways that have emerged to describe the stochastic properties of neural activity. One possible approach relates to the train of spikes as a stochastic point process. For example, in their earlier studies, Alaoglu and Smith [6] and Rodieck et al. [7] suggested that the spontaneous activity of the cochlear nucleus can be described

as a homogeneous Poisson process. Further investigations of the auditory system described the neural response as a nonhomogeneous Poisson point process (NHPP) whose instantaneous rate depends on the input stimuli [8, 9].

A meaningful characterization of neural activity can be derived by using stochastic properties in order to predict human performance. Up to the 19th century, when medical science was still in its infancy and the concept of neural activity was unknown, the only method of understanding and researching the brain was through a black-box approach based on psychoacoustical experiments. While these psychoacoustical experiments provided valuable information, they were regarded as limited since they only produced qualitative information. It was argued that the activities and the contents of the mind could not be measured and therefore could not be objective. This view began to change in the early 1800s when Ernst Weber (1795–1878) demonstrated two measures for quantifying psychological data that he obtained from testing subjects psychoacoustically: (1) the two-point threshold, in which the smallest distance noticeable to touch at various parts of the body is measured, and (2) the just-noticeable difference (JND), in which the smallest difference in weight a person is capable of distinguishing is measured.

In the mid-20th century, several classes of standard adaptive tests for psychoacoustic measurements were introduced for evaluating auditory resolution [10–12]. These measurements are used for comparing the relationship between prediction of neural models and psychoacoustical performances. In such psychoacoustical tests, subjects are asked to distinguish between close values of one of the signal's parameters, such as the signal's frequency or level in monaural stimulation, and the interaural level difference (ILD), or the interaural time difference (ITD) in binaural stimulation. The results of such experiments are the JND of the investigated parameter. Such experiments have been repeatedly performed and reported in the literature (e.g., [13–20]).

Comparing the behavioural JND and the neural activity is possible if one assumes that the neural system estimates the measured parameters. Siebert [21, 22] obtained such a comparison when the JND of a single tone's frequency and level was compared to the neural activity of the auditory nerve. Siebert's findings were based on the assumption that the auditory nerve (AN) response behaves as a NHPP, and the brain acts as an unbiased optimal estimator of the physical parameters. Thus, the JND is equal to the standard deviation of the estimated parameter and can be derived by lower bounds such as the Cramer-Rao lower bound. Heinz et al. [23, 24] generalized Siebert's results to a larger range of frequencies and levels. Colburn and his colleagues [25–29] obtained similar evaluations for binaural signals, where the JND of ITD and ILD was compared to the neural activity of the auditory nerves of both ears.

This approach was extended to analyze brainstem nuclei such as the superior olivary complex (SOC) and the inferior colliculus (IC). These nuclei receive inputs from both ears, integrate the information, and send it by means of neural spike trains to the upper nuclei in the auditory pathway [30–36].

The neural cells in the SOC and IC are frequently described as coincidence detector (CD) cells. These cells receive independent excitatory and inhibitory inputs and generate a spike if the number of excitatory inputs exceeds the number of inhibitory inputs by a known number during a short interval. Krips and Furst [37] showed that the CD cells behave as NHPP if their inputs are NHPP. Therefore, the JND of the binaural parameters such as ITD and ILD, which are presumably estimated at the level of the SOC or IC, can be derived on the basis of the CD cell outputs [38].

Two main types of CD cells are identified in the brainstem auditory pathway: excitatory-excitatory (EE) cells and excitatory-inhibitory (EI) cells. EE cells receive excitatory inputs from both (right and left) anteroventral cochlear nuclei (AVCN) and they fire when both inputs are received within a time interval of less than 50 μ sec [36, 39–41]. These types of cells are sensitive mainly to ITD. EI cells, on the other hand, are sensitive to the balance of intensity at the ears because the excitation, due to ipsilateral stimuli, is reduced by increasing levels of contralateral stimuli [40, 42–45].

The human ability to localize sound depends on ITD, ILD, and spectral information of the acoustic stimulus. The goal of this paper is to test whether the prediction of human performance in this task is possible from the AN response or whether the processing of higher auditory brainstem nuclei is required. We compare the prediction of human performance based on the stochastic properties of the spike trains at the level of the auditory nerve and at the level of the SOC.

2. Minimum Audible Angle

The minimum audible angle (MAA) test is a common means of evaluating human localization ability. In this test, two successive signals from different directions are aimed at a listener. The order of the two signals is random. The listener is instructed to indicate the direction of the two signals relative to each other. For example, in the horizontal plane, the subject is asked if the signal moved from right to left, or vice versa.

MAA experiments have been conducted with various experimental setups and testing procedures for different stimuli conditions [11, 46–50]. For a single-tone MAA in the horizontal plane, Mills' measurements [11] have become the generally accepted standard. MAA as a function of frequency at an azimuth of 0° is redrawn from Mills' [11] measurements in Figure 1. The MAA exhibits the following properties: (1) an increase of MAA as a function of frequency above 1 kHz and (2) an oscillatory behavior as a function of frequency with local maxima at about 1.5 and 8 kHz.

In a typical MAA experiment, the audio signal $S(t, \theta)$ enters both ears from a direction θ relative to the nose. The incoming sounds to each ear are transformed as a function of the shape and size of the head, torso, and the pinna of the outer ears. These anatomical features are known as the head-related-transfer-function (HRTF) that can be measured and synthesized in the form of linear time-invariant filters.

In Figure 2, typical right and left HRTFs for an elevation of 0° are presented. They were obtained from Knowles Electronic Manikin for Acoustics Research (KEMAR) [51].

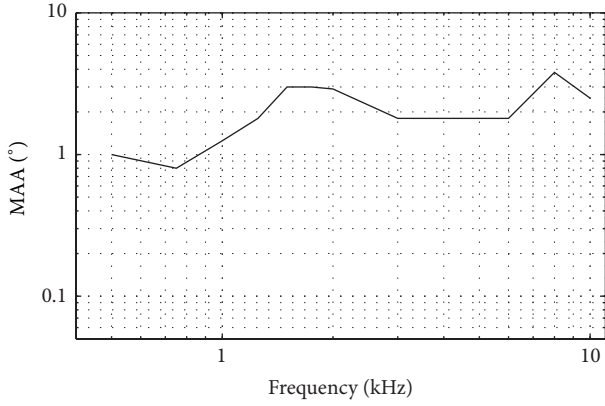


FIGURE 1: MAA experimental results as a function of frequency for a reference azimuth of 0° , (redrawn from [1]).

Both gain (Figure 2(a)) and phase (Figure 2(b)) are demonstrated in Figure 2 by color-coded scales. They are plotted as a function of both frequency (f) and direction (θ). The maximum gain for the right HRTF is obtained at high frequencies when the speaker is located in front of the right ear, that is, a direction of 90° . Similarly, maximum gain for the left HRTF is obtained when the speaker is located in front of the left ear, which corresponds to a direction of 270° . At frequencies above 1 kHz, the phase becomes ambiguous since different directions yield similar phases.

Formally, the signals that are conveyed to the left and right cochleae are

$$\begin{aligned} S_L(t, \theta) &= S(t, \theta) * \text{HRIR}_L(t, \theta), \\ S_R(t, \theta) &= S(t, \theta) * \text{HRIR}_R(t, \theta), \end{aligned} \quad (1)$$

where $*$ represents a convolution and $\text{HRIR}_L(t, \theta)$ and $\text{HRIR}_R(t, \theta)$ are the left and right head-related impulse responses, respectively.

In this study, we refer only to signals that are composed of simple tones, that is, $S(t) = A \sin(2\pi ft)$, where A is the signal amplitude and f represents its frequency. The effects of the HRTF on such a signal are phase shifts and amplitude alterations that yield

$$\begin{aligned} S_L(t, \theta) &= A_L(\theta) \sin(2\pi ft + \varphi_L(\theta)), \\ S_R(t, \theta) &= A_R(\theta) \sin(2\pi ft + \varphi_R(\theta)). \end{aligned} \quad (2)$$

Therefore, the resulting interaural differences are a phase difference (IPD) that is obtained by

$$\text{IPD}(\theta) = \varphi_R(\theta) - \varphi_L(\theta), \quad (3)$$

which corresponds to ITD by

$$\text{ITD}(\theta) = \frac{\text{IPD}(\theta)}{2\pi f}, \quad (4)$$

and interaural level difference (ILD) in dB is given by

$$\text{ILD}(\theta) = 20 \log_{10} \left(\frac{A_R(\theta)}{A_L(\theta)} \right) = 20 \log_{10}(\delta), \quad (5)$$

where $\delta = A_R(\theta)/A_L(\theta)$.

3. Estimating MAA on the Basis of the Stochastic Properties of Neural Spike Trains

We assume that during an MAA experiment, the brain's task is to estimate θ . The resultant unbiased estimator is $\hat{\theta}$, which yields

$$E[\hat{\theta} | \theta^*] = \theta^*, \quad (6)$$

where θ^* is the true direction of the incoming signal. Generally, in a psychoacoustical JND experiment, the yielded JND value is obtained when $d' = 1$, where in an MAA experiment

$$d' = \frac{E[\hat{\theta} | \theta^*] - E[\hat{\theta} | (\theta^* + \Delta\theta)]}{\text{std}(\hat{\theta} | \theta^*)} = \frac{\Delta\theta}{\text{std}(\hat{\theta} | \theta^*)}. \quad (7)$$

Therefore, $d' = 1$, yields the relations:

$$\Delta\theta = \text{MAA} = \text{std}(\hat{\theta} | \theta^*). \quad (8)$$

In an optimal system, the standard deviation of the estimator, $\text{std}(\hat{\theta} | \theta^*)$, can be obtained by the Cramer-Rao lower bound (CRLB). This bound is achievable when the estimator uses information from the vicinity of the true value, θ^* . However, in estimating the direction of sine waves when their phase information is ambiguous (Figure 2), the brain might consider different directions as the true ones. For example, when a continuous 2 kHz tone reaches both ears from either one of the sides or from the front of the head, the produced ITD in all cases will be 0. Thus, when the signal is coming from either of those directions, an optimal estimator can choose any of those possibilities. Since the Barankin lower bound (BLB) [52] takes into account different possible values of the estimated parameter other than those located in the proximity of the true one, the BLB might be a better choice in deriving a lower bound of $\text{std}(\hat{\theta} | \theta^*)$.

Let us define $\text{CRLB}(\theta^*)$ and $\text{BLB}(\theta^*)$ as the CRLB and the BLB of θ^* , respectively. In general,

$$\text{MAA} = \text{std}(\hat{\theta} | \theta^*) \geq \text{BLB}(\theta^*) \geq \text{CRLB}(\theta^*). \quad (9)$$

In order to derive both $\text{CRLB}(\theta^*)$ and $\text{BLB}(\theta^*)$, one should consider the probability density function of the estimator $\hat{\theta} | \theta^*$. The stochastic properties of the estimator $\hat{\theta} | \theta^*$ are initiated by the probabilistic behavior of the neural spike trains along the auditory pathway. Thus, the lower bounds can be derived from the probability density function of the neural spike trains.

The stochastic properties of the neural spike are described by the probability of getting N successive spikes during T seconds at the time instances $\{t_1, \dots, t_N\}$ following an acoustic stimulus. As was stated earlier [8, 9], this behavior can be described as NHPP; therefore,

$$p(t_1, \dots, t_N) = \frac{1}{N!} \prod_{n=1}^N \lambda(t_n, \Theta) \exp \left\{ - \int_0^T \lambda(t, \Theta) dt \right\}, \quad (10)$$

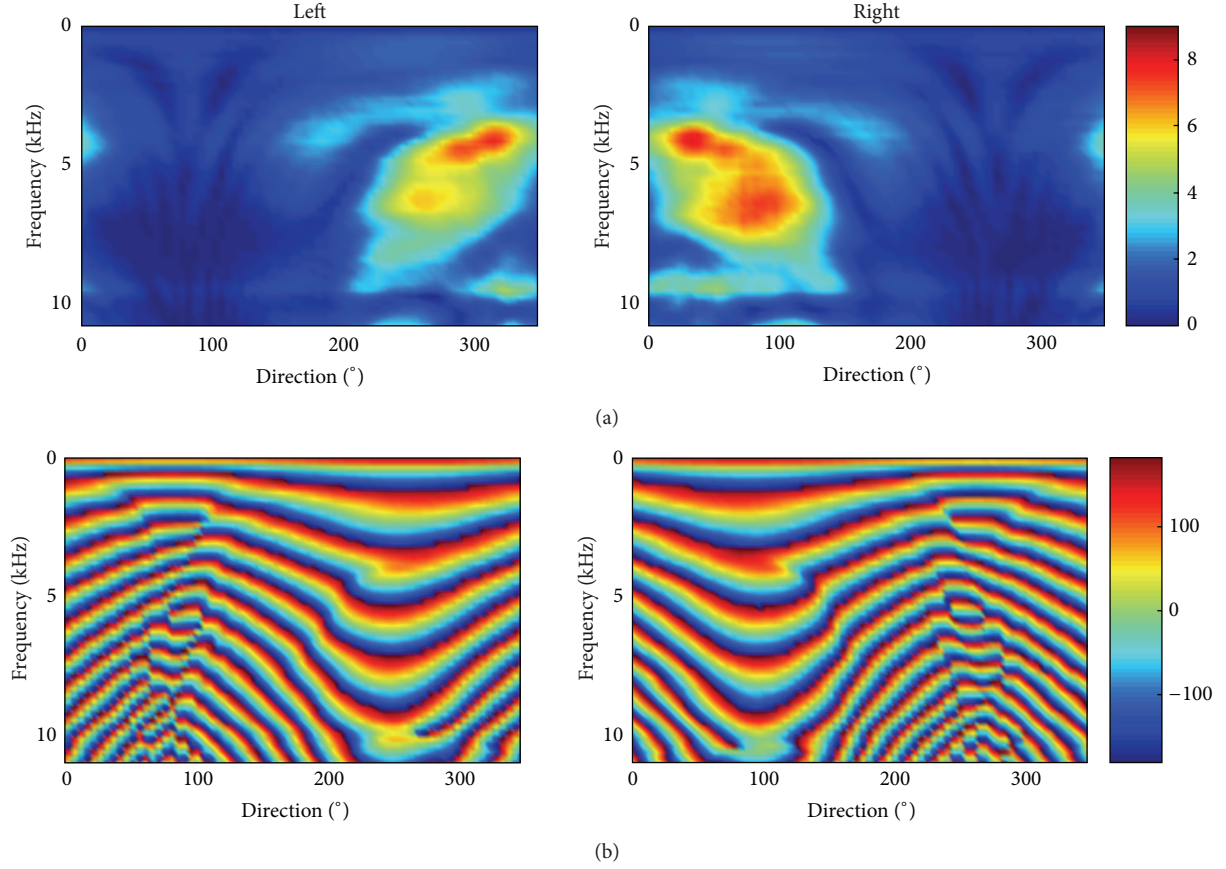


FIGURE 2: A sample HRTF gain in dB (a) and phase in degrees (b) as a function of azimuth and frequency for 0° elevation.

where $\lambda(t, \underline{\Theta})$ is the instantaneous rate of the neural point process and $\underline{\Theta}$ is a vector that includes all the physical parameters of the audio signal. In this study, since we relate to MAA, we choose $\underline{\Theta} = \theta$ as the direction of the incoming signal.

In NHPP, both lower bounds, CRLB and BLB, depend only on the instantaneous rate. The CRLB for a NHPP was derived by Bar David [53] and is given by

$$\text{CRLB}(\theta^*) = \left\{ \int_0^T \frac{1}{\lambda(t, \theta^*)} \left[\frac{\partial \lambda(t, \theta)}{\partial \theta} \bigg|_{\theta=\theta^*} \right]^2 dt \right\}^{-1/2}. \quad (11)$$

For deriving BLB, we define a vector of L that includes all the nontrue but possible values $\underline{\Phi} = [\theta_1, \dots, \theta_L]$. In [37] the BLB was derived for an NHPP which is given by

$$\text{BLB}(\theta^*) = \text{CRLB}(\theta^*) + (\underline{\Phi} - \text{CRLB}(\theta^*) \cdot \underline{A}) \Delta^{-1} \times (\underline{\Phi} - \text{CRLB}(\theta^*) \cdot \underline{A})^T, \quad (12)$$

where $\underline{A} = [A_1, \dots, A_L]$ is a vector of length L , when each A_l is given by

$$A_l = \int_0^T \left[\frac{\lambda(t, \theta_l)}{\lambda(t, \theta^*)} - 1 \right] \cdot \frac{\partial \lambda(t, \theta)}{\partial \theta} \bigg|_{\theta=\theta^*} dt. \quad (13)$$

The matrix $\Delta = B - \underline{A}^T \text{CRLB}(\theta^*) \underline{A}$, where B is a symmetric matrix whose size is $L \times L$. Each element in the matrix B is obtained by

$$B_{ij} = \exp \left(\int_0^T \left[-\lambda(t, \theta_i) - \lambda(t, \theta_j) + \lambda(t, \theta^*) + \frac{\lambda(t, \theta_i) \lambda(t, \theta_j)}{\lambda(t, \theta^*)} \right] dt \right). \quad (14)$$

The vector $\underline{\Phi}$ is essential in BLB derivation. If the size of the vector is predetermined, the actual values $\theta_1, \dots, \theta_L$ can be obtained by deriving BLB for all the possibilities. The L directions that yield the maximum BLB are then chosen for vector $\underline{\Phi}$. Such a straightforward approach is a time-consuming process that requires calculating enormous number of possible sets. For example, for $L = 4$ with a resolution of 1° , there are 360^4 sets to consider. In order to reduce the number of calculations, a two-stage procedure was designed. In the first stage, for every frequency, BLB predicted MAAs based on a single ambiguity. In a 1° resolution, a total of 360 BLB derivations were obtained. In the second stage, for every frequency, the number of ambiguous directions (L) was defined and the vector $[\theta_1, \dots, \theta_L]$ of the ambiguous directions was chosen according to directions that yielded maximum MAA in the first stage.

4. MAA Prediction Based on Auditory Nerve Response

Since the auditory nerve (AN) is the initial stage in the auditory neural pathway, we first tested the prediction of MAA on the basis of its response.

There are about 30,000 AN fibers that innervate each ear. The different location of each fiber's attachment on the cochlear partition determines its frequency sensitivity since each point along the cochlea has a different resonance frequency.

The auditory nerve's instantaneous rate (IR) for a simple tone stimulus $s(t) = A \cdot \sin(2\pi ft + \varphi)$ is commonly expressed with an exponential function [21, 24, 25, 29, 38] which is obtained by

$$\lambda_{AN}(t) = \gamma(f, A) \cdot \exp\{\gamma(f, A) \cdot B(f) \cdot \sin(2\pi ft + \varphi + \phi(f))\}. \quad (15)$$

Generally, $\gamma(f, A)$ is a nonlinear function of both the level and frequency of the stimulus. Its minimum value equals the fiber's spontaneous rate while its maximum value is equal to the fiber's saturation rate. For stimuli whose levels are in the mid-range ($20 \leq A \leq 50$ dB SPL), as used in this MAA experiment, $\gamma(f, A)$ is proportional to the stimulus level; that is, $\gamma(f, A) = A \cdot \gamma_0(f)$, where $\gamma_0(f)$ is different for every fiber as determined by the location along the cochlear partition that the fiber innervates.

The function $B(f)$ governs the synchronization of the fiber response which decreases with the increase of both frequency and the level of the simple tone stimuli. In this study we refer only to the dependence of the synchronization on frequency. The AN synchronization data [30, 54, 55] is commonly modelled by a sigmoid function of the form

$$B(f) = 1.5 \frac{e^{-\beta \cdot f}}{1 + e^{-\beta \cdot f}}, \quad (16)$$

where β is a constant that determines the loss of the fiber's synchrony as a function of frequency. We chose $\beta = 10^{-5}$ which corresponds to a loss of synchrony at around 3 kHz [38, 54–56].

Since in a MAA experiment both ears are involved, the derivation of MAA will take into account those fibers from the right and left cochleae that are most sensitive to the stimulus frequency. We ignore all other fibers whose IRs are significantly reduced in comparison to the most sensitive fiber. Since the AN fibers are statistically independent [2], therefore the d' theorem can be applied in order to obtain the MAA from N fibers:

$$(d')^2 = \sum_{n=1}^N (d'_n)^2, \quad (17)$$

where N is the number of independent nerve fibers and d'_n is the d' (see (7)) that was derived for the n th fiber. Since MAA is obtained when $d' = 1$, this implies that

$$\begin{aligned} \text{MAA} &= \text{std}(\hat{\theta} | \theta^*) \\ &= \frac{1}{\sqrt{\sum_{n=1}^{N_R(f)} \{\text{std}_R^n(\hat{\theta} | \theta^*)\}^{-2} + \sum_{n=1}^{N_L(f)} \{\text{std}_L^n(\hat{\theta} | \theta^*)\}^{-2}}}, \end{aligned} \quad (18)$$

where $\text{std}_R^n(\hat{\theta} | \theta^*)$ and $\text{std}_L^n(\hat{\theta} | \theta^*)$ are the standard deviations of the estimator as obtained by the right and left n th AN fibers, respectively, while $N_R(f)$ and $N_L(f)$ are the number of fibers of the right and left auditory nerve, respectively. When the optimal estimation is considered, the standard deviation is replaced by the correspondent CRLB (see (11)) or BLB (see (12)).

Figure 3 represents the prediction of MAA based on a BLB derivation with a single ambiguity ($L = 1$) as a function of both frequency and direction. The derivations were obtained by substituting (15) in (12). Equation (15) was derived for both right and left stimulations by using the correspondent HRIRs (see (2) that yields $\lambda_R(t, \theta)$ and $\lambda_L(t, \theta)$, the right and left auditory nerve instantaneous rates, respectively. In practice, only the fibers with a characteristic frequency equal to the stimulus frequency contribute to the MAA prediction. For the sake of simplicity, we chose $N_R(f) = N_L(f) = N_0$; $\gamma_0(f) = 1$ and $A = 1$. The number of fibers N_0 was chosen so that CRLB at 500 Hz yielded MAA of 1° .

Throughout the frequency range, high values of MAA were obtained at the rear of the head ($\theta = 180^\circ, \theta = -180^\circ$). This is most likely due to front-back confusion. At approximately 2 kHz and its harmonics (4 and 8 kHz), relatively high values of MAA were obtained at approximately $\theta = 90^\circ$ and $\theta = -90^\circ$. This most likely corresponds to the confusion between right and left. At directions that did not correspond to ambiguity, the values of the bound decreased with frequency.

Figure 4 represents the simulation results of MAA, derived by both lower bounds, CRLB and BLB, as a function of frequency when the reference direction was in front ($\theta^* = 0$). BLB was derived with at most 4 possible directions ($L = 4$). As can be expected, the estimated MAA according to BLB is greater than the CRLB estimates for all frequencies. At low frequencies, below 1 kHz, MAA according to BLB is about 10 times greater than the one yielded by CRLB. However, a more interesting difference between the two predictions is their dependence on frequency. CRLB derivation yielded a constant MAA of up to about 1 kHz and a monotonic decrease with increasing frequency for frequencies above 1 kHz. The BLB derivation yielded multiple peaks of MAA, in particular around 2, 4, 7, and 9 kHz.

The front-back confusion that exists throughout the whole frequency range is probably the reason for the difference in the MAA estimate according to the BLB and CRLB at low frequencies. Peaks at high frequencies (2, 4, 7, 9 kHz) can be attributed to the ambiguities that correspond to the similar phase obtained from tones coming from the sides or from the front of the head. According to the anthropometric data of

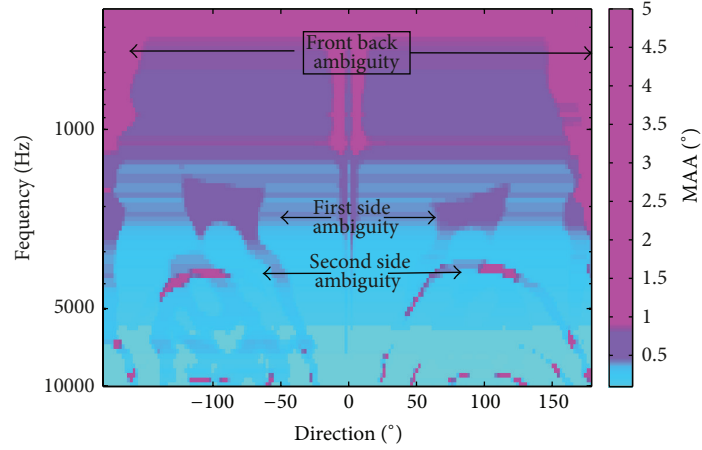


FIGURE 3: Prediction of MAA according to BLB derivation with a single ambiguity direction for all frequencies and directions based on the AN response (color online).

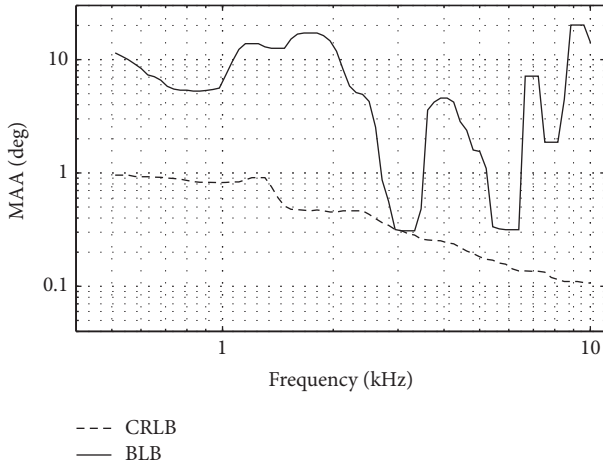


FIGURE 4: Normalized MAA according to CRLB and BLB as a function of frequency for a reference azimuth of 0° at AN.

the KEMAR dummy head [51], the head width is about 14 cm, which corresponds to a wavelength of tones with frequencies between 1.6 and 2.4 kHz [57]. These are the frequencies that yielded maximum MAA according to the BLB derivation.

While comparing the computational results of Figure 4 to the human performance shown in Figure 1, it seems that neither CRLB nor BLB is good predictors of human performance. By deriving MAA from CRLB, the dependence on frequency is totally different from data based on human performance. According to CRLB, MAA decreases monotonically as opposed to an oscillatory dependence in human experimental data. Although BLB reveals an oscillatory behaviour as a function of frequency, the predicted MAA has more oscillations as a function of frequency than human performance. In the next section we test whether this contradiction can be resolved by taking into account the binaural processing performed by CD cells in the brainstem nuclei such as SOC and IC.

5. MAA Prediction Based on the Superior Olivary Complex CD Cells

Figure 5 presents a schematic representation of part of the brainstem auditory pathway that is involved in binaural processing. The acoustic stimulus entering both ears innervates the auditory nerves. In Figure 5, the auditory nerves are represented by the left and right IRs, $\lambda_{AN}^{(L)}$ and $\lambda_{AN}^{(R)}$, respectively. The ANs stimulate both right and left SOC. In each SOC, the two types of CD cells, EE and EI, are indicated.

Both EE and EI cells receive two independent inputs, one from each ear as Figure 5 indicates. Following [38], the output of both EE and EI cells is NHPP if the time interval (Δ) in which the two inputs can interact satisfies the condition $\Delta \ll \min\{\tau_R, \tau_L\}$, where τ_R and τ_L are refractory periods of the right and left inputs.

The IR of the EE cells is obtained by

$$\lambda_{EE}(t, \theta) = \lambda_{AN}^{(L)}(t, \theta) \int_{t-\Delta_{EE}}^t \lambda_{AN}^{(R)}(t', \theta) dt' + \lambda_{AN}^{(R)}(t, \theta) \int_{t-\Delta_{EE}}^t \lambda_{AN}^{(L)}(t', \theta) dt'. \quad (19)$$

Since both right and left EE cells receive similar inputs, their output IRs are also identical; that is,

$$\lambda_{EE}^{(R)}(t, \theta) = \lambda_{EE}^{(L)}(t, \theta) = \lambda_{EE}(t, \theta). \quad (20)$$

A possible coincidence window length is $\Delta_{EE} = 20 \mu\text{sec}$ [58]. The value of this length, which was previously used in theoretical models [24, 25, 29], satisfies the condition $\Delta_{EE} \ll \min\{\tau_R, \tau_L\}$, since the refractory period at the auditory nerve is in the order of 500 μsec to 1 m sec [59–62].

EI cells receive excitatory and inhibitory inputs. An EI in the right SOC (Figure 5) receives an excitatory input from the left side and an inhibitory input from the right side that yields

$$\lambda_{EI}^{(R)}(t, \alpha) = \lambda_L(t, \alpha) \left(1 - \int_{t-\Delta_{EI}}^t \lambda_R(t', \alpha) dt' \right). \quad (21)$$

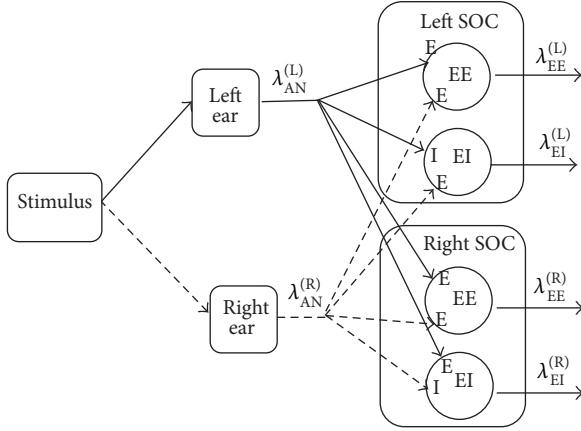


FIGURE 5: A schematic diagram of the binaural processing in the brainstem auditory pathway.

On the other hand, an EI cell in the left SOC receives the antisymmetric inputs, that is, an excitatory input from the right side and an inhibitory input from the left side (Figure 5) that yields

$$\lambda_{EI}^{(L)}(t, \alpha) = \lambda_R(t, \alpha) \left[1 - \int_{t-\Delta_{EI}}^t \lambda_L(t', \alpha) dt' \right]. \quad (22)$$

A possible coincidence window length is $\Delta_{EI} = 200 \mu\text{sec}$ [34]. This length is ten times longer than what was used in EE cells. However, it satisfies the condition $\Delta_{EI} < \min\{\tau_R, \tau_L\}$, which guarantees that EI cells behave as NHPP if their inputs also behave as NHPP [37].

In deriving MAA from the SOC CD cells, we assume that the outputs of the EE and EI cells are statistically independent. Therefore, MAA can be derived by using the d' theorem (see (17)) which implies that

$$\begin{aligned} \text{MAA} &= \text{std}(\hat{\theta} | \theta^*) \\ &= \left(\frac{N_{EE}^{(R)}(f)}{\{\text{std}_{EE}^{(R)}(\hat{\theta} | \theta^*)\}^2} + \frac{N_{EE}^{(L)}(f)}{\{\text{std}_{EE}^{(L)}(\hat{\theta} | \theta^*)\}^2} \right. \\ &\quad + \frac{N_{EI}^{(R)}(f)}{\{\text{std}_{EI}^{(R)}(\hat{\theta} | \theta^*)\}^2} \\ &\quad \left. + \frac{N_{EI}^{(L)}(f)}{\{\text{std}_{EI}^{(L)}(\hat{\theta} | \theta^*)\}^2} \right)^{-1/2}, \end{aligned} \quad (23)$$

where $\text{std}_{EE}^{(R)}(\hat{\theta} | \theta^*)$, $\text{std}_{EE}^{(L)}(\hat{\theta} | \theta^*)$, $\text{std}_{EI}^{(R)}(\hat{\theta} | \theta^*)$, and $\text{std}_{EI}^{(L)}(\hat{\theta} | \theta^*)$ are the standard deviations of the estimator that were obtained by the right and left EEs and the right and left EI cells, respectively. The values $N_{EE}^{(R)}(f)$, $N_{EE}^{(L)}(f)$, $N_{EI}^{(R)}(f)$, and $N_{EI}^{(L)}(f)$ are the number of the right and left EE and EI cells at the SOC. When the optimal estimation is considered, the standard deviations are replaced by the correspondent lower bounds, CRLB (see (11)) or BLB with $L = 4$ (see (12)).

The relevant instantaneous rates are obtained by substituting the EE IRs (see (19)) and the EI IRs (see (21) and (22)).

In order to demonstrate the difference between the MAA derivations as obtained by EE and EI cells, we calculated (23) with either a single EE cell ($N_{EE}^{(R)}(f) = N_{EE}^{(L)}(f) = 1$; $N_{EI}^{(R)}(f) = N_{EI}^{(L)}(f) = 0$) or a single EI cell ($N_{EE}^{(R)}(f) = N_{EE}^{(L)}(f) = 0$; $N_{EI}^{(R)}(f) = N_{EI}^{(L)}(f) = 1$). Figure 6 exhibits the resulting derivations as a function of frequency for a reference azimuth of 0° . According to CRLB, both EE and EI yielded a monotonic decrease as a function of frequency. But EE yielded a MAA with an order of magnitude greater than the one predicted from the EI response. On the other hand, the MAA that the EI yielded was similar to the one obtained by the AN response (Figure 4). One can then conclude that at the SOC level, EI processing caused minor information loss. However, due to EE processing some of the information that was included in the AN was lost.

When the MAA prediction was based on the BLB derivation (Figure 6(b)), both EE and EI yielded an oscillatory behavior as a function of frequency. When EE response was considered, the predicted MAA revealed local maxima at around 1.3 and 8 kHz, whereas the EI response yielded local maxima at 3.5 and 8 kHz.

It is clear from Figure 6 that both EE and EI are required in order to match the experimental results (Figure 1). Figure 7 represents the predicted MAA according to BLB with the following choice EE and EI cells:

$$\begin{aligned} N_{EE}^{(R)}(f) &= N_{EE}^{(L)}(f) = \begin{cases} 200 & f < 1250 \text{ Hz} \\ 25 & 1250 \leq f < 4000 \\ 0 & f \geq 4000 \end{cases} \\ N_{EI}^{(R)}(f) &= N_{EI}^{(L)}(f) = \begin{cases} 0 & f < 4000 \\ 3 & f \geq 4000. \end{cases} \end{aligned} \quad (24)$$

Note that (24) is consistent with physiological data indicating that EE cells mostly innervate signals with low frequencies, while EI cells are most sensitive to signals with high frequencies (e.g., [31, 32]). The predicted MAA has 2 peaks at about the same locations as the experimental data.

6. Discussion

The stochastic properties of neural spike trains in the auditory pathway were used in order to predict human performance in sound localization. We have shown that it is possible to predict human performance in an MAA experiment based on simple tones.

As in any JND experiment, predicting human performance was based on two main assumptions: (1) the brain is an unbiased optimal processor and (2) the neural spike trains behave as NHPP. The methodology involved deriving lower bounds based on the stochastic properties of neural spike trains [21, 22, 24, 25, 38].

When JND is predicted by deriving a lower bound, its significance is obtained by comparing it to experimental results as a function of a physical parameter. In this paper,

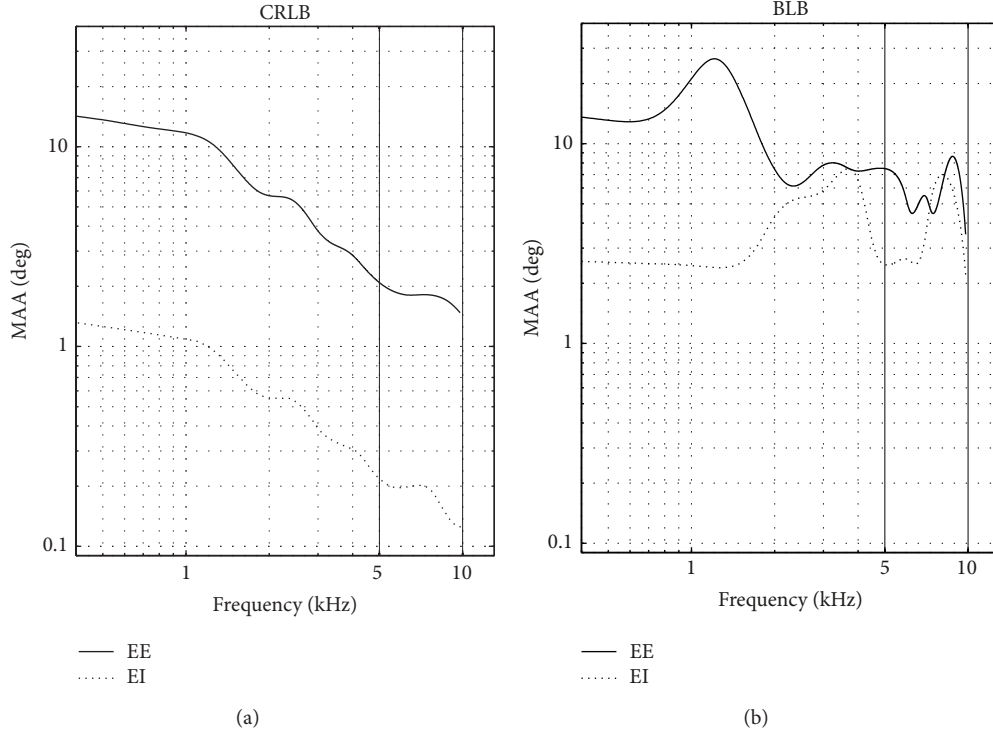


FIGURE 6: Predicted MAA according to CRLB (a) and BLB (b) as a function of frequency for a reference azimuth of 0° at the SOC level with either single EE cell or single EI cell.

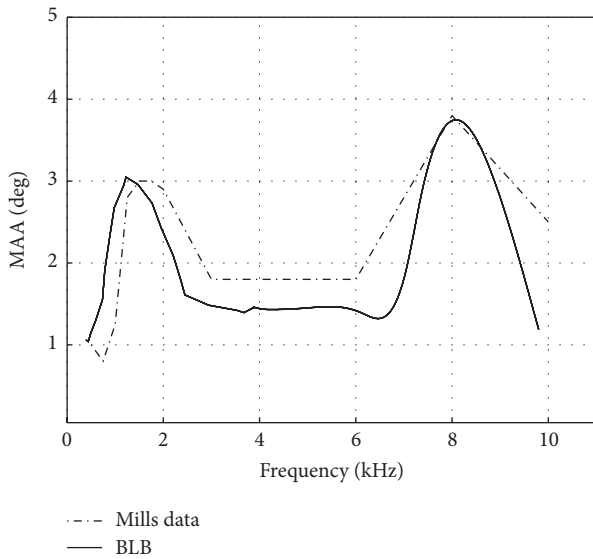


FIGURE 7: Predicted MAA according to BLB as a function of frequency for reference azimuth of 0° at the SOC level along with Mills' experimental data.

we compared the bound prediction of MAA as a function of the stimulus frequency.

In an MAA experiment with simple tones, the information about the origin of the stimulus might be ambiguous at high frequencies. We have shown that the ambiguous

interpretation of the HRTF phase data is probably the reason for the oscillatory behaviour of MAA as a function of frequency in human performance. This was demonstrated by the usage of two lower bounds, CRLB and BLB. In general, one can expect that the predictions of BLB will be greater than those obtained by CRLB. But the derivation demonstrated in this study reveals a totally different dependency on frequency. CRLB that took into account only the approximate true origin of the stimulus failed to predict oscillatory behavior. On the other hand, BLB, which considered ambiguity, succeeded in predicting the oscillatory behavior.

We further compared the predictions that were based on the AN outputs with those obtained by the SOC outputs. Although, both BLB predictions yielded an oscillatory behavior, it seems that the SOC output obtained a better prediction in respect to psychoacoustical data. When the AN output was considered, MAA local maxima were derived at frequencies 1.5, 2, 4, 7, and 9 kHz (Figure 4). When SOC was considered, some of the local maxima disappeared. It seems that loss of information due the SOC processing reduced the effect of the phase ambiguity.

The SOC outputs were derived by CD cells that processed the binaural information. The main task of CD cells is probably to extract binaural cues, with EE cells most likely extracting ITD and the EI cells extracting ILD [31, 32, 38]. Both ITD and ILD contribute to estimating the signal direction. In fact, our calculation of MAA, as derived by BLB, has shown that both EE and EI cells are required for predicting the experimental results.

We expect that the brain as an optimal system seeks to achieve a monotonic descending dependency of MAA as a function of frequency as predicted by CRLB. However, the physical constraints (i.e., the ambiguous phases) prevent the brain from achieving this goal.

When MAA was derived from the AN response or a single EI response, the predicted MAA according to BLB derivation was ten times greater than that predicted by CRLB at low frequencies (below 1.5 kHz). This difference was explained by the front-back confusion. However, this effect was almost eliminated when the EE response was considered. At low frequencies, the nerve response is synchronized to the stimulus; that is, $B(f) \rightarrow 1$ in (16). Although the synchronization exists in the AN response, it was not sufficient for overcoming the front-back confusion. However, it played an important role at the SOC level where ITD was efficiently extracted [38]. Therefore, at the SOC level, at low frequencies, both lower bounds, CRLB and BLB, yielded almost the same MAA.

The methodology used in this study was to express the lower bounds by analytical expressions as derived in [21, 24, 25, 38]. This was possible, since we assumed that the neural spike trains behave as NHPP at AN. However, in using this assumption, the discharge history including the refractory period, which is a basic characteristic of every neuron [63–68], was ignored. Other models of neural spike trains that take into account the refractory period might be a better choice for describing the neural spike trains [66, 69–73]. Yet, the usage of the NHPP model and the outcome lower bounds approach has been successful in predicting several additional properties of the auditory system. Consider, for example, (1) the prediction of level and frequency discrimination as a function of the stimulus duration, level, and frequency [21, 24], and (2) ITD and ILD as a function of frequency [38]. In these studies it seems that the NHPP approach is adequate for describing steady-state responses of continuous stimuli [74–76].

This study aimed to show the potential in using lower bounds in order to predict human performance in psychoacoustical experiments and particularly the importance of considering ambiguous information. However, we do not claim that the simple model presented in this paper is the exact biological model. Further research is required in order to quantitatively predict human performances. Such studies should include a generalized cochlear model, a synapse model, and models of the brainstem nuclei.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This study was partially supported by the Israel Science Foundation Grant no. 563/12.

References

- [1] E. D. Adrian, "The impulses produced by sensory nerve endings," *The Journal of Physiology*, vol. 61, no. 2, pp. 151–171, 1926.
- [2] N. Y. S. Kiang, T. Watanabe, E. C. Thomas, and L. F. Clark, *Discharge Patterns of Single Fibers in the Cat's Auditory Nerve*, MIT Press, Cambridge, Mass, USA, 1965.
- [3] A. Manwani and C. Koch, "Detecting and estimating signals in noisy cable structures, I: neuronal noise sources," *Neural Computation*, vol. 11, no. 8, pp. 1797–1829, 1999.
- [4] J. A. White, J. T. Rubinstein, and A. R. Kay, "Channel noise in neurons," *Trends in Neurosciences*, vol. 23, no. 3, pp. 131–137, 2000.
- [5] E. Schneidman, B. Freedman, and I. Segev, "Channel stochasticity may be critical in determining the reliability and precision of spike timing," *Neural Computation*, vol. 10, no. 7, pp. 1679–1703, 1998.
- [6] L. Alaoglu and N. M. Smith Jr., "Statistical theory of a scaling circuit," *Physical Review*, vol. 53, no. 10, pp. 832–836, 1938.
- [7] R. W. Rodieck, N. Y.-S. Kiang, and G. L. Gerstein, "Some quantitative methods for the study of spontaneous activity of single neurons," *Biophysical Journal*, vol. 2, pp. 351–368, 1962.
- [8] P. R. Gray, "Conditional probability analyses of the spike activity of single neurons," *Biophysical Journal*, vol. 7, no. 6, pp. 759–777, 1967.
- [9] F. Rieke, D. Warland, R. D. R. van Steveninck, and W. Bialek, *Spikes Exploring the Neural Code*, MIT Press, Cambridge, Mass, USA, 1997.
- [10] A. Wald, "Foundations of a general theory of sequential decision functions," *Econometrica*, vol. 15, no. 4, pp. 279–313, 1947.
- [11] A. W. Mills, "On the minimum audible angle," *The Journal of the Acoustical Society of America*, vol. 30, no. 2, pp. 237–246, 1958.
- [12] H. Levitt, "Transformed up-down methods in psychoacoustics," *The Journal of the Acoustical Society of America*, vol. 49, no. 2, pp. 467–477, 1971.
- [13] R. G. Klumpp and H. R. Eady, "Some measurements of interaural time difference thresholds," *The Journal of the Acoustical Society of America*, vol. 28, no. 5, pp. 859–860, 1956.
- [14] J. Zwislöcki and R. S. Feldman, "Just noticeable differences in dichotic phase," *The Journal of the Acoustical Society of America*, vol. 28, no. 5, pp. 860–864, 1956.
- [15] A. W. Mills, "Lateralization of high-frequency tones," *The Journal of the Acoustical Society of America*, vol. 32, no. 1, pp. 132–134, 1960.
- [16] R. M. Hershkowitz and N. I. Durlach, "Interaural time and amplitude jnds for a 500-Hz tone," *The Journal of the Acoustical Society of America*, vol. 46, no. 6, pp. 1464–1467, 1969.
- [17] D. H. Klatt, "Discrimination of fundamental frequency contours in synthetic speech: implications for models of pitch perception," *The Journal of the Acoustical Society of America*, vol. 53, no. 1, pp. 8–16, 1973.
- [18] L. Demany and C. Semal, "Detection thresholds for sinusoidal frequency modulation," *The Journal of the Acoustical Society of America*, vol. 85, no. 3, pp. 1295–1301, 1989.
- [19] J. H. Johnson, C. W. Turner, J. J. Zwislöcki, and R. H. Margolis, "Just noticeable differences for intensity and their relation to loudness," *The Journal of the Acoustical Society of America*, vol. 93, no. 2, pp. 983–991, 1993.
- [20] J. Lyzenga and J. W. Horst, "Frequency discrimination of bandlimited harmonic complexes related to vowel formants," *The Journal of the Acoustical Society of America*, vol. 98, no. 4, pp. 1943–1955, 1995.

- [21] W. M. Siebert, "Stimulus transformation in the peripheral auditory system," in *Recognizing Patterns*, P. A. Kolars and M. Eden, Eds., pp. 104–133, MIT Press, Cambridge, Mass, USA, 1968.
- [22] W. M. Siebert, "Frequency discrimination in the auditory system: place or periodicity mechanisms?" vol. 58, no. 5, pp. 723–730, 1970.
- [23] M. G. Heinz, *Quantifying the effects of the cochlear amplifier on temporal and average-rate information in the auditory nerve [Ph.D. dissertation]*, Massachusetts Institute of Technology, Cambridge, Mass, USA, 2000.
- [24] M. G. Heinz, H. S. Colburn, and L. H. Carney, "Evaluating auditory performance limits: I. One-parameter discrimination using a computational model for the auditory nerve," *Neural Computation*, vol. 13, no. 10, pp. 2273–2316, 2001.
- [25] H. S. Colburn, "Theory of binaural interaction based on auditory nerve data. I. General strategy and preliminary results on interaural discrimination," *The Journal of the Acoustical Society of America*, vol. 54, no. 6, pp. 1458–1470, 1973.
- [26] R. M. Stern Jr. and H. S. Colburn, "Theory of binaural interaction based on auditory-nerve data. IV. A model for subjective lateral position," *The Journal of the Acoustical Society of America*, vol. 64, no. 1, pp. 127–140, 1978.
- [27] L. G. Huettel and L. M. Collins, "A theoretical comparison of information transmission in the peripheral auditory system: normal and impaired frequency discrimination," *Speech Communication*, vol. 39, no. 1–2, pp. 5–21, 2003.
- [28] L. G. Huettel and L. M. Collins, "Predicting auditory tone-in-noise detection performance: the effects of neural variability," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 2, pp. 282–293, 2004.
- [29] O. Cohen, M. Furst, and R. Krips, "ITD and ILD estimation based on neural stochastic analysis," in *Proceedings of the 23rd IEEE Convention of Electrical and Electronics Engineers in Israel*, pp. 185–188, September 2004.
- [30] J. E. Rose, J. F. Brugge, D. J. Anderson, and J. E. Hind, "Phase-locked response to low-frequency tones in single auditory nerve fibers of the squirrel monkey," *Journal of Neurophysiology*, vol. 30, no. 4, pp. 769–793, 1967.
- [31] J. M. Goldberg and P. B. Brown, "Response of binaural neurons of dog superior olivary complex to dichotic tonal stimuli: some physiological mechanisms of sound localization," *Journal of Neurophysiology*, vol. 32, no. 4, pp. 613–636, 1969.
- [32] T. C. Yin and J. C. Chan, "Interaural time sensitivity in medial superior olive of cat," *Journal of Neurophysiology*, vol. 64, no. 2, pp. 465–488, 1990.
- [33] D. McAlpine, D. Jiang, T. M. Shackleton, and A. R. Palmer, "Convergent input from brainstem coincidence detectors onto delay-sensitive neurons in the inferior colliculus," *The Journal of Neuroscience*, vol. 18, no. 15, pp. 6026–6039, 1998.
- [34] T. J. Park, "IID sensitivity differs between two principal centers in the interaural intensity difference pathway: the LSO and the IC," *Journal of Neurophysiology*, vol. 79, no. 5, pp. 2416–2431, 1998.
- [35] A. J. Smith, S. Owens, and I. D. Forsythe, "Characterisation of inhibitory and excitatory postsynaptic currents of the rat medial superior olive," *The Journal of Physiology*, vol. 529, no. 3, pp. 681–698, 2000.
- [36] A. R. Palmer, T. M. Shackleton, and D. McAlpine, "Neural mechanisms of binaural hearing," *Acoustical Science and Technology*, vol. 23, no. 2, pp. 61–68, 2002.
- [37] R. Krips and M. Furst, "Stochastic properties of coincidence-detector neural cells," *Neural Computation*, vol. 21, no. 9, pp. 2524–2553, 2009.
- [38] R. Krips and M. Furst, "Stochastic properties of auditory brainstem coincidence detectors in binaural perception," *The Journal of the Acoustical Society of America*, vol. 125, no. 3, pp. 1567–1583, 2009.
- [39] W. B. Warr, "Fiber degeneration following lesions in the anteroventral cochlear nucleus of the cat," *Experimental Neurology*, vol. 23, pp. 140–155, 1966.
- [40] J. J. Guinan, S. S. Guinan, and B. E. Norris, "Single auditory units in the superior olivary complex. I: responses to sounds and classifications based on physiological properties," *International Journal of Neuroscience*, vol. 4, no. 3, pp. 101–120, 1972.
- [41] P. X. Joris, L. H. Carney, P. H. Smith, and T. C. T. Yin, "Enhancement of neural synchronization in the anteroventral cochlear nucleus. I. Responses to tones at the characteristic frequency," *Journal of Neurophysiology*, vol. 71, no. 3, pp. 1022–1036, 1994.
- [42] J. C. Boudreau and C. Tsuchitani, "Cat superior olive S-segment cell discharge to tonal stimulation," *Contributions to Sensory Physiology*, vol. 4, pp. 143–213, 1970.
- [43] D. Caird and R. Klinke, "Processing of binaural stimuli by cat superior olivary complex neurons," *Experimental Brain Research*, vol. 52, no. 3, pp. 385–399, 1983.
- [44] D. M. Caspary and P. G. Finlayson, "Superior olivary complex: functional neuropharmacology of the principal cell types," in *Neurobiology of Hearing: The Central Auditory System*, R. A. Altschuler, R. P. Bobbin, B. M. Clopton, and D. W. Hoffman, Eds., pp. 141–161, Raven Press, New York, NY, USA, 1991.
- [45] D. J. Tollin and T. C. T. Yin, "The coding of spatial location by single units in the lateral superior olive of the cat. I. Spatial receptive fields in azimuth," *The Journal of Neuroscience*, vol. 22, no. 4, pp. 1454–1467, 2002.
- [46] J. D. Harris, "A florilegium of experiments on directional hearing," *Acta Oto-Laryngologica*, vol. 298, pp. 1–26, 1972.
- [47] D. W. Grantham, "Detection and discrimination of simulated motion of auditory targets in the horizontal plane," *The Journal of the Acoustical Society of America*, vol. 79, no. 6, pp. 1939–1949, 1986.
- [48] E. R. Hafer, T. N. Buell, D. A. Basiji, and E. E. Shriberg, "Discrimination of direction for complex sounds presented in the free-field," in *Basic Issues in Hearing: Proceedings of the 8th International Symposium on Hearing*, H. Duifhuis, J. W. Horst, and H. P. Wit, Eds., pp. 394–401, Academic Press, London, UK, 1988.
- [49] R. Y. Litovsky and N. A. Macmillan, "Sound localization precision under conditions of the precedence effect: effects of azimuth and standard stimuli," *The Journal of the Acoustical Society of America*, vol. 96, no. 2, pp. 752–758, 1994.
- [50] D. W. Grantham, B. W. Y. Hornsby, and E. A. Erpenbeck, "Auditory spatial resolution in horizontal, vertical, and diagonal planes," *The Journal of the Acoustical Society of America*, vol. 114, no. 2, pp. 1009–1022, 2003.
- [51] V. R. Algazi, R. O. Duda, D. M. Thompson, and C. Avendano, "The CIPIC HRTF database," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 99–102, New Platz, NY, USA, October 2001.
- [52] E. W. Barankin, "Locally best unbiased estimates," *The Annals of Mathematical Statistics*, vol. 20, no. 4, pp. 477–628, 1949.
- [53] I. Bar David, "Communication under the Poisson regime," *IEEE Transactions on Information Theory*, vol. 15, no. 1, pp. 31–37, 1969.

- [54] A. R. Palmer and I. J. Russell, "Phase-locking in the cochlear nerve of the guinea-pig and its relation to the receptor potential of inner hair-cells," *Hearing Research*, vol. 24, no. 1, pp. 1–15, 1986.
- [55] D. H. Johnson, "The relationship between spike rate and synchrony in responses of auditory-nerve fibers to single tones," *The Journal of the Acoustical Society of America*, vol. 68, no. 4, pp. 1115–1122, 1980.
- [56] A. D. Reyes, E. W. Rubel, and W. J. Spain, "In vitro analysis of optimal stimuli for phase-locking and time-delayed modulation of firing in avian nucleus laminaris neurons," *The Journal of Neuroscience*, vol. 16, no. 3, pp. 993–1007, 1996.
- [57] G. F. Kuhn, "Physical acoustics and measurements pertaining to directional hearing," in *Directional Hearing*, W. A. Yost and G. Gourevitch, Eds., pp. 3–25, Springer, New York, NY, USA, 1987.
- [58] H. Agmon-Snir, C. E. Carr, and J. Rinzel, "The role of dendrites in auditory coincidence detection," *Nature*, vol. 393, no. 6682, pp. 268–272, 1998.
- [59] C. A. Miller, P. J. Abbas, and B. K. Robinson, "Response properties of the refractory auditory nerve fiber," *Journal of the Association for Research in Otolaryngology*, vol. 2, no. 3, pp. 216–232, 2001.
- [60] S. Dynes, *Discharge characteristics of auditory nerve fibers for pulsatile electrical stimuli [Ph.D. thesis]*, Massachusetts Institute of Technology, Cambridge, Mass, USA, 1996.
- [61] I. C. Bruce, L. S. Irlicht, M. W. White et al., "A stochastic model of the electrically stimulated auditory nerve: pulse- train response," *IEEE Transactions on Biomedical Engineering*, vol. 46, no. 6, pp. 630–637, 1999.
- [62] C. J. Brown and P. J. Abbas, "Electrically evoked whole-nerve action potentials: parametric data from the cat," *The Journal of the Acoustical Society of America*, vol. 88, no. 5, pp. 2205–2210, 1990.
- [63] R. P. Gaumond, D. O. Kim, and C. E. Molnar, "Response of cochlear nerve fibers to brief acoustic stimuli: role of discharge-history effects," *The Journal of the Acoustical Society of America*, vol. 74, no. 5, pp. 1392–1398, 1983.
- [64] M. C. Teich, L. Matin, and B. I. Cantor, "Refractoriness in the maintained discharge of the cat's retinal ganglion cell," *Journal of the Optical Society of America*, vol. 68, no. 3, pp. 386–402, 1978.
- [65] M. J. Berry, D. K. Warland, and M. Meister, "The structure and precision of retinal spike trains," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 94, no. 10, pp. 5411–5416, 1997.
- [66] M. J. Berry II and M. Meister, "Refractoriness and neural precision," *The Journal of Neuroscience*, vol. 18, no. 6, pp. 2200–2211, 1998.
- [67] D. R. Cox and P. A. W. Lewis, *The Statistical Analysis of Series of Events*, Methuen, London, UK, 1966.
- [68] D. H. Perkel, G. L. Gerstein, and G. P. Moore, "Neuronal spike trains and stochastic point processes. I. The single spike train," *Biophysical Journal*, vol. 7, no. 4, pp. 391–418, 1967.
- [69] D. H. Johnson and A. Swami, "The transmission of signals by auditory-nerve fiber discharge patterns," *The Journal of the Acoustical Society of America*, vol. 74, no. 2, pp. 493–501, 1983.
- [70] L. H. Carney, "A model for the responses of low-frequency auditory-nerve fibers in cat," *The Journal of the Acoustical Society of America*, vol. 93, no. 1, pp. 401–417, 1993.
- [71] M. C. Teich, "Fractal character of the auditory neural spike train," *IEEE Transactions on Biomedical Engineering*, vol. 36, no. 1, pp. 150–160, 1989.
- [72] M. C. Teich and S. B. Lowen, "Fractal patterns in auditory nerve-spike trains," *IEEE Engineering in Medicine and Biology Magazine*, vol. 13, no. 2, pp. 197–202, 1994.
- [73] J. Keat, P. Reinagel, R. C. Reid, and M. Meister, "Predicting every spike: a model for the responses of visual neurons," *Neuron*, vol. 30, no. 3, pp. 803–817, 2001.
- [74] P. A. W. Lewis and G. S. Shedler, "Simulation methods for Poisson processes in nonstationary systems," in *Proceedings of the 10th Conference on Winter Simulation (WSC '78)*, vol. 1, pp. 155–163, 1978.
- [75] T. C. Brown, "Poisson approximations and the definition of the Poisson process," *The American Mathematical Monthly*, vol. 91, no. 2, pp. 116–123, 1984.
- [76] D. L. Snyder and M. I. Miller, *Random Point Processes in Time and Space*, Springer, Berlin, Germany, 1991.

Research Article

Homogenous Chaotic Network Serving as a Rate/Population Code to Temporal Code Converter

Mikhail V. Kiselev

Megaputer Intelligence Ltd., Office 403 Building 1, 69 Bakuninskaya Street, Moscow 105082, Russia

Correspondence should be addressed to Mikhail V. Kiselev; mkiselev@megaputer.ru

Received 14 September 2013; Revised 31 January 2014; Accepted 14 February 2014; Published 23 March 2014

Academic Editor: Zhe (Sage) Chen

Copyright © 2014 Mikhail V. Kiselev. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

At present, it is obvious that different sections of nervous system utilize different methods for information coding. Primary afferent signals in most cases are represented in form of spike trains using a combination of rate coding and population coding while there are clear evidences that temporal coding is used in various regions of cortex. In the present paper, it is shown that conversion between these two coding schemes can be performed under certain conditions by a homogenous chaotic neural network. Interestingly, this effect can be achieved without network training and synaptic plasticity.

1. Introduction

Nervous system codes information in form of sequences of spikes or spike trains. Therefore, analysis of information processing in the brain is impossible without understanding principles of information coding and principles of conversion between different coding schemes, because it is well known that nervous system uses different coding schemes for transmitting information about stimuli, patterns, muscular commands, and so on. These coding schemes are based on two main approaches. In the first class of coding methods, the exact relative position of different spikes on the time axis is not taken into account, only their frequency or the sets of neurons emitting them are important. On the contrary, the other coding methods utilize exact delays between individual spikes. Let us call these two classes of codes asynchronous and synchronous codes.

There are several schemes of asynchronous coding and they are often used in combination. Rate coding is used in many afferent and efferent parts of nervous system. In this approach, intensity of a stimulus or command sent to a muscle is represented as number of spikes per unit time. It is the most explored coding method. Another asynchronous coding method, population coding, is based on representation of a stimulus as episodes of increased activity of a certain neuronal ensemble specific for this stimulus. It may be used

to code the fact of presence of some stimulus as well as its strength, as a number of active neurons. For example, it is known [1] that visual image moving direction is encoded as activity of the respective neuronal ensembles in middle temporal visual area of primates' brain. Rate and population coding can be considered as two sides of the general coding scheme when presence and/or intensity of some stimulus is expressed by increasing firing rate in certain population of neurons. We will call this scheme rate/population coding. In a specific variant of this scheme (sometimes referred to as position coding), numeric parameter of a stimulus is coded as position of the most active neurons in the ensemble. It was noted that this type of coding has a number of advantages compared to rate coding [2].

Synchronous coding (usually called temporal coding [3]) is based on the idea that precise relative timing of individual spikes inside spike trains emitted by different neurons can contain information about stimuli. This representation may have different forms. The fact of presence of some stimulus can be represented as a stable combination of spikes emitted by certain neurons with fixed delays of spikes with respect to other spikes (and therefore we can call this kind of temporal coding the spatiotemporal coding). Continuous value can be encoded as a time interval between two spikes, as a phase shift between two spike trains, or a spike phase relative to some global synchronizing signal. For example, hippocampal

CA1 pyramidal cells code body spatial location by their firing phase relatively to theta rhythm [4]. This coding method can be called the phase coding. The fact that temporal resolution of the neural code often has millisecond order of magnitude [5] is an evidence of wide usage of temporal coding in the brain. Most commonly used synaptic plasticity model, spike-timing dependent plasticity (STDP [6]), assumes this information coding scheme. Models of working memory based on neuronal polychronization effect [7, 8] are also naturally based on this coding method. Let us note that in this paper we will consider spatiotemporal variant of temporal coding only.

At present, it is evident that different sections of the nervous system utilize different information coding schemes. Primary afferent information encoded using rate- or population-based schemes is passed for processing to the cortex zones where temporal coding is widely used. But commands to muscles again should be represented as rate coded signals. It is also very probable that future intelligent systems and devices based on spiking neural networks (SNNs), for example, in robotics, will include components using various coding schemes. Therefore, SNNs performing functions of converter between different information coding forms should be a necessary part of nervous system as well as of these devices. However, in contrast with vast literature devoted to information coding in SNNs, the number of works considering conversion between different coding schemes is surprisingly few. For example, in [9] it was discussed how cortical bursting neurons could translate phase information contained in precisely timed spike sequences into rate coded signal. Relationship between rate and phase coding schemes in ensembles of hippocampal pyramidal neurons and translation from former to latter was explored in [10]. The question about which kind of networks could perform translation from rate/population coding to spatiotemporal coding seems to remain insufficiently explored until now. General approach to solution of this problem was presented by Izhikevich in [11] in relation to the so-called polychronous neuronal groups (PNGs). The idea is that desired conversion is performed by polychronous neuronal groups (populations of neurons which are being activated emit spike trains with precisely reproduced delays between individual spikes) existing or spontaneously emerging in chaotic neural network. However, this work as well as the subsequent works devoted to polychronization neither considered, to the extent of my knowledge, concrete conditions under which this conversion could be realized nor reported an experimental evidence of its realization. Achievement of these goals was motivation for the research reported here. The present work also uses polychronization effect as a basis, like [11], but, besides that, as we will see, it is shown in it that the network performing conversion from rate/population coding to spatiotemporal coding

- (1) may consist of leaky integrate-and-fire (LIF) neurons which are much simpler than the neuron model used in [11],
- (2) may not be plastic (while STDP plasticity was used in [11], it was noted there that synaptic plasticity can be harmful because it makes the conversion unstable),

- (3) does not need to be involved in global rhythmic activity like theta rhythm in [4].

2. Materials and Methods

In this research we utilized one of the simplest and in the same time the most widely used neuron model, leaky integrate-and-fire (LIF) neuron with absolute refractory period (see, e.g., [6]).

There are two kinds of neurons in the network: excitatory and inhibitory neurons. Axons of the excitatory and inhibitory neurons are connected only to excitatory or inhibitory synapses of other neurons, respectively. It is essential that postsynaptic spike emission is a result of collective activity of sufficient number (we set this number equal to 6) of presynaptic neurons. In order to use dimensionless units, we assume that the threshold membrane potential value is always equal to 1. To satisfy the above mentioned condition, the maximum excitatory synaptic weight value was selected equal to 0.19. Every individual synaptic weight was randomly selected using uniform distribution in the range $(0, 0.19)$. Inhibitory synapse weights were also assigned with randomly generated values uniformly distributed in the range $(0, W^-)$. Value of the maximum inhibitory weight W^- was used as a regulator necessary for maintaining balance of excitation and inhibition in the network while numbers of inhibitory and excitatory neurons were fixed. Namely, we used the network consisting of 700 excitatory and 300 inhibitory neurons in all the experiments. In case of small W^- even slight input signal or infrequent spontaneous firing causes avalanche of spike emissions leading the network to the state of constant self-sustaining activity with very high firing frequency. Great W^- values cause immediate suppression of network reaction to any external signal. For every network configuration there exists a threshold value of W^- above which self-sustaining network activity is impossible under any conditions. We selected value of W^- slightly higher than this threshold. In our case, it corresponded to $W^- = 10$.

Time of spike propagation from neuron to neuron lied in the range 1–10 ms for excitatory connections and 1–3 ms for inhibitory connections that is close to physiological values. Setting the spike propagation delays is considered below.

For sake of generality, the network should not have any structure a priori taking into account properties of input signal. In fact, the considered network has no intrinsic structure at all; it is completely homogenous and chaotic in the sense that all neurons of the same kind and all connections between the same kinds of neurons (excitatory \rightarrow excitatory, excitatory \rightarrow inhibitory, etc.) have the same distributions of weights, delays, connection probabilities, and so forth. Besides, neuron's axon can never be connected to a synapse of the same neuron.

Source of external signals received by the network is an array of its *input nodes*. Neurons are connected to them via excitatory or inhibitory synapses. Through these connections (we will call them afferent connections) neurons receive the signal consisting of noise (random spikes with constant mean frequency) and stimuli represented as short episodes of high

frequency spiking of certain groups of input nodes. The total number of input nodes was always the same and was equal to 1000. Ratio of excitatory and inhibitory input nodes was the same as for neurons, 700/300.

Provided that the described conditions are met, selection of sets of presynaptic neurons and input nodes was absolutely random for every neuron.

Many SNN computer simulation experiments show that distribution of synaptic delays is an equally important factor determining network behavior as distribution of synaptic weights. For example, it is crucial that the propagation delay of inhibitory connections would be substantively less than of excitatory connections; it is necessary to prevent development of the powerful permanent global oscillatory network activity which can suppress network reaction to external stimuli (as it follows, e.g., from theoretical model considered in [12]). However, if this requirement is satisfied, the exact distribution of inhibitory connection delays does not influence network properties significantly. On the contrary, selection of excitatory connection delays was found to be very important so that we consider it more in detail. As it was mentioned above, in our approach the key role in realization of rate/population to temporal coding conversion is played by PNGs. In [8], it was proposed to use SNNs artificially enriched by potential PNGs due to specially tuned excitatory propagation delays. Namely, excitatory neurons were considered as located at random points on surface of sphere or N-dimensional hypersphere and connection delays were made proportional to the spherical distance between the neurons connected. Since PNGs are characterized by great number of short paths between the same pair of neurons such that the total delay in every path is (almost) identical, this distribution of delays gives much greater number of PNGs than totally random distribution. Similar to [8], in this work we used 4-dimensional sphere neuron placement. Experimental results considered in next section confirm importance of this choice.

The input signal consisted of sequence of different stimuli mixed with noise. Every stimulus was 30 ms long and was presented after network reaction to previous stimulus that faded away completely (that was achieved due to proper selection of W^- —as was discussed earlier). Every stimulus was a sequence of randomly generated spikes from set of input nodes corresponding to this stimulus. The stimuli were characterized by significantly higher spike frequency comparatively to the background noise.

Advantage of the LIF neuron model is that it is very simple. Model of its soma includes only two parameters: the length of refractory period T and the membrane potential decay constant τ . The former limits the maximum firing frequency and is usually selected equal to few milliseconds. We set $T = 6$ ms. The latter determines the size of time period during which arriving presynaptic spikes act together to produce postsynaptic spike. This parameter varies in different kinds of neurons [6], however, obviously, it cannot be great in neurons forming PNGs which are based on very exact firing timings. For this reason we set it to 3 ms.

The general structural properties of the network are summarized in Table 1.

TABLE 1: Parameters determining balance of excitation and inhibition in the network.

	Excitatory neurons	Inhibitory neurons
Amount	700	300
Maximum synaptic weight (for postsynaptic neurons)	0.19	10
Number of excitatory afferent synapses/total effective weight ¹	300/28.5	300/28.5
Number of inhibitory afferent synapses/total effective weight	10/50	30/150
Number of nonafferent excitatory synapses/total effective weight	100/9.5	100/9.5
Number of nonafferent inhibitory synapses/total effective weight	10/50	3/15
Synaptic propagation delays (for postsynaptic neurons), ms	2–10	1–3

¹ It is the mean weight multiplied by the number of synapses.

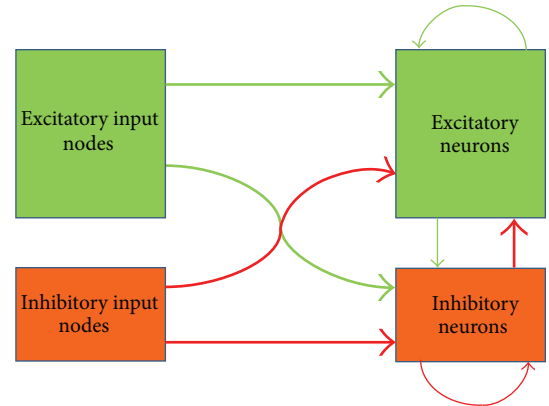


FIGURE 1: Excitatory (green) and inhibitory (red) neurons, input nodes, and synaptic connections. Size of blocks corresponds to relative amounts of neurons and input nodes. Thickness of arrows reflects total effective weights of the respective connections.

This table displays another important feature of the described network related to the role of inhibitory neurons. These neurons should not block network response to external stimuli but should efficiently stop network self-sustaining activity after end of stimulation. To reach this goal, the inhibitory neurons are themselves strongly inhibited by afferent signals, so that they almost do not fire during stimulation. But their mutual inhibition is much weaker than their inhibitory effect on the excitatory neurons (they have only 3 nonafferent inhibitory synapses, while excitatory neurons have 10). Therefore, just after the stimulus end the inhibitory neurons begin to fire extensively and suppress the whole network activity. The relative strength of excitation and inhibition in the network is shown schematically on Figure 1. The effect of this distribution of interneuron connections is depicted on Figure 2. It shows averaged firing frequency of excitatory and inhibitory neurons at different moments after beginning of stimulus presentation.

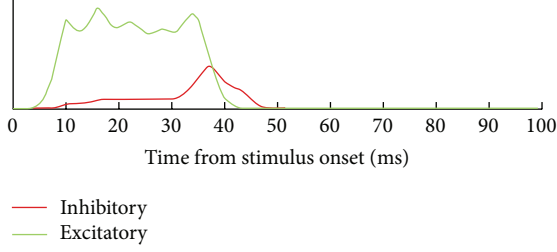


FIGURE 2: Dynamics of mean firing rate of excitatory (green) and inhibitory (red) neurons after stimulus presentation. Stimulus duration is 30 ms.

Now let us return to the final goal of this work. We see that the input signal is encoded in form of increased firing frequency of certain populations of the network input nodes. If every presentation of some stimulus activates a PNG specific for this stimulus it means, in our approach, that this stimulus is recoded to temporal form, since neurons belonging to active PNG fire in strict sequence with exact timings. Therefore, pursuing our goal we should solve the problem of finding PNGs in the network, and, in particular, the PNGs specific for the given patterns.

Basically, there are two different approaches to determination of PNGs inside a neural network [13]. The first method is based on analysis of the network structural properties such as synaptic delays and weights. In the second method the recordings of firing times of neurons are analyzed in order to determine stable repeating time-locked sequences of spikes associated with active PNGs. We used the second approach but implemented an alternative algorithm for PNG detection.

In my terms, PNG is defined by a sequence (neuron id, firing time). Only excitatory neurons are included in PNGs. Let us consider the stimulus A . Let P_{Ai} be a set of pairs (neuron id, time after the beginning of i th presentation of the stimulus A) corresponding to all spikes emitted before presentation of next stimulus. Set of all such sets corresponding to the stimulus A will be denoted as $\mathcal{P}_A = \{P_{Ai}\}$. Then the algorithm for finding in \mathcal{P}_A a PNG with support $n(G(\mathcal{P}_A, n))$ is the following:

- (1) create the matrix C_{at} , initializing by 1s all its elements, for which $\langle a, t \rangle \in P_{A1}$, and by 0s, all the rest elements.
- (2) Iteratively for each i , $2 \leq i \leq N_A$, find the shift s , for which the value of $\sum_{\langle a, t+s \rangle \in P_{Ai}} C_{at}$ is the greatest, then increment by 1 those C_{at} , for which $\langle a, t+s \rangle \in P_{Ai}$.
- (3) If $\forall t \forall a C_{at} < n$, then $G(\mathcal{P}_A, n) = \emptyset$, else if \tilde{t} is the least t , for which $C_{at} \geq n$, then $\langle a, t - \tilde{t} \rangle \in G(\mathcal{P}_A, n) \Leftrightarrow C_{at} \geq n$.

This algorithm can be illustrated by the following simple example (Figure 3). The upper row of this figure represents 3 fragments of firing history of 4 neurons (x -axis represents time), each fragment includes 4 time steps. Filled squares denote firing. The lower row displays the matrix C_{at} after step 1 of the algorithm and after 2 iterations on step 2. Gray squares in its final variant correspond to $G(\mathcal{P}_A, 2)$.

Using this algorithm the PNGs $G(\mathcal{P}_A, n)$ are found. But we are interested only in the PNGs specifically reacting to

only one stimulus. Let us define activity of the PNG G in the history fragment P_{Ai} as $A_{Ai}(G) = \max_s |\text{shift}(G, s) \cap P_{Ai}|$, where $\text{shift}(G, s)$ is built from G by addition of s to second elements in all pairs in G . Then the strength of reaction of G to the stimulus A can be defined as $R_A(G) = \min_i A_{Ai}(G)$, and the measure of selectivity of $G(\mathcal{P}_A, n)$ as $S(A, n) = N_A / |\{ \langle B, i \rangle : A_{Bi}(G(\mathcal{P}_A, n)) \geq R_A(G(\mathcal{P}_A, n)) \}|$. It is evident that if reaction of $G(\mathcal{P}_A, n)$ to any stimulus different from A is weaker than $R_A(G(\mathcal{P}_A, n))$, then $S(A, n) = 1$. It can be readily seen that the algorithm determining PNGs and their selectivity has complexity $O(TN_F N_S N_P^2)$, where T is time between presentation of consecutive stimuli, N_F is number of all stimuli presentations, N_S is total number of spikes in the whole firing protocol analyzed, and N_P is number of different stimuli.

Recoding process was declared as successful if for every stimulus a PNG with selectivity 1 was found. For every PNG we recorded its size and strength of its reaction to the corresponding stimulus (in the relative units, divided by the size of this PNG). The latter parameter has meaning of the minimum part of the PNG becoming active after stimulus presentation. Besides that, it is important to know to what degree these PNGs are independent; how the fact that a neuron that belongs to one PNG changes probability to find it inside some other PNG. If PNGs are independent it means that network has enough informational capacity to be able to convert greater number of different stimuli without loss of accuracy. Also, in case of numerous stimuli, independent PNGs are less similar and therefore permit more reliable recognition of encoded stimulus. To characterize the degree of independence of two PNGs we use sets of neurons belonging to them, S_1 and S_2 . If the PNGs are independent then size of the intersection of S_1 and S_2 equals approximately $(|S_1||S_2|)/N^+$, where N^+ is the total number of excitatory neurons. As a measure of dependence between two PNGs we take the ratio of real size of their intersection and this value. Proximity of the calculated number to 1 is an indication of their independence.

In all described experiments, determination of PNGs and measurement of their parameters were performed for 100 presentations of every stimulus. Results of these experiments are discussed in next section.

3. Results and Discussion

In the discussed experiments we varied 3 basic parameters of the input signal: number of different stimuli, intensity of stimuli (in terms of spike rate and population size), and signal/noise ratio. Only stimulus duration was always equal to 30 ms. It is close to duration of shortest stimuli recognized by living neural systems [14, 15].

The main goal of these experiments is to demonstrate that the desired conversion can be performed by the network described in previous section and that the effect is stable and observed under wide range of conditions, not just for carefully prepared specific signal. For this purpose it is quite sufficient to use star experiment design scheme starting from some point and varying different parameters separately;

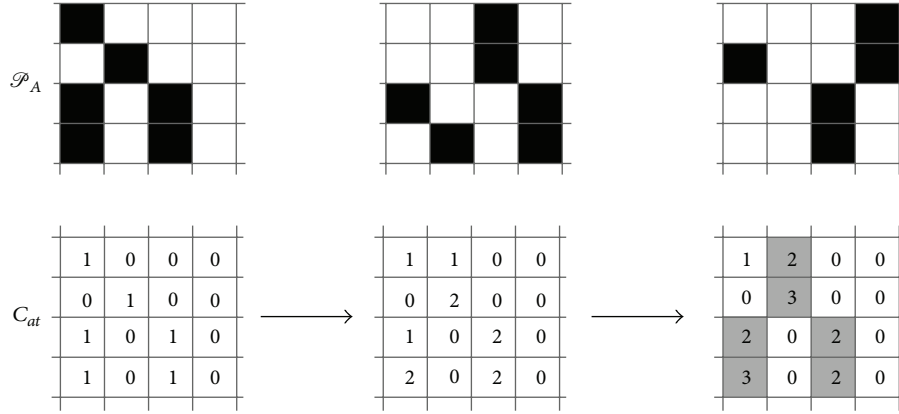


FIGURE 3: Determination of PNG with support 2 on 3 fragments of firing history of 4 neurons.

here we are not interested in exact dependencies of conversion characteristics on signal parameters, coupling effect of parameters, and so forth. The starting point corresponded to 10 stimuli, 100 input nodes per one stimulus, 300 Hz stimulus spike frequency, and 3 Hz background noise (that corresponds to signal/noise ratio = 10). Effect of variation of different parameters is considered in the following subsections.

3.1. Informational Capacity

3.1.1. Dependence of Conversion Quality on Number of Different Stimuli. The experiments were performed with number of different stimuli varying from 3 to 1000 (which is greater than the number of excitatory neurons in the network!). In all the experiments the conversion was successful; a PNG with selectivity equal to 1 was found for every stimulus. The detailed results for this experiment series are shown on Figure 4.

The most unexpected result is weak dependence of conversion quality on number of stimuli converted, even when there are more stimuli than excitatory neurons in the network. Average size of the polychronous groups performing the conversion was about 130. At least 20–25% of the respective PNG is activated after every presentation of the stimulus converted. All these PNGs are almost independent, although all points on the bottom plot are slightly above 1: it means that if a neuron belongs to some PNG it has a bit more chances to enter some other PNGs. Nevertheless, proximity of the average relative PNG intersection to 1 in all experiments is an indication that the network could convert successfully the number of stimuli significantly greater than the number of its neurons (however, in order to prove it experimentally much longer computation is required because computation time of the PNG detection algorithm described above is proportional to square of the number of different stimuli and even for 1000 stimuli the computation time was about 1.5 days).

3.2. Dependence of Conversion Quality on Stimulus Intensity. In case of uncorrelated presynaptic activity, LIF neuron

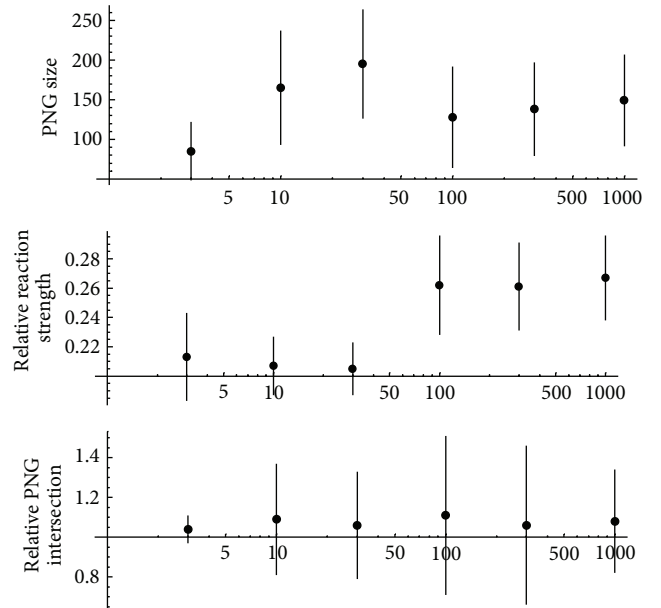


FIGURE 4: Conversion from rate/population coding to temporal coding in case of various numbers of different stimuli. The x-axis displays the number of stimuli using logarithmic scale. Length of the vertical lines drawn from the experimental points corresponds to standard deviation of the respective measured parameter.

behaves like a unit with sigmoid transfer function (with respect to spike rate). It is silent (in models without spontaneous firing) when presynaptic spikes are rare and fires with the maximum possible frequency determined by its refractory period in case of very frequent presynaptic spikes. Transfer between these “nothing” and “all” states may be more or less sharp: it depends on membrane potential decay time τ and average contribution of one presynaptic spike to membrane potential. In our case when τ is small (3 ms) and average synapse contribution is below 0.1 (while threshold membrane potential is set to 1), the sigmoid is rather similar to step function. For example, by decreasing stimulus spike frequency to 100 Hz, we observed that considerable number of stimulus presentations did not cause any network

TABLE 2: Effect of increased stimulus intensity.

Number of input nodes per stimulus	PNG size	Relative PNG reaction strength	Relative PNG intersection
100	165 ± 72	0.207 ± 0.02	1.09 ± 0.28
300	121 ± 55	0.247 ± 0.12	1.15 ± 0.6

reaction. The same effect took place when we decreased number of input nodes per stimulus to 30. Naturally, under these conditions the network cannot operate as a converter. Probably, ability of the network to convert weaker stimuli could be facilitated by using more complex neuron models with threshold membrane potential adaptation [16] or based on homeostatic synaptic plasticity [17].

On the contrary, increasing intensity of stimuli due to enlarging subset of input nodes corresponding to one stimulus only improves the conversion quality. In experiments with 300 input nodes per stimulus all stimuli had PNGs with absolute selectivity, and reaction strength of these PNGs was significantly greater than that for 100 input nodes per stimulus, although some PNGs showed tendency to stick together under this condition. The corresponding data are gathered in Table 2.

3.3. Influence of Background Noise. In the last series of the experiments we varied level of background noise in the range 1 Hz–30 Hz. It was senseless to perform experiments with noise more intensive than 30 Hz because under this condition inhibition level in the network became insufficient and the network demonstrated ceaseless strong activity. The results of these experiments show that the conversion process is very stable with respect to noise. Although, naturally, conversion under condition of strong noise (30 Hz noise corresponds to the signal/noise ratio equal to 1) has lower quality in terms of PNG response strength and degree of PNG independence, but, nevertheless, for all stimuli in all experiments PNGs with absolute selectivity were found.

The respective experimental data are represented in Figure 5.

3.4. Randomization of Excitatory Connection Delays. Probably, the most unexpected result obtained in this study is that synaptic plasticity was found to be unnecessary for achieving our goal. Indeed, the inventor of the term “polychronization”, Izhikevich, used synaptic plasticity (in fact, two kinds of it: long-term and short-term) in his experiments ([7] together with Szatmáry, [11], and others). Plasticity helped to highlight relatively rare neuron connections constituting PNGs in the ocean of other chaotic connections. We can hypothesize that since in our case the special selection of excitatory synaptic delays discussed above makes relative amount of PNGs much greater, it makes the positive effect of plasticity less important. In order to confirm this hypothesis, we performed experiments similar to ones considered above but with randomized values of delays in excitatory connections. Namely, after the network had been created using the rules described in the

previous section, the delays of all its excitatory synapses were randomly permuted that made the network completely chaotic. Under these conditions, PNGs were detected but they lost their selectivity. To illustrate it quantitatively, we measured part of stimuli for which selective PNGs were found. Values of this parameter for different number of stimuli are shown in Figure 6. For each number of stimuli the experiment was repeated 10 times. We see that only the easiest experiment with 3 different stimuli was successful from the point of view of our selectivity criterion. It would be interesting to understand why the observed dependence is not monotonous but detailed exploration of properties of completely chaotic networks has no direct relation to main subject of this research.

Also, it should be noted that in this study we used the very simple simulation of input signal; it is possible that future research where we plan to work with more realistic sensory signals will require implementation of some forms of synaptic plasticity in my model. Indeed, the primary purpose of this study was to demonstrate how a simple homogenous SNN can convert signal from rate/population coding form to temporal code. But, if to consider this work in context of research efforts directed at simulation of integration and processing of multimodal real-world sensory information flows, then the next step should be creation of software model of sufficiently rich informational environment for the studied SNNs and reproduction of the reported results under these more realistic conditions. It would make possible incorporation of working memory mechanisms based on PNGs [7, 8] as the next layer of the whole SNN-based information processing system because these mechanisms assume temporal coding of stimuli memorized.

4. Conclusions

Thus, it was discovered in this work that under certain conditions chaotic and homogenous network consisting of simple LIF neurons can convert signal encoded using rate/population-based scheme to a form based on temporal coding. It is important because each of these two forms of information coding is very common for many (but different) parts of central nervous system. It is interesting that synaptic plasticity and learning are not required for successful recoding. Presence of global synchronizing signal propagated across the whole network is also not necessary.

In my approach the recoding process is considered as selective activation of a polychronous neuronal group specific for the given stimulus encoded using rate/population coding scheme. Therefore, it is essential that the network is enriched by potential PNGs due to special selection of propagation delays in excitatory interneuron connections; namely, these delays have values proportional to distances between the neurons as if they were placed at random points of imaginary sphere (see the details in [8]). Appropriate choice of numbers of inhibitory synapses for excitatory and inhibitory neurons, their weights, and propagation delays (see Table 1 and Figure 1) are also very significant, because inhibitory neurons play an important role in this construction; they

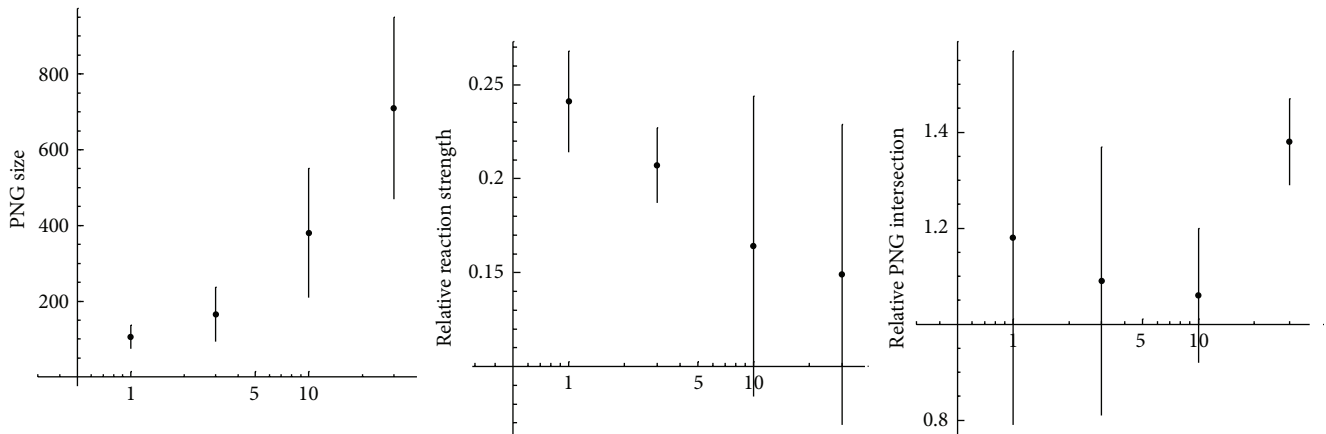


FIGURE 5: Conversion from rate/population coding to temporal coding under conditions of different background noise intensity (in Hz). It is displayed on the x -axes using logarithmic scale. Length of vertical lines corresponds to standard deviation of the respective parameter.

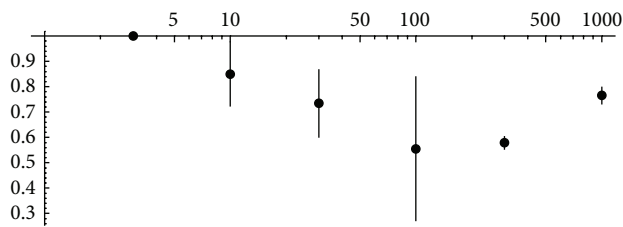


FIGURE 6: Part of stimuli for which PNGs with selectivity equal to 1 were found in case of SNN with randomized delays. The x -axis displays the number of stimuli using logarithmic scale. Each point corresponds to 10 experiments.

should stop uncontrolled growth of excitation leading to permanent senseless activity of the network while permitting the pronounced network reaction to stimulus presentation (Figure 2).

In our approach, the selective PNGs are determined by a specially designed novel algorithm. It has linear complexity with respect to the main dimensions of the problem except the number of different stimuli (it has complexity proportional to square of this parameter).

The described above computational experiments confirmed that stable and quality conversion is performed by the described network in great range of stimuli parameters.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The author would like to thank Eugene Izhikevich, Botond Szatmáry, Oleg Sinyavsky, and other members of the Brain Corporation research team for useful remarks and discussion on this research.

References

- [1] J. H. R. Maunsell and D. C. van Essen, "Functional properties of neurons in middle temporal visual area of the macaque monkey. I. Selectivity for stimulus direction, speed, and orientation," *Journal of Neurophysiology*, vol. 49, no. 5, pp. 1127–1147, 1983.
- [2] A. Mathis, A. V. Herz, and M. B. Stemmler, "Resolution of nested neuronal representations can be exponential in the number of neurons," *Physical Review Letters*, vol. 109, no. 1, Article ID 018103, 2012.
- [3] P. Dayan and L. F. Abbott, *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*, Massachusetts Institute of Technology Press, 2001.
- [4] O. J. Ahmed and M. R. Mehta, "The hippocampal rate code: anatomy, physiology and theory," *Trends in Neurosciences*, vol. 32, no. 6, pp. 329–338, 2009.
- [5] D. A. Butts, C. Weng, J. Jin et al., "Temporal precision in the neural code and the timescales of natural vision," *Nature*, vol. 449, no. 7158, pp. 92–95, 2007.
- [6] W. Gerstner and W. Kistler, *Spiking Neuron Models: Single Neurons, Populations, Plasticity*, Cambridge University Press, Cambridge, UK, 2002.
- [7] B. Szatmáry and E. M. Izhikevich, "Spike-timing theory of working memory," *PLoS Computational Biology*, vol. 6, no. 8, Article ID e1000879, 2010.
- [8] M. V. Kiselev, "Self-organization process in large spiking neural networks leading to formation of working memory mechanism—part I," in *Proceedings of the 12th International Work-Conference on Artificial Neural Networks (IWANN '13)*, pp. 510–517, Tenerife, Spain, 2013.
- [9] I. Samengo and M. A. Montemurro, "Conversion of phase information into a spike-count code by bursting neurons," *PLoS ONE*, vol. 5, no. 3, Article ID e9669, 2010.
- [10] M. R. Mehta, A. K. Lee, and M. A. Wilson, "Role of experience and oscillations in transforming a rate code into a temporal code," *Nature*, vol. 417, no. 6890, pp. 741–746, 2002.
- [11] E. M. Izhikevich, "Polychronization: computation with spikes," *Neural Computation*, vol. 18, no. 2, pp. 245–282, 2006.
- [12] M. Zeitler, A. Daffertshofer, and C. C. A. M. Gielen, "Asymmetry in pulse-coupled oscillators with delay," *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, vol. 79, no. 6, Article ID 065203, 2009.

- [13] R. Martinez and H. Paugam-Moisy, "Algorithms for structural and dynamical polychronous groups detection," in *Proceedings of the 19th International Conference on Artificial Neural Networks (CANN '09)*, vol. 5769 of *Lecture Notes in Computer Science*, pp. 75–84, Limassol, Cyprus, 2009.
- [14] F. Rieke, D. Warland, R. de Ruyter van Steveninck, and W. Bialek, *Spikes—Exploring the Neural Code*, MIT Press, Cambridge, UK, 1996.
- [15] C. Keysers, D. K. Xiao, P. Földiák, and D. I. Perrett, "The speed of sight," *Journal of Cognitive Neuroscience*, vol. 13, no. 1, pp. 90–101, 2001.
- [16] R. Brette and W. Gerstner, "Adaptive exponential integrate-and-fire model as an effective description of neuronal activity," *Journal of Neurophysiology*, vol. 94, no. 5, pp. 3637–3642, 2005.
- [17] G. G. Turrigiano and S. B. Nelson, "Homeostatic plasticity in the developing nervous system," *Nature Reviews Neuroscience*, vol. 5, no. 2, pp. 97–107, 2004.

Review Article

An Overview of Bayesian Methods for Neural Spike Train Analysis

Zhe Chen^{1,2}

¹ Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, 43 Vassar Street, Cambridge, MA 02139, USA

² Picower Institute for Learning and Memory, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

Correspondence should be addressed to Zhe Chen; zhechen@mit.edu

Received 16 May 2013; Revised 10 September 2013; Accepted 23 September 2013

Academic Editor: Wei Wu

Copyright © 2013 Zhe Chen. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Neural spike train analysis is an important task in computational neuroscience which aims to understand neural mechanisms and gain insights into neural circuits. With the advancement of multielectrode recording and imaging technologies, it has become increasingly demanding to develop statistical tools for analyzing large neuronal ensemble spike activity. Here we present a tutorial overview of Bayesian methods and their representative applications in neural spike train analysis, at both single neuron and population levels. On the theoretical side, we focus on various approximate Bayesian inference techniques as applied to latent state and parameter estimation. On the application side, the topics include spike sorting, tuning curve estimation, neural encoding and decoding, deconvolution of spike trains from calcium imaging signals, and inference of neuronal functional connectivity and synchrony. Some research challenges and opportunities for neural spike train analysis are discussed.

1. Introduction

Neuronal action potentials or spikes are the basic language that neurons use to represent and transmit information. Understanding neuronal representations of spike trains remains a fundamental task in computational neuroscience [1, 2]. With the advancement of multielectrode array and imaging technologies, neuroscientists have been able to record a large population of neurons at a fine temporal and spatial resolution [3]. To extract (“read out”) information from or inject/restore (“write in”) signals to neural circuits [4], there are emerging needs for modeling and analyzing neural spike trains recorded directly or extracted indirectly from neural signals, as well as building closed-loop brain-machine interfaces (BMIs). Many good examples and applications can be found in the volumes of the current or other special issues [5, 6].

In recent years, cutting-edge Bayesian methods have gained increasing attention in the analysis of neural data and neural spike trains. Despite its well-established theoretic principle since the inception of Bayes’ rule [7], Bayesian machinery has not been widely used in large-scaled data analysis

until very recently. This was partially ascribed to two facts: first, the development of new methodologies and effective algorithms; second, the ever-increasing computing power. The major theoretic or methodological development has been reported in the field of statistics, and numerous algorithms were developed in applied statistics and machine learning for successful real-world applications [8]. It is time to push this research frontier to neural data analysis. With this purpose in mind, this paper provides a tutorial review on the basic theory and the state-of-the-art Bayesian methods for neural spike train analysis.

The rest of the paper is organized as follows. Section 2 presents the background information about statistical inference and estimation, Bayes’ theory, and statistical characterization of neural spike trains. Section 3 reviews several important Bayesian modeling and inference methods in light of different approximation techniques. Section 4 reviews a few representative applications of Bayesian methods for neural spike train analysis. Finally, Section 5 concludes the paper with discussions on a few challenging research topics in neural spike train analysis.

2. Background

2.1. Estimation and Inference: Statistic versus Dynamic. Throughout this paper, we denote by Y the observed variables, by X the hidden variables and by θ an unknown parameter vector, and by $^\top$ the transpose operator for vector or matrix. We assume that $p(Y | X, \theta)$ has a regular and well-defined form of the likelihood function. For neural spike train analysis, Y typically consists of time series of single or multiple spike trains. Given a fixed time interval $(0, T]$, by time discretization we have $Y = \{Y_1, Y_2, \dots, Y_K\}$ (where $K = T/\Delta$ and Δ denotes the temporal bin size). A general statistical inference problem is stated as follows: given observations Y , estimate the unknown hidden variable X with a known θ , or estimate θ alone, or jointly estimate θ and X . The unknown variables θ and X can be either static or dynamic (e.g., time-varying with a Markovian structure). We will review the approaches that tackle these scenarios in this paper.

There are two fundamental approaches to solve the inference problem: likelihood approach and Bayesian approach. The likelihood approach [9] computes a point estimate by maximizing the likelihood function and represents the uncertainty of estimate via confidence intervals. The maximum likelihood estimate (m.l.e.) is asymptotically consistent, normal, and efficient, and it is invariant to reparameterization (i.e., functional invariance). However, the m.l.e. is known to suffer from overfitting, and therefore model selection is required in statistical data analysis. In contrast, the Bayesian philosophy lets data speak for themselves and models the unknowns (parameters, latent variables, and missing data) and uncertainties (which are not necessarily random) with probabilities or probability densities. The Bayesian approach computes the full posterior of the unknowns based on the rules of probability theory; the Bayesian approach can resolve the overfitting problem in a principled way [7, 8].

2.2. Bayesian Inference. The foundation of Bayesian inference is given by Bayes' rule, which consists of two rules: *product rule* and *sum rule*. Bayes' rule provides a way to compute the conditional, joint, and marginal probabilities. Specifically, let X and Y be two continuous random variables (r.v.); the conditional probability $p(X | Y)$ is given by

$$p(X | Y) = \frac{p(X, Y)}{p(Y)} = \frac{p(Y | X) p(X)}{\int p(Y | X) p(X) dX}. \quad (1)$$

If $X = \{X_i\}$ is discrete, then (1) is rewritten as

$$p(X_i | Y) = \frac{p(X_i, Y)}{p(Y)} = \frac{p(Y | X_i) p(X_i)}{\sum_j p(Y | X_j) p(X_j)}. \quad (2)$$

In Bayesian language, $p(Y | X)$, $p(X)$, and $p(X | Y)$ are referred to as the likelihood, prior and posterior, respectively. The Bayesian machinery consists of three types of basic operations: *normalization*, *marginalization*, and *expectation*, all of which involve integration. And the optimization problem consists in maximizing the posterior $p(X | Y)$ and finding the maximum a posteriori (MAP) estimate $X_{\text{MAP}} = \arg_X \max p(X | Y)$. Notably, except for very few scenarios

(i.e., Gaussianity), most integrations are computationally intractable or costly when dealing with high-dimensional problems. Therefore, for the sake of computational tractability, various types of approximations are often assumed at different stages of the inference procedure.

More specifically, for the state and parameter estimation problem, Bayesian inference aims to infer the joint posterior of the state and the parameter using Bayes' rule

$$\begin{aligned} p(X, \theta | Y) &\approx p(X | Y) p(\theta | Y) \\ &= \frac{p(Y | X, \theta) p(X) p(\theta)}{p(Y)} \\ &= \frac{p(Y | X, \theta) p(X) p(\theta)}{\iint p(Y | X, \theta) p(X) p(\theta) dX d\theta}, \end{aligned} \quad (3)$$

where the first equation assumes a factorial form of the posterior for the state and the parameter (first-stage approximation) and $p(X)$ and $p(\theta)$ denote the prior distributions for the state and parameter, respectively. The denominator of (3) is a normalizing constant known as the partition function. When dealing with a prediction problem for unseen data Y^* , we compute the posterior predictive distribution

$$p(Y^* | Y) = \iint p(Y^* | Y, \theta, X) p(X | Y) p(\theta | Y) dX d\theta \quad (4)$$

or its expected mean $\hat{Y}^* = \mathbb{E}_{p(Y^* | Y)}[Y^*] = \iiint Y^* p(Y^* | Y, \theta, X) p(X | Y) p(\theta | Y) dX d\theta dY^*$.

Sometimes, instead of maximizing the posterior, Bayesian inference attempts to maximize the marginal likelihood (also known as "evidence") $p(Y)$ as follows:

$$p(Y) = \iint p(Y | X, \theta) p(X) p(\theta) dX d\theta. \quad (5)$$

The second-stage approximation in approximate Bayesian inference deals with the integration in computing (3), (4), or (5), which will be reviewed in Section 3.

Note. Maximum likelihood inference can be viewed as a special case of Bayesian inference, in which θ is represented by a Dirac-delta function centered at the point estimate $\hat{\theta}_{\text{m.l.e.}}$; namely, $p(\theta) = \delta(\theta - \hat{\theta}_{\text{m.l.e.}})$. Nevertheless, Bayesian inference can still be embedded into likelihood inference to estimate $p(X)$ given a point estimate of θ . The $p(X)$ can either have an analytic form (with finite natural parameters) or be represented by Monte Carlo samples; the latter approach may be viewed as a specific case of Monte Carlo expectation-maximization (EM) methods.

2.3. Characterization of Neural Spike Trains. Neural spike trains can be modeled as a simple (temporal) point process [10]. For a single neural spike train observed in $(0, T]$, we often discretize it with a small bin size Δ such that each bin contains no more than one spike. The conditional intensity

TABLE 1: Probability distributions for modeling neuronal spike count observations.

Distribution	Mean statistic	Variance	Note for observations Y
Binomial (p)	$\mathbb{E}[Y] = p$	$p(1 - p)$	$Y \in \{0, 1\}$
Poisson (λ)	$\mathbb{E}[Y] = \lambda$	λ	$Y \geq 0, Y \in \mathbb{Z}^+$
Negative binomial (r, p)	$\mathbb{E}[Y] = pr/(1 - p)$	$pr/(1 - p)^2$	$Y \geq 0, Y \in \mathbb{Z}^+$ (overdispersed Poisson)
Skellam (μ_1, μ_2)	$\mathbb{E}[Y] = \mu_1 - \mu_2$	$\mu_1 + \mu_2$	$Y \in \mathbb{Z}$ (difference between two Poissons)

function (CIF), denoted as $\lambda(t \mid H_t)$, is used to characterize the spiking probability of a neural point process as follows:

$$\lambda(t \mid H_t) = \lim_{\Delta \rightarrow 0} \frac{\Pr \{\text{spike in } (t, t + \Delta] \mid H_t\}}{\Delta}, \quad (6)$$

where H_t denotes all history information available up to time t (that may include spike history, stimulus covariate, etc.). When $\lambda(t \mid H_t)$ is history independent, then the stochastic process is an inhomogeneous Poisson process. For notation simplicity, we sometimes use λ_t to replace $\lambda(t \mid H_t)$ when no confusion occurs. When Δ is sufficiently small, the product $\lambda(t \mid H_t)\Delta$ is approximately equal to the probability of observing a spike within the interval $((t - 1)\Delta, t\Delta]$. Assuming that the CIF λ_t is characterized by a parameter θ and an observed or latent variable X , then the point process likelihood function is given as [11–13]

$$p(Y \mid X, \theta) = \exp \left\{ \int_0^T \log \lambda(\tau \mid \theta, X) dy(\tau) - \int_0^T \lambda(\tau \mid \theta, X) d\tau \right\}, \quad (7)$$

where $dy(t)$ is an indicator function of the spike presence within the interval $((t - 1)\Delta, t\Delta]$. In the presence of multiple spike trains from C neurons, assuming that multivariate point process observations are conditionally independent at any time t given a new parameter θ , one then has

$$\begin{aligned} p(Y_{1:C} \mid X, \theta) &= \prod_{c=1}^C p(Y_c \mid X, \theta) \\ &= \prod_{c=1}^C \exp \left\{ \int_0^T \log \lambda_c(\tau \mid \theta, X) dy_c(\tau) - \int_0^T \lambda_c(\tau \mid \theta, X) d\tau \right\}. \end{aligned} \quad (8)$$

Since neural spike trains are fully characterized by the CIF, the modeling goal is then turned to model the CIF, which can have a parametric or nonparametric form. Identifying the CIF and its associated parameters is essentially a neural encoding problem (Section 4.2). A convenient modeling framework is the generalized linear model (GLM) [14, 15], which can model the binary (0/1) or spike count measurements. Within the exponential family, one can use the logit link function to model the binomial distribution, which has a generic form of $\log(p_t/(1 - p_t)) = \theta^\top X$; one can also use the

log link function to model the Poisson distribution, which has a generic form of $\log(\lambda_t) = \theta^\top X$.

In addition, researchers have used the negative binomial distribution to model spike count observations to capture the overdispersion phenomenon (where the variance is greater than the mean statistic). In many cases, for the purpose of computational tractability, researchers often use a Gaussian approximation for Poisson spike counts through a variance stabilization transformation. Table 1 lists a few population probability distributions for modeling spike count observations.

Another popular statistical model for characterizing population spike trains is the maximum entropy (MaxEnt) model with a log-linear form [16, 17]. Given an ensemble of C neurons, the ensemble spike activity can be characterized by the following form:

$$\begin{aligned} p(X) &= \frac{1}{\mathcal{Z}(X)} \exp \left(\sum_{i=1}^C \theta_c \langle x_c \rangle + \sum_{i,j} \theta_{ij} \langle x_i x_j \rangle \right) \\ &\equiv \frac{1}{\mathcal{Z}(X)} \exp \left(\sum_{i=1}^{C+C^2} \theta_i f_i(X) \right), \end{aligned} \quad (9)$$

where $x_i \in \{-1, +1\}$, $\langle \cdot \rangle$ denotes the sample average, $\langle x_c \rangle$ denotes the mean firing rate of the c th neuron, $f_i(X)$ denotes a generic function of X (where the couplings θ_i have to match the measured expectation values $\langle f_i(X) \rangle$), and $\mathcal{Z}(X)$ denotes the partition function. The basic MaxEnt model (9) assumes the stationarity of the data and includes the first- and second-order moment statistics but no stimulus component, but these assumptions can be relaxed to further derive an extended model.

An important issue for characterizing neural spike trains is model selection and the associated goodness-of-fit assessment. For goodness-of-fit assessment of spike train models, the reader is referred to [11, 18]. In addition, standard statistical techniques such as cross-validation, leave-one-out, and the receiver-operating-characteristic (ROC) curve may be considered. The model selection issue can be resolved by the likelihood principle based on well-established criteria (such as the *Bayesian information criterion* or *Akaike's information criterion*) [9, 11] or resolved by the Bayesian principle. Bayesian model selection and variable selection will be reviewed in Section 3.4.

3. Bayesian Modeling and Inference Methods

The common strategy of Bayesian modeling is to start with specific prior distributions for the unknowns. The prior

distributions are characterized by some hyperparameters, which can be directly optimized or modeled by the second-level hyperpriors. If the prior is conjugate to the likelihood, then the posterior has the same form as the prior [8]. Hierarchical Bayesian modeling characterizes the uncertainties of all unknowns at different levels.

In this section, we will review some, either exact or approximate, Bayesian inference methods. The approximate Bayesian inference methods aim to compute or evaluate the integration by approximation. There are two types of approaches to accomplish this goal: deterministic approximation and stochastic approximation. The deterministic approximation can rely on Gaussian approximation, deterministic sampling (e.g., sigma-point approximation [19, 20]) or variational approximation [21–23]. The stochastic approximation uses Monte Carlo sampling to achieve a point mass representation of the probability distribution. These two approaches have been employed to approximate the likelihood or posterior function in many inference problems, such as model selection, filtering and smoothing, and state and parameter joint estimation. Detailed coverage of these topics can be found in many excellent books (e.g., [24–28]).

3.1. Variational Bayes (VB). VB is based on the idea of variational approximation [21–23] and is also referred to as *ensemble learning* [24]. To avoid overfitting in maximum likelihood estimation, VB aims to maximize the marginal log-likelihood or its lower bound as follows:

$$\begin{aligned}
 \log p(Y) &= \log \int d\theta \int dX p(\theta) p(X, Y | \theta) \\
 &= \log \int d\theta \int dX q(X, \theta) \frac{p(\theta) p(X, Y | \theta)}{q(X, \theta)} \\
 &\geq \int d\theta \int dX q(X, \theta) \log \frac{p(\theta) p(X, Y | \theta)}{q(X, \theta)} \\
 &= \langle \log p(X, Y, \theta) \rangle_q + \mathcal{H}_q(X, \theta) \equiv \mathcal{F}(q(X, \theta)), \tag{10}
 \end{aligned}$$

where $p(\theta)$ denotes the parameter prior distribution, $p(X, Y | \theta)$ defines the complete data likelihood, and $q(X, \theta)$ is called the variational posterior distribution which approximates the joint posterior of the unknown state and parameter $p(X, \theta | Y)$. The term \mathcal{H}_q represents the entropy of the variational posterior distribution q , and $\mathcal{F}(q(X, \theta))$ is referred to as the free energy. The lower bound is derived based on the *Jensen's inequality* [29]. Maximizing the free energy $\mathcal{F}(q(X, \theta))$ is equivalent to minimizing the Kullback-Leibler (KL) divergence [29] between the variational posterior and true posterior (denoted by $\text{KL}(q \parallel p)$); since the KL divergence is nonnegative, we have $\mathcal{F}(q) = \log p(Y) - \text{KL}(q \parallel p) \leq \log p(Y)$. The optimization problem in (10) can be resorted to the VB-EM algorithm [23] in a similar fashion as the standard EM algorithm [30].

A common (but not necessary) VB assumption is a factorial form of the posterior $q(X, \theta) = q(X)q(\theta)$, although one can further impose certain structure within the parameter space. In the case of mean-field approximation, we have

$q(X, \theta) = q(X) \prod_i q(\theta_i)$. With selected priors $p(X)$ and $p(\theta)$, one can maximize the free energy by alternatively solving two equations: $\partial \mathcal{F} / \partial X = 0$ and $\partial \mathcal{F} / \partial \theta = 0$. Specifically, VB-EM inference can be viewed as a natural extension of the EM algorithm, which consists of the following two steps.

- (i) VB-E step: given the available information of $q(\theta)$, maximize the free energy \mathcal{F} with respect to the function $q(X)$ and update the posterior $q(X)$.
- (ii) VB-M step: given the available information of $q(X)$, maximize the free energy \mathcal{F} with respect to the function $q(\theta)$ and update the posterior $q(\theta)$. The posterior update will have an analytic form provided that the prior $p(\theta)$ is conjugate to the complete-data likelihood function (the conjugate-exponential family).

These two steps are alternated repeatedly until the VB algorithm reaches the convergence (say, the incremental change of \mathcal{F} value is below a small threshold). Similar to the iterative EM algorithm, the VB-EM inference has local maxima in optimization. To resolve this issue, one may use multiple random initializations or employ a deterministic annealing procedure [31]. The EM algorithm can be viewed as a variant of the VB algorithm in that the VB-M step replaces the point estimate (i.e., $q(\theta) = \delta(\theta - \theta_{\text{MAP}})$) from the traditional M-step with a full posterior estimate. Another counterpart of the VB-EM is the maximization-expectation (ME) algorithm [32], in which the VB-E step uses the MAP point estimate $q(X) = \delta(X - X_{\text{MAP}})$, while the VB-M step updates the full posterior.

It is noted that when the latent variables and parameters are intrinsically coupled or statistically correlated, the mean-field approximation will not be accurate, and consequently the VB estimate will be strongly biased. To alleviate this problem, the latent-space VB (LSVB) method [33, 34] aims to maximize a tighter lower bound of the marginal log-likelihood from (10) as follows:

$$\begin{aligned}
 \log p(Y) &\geq \int dX q(X) \log \frac{p(X, Y)}{q(X)} \\
 &= \int dX q(X) \log \frac{\int d\theta p(X, Y, \theta) p(\theta)}{q(X)} \tag{11} \\
 &\equiv \mathcal{F}(q(X)) \geq \max_{q(\theta)} \mathcal{F}(q(X) q(\theta)).
 \end{aligned}$$

The reader is referred to [33, 34] for more details and algorithmic implementation.

Note. (i) Depending on specific problems, the optimization bound of VB methods may not be tight, which may cause a large estimate bias or underestimated variance [35]. Desirably, a *data-dependent* lower bound is often tighter (such as the one used in Bayesian logistic regression [25]). (ii) It was shown in [36] that the VB method for statistical models with latent variables can be viewed as a special case of local variational approximation, where the log-sum-exp function is used to form the lower bound of the log-likelihood. (iii) The VB-EM inference was originally developed for the probabilistic models in the conjugate-exponential family, but it can

be extended to more general models based on approximation [37].

3.2. Expectation Propagation (EP). EP is a message-passing algorithm that allows approximate Bayesian inference for factor graphs (one type of probabilistic graphical model that shows how a function of several variables can be factored into a product of simple functions and can be used to represent a posterior distribution) [38]. For a specific r.v. X (either continuous or discrete), the probability distribution $p(X)$ is represented as a product of factors as follows:

$$p(X) = \prod_a f_a(X). \quad (12)$$

The basic idea of EP is to “divide-and-conquer” by approximating the factors one by one as follows:

$$f_a(X) \longrightarrow \tilde{f}_a(X) \quad (13)$$

and then use the product of approximated term as the final approximation as follows:

$$q(X) = \prod_a \tilde{f}_a(X). \quad (14)$$

As a result, EP replaces the global divergence $\text{KL}(p(X) \parallel q(X))$ by the local divergence between two product chains as follows:

$$\begin{aligned} & \text{KL}(p(X) \parallel q(X)) \\ &= \text{KL}\left(f_a(X) \prod_{b \neq a} f_b(X) \parallel \tilde{f}_a(X) \prod_{b \neq a} \tilde{f}_b(X)\right) \\ &\approx \text{KL}\left(f_a(X) \prod_{b \neq a} \tilde{f}_b(X) \parallel \tilde{f}_a(X) \prod_{b \neq a} \tilde{f}_b(X)\right). \end{aligned} \quad (15)$$

To minimize (15), the EP inference procedure is planned as follows.

Step 1. Use message-passing algorithms to pass messages $\tilde{f}_a(X)$ between factors.

Step 2. Given the received message $\tilde{f}_b(X)$ for factor a (for all $b \neq a$), minimize the local divergence to obtain $\tilde{f}_a(X)$, and send it to other factors.

Step 3. Repeat the procedure until convergence.

Note. (i) EP aims to find the closest approximation q such that $\text{KL}(p \parallel q)$ is minimized, whereas VB aims to find the variational distribution to minimize $\text{KL}(q \parallel p)$ (note that the KL divergence is asymmetric, and $\text{KL}(p \parallel q)$ and $\text{KL}(q \parallel p)$ have different geometric interpretations [39]). (ii) Unlike the global approximation technique (e.g., moment matching), EP uses a local approximation strategy to minimize a series of local divergence.

3.3. Markov Chain Monte Carlo (MCMC). MCMC methods are referred to as a class of algorithms for drawing random samples from probability distributions by constructing a Markov chain that has the equilibrium distribution as the desired distribution [40]. The designed Markov chain is reversible and has detailed balance. For example, given a transition probability P , the detailed balance holds between each pair of state i and j in the state space if and only if $\pi_i P_{ij} = \pi_j P_{ji}$ (where $\pi_i = \Pr(X_{t-1} = i)$ and $P_{ij} = \Pr(X_{t-1} = i, X_t = j)$). The appealing use of MCMC methods for Bayesian inference is to numerically calculate high-dimensional integrals based on the samples drawn from the equilibrium distribution [41].

The most common MCMC methods are the random walk algorithms, such as the Metropolis-Hastings (MH) algorithm [42, 43] and Gibbs sampling [44]. The MH algorithm is the simplest yet the most generic MCMC method to generate samples using a random walk and then to accept them with a certain acceptance probability. For example, given a random-walk proposal distribution $g(Z \rightarrow Z')$ (which defines a conditional probability of moving state Z to Z'), the MH acceptance probability $\mathcal{A}(Z \rightarrow Z')$ is

$$\mathcal{A}(Z \rightarrow Z') = \min\left(1, \frac{p(Z') g(Z' \rightarrow Z)}{p(Z) g(Z \rightarrow Z')}\right), \quad (16)$$

which gives a simple MCMC implementation. Gibbs sampling is another popular MCMC method that requires no parameter tuning. Given a high-dimensional joint distribution $p(Z) = p(z_1, \dots, z_n)$, Gibbs sampler draws samples from the individual conditional distribution $p(z_i \mid Z_{-i})$ in turn while holding others fixed (where Z_{-i} denote the $n - 1$ variables in Z except for z_i).

For high-dimensional sampling problems, the random-walk behavior of the proposal distribution may not be efficient. Imagine that there are two directions (increase or decrease in the likelihood space) for a one-dimensional search; there will be 2^n search directions in an n -dimensional space. On average, it will take about $2^n/n$ steps to hit the exact search direction. Notably, some sophisticated MCMC algorithms employ side information to improve the efficiency of the sampler (i.e., the “mixing” of the Markov chain). Examples of non-random-walk methods include successive over-relaxation, hybrid Monte Carlo, gradient-based Langevin MCMC, and Hessian-based MCMC [24, 45–47].

Many statistical estimation problems (e.g., change point detection, clustering, and segmentation) consist in identifying the unknown number of statistical objects (e.g., change points, clusters, and boundaries), which are categorized as the variable-dimensional statistical inference problem. For this kind of inference problem, the so-called reversible jump MCMC (RJ-MCMC) method has been developed [48], which can be viewed as a variant of MH algorithm that allows proposals to change the dimensionality of the space while satisfying the detailed balance of the Markov chain.

Note. As discussed in Section 2.2, since the fundamental operations of Bayesian statistics involve integration, the MCMC methods appear naturally as the most generic techniques for Bayesian inference. On the one hand, the recent

decades have witnessed an exponential growth in the MCMC literature for its own theoretic and algorithmic developments. On the other hand, there has been also an increasing trend in applying MCMC methods to neural data analysis, ranging from spike sorting, tuning curve estimation, and neural decoding to functional connectivity analysis, some of which will be briefly reviewed in Section 4.

3.4. Bayesian Model Selection and Variable Selection. Statistical model comparison can be carried on by Bayesian inference. From Bayes' rule, the model posterior probability is expressed by

$$p(\mathcal{M}_i | D) \propto p(D | \mathcal{M}_i) p(\mathcal{M}_i). \quad (17)$$

Under the assumption of equal model priors, maximizing the model posterior is equivalent to maximizing the model evidence (or marginal likelihood) as follows:

$$\begin{aligned} p(D | \mathcal{M}_i) &= \int_{\theta} p(D, \theta | \mathcal{M}_i) d\theta \\ &= \int_{\theta} p(D | \theta, \mathcal{M}_i) p(\theta | \mathcal{M}_i) d\theta. \end{aligned} \quad (18)$$

The Bayes factor (BF), defined as the ratio of evidence between two models, can be computed as [49]

$$\begin{aligned} \text{BF} &= \frac{p(D | \mathcal{M}_1)}{p(D | \mathcal{M}_2)} = \frac{\int p(D, \theta_1 | \mathcal{M}_1) d\theta_1}{\int p(D, \theta_2 | \mathcal{M}_2) d\theta_2} \\ &= \frac{\int p(\theta_1 | \mathcal{M}_1) p(D | \theta_1, \mathcal{M}_1) d\theta_1}{\int p(\theta_2 | \mathcal{M}_2) p(D | \theta_2, \mathcal{M}_2) d\theta_2}. \end{aligned} \quad (19)$$

Specifically, the BF is treated as the Bayesian alternative to P values for testing hypotheses (in model selection) and for quantifying the degree the observed data support or conflict with a hypothesis [50]. As discussed previously in Section 3.1, the marginal likelihood may be intractable for a large class of probabilistic models. In practice, the BF is often computed based on numerical approximation, such as the Laplace-Metropolis Estimator [51] or sequential Monte Carlo methods [52]. In addition, for a large sample size, the logarithm of the BF can be roughly approximated by the Bayesian information criterion (BIC) [9], whose computation is much simpler without involving numerical integration.

Bayesian model selection can also be directly implemented via the so-called MCMC model composition (MC^3). The basic idea of MC^3 is to simulate a Markov chain $\{\mathcal{M}(t)\}$ with an equilibrium distribution as $p(\mathcal{M}_i | D)$. For each model \mathcal{M} , define a neighborhood $\text{nbrd}(\mathcal{M})$ and a transition matrix q by setting $q(\mathcal{M} \rightarrow \mathcal{M}') = 0$ for all $\mathcal{M}' \notin \text{nbrd}(\mathcal{M})$. Draw a new sample \mathcal{M}' from $q(\mathcal{M} \rightarrow \mathcal{M}')$ and accept the new sample with a probability

$$\min \left\{ 1, \frac{p(\mathcal{M}' | D)}{p(\mathcal{M} | D)} \right\}. \quad (20)$$

Otherwise the chain remains unchanged. Once the Markov chain converges to the equilibrium, one can construct the model posterior based on Monte Carlo samples.

Within a fixed model class, it is often desirable to have a compact or sparse representation of the model to alleviate overfitting. Namely, many coefficients of the model parameters are zeros. A very useful approach for variable selection is the so-called automatic relevance determination (ARD) that encourages sparse Bayesian learning [24, 26, 53]. More specifically, ARD provides a way to infer hyperparameters in hierarchical Bayesian modeling. Given the likelihood $p(Y | \theta)$ and the parameter prior $p(\theta | \omega)$ (where ω denotes the hyperparameters), one can assign a hyperprior $p(\omega | \eta)$ for ω such that the marginal distribution $p(\theta) = \int p(\theta | \omega) p(\omega) d\omega$ is peaked and long-tailed (thereby favoring a sparse solution). The hyperprior $p(\omega)$ can be either identical or different for each element in θ . In the most general form, we can write

$$p(\theta) = \prod_i p(\theta_i) = \prod_i \int p(\theta_i | \omega_i) p(\omega_i | \eta_i) d\omega_i. \quad (21)$$

The hyperprior parameters $\eta = \{\eta_i\}$ can be fixed or optimized from data. Upon completing Bayesian inference, the estimated mean and variance statistics of some coefficients θ_i will be close to zero (i.e., with the least relevance) and therefore can be truncated. The ARD principle has been widely used in various statistical models, such as linear regression, GLM, and the relevance vector machine (RVM) [26].

3.5. Bayesian Model Averaging (BMA). BMA is a statistical technique aiming to account for the uncertainty in the model selection process [54]. By averaging many different competing statistical models (e.g., linear or Cox regression and GLM), BMA incorporates model uncertainties into parameter inference and data prediction.

Consider an example of GLM involving choosing independent variables and the link function. Every possible combination of choices defines a different model, say $\{\mathcal{M}_0, \mathcal{M}_1, \dots, \mathcal{M}_K\}$ (where \mathcal{M}_0 denotes the null model). Upon computing K Bayes factors $\text{BF}_{10} = p(D | \mathcal{M}_1)/p(D | \mathcal{M}_0)$, $\text{BF}_{20} = p(D | \mathcal{M}_2)/p(D | \mathcal{M}_0)$, ..., and $\text{BF}_{K0} = p(D | \mathcal{M}_K)/p(D | \mathcal{M}_0)$, the posterior probability $p(\mathcal{M}_k | D)$ is computed as [54]

$$p(\mathcal{M}_k | D) = \frac{\pi_k \text{BF}_{k0}}{\sum_{i=0}^K \pi_i \text{BF}_{i0}}, \quad (22)$$

where $\pi_k = p(\mathcal{M}_k)/p(\mathcal{M}_0)$ denotes the prior odds for model \mathcal{M}_k against \mathcal{M}_0 . In the case of GLM, the marginal likelihood can be approximated by the Laplace method [55].

3.6. Bayesian Filtering: Kalman Filter, Point Process Filter, and Particle Filter. Bayesian filtering aims to infer a filtered or predictive posterior distribution of temporal data in a sequential fashion, which is often cast within the framework of state space model (SSM) [13, 56, 57]. Without loss of generality, let \mathbf{x}_t denote the state at discrete time t and let $\mathbf{y}_{0:t}$ denote the cumulative observations up to time t . The filtered

posterior distribution of the state, conditional on the observations $\mathbf{y}_{0:t}$, bears a form of *recursive* Bayesian estimation as follows:

$$\begin{aligned}
 p(\mathbf{x}_t | \mathbf{y}_{0:t}) &= \frac{p(\mathbf{x}_t) p(\mathbf{y}_{0:t} | \mathbf{x}_t)}{p(\mathbf{y}_{0:t})} \\
 &= \frac{p(\mathbf{x}_t) p(\mathbf{y}_t, \mathbf{y}_{0:t-1} | \mathbf{x}_t)}{p(\mathbf{y}_t, \mathbf{y}_{0:t-1})} \\
 &= \frac{p(\mathbf{x}_t) p(\mathbf{y}_t | \mathbf{x}_t, \mathbf{y}_{0:t-1}) p(\mathbf{y}_{0:t-1} | \mathbf{x}_t)}{p(\mathbf{y}_t | \mathbf{y}_{0:t-1}) p(\mathbf{y}_{0:t-1})} \\
 &= \frac{p(\mathbf{x}_t) p(\mathbf{y}_t | \mathbf{x}_t, \mathbf{y}_{0:t-1}) p(\mathbf{x}_t | \mathbf{y}_{0:t-1}) p(\mathbf{y}_{0:t-1})}{p(\mathbf{y}_t | \mathbf{y}_{0:t-1}) p(\mathbf{y}_{0:t-1}) p(\mathbf{x}_t)} \\
 &= \frac{p(\mathbf{y}_t | \mathbf{x}_t, \mathbf{y}_{0:t-1}) p(\mathbf{x}_t | \mathbf{y}_{0:t-1})}{p(\mathbf{y}_t | \mathbf{y}_{0:t-1})} \\
 &= \frac{p(\mathbf{y}_t | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{y}_{0:t-1})}{p(\mathbf{y}_t | \mathbf{y}_{0:t-1})}, \tag{23}
 \end{aligned}$$

where the first four steps are derived from Bayes' rule and the last equality of (23) assumes the conditional independence between the observations. The one-step state prediction, also known as the *Chapman-Kolmogorov equation* [58], is given by

$$p(\mathbf{x}_t | \mathbf{y}_{0:t-1}) = \int p(\mathbf{x}_t | \mathbf{x}_{t-1}) p(\mathbf{x}_{t-1} | \mathbf{y}_{0:t-1}) d\mathbf{x}_{t-1}, \tag{24}$$

where the probability distribution (or density) $p(\mathbf{x}_t | \mathbf{x}_{t-1})$ describes a state transition equation and the probability distribution (or density) $p(\mathbf{y}_t | \mathbf{x}_t)$ is the observation equation. Together (23) and (24) provide the fundamental relations to conduct state space analyses. The above formulation of recursive Bayesian estimation holds for both continuous and discrete variables, for either \mathbf{x} or \mathbf{y} or both. When the state variable is discrete and countable (in which we use S_t to replace \mathbf{x}_t), the SSM is also referred to as a hidden Markov model (HMM), with associated $p(S_t | S_{t-1})$ and $p(\mathbf{y}_t | S_t)$. Various approximate Bayesian methods for the HMM have been reported [23, 59, 60]. When the hidden state consists of both continuous and discrete variables, the SSM is referred to as a switching SSM, with associated $p(\mathbf{x}_t | \mathbf{x}_{t-1}, S_t)$ and $p(\mathbf{y}_t | \mathbf{x}_t, S_t)$ [27, 61]. In this case, the inference and prediction involve multiple integrals or summations. For example, the prediction equation (24) will be rewritten as

$$\begin{aligned}
 p(\mathbf{x}_t | \mathbf{y}_{0:t-1}, S_{0:t-1}) &= \int \sum_{S_{t-1}} p(\mathbf{x}_t | \mathbf{x}_{t-1}, S_t) p(S_t | S_{t-1}) \\
 &\quad \times p(\mathbf{x}_{t-1} | \mathbf{y}_{0:t-1}, S_{0:t-1}) d\mathbf{x}_{t-1} \tag{25}
 \end{aligned}$$

whose exact or naive implementation can be computationally prohibitive given a large discrete state space.

When the state and observation equations are both continuous and Gaussian, the Bayesian filtering solution yields the celebrated Kalman filter [62], in which the posterior mean

and posterior variance are updated recursively. In fact, based on a Gaussian approximation of nonnegative spike count observations, the Kalman filter has been long used in spike train analysis [63, 64]. However, such a naive Gaussian approximation does not consider the point process nature of neural spike trains. Brown and his colleagues [65–67] have proposed a point process filter to recursively estimate the state or parameter in a dynamic fashion. Without loss of generality, assume that the CIF (6) is characterized by a parameter θ via an exponential form, namely, $\lambda_t \equiv \lambda(t | \theta_t) = \exp(\theta_t^\top X_t)$, and assume that the parameter follows a random-walk equation $\theta_t = \theta_{t-1} + w_t$ (where w_t denotes random Gaussian noise with zero mean and variance σ^2); then one can use a point process filter to estimate the time-varying parameter θ at arbitrarily fine temporal resolution (i.e., the bin size can be as small as possible for the discrete-time index t) as follows:

$$\theta_{t+1|t} = \theta_{t|t} \text{ (one-step mean prediction)}, \tag{26}$$

$$V_{t+1|t}(\theta) = V_{t+1|t}(\theta) + \sigma^2 \text{ (one-step variance prediction)}, \tag{27}$$

$$\begin{aligned}
 \theta_{t+1|t+1} &= \theta_{t+1|t} + V_{t+1|t}(\theta) \frac{\nabla_\theta \lambda(\theta_{t+1|t})}{\lambda(\theta_{t+1|t})} \\
 &\quad \times [dy_{t+1} - \lambda(\theta_{t+1|t+1}) \Delta] \\
 &= \theta_{t+1|t} + V_{t+1|t}(\theta) X_{t+1} \\
 &\quad \times [dy_{t+1} - \lambda(\theta_{t+1|t+1}) \Delta] \text{ (posterior mode)}, \tag{28}
 \end{aligned}$$

$$\begin{aligned}
 V_{t+1|t+1}(\theta) &= [(V_{t+1|t}(\theta))^{-1} + X_{t+1} X_{t+1}^\top \lambda(\theta_{t+1|t}) \Delta]^{-1} \\
 &\quad \text{(posterior variance)}, \tag{29}
 \end{aligned}$$

where $\theta_{t+1|t+1}$ and $V_{t+1|t+1}(\theta)$ denote the posterior mode and posterior variance for the parameter θ , respectively. Equations (26)–(29) are reminiscent of Kalman filtering. Equations (26) and (27) for one-step mean and variance predictions are the same as Kalman filtering, but (28) and (29) are different from Kalman filtering due to the presence of non-Gaussian observations and nonlinear operation in (28). In (28), $[dy_{t+1} - \lambda(\theta_{t+1|t+1}) \Delta]$ is viewed as the innovations term, and $V_{t+1|t} X_{t+1}$ may be interpreted as a “Kalman gain.” The quantity of the Kalman gain determines the “step size” in error correction. In (29), the posterior state variance is derived by inverting the second derivative of the log-posterior probability density $\log p(\theta_t | Y)$ based on a Gaussian approximation of the posterior distribution around the posterior mode [65–67]. For this simple example, we have

$$\begin{aligned}
 &\log p(\theta_t | Y_{0:t}, H_t) \\
 &\propto -\frac{1}{2}(\theta_t - \theta_{t-1|t-1})^\top V_{t+1|t}^{-1} (\theta_t - \theta_{t-1|t-1}) \\
 &\quad + [\log \lambda_t dy_t - \lambda_t \Delta],
 \end{aligned}$$

$$\begin{aligned}
& \frac{\partial \log p(\theta_t | Y_{0:t}, H_t)}{\partial \theta_t} \\
&= -(\theta_t - \theta_{t-1|t-1})^\top V_{t+1|t}^{-1} \\
&\quad + \frac{1}{\lambda_t} \nabla_{\theta} \lambda_t [dy_t - \lambda_t \Delta], \\
& \frac{\partial^2 \log p(\theta_t | Y_{0:t}, H_t)}{\partial \theta_t \partial \theta_t^\top} \\
&= -V_{t+1|t}^{-1} + \left[\left(\frac{\partial^2 \lambda_t}{\partial \theta_t \partial \theta_t^\top} \frac{1}{\lambda_t} - \left(\frac{\partial \lambda_t}{\partial \theta_t} \right)^2 \frac{1}{\lambda_t^2} \right) \right. \\
&\quad \left. \times [dy_t - \lambda_t \Delta] - \left(\frac{\partial \lambda_t}{\partial \theta_t} \right)^2 \frac{1}{\lambda_t} \Delta \right]. \tag{30}
\end{aligned}$$

Setting the first-order derivative $\partial \log p(\theta_t | Y_{0:t}, H_t) / \partial \theta_t$ to zero and rearranging terms yield (28), and setting $V_{t+1|t+1}(\theta) = -[\partial^2 \log p(\theta_t | Y_{0:t}, H_t) / (\partial \theta_t \partial \theta_t^\top)]^{-1}$ yields (29).

The Gaussian approximation is based on the first-order Laplace method. In theory one can also use a second-order method to further improve the approximation accuracy [68]. However, in practice the performance gain is relatively small in the presence of noise and model uncertainty while analyzing real experimental data sets. Although the above example only considers a univariate point process (i.e., a single neuronal spike train), it is straightforward to extend the analysis to multivariate point processes (multiple neuronal spike trains). When the number of the neurons increases, the accuracy of Gaussian approximation of log-posterior also improves due to the *Law of large numbers*.

An alternative way for estimating a non-Gaussian posterior is to use a particle filter [69]. Several reports have been published in the context of neural spike train analysis [70, 71]. The basic idea of particle filtering is to employ sequential Monte Carlo (importance sampling and resampling) methods and draw a set of independent and identically distributed (i.i.d.) samples (i.e., “particles”) from a *proposal distribution*; the samples are propagated through the likelihood function, weighted, and reweighted after each iteration update. In the end, one can use Monte Carlo samples (or their importance weights) to represent the posterior. For example, to evaluate the expectation of a function $f(\mathbf{x}_t)$ with respect to the posterior $p(\mathbf{x}_t | \mathbf{y}_{0:t})$, we have

$$\begin{aligned}
\mathbb{E}[f(\mathbf{x}_t)] &= \int f(\mathbf{x}_t) \frac{p(\mathbf{x}_t | \mathbf{y}_{0:t})}{q(\mathbf{x}_t | \mathbf{y}_{0:t})} q(\mathbf{x}_t | \mathbf{y}_{0:t}) d\mathbf{x}_t \\
&= \int f(\mathbf{x}_t) W(\mathbf{x}_t) q(\mathbf{x}_t | \mathbf{y}_{0:t}) d\mathbf{x}_t \\
&\approx \frac{\sum_{i=1}^{N_c} f(\mathbf{x}_t^{(i)}) W(\mathbf{x}_t^{(i)})}{\sum_{i=1}^{N_c} W(\mathbf{x}_t^{(i)})} = \hat{f}(\mathbf{x}_t), \tag{31}
\end{aligned}$$

where $W(\mathbf{x}_t) = p(\mathbf{x}_t | \mathbf{y}_{0:t}) / q(\mathbf{x}_t | \mathbf{y}_{0:t})$ denotes the importance weight function and $\{\mathbf{x}_t^{(i)}\}_{i=1}^{N_c}$ denotes the N_c particles drawn from the proposal distribution $q(\mathbf{x}_t | \mathbf{y}_{0:t})$. When the sample size N_c is sufficiently large (depending on the

dimensionality of \mathbf{x}), the estimate $\hat{f}(\mathbf{x}_t)$ will be an unbiased estimate of $\mathbb{E}[f(\mathbf{x}_t)]$. Based on sequential important sampling (SIS), the importance weights of each sample can be recursively updated as follows [69]:

$$W(\mathbf{x}_t^{(i)}) = W(\mathbf{x}_{t-1}^{(i)}) \frac{p(\mathbf{y}_t | \mathbf{x}_t^{(i)}) p(\mathbf{x}_t^{(i)} | \mathbf{x}_{t-1}^{(i)})}{q(\mathbf{x}_t^{(i)} | \mathbf{x}_{0:t-1}^{(i)}, \mathbf{y}_t)}. \tag{32}$$

In practice, choosing a proper proposal distribution $q(\mathbf{x}_t | \mathbf{x}_{0:t-1}, \mathbf{y}_t)$ is crucial (see [69] for detailed discussions). In the neuroscience literature, Brockwell et al. [70] used a transition prior $p(\mathbf{x}_t | \mathbf{x}_{t-1})$ as the proposal distribution, which yields a simple form of update from (32) as follows:

$$W(\mathbf{x}_t^{(i)}) = W(\mathbf{x}_{t-1}^{(i)}) p(\mathbf{y}_t | \mathbf{x}_t^{(i)}). \tag{33}$$

That is, the importance weights $W(\mathbf{x}_t^{(i)})$ are only scaled by the instantaneous likelihood value. Despite its simplicity, the transition prior proposal distribution completely ignores the information of current observation \mathbf{y}_t . To overcome this limitation, Ergun et al. [71] used a filtered (Gaussian) posterior density derived from the point process filter as the proposal distribution, and they reported a significant performance gain in estimation while maintaining the algorithmic simplicity (i.e., sampling from a Gaussian distribution). In addition, the VB approach can be integrated with particle filtering to obtain a variational Bayesian filtering algorithm [72].

Note. (i) If the online operation is not required, we can estimate a smoothed posterior distribution $p(\mathbf{x}_t | \mathbf{y}_{0:T})$ to obtain a more accurate estimate. The above Bayesian filters can be extended to the fixed-lag Kalman smoother, point process smoother, and particle smoother [63, 66, 69]. (ii) For neural spike train analysis, the formulation of Bayesian filtering is applicable not only to simple point processes but also to marked point processes [73] or even spatiotemporal point processes.

3.7. Bayesian Nonparametrics. The contrasting methodological pairs “*frequentist* versus *Bayes*” and “*parametric* versus *nonparametric*” are two examples of dichotomy in modern statistics [74]. The historical roots of Bayesian nonparametrics are dated back to the late 1960s and 1970s. Despite its theoretic development over the past few decades, successful applications of nonparametric Bayesian inference have not been widespread until recently, especially in the field of machine learning [75]. Since nonparametric Bayesian models accommodate a large number of degrees of freedom (infinite-dimensional parameter space) to exhibit a rich class of probabilistic structure, such approaches are very powerful in terms of data representation. The fundamental building blocks are two stochastic processes: Dirichlet process (DP) and Gaussian process (GP). Although detailed technical reviews of these topics are far beyond the scope of this paper, we would like to point out the strengths of these methods in two aspects of statistical data analysis.

- (i) Data clustering, partitioning, and segmentation: unlike the finite mixture models, nonparametric

Bayesian models define a prior distribution over the set of all possible partitions, in which the number of clusters or partitions may grow as the increase of the data samples in both static and dynamic settings, including the infinite Gaussian mixture model, Dirichlet process mixtures, Chinese restaurant process, and infinite HMM [74–76]. The model selection issue is resolved implicitly in the process of infinite mixture modeling.

- (ii) Prediction and smoothing: unlike the fixed finite-dimensional parametric models, the GP defines priors for the mean function and covariance function, where the covariance kernel function determines the smoothness and stationarity between the data points. Since the predictive posterior is Gaussian, the prediction uncertainty can be computed analytically [28, 77].

Therefore, Bayesian nonparametrics offer greater flexibility for modeling complex data structures. Unfortunately, most inference algorithms for Bayesian nonparametric models involve MCMC methods, which can be computationally prohibitive for large-scale neural data analysis. Therefore, exploiting the sparsity structure of specific neural data and designing efficient inference algorithms are two important directions in practical applications [78].

4. Bayesian Methods for Neural Spike Train Analysis

In this section, we review some representative applications of Bayesian methods for neural spike train analysis, with specific emphases on the real experimental data. Nevertheless, the list of the references is by no means complete, and some other complementary references can be found in [79, 80]. Specifically, the strengths of the Bayesian methods are highlighted in comparison with other standard methods; the potentially issues arising from these methods are also discussed.

4.1. Spike Sorting and Tuning Curve Estimation. To characterize the firing property of single neurons, it is necessary to first *identify* and *sort* the spikes from the recorded multiunit activity (MUA) (which is referred to as the discrete ensemble spikes passing the threshold criterion) [81–83]. However, spike sorting is often a difficult and error-prone process. Traditionally, spike sorting is formulated as a clustering problem based on spike waveform features [84]. Parametric and nonparametric Bayesian inference methods have been developed for mixture modeling and inference (e.g., [25, 26]), especially for determining the model size [85, 86]. Unlike the maximum likelihood estimation (which produces a hard label for each identified spike), Bayesian approaches produce a soft label (posterior probability) for individual spike; such uncertainties may be considered in subsequent analyses (such as tuning curve estimation and decoding). Spike sorting can also be formulated as a dynamic model inference problem, in the context of state space analysis [87] or in the presence of nonstationarity [88]. Recent studies have suggested that spike sorting should take into account not only spike waveform

features but also the neuronal tuning property [89, 90], suggesting that these two processes shall be integrated.

At the single neuron level, a Poisson neuronal firing response is completely characterized by its tuning curve or receptive field (RF). Naturally, estimating the neuronal tuning curve is the second step following spike sorting. Standard tuning curve or RF estimation methods include the spike-triggered average (STA) and spike-triggered covariance (STC). The Bayesian versions of the STA and STC have been proposed [91, 92]. Binning and smoothing are two important issues in firing rate estimation. Bayesian methods provide a principled way to estimate the peristimulus time histogram (PSTH) [93]. For estimating a time-varying firing rate profile similar to PSTH, the Bayesian adaptive regression splines (BARS) method offers a principled solution for bin size selection and smoothing based on the RJ-MCMC method [94]. Notably, BARS is more computationally intensive. For similar estimation performance (validated on simulated data), a more computationally efficient approach has been developed using Bayesian filtering-based state space analysis [95]. In addition, Metropolis-type MCMC approaches have been proposed for high-dimensional tuning curve estimation [96, 97].

4.2. Neural Encoding and Decoding. The goal of neural encoding is to establish a statistical mapping (which can be either a biophysical or data-driven model) between the stimulus input and neuronal responses, and the goal of neural decoding is to extract or reconstruct information of the stimulus given the observed neural signals. For instance, the encoded and decoded variables of interest can be a rodent's position during spatial navigation, the monkey's movement kinematics in a reach-to-grasp task, or specific visual/auditory/olfactory stimuli during neuroscience experiments.

Without loss of generality, let $\{\tilde{X}, \tilde{Y}\}$ denote the *observed* stimuli and neuronal responses, respectively, at the encoding stage, and let θ denote the model parameter of a specific encoding model \mathcal{M} ; then the posterior distribution of the model (and model parameters) is written as

$$p(\theta, \mathcal{M} | \tilde{X}, \tilde{Y}) \propto p(\tilde{X}, \tilde{Y} | \theta, \mathcal{M}) p(\theta | \mathcal{M}) p(\mathcal{M}). \quad (34)$$

Once the model \mathcal{M} is determined, one can infer the posterior mean by $\hat{\theta} = \int \theta p(\theta | \tilde{X}, \tilde{Y}, \mathcal{M}) d\theta$. Depending on the selected likelihood or prior, variations of Bayesian neural encoding methods have been developed [98–100].

Given the parameter posterior $p(\theta | \tilde{X}, \tilde{Y}, \mathcal{M})$ from the encoding analysis, decoding analysis aims to infer the latent variable X given new data Y at the decoding stage (with preselected \mathcal{M}). Within the Bayesian framework, it is equivalent to finding the X_{MAP} [101] as follows:

$$\begin{aligned} X_{\text{MAP}} &= \arg \max_X p(X | \theta, Y, \mathcal{M}) \\ &= \arg \max_X \int p(Y | X, \theta, \mathcal{M}) p(\theta | \tilde{X}, \tilde{Y}, \mathcal{M}) p(X) d\theta \\ &\approx \arg \max_X p(Y | X, \hat{\theta}, \mathcal{M}) p(X), \end{aligned} \quad (35)$$

which consists of two numerical problems: *maximization* and *integration*. In the approximation in the last step of (35), we have used $p(\theta | \bar{X}, \bar{Y}, \mathcal{M}) \approx \delta(\theta - \hat{\theta})$, where $\hat{\theta}$ denotes the estimated mean or mode statistic from $p(\theta | \bar{X}, \bar{Y}, \mathcal{M})$. The optimization problem is more conveniently written in the log domain as follows:

$$\log p(X | Y, \hat{\theta}) \propto \log p(Y | X, \hat{\theta}) + \log p(X). \quad (36)$$

If X follows a Markovian process, this can be solved by recursive Bayesian filtering [65, 67] (Section 3.6). When X is non-Markovian but $p(X)$ and the likelihood are both log-concave, this can be resorted to a global optimization problem [57, 102]. Imposing prior information and structure (e.g., sparsity, spatiotemporal correlation) onto $p(X)$ is important for obtaining either a meaningful solution or a significant optimization speedup [103, 104]. In contrast, when $p(X)$ is flat or noninformative, the MAP solution will be similar to the m.l.e.

In the literature, the majority of neural encoding or decoding models fall within two parametric families: linear model (e.g., [63, 105]) and GLM (e.g., [64, 106, 107]), although nonparametric encoding models have also been considered [108, 109]. Methods for Bayesian neural decoding include (i) Kalman filtering [63], (ii) point process filtering [65–67, 110, 111], (iii) particle filtering [70, 71], and (iv) MCMC methods [112]. The areas of experimental neuroscience data include the retina, primary visual cortex, primary somatosensory cortex, auditory periphery (auditory nerves and midbrain auditory neurons), primary auditory cortex, primary motor cortex, premotor cortex, hippocampus, and the olfactory bulb.

It is important to point out that most spike-count or point process based decoding algorithms rely on the assumptions that neural spikes have been properly sorted (some neural decoding algorithms (e.g., [113]) are based on detected MUA instead of sorted single unit activity). Recently, there have been a few efforts in developing spike-sorting-free decoding algorithms, by either estimating the cell identities as missing variables [114] or modeling the spike identities by their proxy based on a spatiotemporal point process [115, 116]. Although this work has been carried out using likelihood inference, it is straightforward to extend it to the Bayesian framework. In the example of decoding the rat's position from recorded ensemble hippocampal spike activity [115, 116], we used a model-free (without θ) and data-driven Bayes' rule as follows:

$$p(X | Y, \bar{X}, \bar{Y}) \propto p(Y | X, \bar{X}, \bar{Y}) p(X), \quad (37)$$

in which $p(X)$ denotes the prior and the likelihood $p(Y | X, \bar{X}, \bar{Y})$ is evaluated nonparametrically (namely, nonparametric neural decoding). By assuming that the joint/marginal/conditional distributions ($p(X, Y)$ and $p(\bar{X}, \bar{Y})$, $p(X)$ and $p(\bar{X})$, and $p(Y | X)$ and $p(\bar{Y} | \bar{X})$) are stationary during

both encoding and decoding phases, the MAP estimate of decoding analysis is obtained by

$$\begin{aligned} X_{\text{MAP}} &= \arg \max_X p(Y | X, \bar{X}, \bar{Y}) p(X) \\ &\approx \arg \max_X f(Y | p(X | \bar{X}), p(X, Y | \bar{X}, \bar{Y})) p(X), \end{aligned} \quad (38)$$

where f is a nonlinear function that involves the marginal and joint pdf's in the argument [115, 116], in which the pdf's are constructed based on a kernel density estimator (KDE). Alternatively, the nonparametric pdf in (38) can be replaced by a parametric form [115] as follows:

$$X_{\text{MAP}} \approx \arg \max_X f(Y | p(X | \theta), p(X, Y | \theta)) p(X), \quad (39)$$

where $p(X | \theta) = \int p(X, Y | \theta) dY$ is the parametric marginal and θ is the point estimate obtained from the training samples $\{\bar{X}, \bar{Y}\}$.

Note. (i) Neural encoding and decoding analyses are established upon the assumption that the neural codes are well understood—namely, how neuronal spikes represent and transmit the information of the external world. Whether being a rate code, a timing code, a latency code, or an independent or correlated population code, Bayesian approach provides a universal strategy to test the coding hypothesis or extract the information [117]. (ii) The sensitivity of spike trains to noise may affect the effectiveness to the encoding-decoding process. From an information-theoretic perspective, various sources of spike noise, such as misclassified spikes (false positives) and misdetected, or misclassified spikes (false negatives), may affect differently the mutual information between the input (stimulus) and output (spikes) channel [118, 119]. In designing a Bayesian decoder, it is important to take into account the noise issue. A decoding strategy that is robust to the noise assumption will presumably yield the best performance [115, 116].

4.3. Deconvolution of Neural Spike Trains. Fluorescent calcium imaging tools have become increasingly popular for observing the spiking activity of large neuronal populations. However, extracting or deconvolving neural spike trains from the raw fluorescence movie or video sequences remains a challenging estimation problem. The standard dF/F or Wiener filtering approaches do not capture the true statistics of neural spike trains and are sensitive to the noise statistics [120].

A principled approach is to formulate the deconvolution problem of a filtered point process via state space analysis and Bayesian inference [121, 122] (see also [123] for another type of Bayesian deconvolution approach using MCMC). Let F_t denote the measured univariate fluorescence time series, which is modeled as a linear Gaussian function of the intracellular calcium concentration ($[Ca^{2+}]$) as follows:

$$F_t = \alpha [Ca^{2+}]_t + \beta + \epsilon_t, \quad (40)$$

where β denotes a constant baseline and $\epsilon_t \sim \mathcal{N}(0, \sigma^2)$ denotes the Gaussian noise with zero mean and variance σ^2 . The calcium concentration can be modeled as a first-order autoregressive (AR) process corrupted by Poisson noise as follows:

$$\alpha[\text{Ca}^{2+}]_t = \alpha[\text{Ca}^{2+}]_{t-1} + n_t, \quad (41)$$

where $n_t \sim \text{Poisson}(\lambda\Delta)$ and the bin size Δ is chosen to assure that the mean firing rate is independent of the imaging frame rate.

Let $\theta = \{\alpha, \beta, \gamma, \sigma^2, \lambda\}$; given the above generative biophysical model, Bayesian deconvolution aims to seek the MAP estimate of spike train as follows:

$$\begin{aligned} \hat{\mathbf{n}} &= \arg \max_{n_t \in \mathbb{N}_0} p(\mathbf{n} | \mathbf{F}, \theta) \\ &= \arg \max_{n_t \in \mathbb{N}_0} p(\mathbf{F} | \mathbf{n}, \theta) p(\mathbf{n} | \theta) \\ &= \arg \max_{n_t \in \mathbb{N}_0} \prod_{t=1}^T p(F_t | \text{Ca}_t^{2+}, \theta) \prod_{t=1}^T p(n_t | \theta). \end{aligned} \quad (42)$$

Within the state space framework, Vogelstein and colleagues [121] proposed a particle filtering method to infer the posterior probability of spikes at each imaging frame, given the entire fluorescence traces. However, the Monte Carlo approach is computationally expensive and may not be suitable for analyses of a large population of neurons. To meet the real-time processing requirement, they further proposed an approximate yet fast solution by replacing the Poisson distribution by an exponential distribution with the same mean (therefore relaxing the nonnegative integer constraint to the nonnegative real number) [122]. And the approximate solution is given by the following optimization problem:

$$\begin{aligned} \hat{\mathbf{n}} &= \arg \max_{n_t \geq 0} \sum_{t=1}^T -\frac{1}{2\sigma^2} (F_t - \alpha \text{Ca}_t^{2+} - \beta)^2 - n_t \lambda \Delta \\ &= \arg \max_{\text{Ca}_t^{2+} - \gamma \text{Ca}_{t-1}^{2+} \geq 0} \sum_{t=1}^T -\frac{1}{2\sigma^2} (F_t - \alpha \text{Ca}_t^{2+} - \beta)^2 \\ &\quad - (\text{Ca}_t^{2+} - \gamma \text{Ca}_{t-1}^{2+}) \lambda \Delta. \end{aligned} \quad (43)$$

The approximation of exponential form makes the optimization problem concave with respect to Ca^{2+} , from which the global optimum can be obtained using constrained convex optimization [102]. Once the estimate of the calcium trace is obtained, the MAP spike train can be simply inferred by a linear transformation.

In a parallel fashion, the parameter θ can be similarly estimated by Bayesian inference as follows:

$$\begin{aligned} \theta_{\text{MAP}} &= \arg \max_{\theta} \int p(\mathbf{F} | \text{Ca}^{2+}, \theta) p(\text{Ca}^{2+} | \theta) d\text{Ca}^{2+} \\ &\approx \arg \max_{\theta} p(\mathbf{F} | \hat{\mathbf{n}}, \theta) p(\hat{\mathbf{n}} | \theta), \end{aligned} \quad (44)$$

where the approximation in the second step assumes that the major mass in the integral is around the MAP sequence $\hat{\mathbf{n}}$

(or equivalently the Ca^{2+} traces). Therefore, the joint estimate $(\hat{\mathbf{n}}, \theta_{\text{MAP}})$ can be computed iteratively from (43) and (44) until convergence.

Note. The output of Bayesian deconvolution yields a probability vector between 0 and 1 of having a spike in a given time frame. Selection of different thresholds on the probability vector leads to different detection errors (a tradeoff between the false positives and false negatives). Nevertheless, the Bayesian solution is much more superior to the standard least-squares method. It is noteworthy that a new fast deconvolution method has recently been proposed based on finite rate of innovation (FRI) theory, with reported performance better than the approximate Bayesian solution [124].

4.4. Inference of Neuronal Functional Connectivity and Synchrony. Identifying the functional connectivity of simultaneously recorded neuronal ensembles is an important research objective in computational neuroscience. This analysis has many functional applications such as in neural decoding [125] and in understanding the collective dynamics of coordinated spiking cortical networks [126]. Compared to the standard nonparametric approaches such as cross-correlogram and joint peristimulus time histogram (JPSTH), parametric model-based statistical approaches offer several advantages in neural data interpretation [127].

To model the spike train point process data, without loss of generality we use the following logistic regression model with a logit link function. Specifically, let c be the index of a target neuron, and let $i = 1, \dots, C$ be the indices of triggered neurons (whose spike activity is assumed to trigger the firing of the target neuron). The Bernoulli (binomial) logistic regression GLM is written as

$$\begin{aligned} \text{logit}(\pi_t) &= \theta_c^T X_t = \theta_0^c + \sum_{j=1}^J \theta_j^c x_{j,t} \\ &= \theta_0^c + \sum_{i=1}^C \sum_{k=1}^K \theta_{i,k}^c x_{i,t-k}, \end{aligned} \quad (45)$$

where $\dim(\theta_c) = J + 1 = C \times K + 1$ for the augmented parameter vector $\theta_c = \{\theta_0^c, \theta_{i,k}^c\}$ and $X_t = \{x_0, x_{i,t-k}\}$. Here, $x_0 \equiv 1$, and $x_{i,t-k}$ denotes the raw spike count from neuron i at the k th time-lag history window (or a predefined smooth basis function such as in [125]). The spike count is nonnegative; therefore $x_{i,t-k} \geq 0$. Alternatively, (45) can be rewritten as

$$\pi_t = \frac{\exp(\theta_c^T X_t)}{1 + \exp(\theta_c^T X_t)} = \frac{\exp(\theta_0^c + \sum_{j=1}^J \theta_j^c x_{j,t})}{1 + \exp(\theta_0^c + \sum_{j=1}^J \theta_j^c x_{j,t})}, \quad (46)$$

which yields the probability of a spiking event at time t . Equation (46) defines a spiking probability model for neuron c based on its own spiking history and that of the other neurons in the ensemble. Here, $\exp(\theta_0^c)$ can be interpreted as the baseline firing probability of neuron c . Depending on the algebraic (positive or negative) sign of coefficient $\theta_{i,k}^c$, $\exp(\theta_{i,k}^c)$ can be viewed as a “gain” factor (dimensionless, >1 or <1) that influences the relative firing probability of neuron c

from another neuron i at the previous k th time lag. Therefore, a negative value of $\theta_{i,k}^c$ will strengthen the inhibitory effect; a positive value of $\theta_{i,k}^c$ will enhance the excitatory effect. Two neurons are said to be functionally connected if any of their pairwise connections is nonzero (or the statistical estimate is significantly nonzero).

For inferring the functional connectivity of neural ensembles, in addition to the standard likelihood approaches [127, 128], various forms of Bayesian inference have been developed for the MaxEnt model, GLM, and Bayesian network [129–132]. In a similar context, a Bayesian method has been developed based on the deconvolved neuronal spike trains from calcium imaging data [133].

Bayesian methods also proved useful in detecting higher-order correlations among neural assemblies [134, 135]. Higher-order correlations are often characterized by synchronous neuronal firing at a timescale of 5–10 ms. These findings have been reported in experimental data from prefrontal cortex, somatosensory cortex, and visual cortex across many species and animals. Consider a set of C neurons. Each neuron is represented by two states: 1 (firing) or 0 (silent). At any time instant, the state of the C neurons is represented by the vector $X = (x_1, x_2, \dots, x_C)$ (the time index is omitted for simplicity), and in total there are 2^C neuronal states. For instance, a general joint distribution of three neurons can be expressed by a log-linear model [134]

$$p(x_1, x_2, x_3) = \exp(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_{12} x_1 x_2 + \theta_{13} x_1 x_3 + \theta_{23} x_2 x_3 + \theta_{123} x_1 x_2 x_3), \quad (47)$$

which is a natural extension of the MaxEnt model described in (9). A nonzero coefficient of θ_{123} would imply the presence of third-order correlation among the three neurons. In experimental data, the number of synchronous events may be scarce in single trials, and the interaction coefficients may also be time-varying. State space analysis and Bayesian filtering offer a principled framework to address these issues [135]. However, the computational bottleneck is the curse of dimensionality when the value of C is moderately large ($2^{20} \approx 10^6$). In the presence of finite data sample size, it is reasonable to impose certain structural priors onto the parameter space for the Bayesian solution.

5. Discussion

We have presented an overview of Bayesian inference methods and their applications to neural spike train analysis. Although the focus of current paper is on neural spike trains, the Bayesian principle is also applicable to other modalities of neural data (e.g., [136]). Due to space limitation, we only cover representative methods and applications in this paper, and the references are reflective of our personal choices from the humongous literature.

In comparison with the standard methods, Bayesian methods provide a flexible framework to address many fundamental estimation problems at different stages of neural data analysis. Regardless of the specific Bayesian approach to

be employed, the common goal of Bayesian solutions consists in replacing a single point estimate (or hard decision label) with a full posterior distribution (or soft decision label). As a tradeoff, Bayesian practitioners have to encounter the increasing cost of computational complexity (especially while using MCMC), which may be prohibitive for large-scale spike train data sets. Furthermore, special attention shall be paid to select the optimal technique among different Bayesian methods that ultimately lead to quantitatively different approximate Bayesian solutions.

Despite the significant progresses made to date, there remain many research challenges and opportunities for applying Bayesian machinery to neural spike trains, and we will mention a few of them below.

5.1. Nonstationarity. Neural spiking activity is highly nonstationary at various timescales. Sources that account for such nonstationarity may include the animal's behavioral variability across trials, top-down attention, learning, motivation, or emotional effects across time. These effects are time-varying across behaviors. In addition, individual neuronal firing may be affected by other unobserved neural activity, such as through modulatory or presynaptic inputs from other nonrecorded neurons. Therefore, it may be important to consider these latent variables while analyzing neural spike trains [137]. Bayesian methods are a natural solution to model and infer such latent variables. Traditional mixed-effects models can be adapted to a hierarchical Bayesian model to capture various sources of randomness.

5.2. Characterization of Neuronal Dependencies. Neural responses may appear correlated or synchronous at different timescales. It is important to characterize such neuronal dependencies in order to fully understand the nature of neural codes. It is also equally important to associate the neural responses to other measurements, such as behavioral responses, learning performance, or local field potentials. Commonly, correlation statistics or information-theoretic measures have been used (e.g., [138]). Other advanced statistical measures have also been proposed, such as the log-linear model [139], Granger causality [140], transfer entropy [141], or copula model [142]. Specifically, the copula offers a universal framework to model statistical dependencies among continuous, discrete, or mixed-valued r.v., and it has an intrinsic link to the mutual information; Bayesian methods may prove useful for selecting the copula class or the copula mixtures [143]. However, because of the nonstationary nature of neural codes (Section 5.1), it remains a challenge to identify the “true” dependencies among the observed neural spike trains, and it remains important to rule out and rule in neural codes under specific conditions.

5.3. Characterization and Abstraction of Neuronal Ensemble Representation. Since individual neuronal spike activity is known to be stochastic and noisy, in the single-trial analysis it is anticipated that the information extracted from neuronal populations is more robust than that from a single neuron. How to uncover the neural representation of population codes in a single-trial analysis has been an active research

topic in neuroscience. This is important not only for abstraction, interpretation, and visualization of population codes but also for discovering invariant neural representations and their links to behavior. Standard dimensionality reduction techniques (e.g., principle component analysis, multidimensional scaling, or locally linear embedding) have been widely used for such analyses. However, these methods have ignored the temporal component of neural codes. In addition, no explicit behavioral correlate may become available in certain modeling tasks. Recently, Bayesian dynamic models, such as the Gaussian process factor analysis (GPFA) [144] and VB-HMM [145–147], have been proposed to visualize population codes recorded from large neural ensembles across different experimental conditions. To learn the highly complex structure of spatiotemporal neural population codes, it may be beneficial to borrow the ideas from the machine learning community and to integrate the state-of-the-art unsupervised and supervised deep Bayesian learning techniques.

5.4. Translational Neuroscience Applications. Finally in the long run, it is crucial to apply basic neuroscience knowledge derived from quantitative analyses of neural data to translational neuroscience research. Many clinical research areas may benefit from the statistical analyses reviewed here, such as design of neural prosthetics for patients with tetraplegia [107], detection and control of epileptic seizures, optical control of neuronal firing in behaving animals, or simulation of neural firing patterns to achieve optimal electrotherapeutic effect [148]. Bridging the gap between neural data analysis and their translational applications (such as treating neurological or neuropsychiatric disorders) would continue to be a prominent mission accompanying the journey of scientific discovery.

Acknowledgments

The author was supported by an Early Career Award from the Mathematical Biosciences Institute, Ohio State University. This work was also supported by the NSF-IIS CRCNS (Collaborative Research in Computational Neuroscience) Grant (no. 1307645) from the National Science Foundation.

References

- [1] E. N. Brown, R. E. Kass, and P. P. Mitra, “Multiple neural spike train data analysis: state-of-the-art and future challenges,” *Nature Neuroscience*, vol. 7, no. 5, pp. 456–461, 2004.
- [2] S. Grün and S. Rotter, *Analysis of Parallel Spike Trains*, Springer, New York, NY, USA, 2010.
- [3] I. H. Stevenson and K. P. Kording, “How advances in neural recording affect data analysis,” *Nature Neuroscience*, vol. 14, no. 2, pp. 139–142, 2011.
- [4] G. B. Stanley, “Reading and writing the neural code,” *Nature Neuroscience*, vol. 16, pp. 259–263, 2013.
- [5] Z. Chen, T. W. Berger, A. Cichocki, K. G. Oweiss, R. Quian Quiroga, and N. V. Thakor, “Signal processing for neural spike trains,” *Computational Intelligence and Neuroscience*, vol. 2010, Article ID 698751, 2 pages, 2010.
- [6] J. Macke, P. Berens, and M. Bethge, “Statistical analysis of multi-cell recordings: linking population coding models to experimental data,” *Frontiers in Computational Neuroscience*, vol. 5, article 35, 2011.
- [7] J. Bernardo and A. F. M. Smith, *Bayesian Theory*, John & Wiley, New York, NY, USA, 1994.
- [8] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian Data Analysis*, Chapman & Hall/CRC, New York, NY, USA, 2nd edition, 2004.
- [9] Y. Pawitan, *In All Likelihood: Statistical Modelling and Inference Using Likelihood*, Clarendon Press, New York, NY, USA, 2001.
- [10] D. J. Daley and D. Vere-Jones, *An Introduction to the Theory of Point Processes*, Springer, New York, NY, USA, 2nd edition, 2003.
- [11] E. N. Brown, R. Barbieri, U. T. Eden, and L. M. Frank, “Likelihood methods for neural data analysis,” in *Computational Neuroscience: A Comprehensive Approach*, J. Feng, Ed., pp. 253–286, CRC Press, New York, NY, USA, 2003.
- [12] E. N. Brown, “Theory of point processes for neural systems,” in *Methods and Models in Neurophysics*, C. C. Chow, B. Gutkin, D. Hansel et al., Eds., pp. 691–727, Elsevier, San Diego, Calif, USA, 2005.
- [13] Z. Chen, R. Barbieri, and E. N. Brown, “State-space modeling of neural spike train and behavioral data,” in *Statistical Signal Processing for Neuroscience and Neurotechnology*, K. Oweiss, Ed., pp. 161–200, Elsevier, San Diego, Calif, USA, 2010.
- [14] W. Truccolo, U. T. Eden, M. R. Fellows, J. P. Donoghue, and E. N. Brown, “A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects,” *Journal of Neurophysiology*, vol. 93, no. 2, pp. 1074–1089, 2005.
- [15] P. McCullagh and A. Nelder, *Generalized Linear Models*, vol. 22 of *Computational Intelligence and Neuroscience*, Chapman & Hall/CRC Press, New York, NY, USA, 2nd edition, 1989.
- [16] E. Schneidman, M. J. Berry II, R. Segev, and W. Bialek, “Weak pairwise correlations imply strongly correlated network states in a neural population,” *Nature*, vol. 440, no. 7087, pp. 1007–1012, 2006.
- [17] H. Nasser, O. Marre, and B. Cessac, “Spatio-temporal spike train analysis for large scale networks using the maximum entropy principle and Monte Carlo method,” *Journal of Statistical Mechanics*, vol. 2013, Article ID P03006, 2013.
- [18] E. N. Brown, R. Barbieri, V. Ventura, R. E. Kass, and L. M. Frank, “The time-rescaling theorem and its application to neural spike train data analysis,” *Neural Computation*, vol. 14, no. 2, pp. 325–346, 2002.
- [19] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte, “A new method for the nonlinear transformation of means and covariances in filters and estimators,” *IEEE Transactions on Automatic Control*, vol. 45, no. 3, pp. 477–482, 2000.
- [20] S. Särkkä, “On unscented Kalman filtering for state estimation of continuous-time nonlinear systems,” *IEEE Transactions on Automatic Control*, vol. 52, no. 9, pp. 1631–1641, 2007.
- [21] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, “Introduction to variational methods for graphical models,” *Machine Learning*, vol. 37, no. 2, pp. 183–233, 1999.
- [22] H. Attias, “A variational Bayesian framework for graphical models,” in *Advances in Neural Information Processing Systems (NIPS) 12*, S. A. Solla, T. K. Leen, and K. R. Müller, Eds., MIT Press, Boston, Mass, USA, 2000.

- [23] M. Beal and Z. Ghahramani, "Variational Bayesian learning of directed graphical models," *Bayesian Analysis*, vol. 1, no. 4, pp. 793–832, 2006.
- [24] D. J. MacKay, *Information Theory, Inference, and Learning Algorithms*, Cambridge University Press, New York, NY, USA, 2003.
- [25] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, New York, NY, USA, 2006.
- [26] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*, MIT Press, Cambridge, Mass, USA, 2012.
- [27] D. Barber, *Bayesian Reasoning and Machine Learning*, Cambridge University Press, New York, NY, USA, 2012.
- [28] D. Barber, A. T. Cemgil, and S. Chiappa, *Bayesian Time Series Models*, Cambridge University Press, New York, NY, USA, 2011.
- [29] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, John Wiley & Sons, New York, NY, USA, 2nd edition, 2006.
- [30] A. Dempster, N. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society B*, vol. 39, pp. 1–38, 1977.
- [31] K. Katahira, K. Watanabe, and M. Okada, "Deterministic annealing variant of variational Bayes method," *Journal of Physics*, vol. 95, no. 1, Article ID 012015, 2008.
- [32] K. Kurihara and M. Welling, "Bayesian k -means as a "maximization-expectation" algorithm," *Neural Computation*, vol. 21, no. 4, pp. 1145–1172, 2009.
- [33] J. Sung, Z. Ghahramani, and S.-Y. Bang, "Latent-space variational bayes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 12, pp. 2236–2242, 2008.
- [34] J. Sung, Z. Ghahramani, and S.-Y. Bang, "Second-order latent-space variational bayes for approximate bayesian inference," *IEEE Signal Processing Letters*, vol. 15, pp. 918–921, 2008.
- [35] R. E. Turner and M. Sahani, "Two problems with variational expectation maximisation for time series models," in *Bayesian Time Series Models*, D. Barber, A. T. Cemgil, and S. Chiappa, Eds., pp. 115–138, Cambridge University Press, New York, NY, USA, 2011.
- [36] K. Watanabe, "An alternative view of variational Bayes and asymptotic approximations of free energy," *Machine Learning*, vol. 86, no. 2, pp. 273–293, 2012.
- [37] A. Honkela, T. Raiko, M. Kuusela, M. Tornio, and J. Karhunen, "Approximate riemannian conjugate gradient learning for fixed-form variational bayes," *Journal of Machine Learning Research*, vol. 11, pp. 3235–3268, 2010.
- [38] T. P. Minka, *A family of algorithms for approximate Bayesian inference [Ph.D. thesis]*, Department of EECS, Massachusetts Institute of Technology, Cambridge, Mass, USA, 2001.
- [39] S.-I. Amari and H. Nagaoka, *Methods of Information Geometry*, Oxford University Press, New York, NY, USA, 2007.
- [40] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, *Markov Chain Monte Carlo in Practice*, Chapman & Hall/CRC, New York, NY, USA, 1995.
- [41] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*, Springer, New York, NY, USA, 2nd edition, 2004.
- [42] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.
- [43] W. K. Hastings, "Monte carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.
- [44] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 721–741, 1984.
- [45] R. M. Neal, "Suppressing random walks in Markov chain Monte Carlo using ordered overrelaxation," Tech. Rep. 9508, University of Toronto; Department of Statistics, 1995.
- [46] T. Marshall and G. Roberts, "An adaptive approach to Langevin MCMC," *Statistics and Computing*, vol. 22, no. 5, pp. 1041–1057, 2012.
- [47] Y. Qi and T. P. Minka, "Hessian-based Markov chain Monte-Carlo algorithms," in *Proceedings of the 1st Cape Cod Workshop on Monte Carlo Methods*, Cape Cod, Mass, USA, September 2002.
- [48] P. J. Green, "Reversible jump Markov chain monte carlo computation and Bayesian model determination," *Biometrika*, vol. 82, no. 4, pp. 711–732, 1995.
- [49] R. E. Kass and A. E. Raftery, "Bayes factors," *Journal of the American Statistical Association*, vol. 90, no. 430, pp. 773–795, 1995.
- [50] M. Lavine and M. J. Schervish, "Bayes factors: what they are and what they are not," *American Statistician*, vol. 53, no. 2, pp. 119–122, 1999.
- [51] S. M. Lewis and A. E. Raftery, "Estimating Bayes factors via posterior simulation with the Laplace-Metropolis estimator," *Journal of the American Statistical Association*, vol. 92, no. 438, pp. 648–655, 1997.
- [52] T. Toni and M. P. H. Stumpf, "Simulation-based model selection for dynamical systems in systems and population biology," *Bioinformatics*, vol. 26, no. 1, pp. 104–110, 2009.
- [53] R. M. Neal, *Bayesian Learning for Neural Networks*, Springer, New York, NY, USA, 1996.
- [54] J. A. Hoeting, D. Madigan, A. E. Raftery, and C. T. Volinsky, "Bayesian model averaging: a tutorial," *Statistical Science*, vol. 14, no. 4, pp. 382–417, 1999.
- [55] A. E. Raftery, "Approximate Bayes factors and accounting for model uncertainty in generalised linear models," *Biometrika*, vol. 83, no. 2, pp. 251–266, 1996.
- [56] Z. Chen and E. N. Brown, "State space model," *Scholarpedia*, vol. 8, no. 3, Article ID 30868, 2013.
- [57] L. Paninski, Y. Ahmadian, D. G. Ferreira et al., "A new look at state-space models for neural data," *Journal of Computational Neuroscience*, vol. 29, no. 1-2, pp. 107–126, 2010.
- [58] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, New York, NY, USA, 4th edition, 2002.
- [59] C. P. Robert, T. Rydén, and D. M. Titterton, "Bayesian inference in hidden Markov models through the reversible jump Markov chain Monte Carlo method," *Journal of the Royal Statistical Society B*, vol. 62, no. 1, pp. 57–75, 2000.
- [60] S. L. Scott, "Bayesian methods for hidden Markov models: recursive computing in the 21st century," *Journal of the American Statistical Association*, vol. 97, no. 457, pp. 337–351, 2002.
- [61] Z. Ghahramani, "Learning dynamic Bayesian networks," in *Adaptive Processing of Sequences and Data Structures*, C. L. Giles and M. Gori, Eds., pp. 168–197, Springer, New York, NY, USA, 1998.
- [62] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME*, vol. 82, pp. 35–45, 1960.
- [63] W. Wu, Y. Gao, E. Bienenstock, J. P. Donoghue, and M. J. Black, "Bayesian population decoding of motor cortical activity using a Kalman filter," *Neural Computation*, vol. 18, no. 1, pp. 80–118, 2006.

- [64] W. Wu, J. E. Kulkarni, N. G. Hatsopoulos, and L. Paninski, "Neural decoding of hand motion using a linear state-space model with hidden states," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 17, no. 4, pp. 370–378, 2009.
- [65] E. N. Brown, L. M. Frank, D. Tang, M. C. Quirk, and M. A. Wilson, "A statistical paradigm for neural spike train decoding applied to position prediction from ensemble firing patterns of rat hippocampal place cells," *Journal of Neuroscience*, vol. 18, no. 18, pp. 7411–7425, 1998.
- [66] A. C. Smith and E. N. Brown, "Estimating a state-space model from point process observations," *Neural Computation*, vol. 15, no. 5, pp. 965–991, 2003.
- [67] U. T. Eden, L. M. Frank, R. Barbieri, V. Solo, and E. N. Brown, "Dynamic analysis of neural encoding by point process adaptive filtering," *Neural Computation*, vol. 16, no. 5, pp. 971–998, 2004.
- [68] S. Koyama, L. Castellanos Pérez-Bolde, C. Rohilla Shalizi, and R. E. Kass, "Approximate methods for state-space models," *Journal of the American Statistical Association*, vol. 105, no. 489, pp. 170–180, 2010.
- [69] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*, Springer, New York, NY, USA, 2001.
- [70] A. E. Brockwell, A. L. Rojas, and R. E. Kass, "Recursive Bayesian decoding of motor cortical signals by particle filtering," *Journal of Neurophysiology*, vol. 91, no. 4, pp. 1899–1907, 2004.
- [71] A. Ergun, R. Barbieri, U. T. Eden, M. A. Wilson, and E. N. Brown, "Construction of point process adaptive filter algorithms for neural system using sequential Monte Carlo methods," *IEEE Transactions on Biomedical Engineering*, vol. 54, pp. 419–428, 2007.
- [72] V. Šmídl and A. Quinn, "Variational Bayesian filtering," *IEEE Transactions on Signal Processing*, vol. 56, no. 10, pp. 5020–5030, 2008.
- [73] Y. Salimpour, H. Soltanian-Zadeh, S. Salehi, N. Emadi, and M. Abouzari, "Neuronal spike train analysis in likelihood space," *PLoS ONE*, vol. 6, no. 6, Article ID e21256, 2011.
- [74] N. L. Hjort, C. Holmes, P. Müller, and S. G. Walker, *Bayesian Nonparametrics*, Cambridge University Press, New York, NY, USA, 2010.
- [75] Z. Ghahramani, "Bayesian nonparametrics and the probabilistic approach to modeling," *Philosophical Transactions on Royal Society of London A*, vol. 371, Article ID 20110553, 2012.
- [76] E. Fox, E. Sudderth, M. Jordan, and A. Willsky, "Bayesian nonparametric methods for learning markov switching processes," *IEEE Signal Processing Magazine*, vol. 27, no. 6, pp. 43–54, 2010.
- [77] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, MIT Press, Cambridge, Mass, USA, 2005.
- [78] J. Van Gael, Y. Saatchi, Y. W. Teh, and Z. Ghahramani, "Beam sampling for the infinite hidden Markov model," in *25th International Conference on Machine Learning*, pp. 1088–1095, fin, July 2008.
- [79] F. Gabbiani and C. Koch, "Principles of spike train analysis," in *Methods in Neuronal Modeling: From Synapses to Networks*, C. Koch and I. Segev, Eds., pp. 313–360, MIT Press, Boston, Mass, USA, 2nd edition, 1998.
- [80] R. E. Kass, V. Ventura, and E. N. Brown, "Statistical issues in the analysis of neuronal data," *Journal of Neurophysiology*, vol. 94, no. 1, pp. 8–25, 2005.
- [81] J. S. Prentice, J. Homann, K. D. Simmons, G. Tkačik, V. Balasubramanian, and P. C. Nelson, "Fast, scalable, bayesian spike identification for Multi-Electrode arrays," *PLoS ONE*, vol. 6, no. 7, Article ID e19884, 2011.
- [82] F. Wood, M. J. Black, C. Vargas-Irwin, M. Fellows, and J. P. Donoghue, "On the variability of manual spike sorting," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 6, pp. 912–918, 2004.
- [83] C. Ekanadham, D. Tranchina, and E. P. Simoncelli, "A blind deconvolution method for neural spike identification," in *Proceedings of the 25th Annual Conference on Neural Information Processing Systems (NIPS '11)*, vol. 23, MIT Press, December 2011.
- [84] M. S. Lewicki, "A review of methods for spike sorting: the detection and classification of neural action potentials," *Network*, vol. 9, no. 4, pp. R53–R78, 1998.
- [85] D. P. Nguyen, L. M. Frank, and E. N. Brown, "An application of reversible-jump Markov chain Monte Carlo to spike classification of multi-unit extracellular recordings," *Network*, vol. 14, no. 1, pp. 61–82, 2003.
- [86] F. Wood and M. J. Black, "A nonparametric Bayesian alternative to spike sorting," *Journal of Neuroscience Methods*, vol. 173, no. 1, pp. 1–12, 2008.
- [87] J. A. Herbst, S. Gammeter, D. Ferrero, and R. H. R. Hahnloser, "Spike sorting with hidden Markov models," *Journal of Neuroscience Methods*, vol. 174, no. 1, pp. 126–134, 2008.
- [88] A. Calabrese and L. Paninski, "Kalman filter mixture model for spike sorting of non-stationary data," *Journal of Neuroscience Methods*, vol. 196, no. 1, pp. 159–169, 2011.
- [89] V. Ventura, "Automatic spike sorting using tuning information," *Neural Computation*, vol. 21, no. 9, pp. 2466–2501, 2009.
- [90] V. Ventura, "Traditional waveform based spike sorting yields biased rate code estimates," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 106, no. 17, pp. 6921–6926, 2009.
- [91] M. Park and J. W. Pillow, "Receptive field inference with localized priors," *PLoS Computational Biology*, vol. 7, no. 10, Article ID e1002219, 2011.
- [92] I. M. Park and J. W. Pillow, "Bayesian spike-triggered covariance analysis," in *Advances in Neural Information Processing Systems (NIPS)*, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Fereira, and K. Q. Weinberger, Eds., vol. 24, pp. 1692–1700, MIT Press, Boston, Mass, USA, 2011.
- [93] D. Endres and M. Oram, "Feature extraction from spike trains with Bayesian binning: 'Latency is where the signal starts,'" *Journal of Computational Neuroscience*, vol. 29, no. 1-2, pp. 149–169, 2010.
- [94] I. Dimatteo, C. R. Genovese, and R. E. Kass, "Bayesian curve-fitting with free-knot splines," *Biometrika*, vol. 88, no. 4, pp. 1055–1071, 2001.
- [95] A. C. Smith, J. D. Scalton, S. Wirth, M. Yanike, W. A. Suzuki, and E. N. Brown, "State-space algorithms for estimating spike rate functions," *Computational Intelligence and Neuroscience*, vol. 2010, Article ID 426539, 2010.
- [96] B. Cronin, I. H. Stevenson, M. Sur, and K. P. Körding, "Hierarchical bayesian modeling and Markov chain Monte Carlo sampling for tuning-curve analysis," *Journal of Neurophysiology*, vol. 103, no. 1, pp. 591–602, 2010.
- [97] H. Taubman, E. Vaadia, R. Paz, and G. Chechik, "A Bayesian approach for characterizing direction tuning curves in the supplementary motor area of behaving monkeys," *Journal of Neurophysiology*, 2013.
- [98] L. Paninski, J. Pillow, and J. Lewi, "Statistical models for neural encoding, decoding, and optimal stimulus design," in *Computational Neuroscience: Theoretical Insights Into Brain Function*, P. Cisek, T. Drew, and J. Kalaska, Eds., Elsevier, 2007.

- [99] S. Gerwin, J. H. Macke, M. Seeger, and M. Bethge, "Bayesian inference for spiking neuron models with a sparsity prior," in *Advances in Neural Information Processing Systems (NIPS)*, J. C. Platt, D. Koller, Y. Singer, and S. Roweis, Eds., vol. 20, pp. 529–536, MIT Press, Boston, Mass, USA, 2008.
- [100] J. W. Pillow and J. G. Scott, "Fully Bayesian inference for neural models with negative-binomial spiking," in *Advances in Neural Information Processing Systems (NIPS)*, P. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., vol. 25, pp. 1907–1915, MIT Press, Boston, Mass, USA, 2012.
- [101] S. Koyama, U. T. Eden, E. N. Brown, and R. E. Kass, "Bayesian decoding of neural spike trains," *Annals of the Institute of Statistical Mathematics*, vol. 62, no. 1, pp. 37–59, 2010.
- [102] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, New York, NY, USA, 2004.
- [103] J. W. Pillow, Y. Ahmadian, and L. Paninski, "Model-based decoding, information estimation, and change-point detection techniques for multineuron spike trains," *Neural Computation*, vol. 23, no. 1, pp. 1–45, 2011.
- [104] A. D. Ramirez, Y. Ahmadian, J. Schumacher, D. Schneider, S. M. N. Woolley, and L. Paninski, "Incorporating naturalistic correlation structure improves spectrogram reconstruction from neuronal activity in the songbird auditory midbrain," *Journal of Neuroscience*, vol. 31, no. 10, pp. 3828–3842, 2011.
- [105] Z. Chen, K. Takahashi, and N. G. Hatsopoulos, "Sparse Bayesian inference methods for decoding 3D reach and grasp kinematics and joint angles with primary motor cortical ensembles," in *Proceedings of the 35th Annual International Conference of the IEEE Engineering in Medicine and Biology (EMBC '13)*, pp. 5930–5933, 2013.
- [106] K. Zhang, I. Ginzburg, B. L. McNaughton, and T. J. Sejnowski, "Interpreting neuronal population activity by reconstruction: unified framework with application to hippocampal place cells," *Journal of Neurophysiology*, vol. 79, no. 2, pp. 1017–1044, 1998.
- [107] W. Truccolo, G. M. Fries, J. P. Donoghue, and L. R. Hochberg, "Primary motor cortex tuning to intended movement kinematics in humans with tetraplegia," *Journal of Neuroscience*, vol. 28, no. 5, pp. 1163–1178, 2008.
- [108] W. Truccolo and J. P. Donoghue, "Nonparametric modeling of neural point processes via stochastic gradient boosting regression," *Neural Computation*, vol. 19, no. 3, pp. 672–705, 2007.
- [109] T. P. Coleman and S. S. Sarma, "A computationally efficient method for nonparametric modeling of neural spiking activity with point processes," *Neural Computation*, vol. 22, no. 8, pp. 2002–2030, 2010.
- [110] M. M. Shanechi, E. N. Brown, and Z. M. Williams, "Neural population partitioning and a concurrent brain-machine interface for sequential control motor function," *Nature Neuroscience*, vol. 12, pp. 1715–1722, 2012.
- [111] M. M. Shanechi, G. W. Wornell, Z. Williams, and E. N. Brown, "A parallel point-process filter for estimation of goal-directed movements from neural signals," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '10)*, pp. 521–524, Dallas, Tex, USA, March 2010.
- [112] Y. Ahmadian, J. W. Pillow, and L. Paninski, "Efficient Markov chain monte carlo methods for decoding neural spike trains," *Neural Computation*, vol. 23, no. 1, pp. 46–96, 2011.
- [113] A. K. Bansal, W. Truccolo, C. E. Vargas-Irwin, and J. P. Donoghue, "Decoding 3D reach and grasp from hybrid signals in motor and premotor cortices: spikes, multiunit activity, and local field potentials," *Journal of Neurophysiology*, vol. 107, no. 5, pp. 1337–1355, 2012.
- [114] V. Ventura, "Spike train decoding without spike sorting," *Neural Computation*, vol. 20, no. 4, pp. 923–963, 2008.
- [115] Z. Chen, F. Kloosterman, S. Layton, and W. A. Wilson, "Transductive neural decoding of unsorted neuronal spikes of rat hippocampus," in *Proceedings of the 34th Annual International Conference of the IEEE Engineering in Medicine and Biology (EMBC '12)*, pp. 1310–1313, August 2012.
- [116] F. Kloosterman, S. Layton, Z. Chen, and M. A. Wilson, "Bayesian decoding of unsorted spikes in the rat hippocampus," *Journal of Neurophysiology*, 2013.
- [117] A. L. Jacobs, G. Fridman, R. M. Douglas et al., "Ruling out and ruling in neural codes," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 106, no. 14, pp. 5936–5941, 2009.
- [118] D. H. Johnson, "Information theory and neural information processing," *IEEE Transactions on Information Theory*, vol. 56, no. 2, pp. 653–666, 2010.
- [119] C. Smith and L. Paninski, "Computing loss of efficiency in optimal Bayesian decoders given noisy or incomplete spike trains," *Network*, vol. 24, no. 2, pp. 75–98, 2013.
- [120] D. S. Greenberg, A. R. Houweling, and J. N. D. Kerr, "Population imaging of ongoing neuronal activity in the visual cortex of awake rats," *Nature Neuroscience*, vol. 11, no. 7, pp. 749–751, 2008.
- [121] J. T. Vogelstein, B. O. Watson, A. M. Packer, R. Yuste, B. Jedynek, and L. Paninski, "Spike inference from calcium imaging using sequential Monte Carlo methods," *Biophysical Journal*, vol. 97, no. 2, pp. 636–655, 2009.
- [122] J. T. Vogelstein, A. M. Packer, T. A. Machado et al., "Fast nonnegative deconvolution for spike train inference from population calcium imaging," *Journal of Neurophysiology*, vol. 104, no. 6, pp. 3691–3704, 2010.
- [123] C. Andrieu, E. Barat, and A. Doucet, "Bayesian deconvolution of noisy filtered point processes," *IEEE Transactions on Signal Processing*, vol. 49, no. 1, pp. 134–146, 2001.
- [124] J. Oñativia, S. R. Schultz, and P. L. Dragotti, "A finite rate of innovation algorithm for fast and accurate spike detection from two-photon calcium imaging," *Journal of Neural Engineering*, vol. 10, Article ID 046017, 2013.
- [125] J. W. Pillow, J. Shlens, L. Paninski et al., "Spatio-temporal correlations and visual signalling in a complete neuronal population," *Nature*, vol. 454, no. 7207, pp. 995–999, 2008.
- [126] W. Truccolo, L. R. Hochberg, and J. P. Donoghue, "Collective dynamics in human and monkey sensorimotor cortex: predicting single neuron spikes," *Nature Neuroscience*, vol. 13, no. 1, pp. 105–111, 2010.
- [127] E. S. Chornoboy, L. P. Schramm, and A. F. Karr, "Maximum likelihood identification of neural point process systems," *Biological Cybernetics*, vol. 59, no. 4–5, pp. 265–275, 1988.
- [128] M. Okatan, M. A. Wilson, and E. N. Brown, "Analyzing functional connectivity using a network likelihood model of ensemble neural spiking activity," *Neural Computation*, vol. 17, no. 9, pp. 1927–1961, 2005.
- [129] F. Rigat, M. de Gunst, and J. van Pelt, "Bayesian modelling and analysis of spatio-temporal neuronal networks," *Bayesian Analysis*, vol. 1, no. 4, pp. 733–764, 2006.
- [130] I. H. Stevenson, J. M. Rebesco, N. G. Hatsopoulos, Z. Haga, L. E. Miller, and K. P. Kording, "Bayesian inference of functional connectivity and network structure from spikes," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 17, no. 3, pp. 203–213, 2009.

- [131] Z. Chen, D. F. Putrino, S. Ghosh, R. Barbieri, and E. N. Brown, "Statistical inference for assessing functional connectivity of neuronal ensembles with sparse spiking data," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 19, no. 2, pp. 121–135, 2011.
- [132] S. Eldawlatly, Y. Zhou, R. Jin, and K. G. Oweiss, "On the use of dynamic Bayesian networks in reconstructing functional neuronal networks from spike train ensembles," *Neural Computation*, vol. 22, no. 1, pp. 158–189, 2010.
- [133] Y. Mishchenko, J. Vogelstein, and L. Paninski, "A Bayesian approach for inferring neuronal connectivity from calcium uorescent imaging data," *Annals of Applied Statistics*, vol. 5, pp. 1229–1261, 2011.
- [134] L. Martignon, G. Deco, K. Laskey, M. Diamond, W. Freiwald, and E. Vaadia, "Neural coding: higher-order temporal patterns in the neurostatistics of cell assemblies," *Neural Computation*, vol. 12, no. 11, pp. 2621–2653, 2000.
- [135] H. Shimazaki, S. Amari, E. N. Brown, and S. Gruen, "State-space analysis of time-varying higherorder spike correlation for multiple neural spike train data," *PLoS Computational Biology*, vol. 8, no. 3, Article ID e1002385, 2012.
- [136] B. M. Turner, B. U. Forstmann, E.-J. Wagenmakers, S. D. Brown, P. B. Sederberg, and M. Steyvers, "A Bayesian framework for simultaneously modeling neural and behavioral data," *NeuroImage*, vol. 72, pp. 193–206, 2013.
- [137] J. W. Pillow and P. Latham, "Neural characterization in partially observed populations of spiking neurons," in *Advances in Neural Information Processing Systems (NIPS)*, J. C. Platt, D. Koller, Y. Singer, and S. Roweis, Eds., vol. 20, pp. 1161–1168, MIT Press, Boston, Mass, USA, 2008.
- [138] L. Li, I. M. Park, S. Seth, J. C. Sanchez, and J. C. Principe, "Functional connectivity dynamics among cortical neurons: a dependence analysis," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 20, no. 1, pp. 18–30, 2012.
- [139] R. E. Kass, R. C. Kelly, and W.-L. Loh, "Assessment of synchrony in multiple neural spike trains using loglinear point process models," *The Annals of Applied Statistics*, vol. 5, no. 2B, pp. 1262–1292, 2011.
- [140] S. Kim, D. Putrino, S. Ghosh, and E. N. Brown, "A Granger causality measure for point process models of ensemble neural spiking activity," *PLoS Computational Biology*, vol. 7, no. 3, Article ID e1001110, 2011.
- [141] R. Vicente, M. Wibral, M. Lindner, and G. Pipa, "Transfer entropy—a model-free measure of effective connectivity for the neurosciences," *Journal of Computational Neuroscience*, vol. 30, no. 1, pp. 45–67, 2011.
- [142] P. Berkes, F. Wood, and J. Pillow, "Characterizing neural dependencies with copula models," in *Advances in Neural Information Processing Systems (NIPS)*, J. C. Platt, D. Koller, Y. Singer, and S. Roweis, Eds., vol. 20, MIT Press, Boston, Mass, USA, 2008.
- [143] M. S. Smith, "Bayesian approaches to copula modelling," in *Bayesian Theory and Applications*, P. Damien, P. Dellaportas, N. Polson, and D. Stephens, Eds., Oxford University Press, New York, NY, USA, 2013.
- [144] B. M. Yu, J. P. Cunningham, G. Santhanam, S. I. Ryu, K. V. Shenoy, and M. Sahani, "Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity," *Journal of Neurophysiology*, vol. 102, no. 1, pp. 614–635, 2009.
- [145] Z. Chen, F. Kloosterman, E. N. Brown, and M. A. Wilson, "Uncovering spatial topology represented by rat hippocampal population neuronal codes," *Journal of Computational Neuroscience*, vol. 33, no. 2, pp. 227–255, 2012.
- [146] Z. Chen, S. N. Gomperts, J. Yamamoto, and W. A. Wilson, "Neural representation of spatial topology in the rodent hippocampus," *Neural Computation*, vol. 26, no. 1, pp. 1–39, 2014.
- [147] Z. Chen and M. A. Wilson, "A variational nonparametric Bayesian approach for inferring rat hippocampal population codes," in *Proceedings of the 35th Annual International Conference of the IEEE Engineering in Medicine and Biology (EMBC '13)*, pp. 7092–7095, 2013.
- [148] K. Famm, B. Litt, K. J. Tracey, E. S. Boyden, and M. Slaoui, "A jump-start for electroceuticals," *Nature*, vol. 496, pp. 159–161, 2013.