Use of Bayesian Filtering and Adaptive Learning Methods to Improve the Detection and Estimation of Pathological and Neurological Disorders

by

Alexander Joseph Maurer

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved June 2016 by the
Graduate Supervisory Committee:

Antonia Papandreou-Suppappola, Chair
Daniel Bliss
Chaitali Chakrabarti
Narayan Kovvali

ARIZONA STATE UNIVERSITY

August 2016

ABSTRACT

Biological and biomedical measurements, when adequately analyzed and processed, can be used to impart quantitative diagnosis during primary health care consultation to improve patient adherence to recommended treatments. For example, analyzing neural recordings from neurostimulators implanted in patients with neurological disorders can be used by a physician to adjust detrimental stimulation parameters to improve treatment. As another example, biosequences, such as sequences from peptide microarrays obtained from a biological sample, can potentially provide presymptomatic diagnosis for infectious diseases when processed to associate antibodies to specific pathogens or infectious agents. This work proposes advanced statistical signal processing and machine learning methodologies to assess neurostimulation from neural recordings and to extract diagnostic information from biosequences.

For locating specific cognitive and behavioral information in different regions of the brain, neural recordings are processed using sequential Bayesian filtering methods to detect and estimate both the number of neural sources and their corresponding parameters. Time-frequency based feature selection algorithms are combined with adaptive machine learning approaches to suppress physiological and non-physiological artifacts present in neural recordings. Adaptive processing and unsupervised clustering methods applied to neural recordings are also used to suppress neurostimulation artifacts and classify between various behavior tasks to assess the level of neurostimulation in patients.

For pathogen detection and identification, random peptide sequences and their properties are first uniquely mapped to highly-localized signals and their corresponding parameters in the time-frequency plane. Time-frequency signal processing methods are then applied to estimate antigenic determinants or epitope candidates for detecting and identifying potential pathogens.

to my parents Roger and Vikki,

my sister Suzanne and my brother-in-law Christian,

my brother James and his fiancée Mini,

my brother Eric,

and also to the Ozarowicz family, Jerry, Sherri, Brian and Michael

ACKNOWLEDGMENTS

I would first like to thank God for my path in life passing through this university.

I would like to thank my advisor, Dr. Antonia Papandreou-Suppappola, for her advice, guidance, and focus on ensuring students have what they need. Her lab is a supportive environment to complete a graduate degree. I would also like to thank my committee, Dr. Daniel Bliss, Dr. Chaitali Chakrabarti, and Dr. Narayan Kovvali for their assistance and comments.

Next, I thank my family for their love, support, phone calls, and snacks in the mail over the years, as well as our family friend John Griffith for Thanksgiving dinners and football, driving me to the airport, and helping out when my family comes to visit.

I thank my friends Samantha Vélez, Erik Posch, Scott Spencer, and Ellie Hakanson for video games and listening to me talk about grad school.

Finally, I thank the Electrical Engineering department, my advisor and Dr. Palais for the Dean's Fellowship and teaching opportunities to support my Ph.D., as well as the ARCS Foundation for their Award and financial support.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

Chapter 1

INTRODUCTION

1.1 Neural Activity Processing

1.1.1 Neural Source Estimation and Challenges

Illnesses of the mind have historically been misunderstood. We are only recently starting to make real progress on understanding the mechanisms and treatment of these diseases. Historically, from trepanation to asylums to the misuse of electroshock therapy, it has been a troubling path towards the modern day in working with the brain [5]. Even with an illness of the brain that shows effects primarily in the body like Parkinson's disease, the functional early treatments had significant risk and were fairly drastic, involving scarring areas of the brain to restore a degree of function [6, 7]. Thanks to new technology, we have been able to make progress since the early 1900s in understanding neurons and how behaviors arise [8–10]. Mapping the brain has been declared one of the "Grand Challenges of the Mind and Brain" by the National Science Foundation, and it is also a "Grand Challenge" according to the National Academy of Engineering [8]. Computing power keeps improving, and many new techniques and technologies can help to solve the problem of mapping the brain [11]. Parkinson's disease treatment has advanced through medications up to deep brain stimulation, which is a focus of this research, along with neural activity localization.

The various scans and measurement techniques available have trade offs, often between time resolution and space resolution [11]. Methods that look at blood flow around neurons, like positron emission tomography (PET), magnetic resonance imaging (MRI), and transcranial optics, have strong spacial resolution but poor time res-

olution. This is because the blood flow, which is the measure of activity for these methods, is a somewhat slow process. On the other hand, methods such as electroencephalograms (EEG) and magnetoencephalograms (MEG) have strong time resolution, as electromagnetic information changes quickly, but poor spacial resolution, since the fields around the entire head are affected by local activity [11–13].

The process of mapping the brain has already lead to advances in the understanding of brain function. The University of Chicago has worked on localizing seizure activity. They found that the orientation of the sources differentiated between seizures from different parts of the temporal cortex [14, 15]. Localized sources are good features to use in brain-machine interfaces (BMI), which, as their name implies, allow a level of communication between the brain and machines. This is useful for applications like virtual reality, synthetic limbs, and other machines interacting with the mind [16]. Researchers have examined source localizations in a variety of mental states, including depression compared to alcoholism [17] and anxiety versus relaxation [18]. Diseases of the brain are challenging to treat, and brain mapping is a great step toward being able to improve and discover treatments [11]. Mental disorders are actually fairly common, and 26.2% of American adults experience mental disorders every year [19]. Mental illnesses are the third most common disability, after hearing and vision problems. [20].

One of the goals of this work is to improve the spacial resolution of EEG by localizing the activity in the brain. This can be done by solving the inverse problem, going from measurements to neural sources. The source model we use is a localized dipole of current with a particular position and orientation. It is intended to approximate the action potential of neurons [2, 11, 21–23]. Many neurons working together form a localized current, resulting in volume currents and the propagation of electromag-

netic fields through the head, which can be measured with sensors. MEG measures the magnetic field and EEG measures the electric potential.

Different methods have been used to solve the inverse problem in literature [4, 11, 13, 24–37], including recursively applied projected multiple signal classification (RAP MUSIC) [13, 24], spatial filtering or beamforming [25], low resolution electromagnetic tomography (LORETA), and Bayesian methods, such as Kalman filtering in [28, 31] and particle filtering in [26, 29, 31]. Methods using the probability hypothesis density filter (PHDF) were considered in [36, 37] to estimate both the unknown number of neural dipole sources and their parameters for real EEG/MEG data, and the probabilistic data association filter was applied in [4].

Artifact signals, which are signals that show up in measurements of the brain, but do not originate from the brain, are important to account for [38–48]. They include physiologic artifacts like muscle movements, breathing and heartbeat, and non-physiologic artifacts like the signal leaking from the power line, electrode popping, sweat on the electrode, or movement of the patient. Artifacts corrupt the measurements of EEG and MEG, impeding analysis. Thus, various methods were used for artifact suppression including independent component analysis (ICA) followed by pattern recognition [39, 49], wavelets [45], regression techniques [41], and using principle component analysis along with minimum norm estimation [40]. Noise canceling filters were used in real time in [47, 48] and blind source separation in [46].

1.1.2 Neurostimulation

An important applicability of realizing relations between function and brain region is to assist in the therapy and monitoring of subjects with implantable neurostimulators. Neurostimulation is a neurosurgical procedure that modifies the brains electrical activity to provide potential treatments for a large spectrum of neurological disor-

ders, such as Parkinson's disease, essential tremor, and most recently, depression and obsessive-compulsive disorder [50]. Neurostimulation modulates local field potential (LFP) oscillations in the deep brain nuclei and affects cortical and subcortical connections, key to decision-making, learning and cognitive association [51, 52]. The complexity of these networks and the placement of the neurostimulation results in a variety of individual specific effects in behavior and cognitive function. These are also dynamic networks, which cause side effects to vary over time. To advance neurostimulation and reduce its side effects, it is crucial to understand inter-subject variability from the neural signals. However, there currently does not exist a method to measure the impact of neurostimulation on cognitive functionality.

Parkinson's disease is a degenerative age-related brain disorder caused by the death of neurons responsible for contributing to dopamine production, resulting in tremors and muscle control difficulties. While there are effective medications for managing the symptoms, deep brain stimulation (DBS) is an option in certain cases, including when medications have significant side effects for a certain patient [6, 7, 53, 54]. DBS is a reversible surgical technique that uses implanted electrodes. It has replaced permanent techniques, surgeries called pallidotomies and thalamotomies, that remove small areas of the brain [7, 55]. The electrodes are used to apply an electrical waveform deep into the brain, similar to a pacemaker for the heart [6, 53, 54, 56]. An important research direction is detecting when the parameters of the stimulation waveform need to change to improve the efficacy of the treatment.

EEG as well as other measurements can be used to monitor brain activity in Parkinson's disease patients, as well as understand what happens in the brain when patients perform activities that can aggravate Parkinson's disease symptoms. We would like to use this monitoring to adjust the DBS parameters and improve the quality of disorder treatment. This will be achieved by using EEG to detect when the

stimulation parameters need adjustment due to changes in the symptoms of Parkinson's disease, to improve upon the current open loop DBS system. Our work toward this end is to classify segments of EEG of patients performing different tasks without DBS, as well as using specific filter methods to remove the DBS artifact and classify the tasks even when the DBS is active. Differences in the beta and gamma bands of neural activity can be seen in Parkinson's disease patients, and can be measured with EEG and used for classification of patient activity. The classification can be used for monitoring how the DBS treatment is working and if adjustments need to be made [16, 28, 57–67].

When drug therapy is no longer sufficient, DBS can alleviate motor symptoms by targeting the subthalamicnucleus (STN) using high-frequency electrical stimulation [6, 53, 54, 56]. Similar to the majority of neurostimulation systems, clinicians are able to use one or a configuration of multiple electrodes to apply electrical stimulation to a small target area. Efficacy of DBS relies on the accuracy of placement of the DBS lead at the target area, which is typically on the order of 5 mm in diameter. DBS leads are inserted into the target area with the guidance of a stereotactic frame while patients are awake, allowing neurophysiological recording of brain areas and intraoperative monitoring of electrical stimulation side effects and clinical efficacy. Note, however, that DBS also has negative side effects in certain people such as impairments in cognitive function [54, 68]. Studies have also demonstrated that DBS impairs verbal fluency [69–72] and reactive inhibition [73, 74].

Current methods for removing DBS artifact in EEG include local curve fitting [75, 76], empirical mode decomposition (EMD) detrending followed by time-frequency filtering [77], undersampling and interpolation [78], Hampel filtering [79], and matched filtering [80].

1.2 Proposed Work for Artifact Suppression in Source Estimation and Neurostimulation

Our approach to localizing neural activity while suppressing artifacts is the use of a particle filter implementation of the probabilistic data association filter (PDAF) [3, 4, 81–84]. The PDAF is designed to take in measurements of uncertain origin, which include source measurements and artifact measurements, where the sources and artifacts have different distributions and average densities in the measurement space. It requires the number of sources to be known a priori, accounting for all possible association possibilities between measurements and sources [3, 4, 81–84]. We propose the use of the PDAF to suppress artifact activity, such as eye blinks and movements, while tracking neural activity. We integrate the PDAF with ICA that is used to obtain separated measurements that lack association to sources, requiring a method such as the PDAF to determine which measurements belong to artifacts or sources and which source the measurement is associated with. The measurements are not preprocessed after ICA other than performing a transformation of the data to allow the PDAF to discriminate between the source model and artifact model, unlike methods that require preprocessing to reject measurements before tracking.

We also look at the probability hypothesis density filter (PHDF) [22, 23, 36, 37, 85–89] to track an unknown number of sources. It can be used to estimate the number of sources present as well as the parameter values for the sources by propagating the first moment of the source random finite set, the probability hypothesis density (PHD). The PHD has the property that when integrated over any region of the measurement space, it provides the expected number of sources in the region. The peaks of the PHD are where the sources are most likely to be [23, 37, 85].

We consider various methods to suppress DBS artifacts that can impede task processing and classification, including matched filters [80], Hampel filters [79], and the empirical mode decomposition [90–92]. We then concentrate on the beta and gamma frequency bands that DBS has been shown to affect [16, 28, 57–67], using the DBS artifact suppressed EEG in an adaptive learning clustering method to discriminate between behavioral tasks in Parkinson's disease patients implanted with neurostimulators. The ultimate goal is to find a biomarker for the Parkinson's disease treatment, in order to perform home monitoring using EEG. We perform classification using Gaussian mixture modeling (GMM) [93]. The advantage to this classifier is that the Dirichlet process Gaussian mixture model (DP-GMM) [94–102] does not need the number of tasks specified beforehand. It can also be used to adaptively classify the tasks when new EEG data becomes available. If the new data is classified as belonging to the same task, then it can be deduced that no changes have occurred with the patient's condition. If new classes are formed, then that could indicate that the DBS parameters need to be adjusted.

1.3 Thesis Organization

The rest of the thesis is organized as follows. In Chapter 2, we present the particle filter implementations of the PDAF and PHDF. In Chapter 3, we present the non-linear EEG model for neural sources, as well as a model for signals arising from artifacts. After proposing a transformation of the observed measurements to better separate sources from artifacts, we tie these models into the PDAF and PHDF equations to show how artifacts can be suppressed. We then present results showing artifact suppression while tracking neural sources. In Chapter 4, we further discuss Parkinson's disease and deep brain stimulation, then present the GMM and DP-GMM for classification using features from the matching pursuit decomposition. We show

results from using the DP-GMM to classify local field potential measurements from different tasks. In Chapter 5, we discuss the artifact signal that is created by the operation of the DBS unit, as well as methods to remove this artifact, including the Hampel filter, EMD, and sinusoidal matched filter. Finally, we present results showing the classification of EEG measurements for different taasks in Parkinson's patients with the DBS unit active and inactive. Chapter 6 contains conclusions and future work. Appendix A presents a previously published book chapter on work completed during this Ph.D., that is unrelated to the neural activity work in the main chapters.

Chapter 2

SEQUENTIAL BAYESIAN FILTERING METHODS FOR DYNAMIC PARAMETER ESTIMATION

2.1 The Particle Filter Algorithm

The particle filter (PF) is a technique used with non-linear, dynamic system models and non-Gaussian noise situations to estimate unknown time-varying parameters in noisy conditions. It uses measurements that are taken over time of a system whose underlying parameters are changing over time. The measurements or processed information received are not directly related to the unknown parameters of interest. The particle filter algorithm uses a system representation in terms of two models: a model that maps the noisy measurements into the parameter space to the measurement space, and a transition model that relates the current and past parameter values. When the models are linear and the measurement noise is Gaussian, the Kalman filter [103] can be employed, which performs the estimation in closed form. For non-linear models and non-Gaussian noise, the particle filter can be used instead of the Kalman filter. A common problem solved with Kalman or particle filters is the tracking of the positions and velocities of targets in a radar system. Another problem is the estimation of the location and orientation of neural activity given EEG/MEG recordings from sensors placed at various points on the head to measure electromagnetic fields. Using acceptable models that relate the physical unknown parameters, estimates of neural activity should give information about the status of a patient [104, 105].

We consider the general problem of estimating the dynamic state parameter vector x_k of a moving object using the noisy measurement vector z_k at time step k. The state space model has two equations, describing how the underlying parameters change over time and how the measurement is related to the underlying parameters. These equations are: [105].

$$\mathbf{x}_k = f_k(\mathbf{x}_{k-1}) + \mathbf{u}_{k-1}. \tag{2.1}$$

$$\mathbf{z}_k = h_k(\mathbf{x}_k) + \mathbf{v}_k. \tag{2.2}$$

In (2.1), $f_k(\mathbf{x}_{k-1})$ is the transition function of the parameters at time $k-1$, and $\mathbf{u}_{k-1}$ is a random process vector that represents possible errors in the function. Specifically, the function describes how the state at time $k-1$ transitions to the new state at time k, and the random process vector includes both the random portions of the model transitions and any error in the transition model. In (2.2), $\mathbf{z}_k$ is the measurement vector obtained from the system, and $h_k(\mathbf{x}_k)$ is the measurement function that relates the state parameter vector, $\mathbf{x}_k$, to the measurement vector, $\mathbf{z}_k$. The measurement noise vector $\mathbf{v}_k$ can include both the noise that comes from taking the measurement as well as any error in the measurement model [84, 105].

If $f_k(\cdot)$ and $h_k(\cdot)$ are linear and $\mathbf{u}_k$ and $\mathbf{v}_k$ are Gaussian, $\mathbf{x}_k$ can be estimated in closed form using the Kalman filter [104]. The extended Kalman filter [106] and the unscented Kalman filter [107] can be used for non-linear functions; however, when $f_k(\cdot)$ and $h_k(\cdot)$ are non-linear and $\mathbf{u}_{k-1}$ and $\mathbf{v}_k$ are non-Gaussian, the particle filter is a popular choice to find a sequential Monte Carlo estimate of the state parameters. [105].

The particle filter arises from taking a state model and measurement model for a system and looking to estimate the state from measurements taken of the system.

The goal is to obtain an estimate,$\hat{x}_k$ of the system state, $\hat{\mathbf{x}}_k$ at time step k, given a set of measurements, $\mathbf{Z}_{1:k} = \{\mathbf{z}_1,\ \mathbf{z}_2,\ \ldots\ \mathbf{z}_k\}$, up to time k. The minimum mean-squared error (MMSE) estimator of $\mathbf{x}_k$ can be given by

$$\hat{\mathbf{x}}_k = E[p(\mathbf{x}_k|\mathbf{Z}_{1:k})], \tag{2.3}$$

where $E[\cdot]$ denotes statistical expectation and $p(\mathbf{x}_k|\mathbf{Z}_{1:k})$ is the posterior distribution of the state. This estimator simply minimizes the error of the estimate by using the posterior distribution. In practice the posterior distribution is not available. The particle filter approximates the posterior distribution as a discrete distribution of samples or "particles." Particles in the state space form a discrete probability distribution as long as each particle has a weight, and the weights sum to one. With enough particles and a good way of finding weights, we can approximate the posterior distribution, $p(\mathbf{x}_k|\mathbf{Z}_{1:k})$.

There are two steps to implementing the particle filter algorithm: a state and particle update called the prediction step, and a weight update called the update step, which arises naturally from a few rules of conditional probability. The prediction step uses Equation (2.1) to predict the new state from the old state. This is done by updating each particle's location in the state space. The update step uses Equation (2.2) to update the weights for each particle. The process begins by drawing an initial set of particles in the state space from a distribution $p(\mathbf{x}_0)$. These steps alternate to estimate the state of the system at each time step, with one iteration of prediction and update per time step.

The prediction step uses the Chapman-Kolmogorov equation to account for the past state possibilities:

$$p(\mathbf{x}_k|\mathbf{Z}_{1:k-1}) = \int_{-\infty}^{\infty} p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{Z}_{1:k-1})\mathrm{d}\mathbf{x}_{k-1}. \tag{2.4}$$

In this equation, the newest measurement, $\mathbf{z}_k$, has not been used yet. The prediction step can be completed using the estimated posterior from the last update step, $p(\mathbf{x}_{k-1}|\mathbf{Z}_{1:k-1})$, and the transition probability density function (pdf), $p(\mathbf{x}_k|\mathbf{x}_{k-1})$, that arises from the state equation, Equation (2.1). Bayes' rule can then be used to complete the update step, and obtain the posterior density as

$$p(\mathbf{x}_k|\mathbf{Z}_{1:k}) = \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{Z}_{1:k-1})}{\int_{-\infty}^{\infty} p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{Z}_{1:k-1})}. \tag{2.5}$$

The integrals in (2.4) and (2.5) cannot be evaluated in closed form. Instead, let $\mathbf{x}_k^{(n)}$ and $w_k^{(n)}$, $n = 1, \ldots, N$ denote a set of N particles and associated weights. Then we obtain the approximate posterior:

$$p(\mathbf{x}_k|\mathbf{Z}_{1:k}) \approx \sum_{n=1}^{N} \mathbf{w}_k^{(n)} \delta\left(\mathbf{x}_k - \mathbf{x}_k^{(n)}\right). \tag{2.6}$$

More particles will yield a better estimate of the posterior pdf and thus a better estimate of the parameters $\mathbf{x}_k$. The state estimate can be found as above in (2.6) as

$$\hat{\mathbf{x}}_k = \int \mathbf{x}_k p(\mathbf{x}_k|\mathbf{Z}_{1:k}) d\mathbf{x}_k \approx \sum_{n=1}^{N} \mathbf{w}_k^{(n)} \mathbf{x}_k^{(n)}. \tag{2.7}$$

We use the sampling importance resampling particle filter (SIR PF) algorithm, since we cannot sample from the posterior pdf. This method uses an importance density $q(\mathbf{x}_k|\mathbf{Z}_{1:k})$ to sample from, which is ideally as close to the posterior as possible, but easily samplable [105]. It is not the true distribution, so we modify the samples we get from it. If $q(\mathbf{x}_k|\mathbf{Z}_{1:k}) = p(\mathbf{x}_k|\mathbf{Z}_{1:k})$, then we do not need to weight the samples to obtain samples from $p(\cdot|\mathbf{Z}_{1:k})$, as we are sampling from the true posterior. However, $q(\mathbf{x}_k|\mathbf{Z}_{1:k}) \neq p(\mathbf{x}_k|\mathbf{Z}_{1:k})$, so we need to weigh our samples. The weighted distribution can be written as:

$$\mathbf{w}_k^{(n)} \propto \frac{p(\mathbf{x}_k^{(n)}|\mathbf{Z}_{1:k})}{q(\mathbf{x}_k^{(n)}|\mathbf{Z}_{1:k})}, \tag{2.8}$$

where the weights sum to 1 over all n, in order to form a valid pdf. A powerful class of proposal distributions have the following property:
$q(\mathbf{x}_k|\mathbf{Z}_{1:k}) = q(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{z}_k)\, q(\mathbf{x}_{k-1}|\mathbf{Z}_{1:k-1})$. Then $\mathbf{x}_k^{(n)}$ can be sampled from
$q(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{z}_k)$. It can then be shown using Bayes' rule and the Chapman-Kolmogorov equation that (2.8) can be factored to obtain

$$\mathbf{w}_k^{(n)} \propto \frac{p(\mathbf{z}_k|\mathbf{x}_k^{(n)})p(\mathbf{x}_k^{(n)}|\mathbf{x}_{k-1}^{(n)})p(\mathbf{x}_{k-1}^{(n)}|\mathbf{Z}_{1:k-1})}{q(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{z}_k)q(\mathbf{x}_{k-1}|\mathbf{Z}_{1:k-1})}. \tag{2.9}$$

Then (2.8) and (2.9) give the result that

$$\mathbf{w}_k^{(n)} \propto \mathbf{w}_{k-1}^{(n)} \frac{p(\mathbf{z}_k|\mathbf{x}_k^{(n)})p(\mathbf{x}_k^{(n)}|\mathbf{x}_{k-1}^{(n)})}{q(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{z}_k)}. \tag{2.10}$$

A simple choice for a known distribution with this property is the distribution derived from the state update equation, Equation (2.1). This results in the prediction of $\mathbf{x}_k$ only depending on $\mathbf{x}_{k-1}$. The proposal distribution is then, $q(\mathbf{x}_k|\mathbf{X}_{k-1}, \mathbf{Z}_k) = p(\mathbf{x}_k|\mathbf{x}_{k-1})$. This causes (2.10) to reduce to

$$\mathbf{w}_k^{(n)} \propto \mathbf{w}_{k-1}^{(n)} p(\mathbf{z}_k|\mathbf{x}_k^{(n)}). \tag{2.11}$$

Particle $\mathbf{x}_k^{(n)}$ is obtained from $\mathbf{x}_{k-1}^{(n)}$ via (2.1), as this is a draw from $p(\mathbf{x}_k|\mathbf{x}_{k-1})$. The associated weight is updated to $\mathbf{w}_k^{(n)}$ from $\mathbf{w}_{k-1}^{(n)}$ via (2.2), which is a draw from $p(\mathbf{z}_k|\mathbf{x}_k^{(n)})$.

A situation called "particle degeneracy" is a common problem for the particle filter. As the particle filter operates, individual particles obtain high weights and the other particles vanish. Resampling the particles can be used to avoid this issue. This is done between time steps after obtaining the final estimate for the time step. Low weight particles are thrown away and high weight particles are split into multiple lower weight particles at the same location in the state space. There are several different ways to employ resampling. A commonly used approach first finds the

cumulative distribution for the posterior distribution by numbering the particles with the integers. For the nth pre-resampling particle, we denote the cumulative sum of the weights as c_n, for $n = 1, \ldots, N$, with N being the number of particles. We then draw a threshold, d_0 from a uniform distribution on $[0, 1/N)$, and we calculate the next threshold, $d_m = d_0 + (m-1)/N$, for $m = 1, \ldots, N$. While $c_n \geq d_m$, the mth resampled particle is set equal to the nth pre-resampling particle, with weight $\frac{1}{N}$; then once $c_n < d_m$, we compare the cumulative weight of the next particle to the threshold [105]. This generates N new particles with weight $\frac{1}{N}$, a valid pdf estimate.

The overall particle filter algorithm presented so far is for the theoretical case of estimating the parameters of a single object, and it is summarized in Algorithm 1 [3, 23, 105]. It can be extended to multiple objects given a known measurement association, so that we know which object generated which measurement. However, such an association is not normally available. If the measurement sources are independent, we can use a separate particle filter for each source.

2.2 Probabilistic Data Association Filter

The particle filter on its own is able to track a fixed and known number of objects when the association of measurement to object is known. When the association of measurement to source is not known, the particle filter can be modified to implement the probabilistic data association filter, which accounts for all association possibilities [3, 4, 81–84].

The PDAF assumes imperfect source detection, with probability of detection P_{D_k}. It also assumes that spurious measurements also appear, called clutter. Clutter is modeled as arising from a point Poisson process, which is distributed in the measurements space according to some probability distribution. In the PDAF, the distribution is uniform. The average number of clutter measurements at a given time is λ, though

Algorithm 1 SIR Particle Filter Algorithm.

Initialization

▷ Sample N particles $\mathbf{x}_0^{(n)} \sim p(\mathbf{x}_0)$ {*chosen uniform over expected state values*}

▷ Compute associate weights $\mathbf{w}_0^{(n)} = 1/N$, $n = 1, \ldots, N$

while $k \leq K$ **do**

Prediction Stage {K *is the maximum time step*}

▷ Predict the state for each particle according to $\mathbf{x}_k^{(n)} = \mathbf{x}_{k-1}^{(n)} + \mathbf{u}_{k-1}^{(n)}$

▷ Maintain the weights $\mathbf{w}_{k-1}^{(n)} = \mathbf{w}_{k-1}^{(n)}$

Update Stage

▷ Maintain particle $\mathbf{x}_k^{(n)}$

▷ Update corresponding weight $\mathbf{w}_k^{(n)} = \mathbf{w}_{k-1}^{(n)}\, p(\mathbf{z}_k|\mathbf{x}_k^{(n)})$

▷ Normalize the particle weights to sum up to 1

Resampling Stage

▷ Evaluate the cumulative sampled distribution c_n

▷ Draw threshold d_0 from a uniform distribution between 0 and $1/N$

▷ Initialize $m = 1$ and $n = 1$

for $m = 1 : N$ **do**

$d_m = d_0 + (m-1)/N$

if $c_n \geq d_m$ **then**

Replicate the particle

$\mathbf{x}_{k,\text{resamped}}^{(m)} = \mathbf{x}_k^{(n)}$

$\mathbf{w}_{k,\text{resamped}}^{(m)} = 1/N$

else

Discard the particle

Set $n = n + 1$

end if

end for

▷ $\mathbf{x}_k^{(m)} = \mathbf{x}_{k,\text{resamped}}^{(m)}$, $m = 1, ..., N$

▷ $\mathbf{w}_k^{(m)} = \mathbf{w}_{k,\text{resamped}}^{(m)}$, $m = 1, ..., N$

▷ Set $k = k + 1$

end while

the number is Poisson distributed. The PDAF assumes the distribution of the clutter measurements over the measurement space is uniform, resulting in a clutter density of ϱ, and also assumes a validation region in the measurement space around the expected object locations, of volume V_k. So, the probability of ζ_k clutter measurements arising at at time k is given by [81–84]:

$$\psi(\zeta_k) = \frac{\exp(-\varrho\, V_k)(\varrho\, V_k)^{\zeta_k}}{(\zeta_k)!}, \tag{2.12}$$

Since the PDAF has clutter measurements and may have missed detections, we obtain $m_k \geq 0$ measurement components at time k, $\mathbf{Z}_k = \{\mathbf{z}_{k,1}\mathbf{z}_{k,2} \ldots \mathbf{z}_{k,m_k}\}$. This includes measurements from objects and clutter, but we cannot be sure how many are from the known number of objects. If we assume that all different possible associations for a given number of detections are independent and equally likely, then we can combine their likelihoods. If the objects are well separated in the measurement space, their validation regions do not overlap, and the objects are tracked independently. For the lth object, we have [81–84]:

$$p(\mathbf{Z}_k|\mathbf{x}_{k,l}) \propto (1 - P_{d_k})\psi(m_k)V_k^{-m_k} + \frac{P_{d_k}\psi(m_k - 1)V_k^{-(m_k-1)}}{m_k} \sum_{j=1}^{m_k} p(\mathbf{z}_{k,j}|\mathbf{x}_{k,l})\,.$$

which simplifies to

$$p(\mathbf{Z}_k|\mathbf{x}_{k,l}^{(n)}) \propto (1 - P_{d_k}) + \frac{P_{d_k}}{\varrho} \sum_{j=1}^{m_k} p(\mathbf{z}_{k,j}|\mathbf{x}_{k,l})\,. \tag{2.13}$$

For more than one object, at each time step, the objects are tracked jointly if some measurements fall within the validation regions of more than one object. The possible associations at time step k can be given by [81]

$$B_k = \bigcap_{l=1}^{m_k} B_{k,j,l_j} \,,$$

where B_{k,j,l_j} is the event that the jth measurement came from the l_jth object and m_k is the number of measurements at time step k. The association probability is [81],

$$p(B_k|\mathbf{Z}_k) \propto \prod_j (\frac{1}{\varrho} p(\mathbf{z}_{k,j}|\mathbf{x}_{k,l_j}))^{\tau_j} \prod_l (P_{d_k})^{\delta_l} (1 - P_{d_k})^{1-\delta_l}, \tag{2.14}$$

where ϱ is the clutter density, $\mathbf{z}_{k,j}$ is the jth measurement, $\mathbf{x}_{k,l_j}$ is the state for object l_j, τ_j is the jth measurement association indicator, and δ_l is the lth object detection indicator.

For the case of two objects, we obtain the joint probability,

$$\begin{aligned} p(\mathbf{Z}_k|\mathbf{x}_{k,1}, \mathbf{x}_{k,2}) \propto\ & (1 - P_{d_k})^2 \\ & + \frac{P_{d_k}(1 - P_{d_k}) m_k}{\varrho n_1} \sum_{i \in J_1} p(\mathbf{z}_{k,i}|\mathbf{x}_{k,1}) + \frac{P_{d_k}(1 - P_{d_k}) m_k}{\varrho n_2} \sum_{j \in J_2} p(\mathbf{z}_{k,j}|\mathbf{x}_{k,2}) \\ & + \frac{P_{d_k}^2 m_k (m_k - 1)}{\varrho n} \sum_{i \in J_1 \cup J_3, j \in J_2 \cup J_3, i \neq j} p(\mathbf{z}_{k,i}|\mathbf{x}_{k,1}) p(\mathbf{z}_{k,j}|\mathbf{x}_{k,2}) \,. \end{aligned} \tag{2.15}$$

in this expression, $n = [n_1 n_2 + n_1 n_3 + n_2 n_3 + n_3(n_3 - 1)]$ and J_1, J_2, and J_3 are the sets of measurement indices that fall within the validation regions of object 1, object 2, and both objects. Additionally, n_1, n_2, and n_3 are the number of indices in each set. This can be generally extended to more jointly tracked objects, as discussed in [81–84]. Marginal distributions can be calculated and used in particle filtering [82, 84]. The data association algorithm is summarized in Algorithm 2.

2.3 Probability Hypothesis Density Filter

While data association is able to handle unknown measurement associations, it requires that the number of objects be known. When this is not the case, the probability

Algorithm 2 SIR Particle Probabilistic Data Association Algorithm.

Initialization

Given known number of objects L

▷ Sample N particles for each object $\mathbf{x}_{0,l}^{(n)} \sim p(\mathbf{x}_0)$ {*chosen uniform over expected state values*}

▷ Compute associate weights $\mathbf{w}_{0,l}^{(n)} = 1/N$, $n = 1, \ldots, N$

while $k \leq K$ **do**

Prediction Stage {K *is the maximum time step*}

▷ Predict the state for each particle according to $\mathbf{x}_{k,l}^{(n)} = \mathbf{x}_{k-1,l}^{(n)} + \mathbf{u}_{k-1}^{(n)}$

▷ Maintain the weights $\mathbf{w}_{k-1,l}^{(n)} = \mathbf{w}_{k-1,l}^{(n)}$

Update Stage

▷ Maintain particle $\mathbf{x}_{k,l}^{(n)}$

▷ Perform measurement validation for each object based on its prediction location at time $k-1$

For all objects l that can be tracked independently at time k

▷ Update corresponding weight $\mathbf{w}_{k,l}^{(n)} = \mathbf{w}_{k-1,l}^{(n)} \, p(\mathbf{Z}_k | \mathbf{x}_{k,l}^{(n)})$ (Eq. 2.13)

▷ Normalize the particle weights to sum up to 1

For all groups of objects $l_1, l_2, ..., l_\ell$ that must be jointly tracked at time k

▷ Calculate joint weights $\mathbf{w}_{k,(l_1,\ldots,l_\ell)}^{(n_1,\ldots n_\ell)} = (\prod_{i=1}^{\ell} \mathbf{w}_{k-1,l_i}^{(n_i)}) \, p(\mathbf{Z}_k | \mathbf{x}_{k,l_1}^{(n_1)}, ..., \mathbf{x}_{k,l_\ell}^{(n_\ell)})$ (Eq. 2.14 and 2.15)

▷ Marginalize out corresponding weights $\mathbf{w}_{k,l_i}^{(n_i)}$

▷ Normalize the particle weights to sum up to 1

Resampling Stage

▷ Evaluate the cumulative sampled distribution c_n for each object l, one at a time

▷ Draw threshold d_0 from a uniform distribution between 0 and $1/N$

▷ Initialize $m = 1$ and $n = 1$

for $m = 1 : N$ **do**

$d_m = d_0 + (m-1)/N$

if $c_n \geq d_m$ **then**

Replicate the particle

$\mathbf{x}^{(m)}_{k,\text{resamped}} = \mathbf{x}^{(n)}_k$

$\mathbf{w}^{(m)}_{k,\text{resampled}} = 1/N$

else

Discard the particle

Set $n = n + 1$

end if

end for

▷ $\mathbf{x}^{(m)}_{k,l} = \mathbf{x}^{(m)}_{k,l,\text{resamped}}$, $m = 1, ..., N$

▷ $\mathbf{w}^{(m)}_{k,l} = \mathbf{w}^{(m)}_{k,l,\text{resamped}}$, $m = 1, ..., N$

▷ Set $k = k + 1$

end while

hypothesis density filter (PHDF) can be used to estimate the number of objects, even when the number varies with time [22, 23, 36, 37, 85–89]. The PHDF uses the idea of random finite sets (RFS) to handle the varying number of objects to track [85]. The underlying state has a varying number of objects with corresponding parameter vector set given by, $\mathbf{X}_k = \{\mathbf{x}_{k,1}, ..., \mathbf{x}_{k,N_k}\}$, where N_k is the number of objects at time at each time step k. There is also varying number of measurements, $\mathbf{Z}_k = \{\mathbf{z}_{k,1}, ..., \mathbf{z}_{k,M_k}\}$ at time step k, where M_k is the number of measurements at time k. In the PHDF algorithm, these measurements are not known to be associated to any object, some objects may fail to generate a measurement, and some measurements may originate from clutter, instead of an object [85, 87–89]. Measurements from clutter become part of the RFS and a probability of object detection is included in the PHDF algorithm.

2.3.1 PHDF Formulation

The state RFS $\mathbf{X}_k$ and $\mathbf{x}_k \in \mathbf{X}_k$, as well as the measurement RFS $\mathbf{Z}_k$, can be characterized using the probability hypothesis density or complexity function, $\zeta(\mathbf{x}_k|\mathbf{Z}_k)$. Integrating this function on a region $\mathcal{R}$ gives the expected number of objects within the region. The peaks in the distribution are estimates of the state parameters of the objects [85–89].

We form a new state RFS $\mathbf{X}_k$ from the RFS at the previous time step k, $\mathbf{X}_{k-1}$. The new state RFS consists of the states of the objects still present from the previous time step, represented as $\mathbf{X}_{k|k-1}^{prev}$, the states of newly appearing objects, represented as $\mathbf{X}_k^{new}$, and the state of objects that spawned from objects in the previous time step, represented as $\mathbf{X}_{k|k-1}^{spn}$. The PHDF assumes that the prior and posterior densities for the states of the multiple objects can be characterized by their first moments. The prior intensity function at time k, $\zeta(\mathbf{x}_k|\mathbf{Z}_{1:k-1})$, can then be obtained from the

posterior intensity at time step $k-1$, $\zeta(\mathbf{x}_{k-1}|\mathbf{Z}_{1:k-1})$ as,

$$\begin{aligned}\zeta(\mathbf{x}_k|\mathbf{Z}_{1:k-1}) = &\zeta(\mathbf{x}_k^{new}) \\ &+ \int [P_{k|k-1}(\tilde{\mathbf{x}}_{k-1})p(\mathbf{x}_k|\tilde{\mathbf{x}}_{k-1}) + \zeta(\mathbf{x}_k^{spn}|\tilde{\mathbf{x}}_{k-1})] \\ &\qquad \zeta(\tilde{\mathbf{x}}_{k-1}|\mathbf{Z}_{1:k-1})d\tilde{\mathbf{x}}_{k-1},\end{aligned} \tag{2.16}$$

where $\mathbf{x}_k^{new} \in \mathbf{X}_k^{new}, \mathbf{x}_k^{spn} \in \mathbf{X}_{k|k-1}^{new}$, and $P_{k|k-1}(\tilde{\mathbf{x}}_{k-1})$ is the probability of a object present at time step $k-1$ remaining at time step k. Then we can determine the posterior as,

$$\begin{aligned}\zeta(\mathbf{x}_k|\mathbf{Z}_{1:k}) = &(1 - P_{D_k}(\mathbf{x}_k))\zeta(\mathbf{x}_k|\mathbf{Z}_{1:k-1}) \\ &+ \sum_{\mathbf{z}_{k,m}\in\mathbf{Z}_k} \frac{P_{D_k}(\mathbf{x}_k)p(\mathbf{z}_{k,m}|\mathbf{x}_k)\zeta(\mathbf{x}_k|\mathbf{Z}_{1:k-1})}{\zeta(\mathbf{Z}_k^{clt}) + \int P_{D_k}(\tilde{\mathbf{x}}_k)p(\mathbf{z}_{k,m}|\tilde{\mathbf{x}}_k)\zeta(\tilde{\mathbf{x}}_k|\mathbf{Z}_{1:k-1})d\tilde{\mathbf{x}}_k},\end{aligned} \tag{2.17}$$

where $P_{D_k}(\mathbf{x}_k)$ is the detection probability at time step k, and $\zeta(\mathbf{Z}_k^{clt})$ is the clutter intensity. In this forumlation, measurements are assumed to be mutually independent and the clutter and objects also assumed independent.

2.3.2 Implementation of PHDF using PF

The PHDF can be implemented using a particle filter, as shown in a radar context in [85, 87–89]. The intensity function, $\zeta(\mathbf{x}_{k-1}|\mathbf{Z}_{1:k-1})$ at time step $k-1$ can be approximated with N_p particles and associated weights, $\mathbf{x}_{k-1}^{(\ell)}, \mathbf{w}_{k-1}^{(\ell)}$ for $\ell = 1, ..., N_p$, as follows,

$$\zeta(\mathbf{x}_{k-1}|\mathbf{Z}_{1:k-1}) = \sum_{\ell=1}^{N_p} \mathbf{w}_{k-1}^{(\ell)}\, \delta\left(\mathbf{x}_k - \mathbf{x}_k^{(n)}\right). \tag{2.18}$$

The prediction step can be written as

$$\zeta(\mathbf{x}_k|\mathbf{Z}_{1:k-1}) = \sum_{\ell=1}^{N_p} \mathbf{w}_{k-1}^{(\ell)}[P_{k|k-1}(\tilde{\mathbf{x}}_{k-1}^{(\ell)})p(\mathbf{x}_k|\tilde{\mathbf{x}}_{k-1}^{(\ell)}) + \zeta(\mathbf{x}_k^{spn}|\tilde{\mathbf{x}}_{k-1}^{(\ell)})] + \zeta(\mathbf{x}_k^{new}). \tag{2.19}$$

We then apply importance sampling as described in Section 2.1 for the particle filter algorithm. The first N_p samples are drawn from $q_k(\mathbf{x}_k^{(\ell)}|\mathbf{x}_{k-1}^{(\ell)}, \mathbf{Z}_{1:k-1})$, and the

next N_q samples are drawn from an importance intensity function $E_k(\mathbf{x}_k^{(\ell)}|\mathbf{Z}_{1:k-1})$. Note that N_p and N_q are user-specified. We then obtain a representation with particles and associated weights, $\mathbf{x}_k^{(\ell)}, \mathbf{w}_{k|k-1}^{(\ell)}$ for $\ell = 1, ..., N_p + N_q$:

$$\zeta(\mathbf{x}_k|\mathbf{Z}_{1:k-1}) = \sum_{\ell=1}^{N_p+N_q} \mathbf{w}_{k|k-1}^{(\ell)} \delta(\mathbf{x}_k - \mathbf{x}_k^{(\ell)}), \tag{2.20}$$

where

$$\mathbf{w}_{k|k-1}^{(\ell)} = \begin{cases} \mathbf{w}_{k-1}^{(\ell)} \frac{[P_{k|k-1}(\mathbf{x}_{k-1}^{(\ell)}) p(\mathbf{x}_k^{(\ell)}|\mathbf{x}_{k-1}^{(\ell)})] + \zeta(\mathbf{x}_k^{(\ell),spn}|\mathbf{x}_{k-1}^{(\ell)})}{q_k(\mathbf{x}_k^{(\ell)}|\mathbf{x}_{k-1}^{(\ell)}, \mathbf{Z}_{1:k-1})}, & \ell = 1, ..., N_p \\ \frac{\zeta(\mathbf{x}_k^{(\ell),new})}{N_q \, E_k(\mathbf{x}_k^{(\ell)}|\mathbf{Z}_{1:k-1})}, & \ell = N_p + 1, ..., (N_p + N_q) \end{cases}$$

If spawned objects are not allowed by the model, and the proposal distributions are chosen to be $q_k(\mathbf{x}_k^{(\ell)}|\mathbf{x}_{k-1}^{(\ell)}, \mathbf{Z}_{1:k-1}) = p(\mathbf{x}_k^{(\ell)}|\mathbf{x}_{k-1}^{(\ell)})$ and $E_k(\mathbf{x}_k^{(\ell)}|\mathbf{Z}_{1:k-1}) = \zeta(\mathbf{x}_k^{new}|\mathbf{Z}_{1:k})$, then this simplifies to

$$\mathbf{w}_{k|k-1}^{(\ell)} = \begin{cases} \mathbf{w}_{k-1}^{(\ell)} [P_{k|k-1}(\mathbf{x}_{k-1}^{(\ell)})], & \ell = 1, ..., N_p \\ \frac{\nu}{N_q}, & \ell = N_p + 1, ..., (N_p + N_q), \end{cases}$$

where ν is the expected number of new objects. We then obtain via substitution the representation for the posterior intensity,

$$\zeta(\mathbf{x}_k|\mathbf{Z}_{1:k}) = \sum_{\ell=1}^{N_p+N_q} \mathbf{w}_k^{(\ell)} \delta(\mathbf{x}_k - \mathbf{x}_k^{(\ell)}), \tag{2.21}$$

where

$$\mathbf{w}_k^{(\ell)} = \mathbf{w}_{k|k-1}^{(\ell)} \left[1 - P_k^{det}(\mathbf{x}_k^{(\ell)}) + \sum_{\mathbf{z}_{k,m} \in \mathbf{Z}_k} \frac{P_k^{det}(\mathbf{x}_k^{(\ell)}) p(\mathbf{z}_{k,m}|\mathbf{x}_k^{(\ell)})}{\zeta(\mathbf{Z}_k^{clt}) + C_k(\mathbf{Z}_k)} \right] \tag{2.22}$$

and

$$C_k(\mathbf{z}_{k,m}) = \sum_{\ell=1}^{N_p+N_q} \mathbf{w}_{k|k-1}^{(\ell)} P_k^{det}(\mathbf{x}_k^{(\ell)}) p(\mathbf{z}_{k,m}|\mathbf{x}_k^{(\ell)}). \tag{2.23}$$

This enables us to estimate the state parameters at any time step from the particle approximation at the previous time step. The PHDF is able to track multiple objects and estimate the number of objects present. The final localization at each time step

requires finding the peaks of the PHD. This can be done in a few ways, but the most straightforward is to take the estimated number of objects from summing the weights, and use a Gaussian mixture model or k-means clustering to find the clusters within the particles. It is an algorithm used often in radar and sonar applications, but can be applied to the neural tracking problem as long as separate measurements can be provided. Obtaining good separation from the ICA algorithm is critical. Pseudocode for the particle filter implementation of the PHDF is provided in Algorithm 3.

2.3.3 OSPA Error Metric

Since the PHDF must estimate the number of objects, the mean square error is not directly applicable. Different error metrics on random finite sets have been proposed, including the OSPA metric [108]. The OSPA metric was proposed with the intent of improving the OMAT metric [109], which it uses within. The OSPA metric takes in two sets and uses the OMAT metric to determine the best association in terms of distance between the elements of the set, though if the sets have different sizes, some elements of the larger set are not associated to any element of the smaller set. The distance between each associated pair is computed, and the distance is set to infinity for the unassociated elements of the larger set. Each distance is then compared to a user specified threshold c, and the minimum of the absolute value of the distance and c is kept. These values are then raised to a user specified power p, averaged, and pth-rooted to obtain the final OSPA metric. The cutoff can be thought of as the distance past which an element of the set is considered unassociatable. It is presented in Algorithm 4. Choosing order $p = 2$ resembles an RMSE.

Algorithm 3 Particle Implementation of the Probability Hypothesis Density Filter.

Initialization

▷ Define number of particles per continued object, N, and for new objects N_q

▷ Make initial guess about number of objects $\hat{T}_0$

▷ Sample $N_{p,0} = N\,\hat{T}_0$ particles $\mathbf{x}_0^{(\ell)} \sim p(\mathbf{x}_0)$ {*chosen uniform over expected state values*}

▷ Compute associate weights $\mathbf{w}_0^{(\ell)} = \hat{T}_0/N_{p,0}$, $\ell = 1, \ldots, N_{p,0}$

while $k \leq K$ **do**

Prediction Stage {K *is the maximum time step*}

▷ Predict the state for each particle according to $\mathbf{x}_k^{(\ell)} = \mathbf{x}_{k-1,\text{resamped}}^{(\ell)} + \mathbf{u}_{k-1}^{(\ell)}$

▷ Draw N_q particles for new objects {*chosen uniform over expected state values*}

▷ Calculate the prediction weights,

$$\mathbf{w}_{k|k-1}^{(\ell)} = \begin{cases} \mathbf{w}_{k-1,resampled}^{(\ell)}[P_{k|k-1}(\mathbf{x}_{k-1}^{(\ell)})], & \ell = 1, ..., N_{p,k-1} \\ \frac{\nu}{N_q}, & \ell = N_{p,k-1}+1, ..., N_{p,k-1}+N_q \end{cases}$$

Update Stage

▷ Maintain particle $\mathbf{x}_k^{(n)}$

▷ Update corresponding weight

$$\mathbf{w}_k^{(\ell)} = \mathbf{w}_{k|k-1}^{(\ell)}[1 - P_k^{det}(\mathbf{x}_k^{(\ell)}) + \sum_{\mathbf{z}_{k,m} \in \mathbf{Z}_k} \frac{P_k^{det}(\mathbf{x}_k^{(\ell)}) p(\mathbf{z}_{k,m}|\mathbf{x}_k^{(\ell)})}{\zeta(\mathbf{Z}_k^{clt}) + C_k(\mathbf{Z}_k)}] \tag{2.24}$$

▷ Calculate the estimated number of objects, $\hat{T}_k = \sum_{\ell=1}^{N_{p,k-1}+N_q} \mathbf{w}_k^{(\ell)}$

▷ Find T_k, which is $\hat{T}_k$ rounded to the nearest whole number of objects

Resampling Stage

▷ Divide all particle weights by $\hat{T}_k$

▷ Calculate $N_{p,k} = N\,T_k$

▷ Evaluate the cumulative sampled distribution c_n

▷ Draw threshold d_0 from a uniform distribution between 0 and $1/(N_{p,k})$

▷ Initialize $m = 1$ and $n = 1$

for $m = 1 : N_{p,k}$ **do**

$d_m = d_0 + (m-1)/(N_{p,k})$

if $c_n \geq d_m$ **then**

Replicate the particle

$\mathbf{x}^{(m)}_{k,\text{resamped}} = \mathbf{x}^{(n)}_k$

$\mathbf{w}^{(m)}_{k,\text{resamped}} = \hat{T}_k/N_{p,k}$

Set $m = m + 1$

else

Discard the particle

Set $n = n + 1$

end if

end for

▷ $\mathbf{x}^{(m)}_k = \mathbf{x}^{(m)}_{k,\text{resamped}}$, $m = 1, ..., N_{p,k}$

▷ $\mathbf{w}^{(m)}_k = \mathbf{w}^{(m)}_{k,\text{resamped}}$, $m = 1, ..., N_{p,k}$

▷ Set $k = k + 1$

end while

Algorithm 4 OSPA Metric.

Inputs: Two sets $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_M\}$ and $\mathbf{Y} = \{\mathbf{y}_1, ..., \mathbf{y}_N\}$, whose elements are vectors of the same size, $N \geq M$

$d(\mathbf{x}, \mathbf{y})$ is a distance metric

Π_k is the set of all permutations of {1,2,...,k}

$c > 0$ is a distance cutoff

$d^{(c)}(\mathbf{x}, \mathbf{y}) = \min(c, d(\mathbf{x}, \mathbf{y}))$

Final metric:

$$d_p^{(c)}(\mathbf{X}, \mathbf{Y}) = \left(\frac{1}{n} \left(\min_{\pi \in \Pi_m} \sum_{i=1}^{m} d^{(c)}(\mathbf{x}_i, \mathbf{y}_{\pi(i)})^p + c^p(m-n) \right) \right)^{1/p}$$

Chapter 3

NEURAL ACTIVITY ESTIMATION AND ARTIFACT SUPPRESSION

3.1 Neural Activity Dipole Source Model

The fundamental mechanics of brain function are fairly straightforward. Special cells called neurons activate, firing electric currents called action potentials, in response to other neurons and electrochemical conditions. Coordinated neuron activity controls every brain function, from giving instructions to muscles to considering the mysteries of the universe. These clusters of neural activity can each be modeled as single dipole current sources. Kirchoff's current law requires that the net current within the head must be zero, so the dipole's local primary current must be closed to a weaker volume current throughout the brain, skull, and scalp [11, 22, 23]. Brain current then generates an electric potential and magnetic field that can be measured with a sensor net. The goal is then to solve for the parameters of the current that generated the field, though it is a challenge to do so. [11]. The solution is not unique [11], as a continuous distribution is sampled at the sensor locations, and the Nyquist theorem on the spacial frequencies present cannot be guaranteed. One method, the one used in this work, uses the assumption that the neural currents are accurately modeled as dipole sources to look for the unique solution that then arises [11, 23, 110].

A current density, $\mathbf{J}_n$, formed by neuron current in the head generates a measurement vector of electric potentials. The forward problem is solved by calculating the potentials at the electrodes when given an electrical source. The number of dipole sources can vary as the brain performs different tasks. Mathematically, L dipolar

current sources form a current density at time n of [11, 110]

$$\mathbf{J}_n = \sum_{l=1}^{L} \delta(\mathbf{r} - \mathbf{r}_{n,l})\, \boldsymbol{\theta}_{n,l}, \tag{3.1}$$

where $\mathbf{r}_{n,l} = [r_{n,l}^{(x)}\ r_{n,l}^{(y)}\ r_{n,l}^{(z)}]^{\mathsf{T}}$ and $\boldsymbol{\theta}_{n,l} = [\theta_{n,l}^{(x)}\ \theta_{n,l}^{(y)}\ \theta_{n,l}^{(z)}]^{\mathsf{T}}$ are the position and moment, respectively, of the lth dipole source in the brain. In this model, the amplitude, $s_{n,l}$, and orientation, $\mathbf{q}_{n,l} = [q_{n,l}^{(x)}\ q_{n,l}^{(y)}\ q_{n,l}^{(z)}]^{\mathsf{T}}$ are folded into the moment $\boldsymbol{\theta}_{n,l}$. The moment is then given by $\boldsymbol{\theta}_{n,l} = s_{n,l}\, \mathbf{q}_{n,l}$. We model the head as concentric spheres for the brain, cerebral fluid, skull, and scalp and use the static equations of electromagnetics due to the low frequency nature of the electroencephalogram (EEG) and magnetoencephalogram (MEG) measurements [11]. The dipole sources then generate measurements according to the Biot-Savart law, Ohm's law, and the definition of electric potential. The Biot-Savart law for the magnetic field at time n, $\mathbf{B}_n(\mathbf{r})$, for a region V and current density $\mathbf{J}_n(\mathbf{r}')$ is [11]

$$\mathbf{B}_n(\mathbf{r}) = \frac{\mu_0}{4\pi} \iiint_{\mathbf{r}' \in V} \mathbf{J}_n(\mathbf{r}') \times \frac{(\mathbf{r} - \mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|^3}\, \mathrm{d}\mathbf{r}', \tag{3.2}$$

where μ_0 is the permittivity of free space and $(\mathbf{a} \times \mathbf{b})$ denotes the cross product between vectors $\mathbf{a}$ and $\mathbf{b}$. We assume an isotropic current flowing through the layers of the head with conductivity $\sigma_s(\mathbf{r})$ for the brain, skull, and scalp [2], and that the current generates an electric field at time n, $\mathbf{E}_n(\mathbf{r})$. Using Ohm's law,

$$\mathbf{J}_n(\mathbf{r}) = \sigma_{n,s}(\mathbf{r})\, \mathbf{E}_n(\mathbf{r}). \tag{3.3}$$

The electric potential and electric field are related as follows:

$$V_n(\mathbf{r}) = \int_C \mathbf{E}_n(\mathbf{r}')\, \mathrm{d}\mathbf{r}'. \tag{3.4}$$

Combining these equations forms a model for the measurements obtained from neural currents. The measurements can be placed in vector form: [22, 23, 110]

$$\mathbf{y}_n = \mathbf{A}_n \mathbf{s}_n + \mathbf{v}_n, \tag{3.5}$$

where $\mathbf{y}_n$ is an $M \times 1$ column vector of measurements, M is the number of sensors, $\mathbf{A}_n$ is an $M \times L$ lead-field matrix that is a function of the positions $\mathbf{r}_{n,l}$ and moments $\boldsymbol{\theta}_{n,l}$ of the L dipoles, $\mathbf{s}_n = [s_{n,1} \ \ldots \ s_{n,L}]^{\mathsf{T}}$ is an $L \times 1$ column vector of the dipole amplitudes, and $\mathbf{v}_n$ is an $M \times 1$ column vector of measurement noise, all at time n [22, 23, 110].

EEG and MEG have different models for $\mathbf{A}_n$. In the case of EEG, the elements $a_{n,m,l}$, mapping dipole l to sensor m at time n, which compose matrix $\mathbf{A}_n$ in the case of a single sphere model, are [22, 23, 110, 111]

$$\begin{aligned} a_{n,m,l} =& \frac{1}{4\pi\sigma_s}\cos(\alpha_{n,l})\left[\frac{2}{d_{n,m,l}^3}\big(|\mathbf{r}_{n,l}|\cos(\gamma_{n,m,l}) - r\big) + \frac{1}{|\mathbf{r}_{n,l}|d_{n,m,l}} - \frac{1}{|\mathbf{r}_{n,l}|r}\right] \\ &+ \frac{1}{4\pi\sigma_s}\sin(\alpha_{n,l})\cos(\beta_{n,m,l})\sin(\gamma_{n,m,l}) \\ &\left[\frac{2r}{d_{n,m,l}^3} + \frac{d_{n,m,l} + r}{r\, d_{n,m,l}(r - |\mathbf{r}_{n,l}|\cos(\gamma_{n,m,l}) + d_{n,m,l})}\right]. \end{aligned} \tag{3.6}$$

The elements of the lead-field matrix for MEG, $\mathbf{A}_n$, are [22, 23, 110, 111]

$$a_{n,m,l} = \left[\frac{\mu_0}{4\pi f(\mathbf{r}_{n,l}, \mathbf{r}_m)^2}\Big(f(\mathbf{r}_{n,l}, \mathbf{r}_m)\,\boldsymbol{\theta}_m \times \mathbf{r}_{n,l} - \boldsymbol{\theta}_m \times \mathbf{r}_{n,l} \cdot \mathbf{r}_m \mathbf{F}(\mathbf{r}_{n,l}, \mathbf{r}_m)\Big)\right]^{\mathsf{T}} \mathbf{q}_{n,l}, \tag{3.7}$$

where the position relative to the origin and moment of the M sensors are $\mathbf{r}_m$ and $\boldsymbol{\theta}_m$ respectively, $\mathbf{r}_{n,l}$ is the position of dipole l relative to the center of the sphere at time n, $\mathbf{q}_{n,l}$ is the orientation of dipole l at time n [22, 23, 110, 111], r is the radius of the spherical head model, $d_{n,m,l}$ is the distance between dipole l and sensor m, $\gamma_{n,m,l}$ is the angle between $\mathbf{r}_{n,l}$ and $\mathbf{r}_m$, $\alpha_{n,l}$ is the angle between $\mathbf{r}_{n,l}$ and $\mathbf{q}_{n,l}$, and $\beta_{n,m,l}$ is the angle between two planes, the plane formed by $\mathbf{r}_l$ and $\mathbf{q}_{n,l}$ and the plane formed by $\mathbf{r}_l$ and $\mathbf{r}_m$,

$$\begin{aligned} f(\mathbf{r}_{n,l}, \mathbf{r}_m) &= d_{n,m,l}(d_{n,m,l}\,|\mathbf{r}_m| + |\mathbf{r}_m|^2 - \mathbf{r}_{n,l}^{\mathsf{T}}\,\mathbf{r}_m), \\ \mathbf{F}(\mathbf{r}_{n,l}, \mathbf{r}_m) &= \left(\frac{d_{n,m,l}^2}{|\mathbf{r}_m|} + \frac{(\mathbf{r}_m - \mathbf{r}_{n,l})^{\mathsf{T}}\,\mathbf{r}_m}{d_{n,m,l}} + 2\,d_{n,m,l} + 2|\mathbf{r}_m|\right)\mathbf{r}_m \\ &\quad - \left(d_{n,m,l} + 2\,|\mathbf{r}_m| + \frac{(\mathbf{r}_m - \mathbf{r}_{n,l})^{\mathsf{T}}\,\mathbf{r}_m}{d_{n,m,l}}\right)\mathbf{r}_{n,l}, \end{aligned}$$

μ_0 is the permittivity of free space, and σ_s is permeability. We use a four spherical shell model, for brain, fluid, skull, and scalp, which is more complex. Figure 3.1 depicts an example of the relationship between the dipoles and sensors.

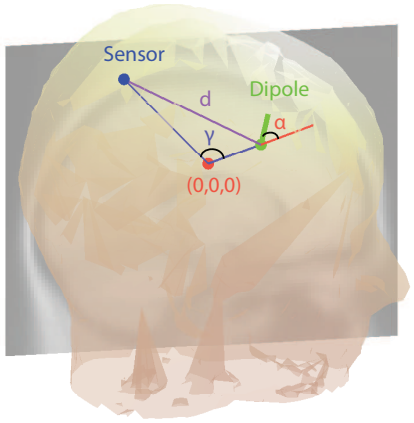


Figure 3.1: Demonstration of the Three-Dimensional Parameters Used in the EEG Lead-Field Matrix Formulation, Created Using EEGLAB [2, 3].

Equation (3.5) can also be written as

$$\mathbf{y}_n = \sum_{l=1}^{L} \mathbf{h}(\mathbf{r}_{n,l}, \mathbf{q}_{n,l}, s_{n,l}) + \mathbf{v}_n, \tag{3.8}$$

with $\mathbf{h}(\mathbf{r}_{n,l}, \mathbf{q}_{n,l}, s_{n,l})$ a highly nonlinear function of $\mathbf{r}_{n,l}$, but linear in $\mathbf{q}_{n,l}$ and $s_{n,l}$, according to (3.5). It can also be written as

$$\mathbf{y}_n = \sum_{l=1}^{L} \mathbf{D}(\mathbf{r}_{n,l})\mathbf{q}_{k,l}s_{n,l} + \mathbf{v}_n, \tag{3.9}$$

where $\mathbf{D}(\mathbf{r}_{k,l})$ is $M \times 3$ vector function of the dipole position $\mathbf{r}_{k,l}$, and $\mathbf{A}_{k,l} = \mathbf{D}(\mathbf{r}_{k,l})\mathbf{q}_{k,l}$. Estimating $\mathbf{r}_{n,l}$, $\mathbf{q}_{n,l}$, and $s_{n,l}$ for all times n and all L dipoles is referred to as the inverse problem [11]. Next, we examine our use of particle filtering to solve the inverse problem [22, 23].

3.2 Dipole Source Parameter Estimation

In the simplest case for EEG source localization, we would obtain many separate measurements, all of which are generated by single dipolar neural activity that we want to track, we would know which measurement comes from which dipole, and we would know that the number of dipoles does not change. Then we could directly use the particle filter to estimate the source parameters, without requiring more advanced algorithms. We will use independent component analysis as a tool to obtain separate measurements, but we still have to account for a changing number of dipoles and lack of association between dipole and measurement. This requires the use of the probability hypothesis density filter (PHDF).

We assume that there are L dipoles at time step n, and that a raw EEG measurement is obtained from M sensors placed over the head of the form $\mathbf{y}_n = \sum_{l=1}^{L} \mathbf{D}(\mathbf{r}_{n,l})\mathbf{q}_{n,l}s_{n,l}+\mathbf{v}_n$. Here, $s_{n,l}$, $\mathbf{r}_{n,l} = [r_{n,l}^{(x)}\ r_{n,l}^{(y)}\ r_{n,l}^{(z)}]^{\mathrm{T}}$ and $\mathbf{q}_{n,l} = [q_{n,l}^{(x)}\ q_{n,l}^{(y)}\ q_{n,l}^{(z)}]^{\mathrm{T}}$ are the lth dipole amplitude, position vector, and orientation vector, respectively, for $l = 1, \ldots, L$. The particle filter (PF) requires a state transition equation and measurement equation of the forms in Equations (2.1) and (2.2) to obtain an estimate of the dipole parameters. Determining the measurement equation involves taking the EEG/MEG lead-field equations in (3.5) and finding a way to obtain separate measurements from each dipole source, to match up to the model:

$$\mathbf{y}_n = \mathbf{A}_n\mathbf{s}_n + \mathbf{v}_n = \sum_{l=1}^{L} \mathbf{D}(\mathbf{r}_{n,l})\mathbf{q}_{n,l}s_{n,l} + \mathbf{v}_n, \tag{3.10}$$

where the elements of $\mathbf{A}_n$ for EEG and MEG are provided in (3.6) and (3.7), respectively, $\mathbf{s}_n$ is a column vector of dipole amplitudes, and $\mathbf{v}_n$ is measurement noise. Then, each column of $\mathbf{A}_n$ can we written as $\mathbf{A}_{n,l} = \mathbf{D}(\mathbf{r}_{n,l})\mathbf{q}_{n,l}$.

3.2.1 Independent Component Analysis

The measurements obtained from EEG/MEG contain a mixture of all of the neural activity. Separating them into measurements from individual sources is difficult. A method used by the group who made the EEG processing program EEGLAB [2] is independent component analysis (ICA). This method is based on finding the most temporally independent components in the data. It can be applied to EEG since the time courses of dipoles in different areas of the brain should be independent, even if there is a causal relationship between activity. Some use of ICA on EEG data can be found in [2, 49, 112]. ICA finds individual measurements that have removed the large dependencies that appear in the sensors across the head. They should tend to correspond to particular sources, allowing us to use them in the particle filter to estimate the dipole position, orientation, and amplitude parameters [22, 23]. ICA maximizes temporal independence using one of several possibilities, including minimizing the mutual information as well as maximizing the non-Gaussianity of the data using metrics like the kurtosis of the sources. In order to apply ICA, we need a segment of data on which the dipole position and orientation are approximately constant.

Not all of the separated components come from brain activity, however. There can be other electrical activity that shows up on the EEG/MEG, including eye movements, muscle movements, line noise from a power source, or electrodes popping. These components can interfere with neural analysis unless they are suppressed. After ICA, these components appear at the output of the ICA [113] and can be suppressed with signal processing techniques.

First, we inspect some data provided with EEGLAB and see the results of using ICA. The provided real data comes from a visual reaction experiment, and has the following description on the EEGLAB wiki [114]:

> In this experiment, there were two types of events "square" and "rt"; "square" events correspond to the appearance of a green colored square in the display and "rt" to the reaction time of the subject. The square could be presented at five locations on the screen distributed along the horizontal axis. Here we only considered presentation on the left, i.e. position 1 and 2 as indicated by the position field (at about 3 degree and 1.5 degree of visual angle respectively). In this experiment, the subject covertly attended to the selected location on the computer screen responded with a quick thumb button press only when a square was presented at this location. They were to ignore circles presented either at the attended location or at an unattended location. To reduce the amount of data required to download and process, this dataset contains only targets (i.e., "square") stimuli presented at the two left-visual-field attended locations for a single subject [114].

The data comes with markers for events, including the presentation of the visual cue. Using these markers, we select a three second segment that begins one second before the cue, and two seconds after. There are 80 such intervals, giving 240 seconds of data with 32 sensors placed over the head. The sensor locations are shown in three-dimension (3-D) and in a two-dimensional (2-D) projection in Figure 3.3. Detrending, baseline removal, and average referencing follow the procedure on the EEGLAB wiki [114]. We first run ICA on the full 240 second interval and examine

the components that arise. Three "epochs," which are the three second segments around each visual cue, are shown in Figure 3.3 [2].

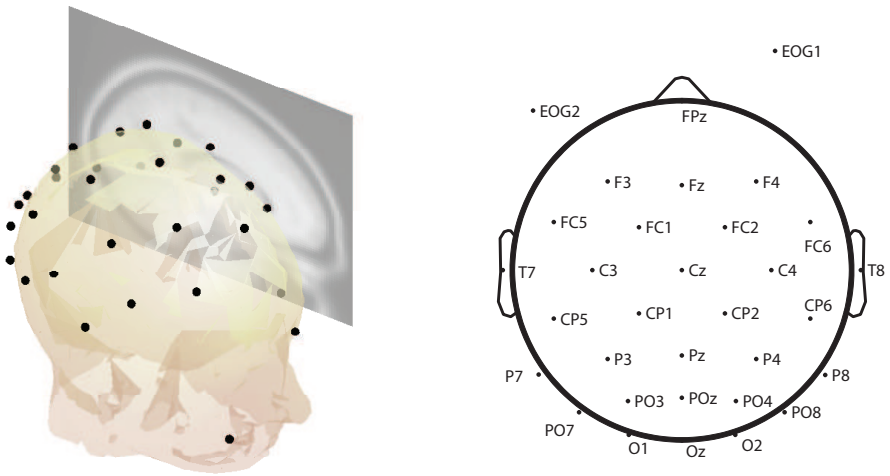


Figure 3.2: Diagram of EEG Sensor Locations, Created Using EEGLAB [2, 3].

The third interval appears to include an artifact as there is a large spike around a half second into the interval. It is most prominent in channels at the top of the plot, which tend to be near the front of the head, so the strongest potential is found near the face, indicating the presence of an eye-related artifact. The way that the signal rises and falls to the same level indicates an eye blink instead of a movement [38]. Using ICA, we separate the artifact component using the linear transformation used by the algorithm to obtain independent components [112]. The creators of EEGLAB have had success in using ICA to separate artifacts, even though ICA does not use information about the underlying neural model [49, 112]. Figure 3.4 shows the final resulting ICA components. The spike observed before appears to have separated out to component 6. The blink also appears in the same component more times over all epochs. Some more artifacts appear in the final components. There is another eye artifact in component 10, though its amplitude is different before and after the jump,

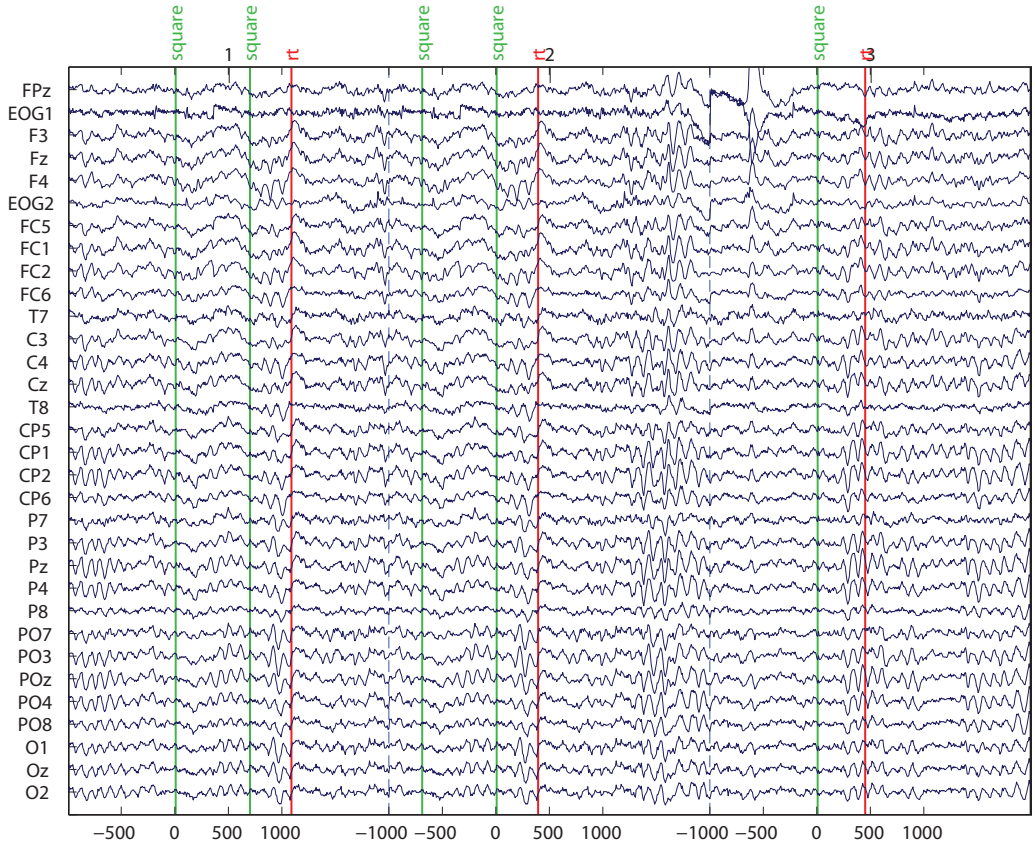


Figure 3.3: EEG Readings from Channels around Visual Stimulus from -1 to 2 s, Created Using EEGLAB [2, 3].

which is more likely to be an eye movement than a blink [38]. Component 28 is a jump that looks like an electrode pop or some other temporary mean shift but does not show up clearly in the original data. A useful way to examine artifacts is to plot the mixture coefficients at each sensor to try and identify the location of the current generating the component. Figure 3.5 and Figure 3.6 show these plots for components 6 and 10, respectively. The large mixture coefficients near the eyes are clear. This strongly indicates their classification as eye artifacts. Once the measurements are

separated, the PF algorithm can be applied as the measurements are associated to their corresponding sources.

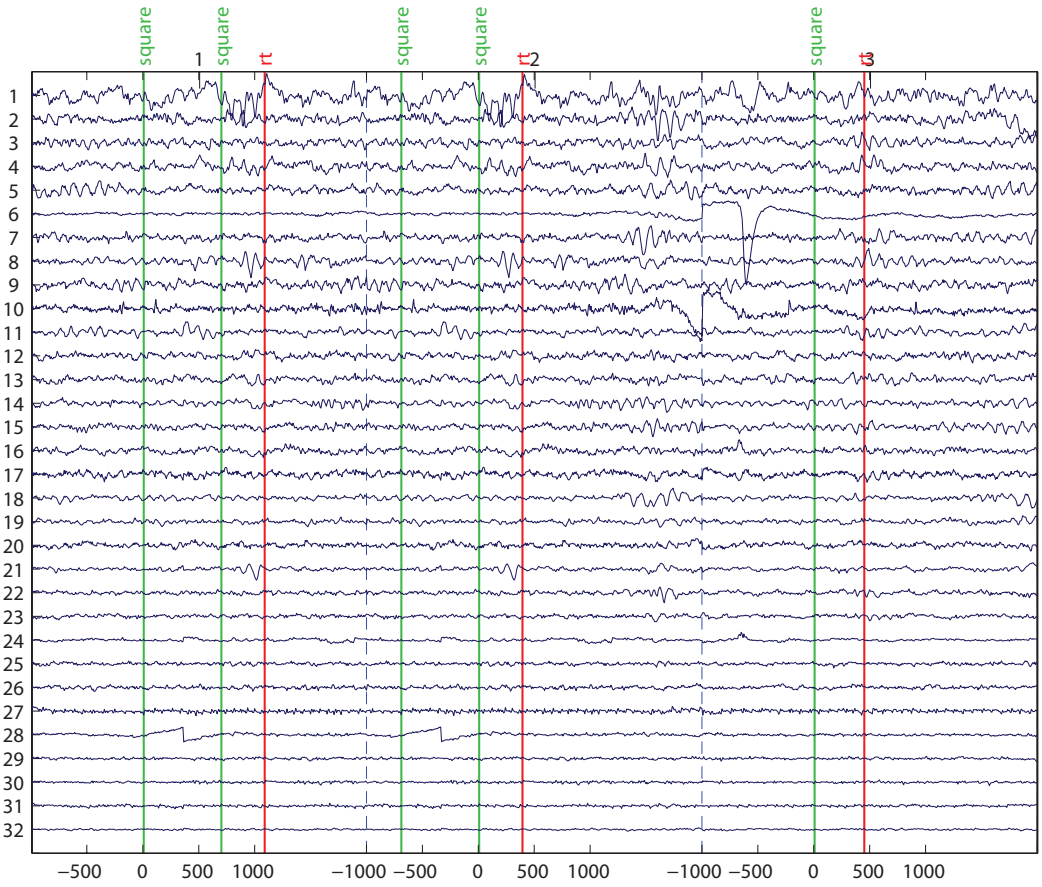


Figure 3.4: EEG Readings of 32 Independent Components around Visual Stimulus from -1 to 2 s, Created Using EEGLAB [2–4].

3.2.2 Particle Filter Tracking of a Known Number of Neural Sources

As stated before, to apply ICA, an interval on which the dipole position and orientation are approximately constant is required. We assume that this is true on intervals of length $\mathcal{L}$, so that the matrices from Equations (3.6) and (3.7) can be constructed. We have $A_n \approx A_k$ for $n = (k-1) * \mathcal{L} + 1, ..., k * \mathcal{L}$, and we extract

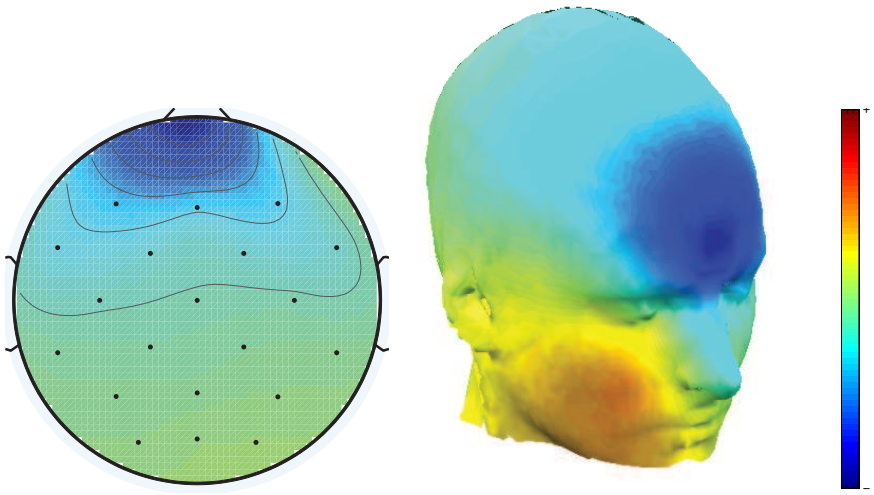


Figure 3.5: Strength of Channels in Independent Component 6, Created Using EEGLAB [2–4].

L_k separated measurements at a particular time interval k, where $k = 1, ..., K$. We denote these separated measurements as $\boldsymbol{\zeta}_{k,l}$ for $l = 1, ..., L_k$. These are extracted via ICA on the raw measurement $\{\mathbf{y}_{k*\mathcal{L}+1}, ..., \mathbf{y}_{(k+1)*\mathcal{L}}\}$. With the modification of the measurement equation in terms of the separated measurements and the separated state vectors $\mathbf{x}_{k,l}$, the state parameters of the lth dipole at time interval k becomes

$$\boldsymbol{\zeta}_{k,l} = \mathbf{A}_{k,l}\mathbf{s}_{k,l} + \mathbf{v}_{k,l} = \mathbf{B}(\mathbf{r}_{k,l})\mathbf{q}_{k,l}\mathbf{s}_{k,l} + \mathbf{v}_{k,l}, \tag{3.11}$$

where $\mathbf{A}_{k,l}$ is the lth column of the EEG/MEG matrix from Equations (3.6) and (3.7), $\mathbf{s}_{k,l}$ is the $1 \times \mathcal{L}$ time course on independent component l, and $\mathbf{v}_{k,l}$ is a Gaussian noise vector. ICA provides an estimate of the $\mathbf{A}_k$ matrix and separated signals $\mathbf{s}_k$. In the particle filter, we can use $A_{k,l}$ to estimate the lth dipole position, $\mathbf{r}_{k,l}$. We can use $\mathbf{s}_{k,l}$ to discriminate brain activity from artifacts, since the amplitude contains frequency information and the overall waveform of the independent components. We assume that $\boldsymbol{\sigma}_{k,l}$ is a vector of features that are functions of $\mathbf{s}_{k,l}$ and discriminate brain activity from artifacts. If the measurement does not include artifacts and the

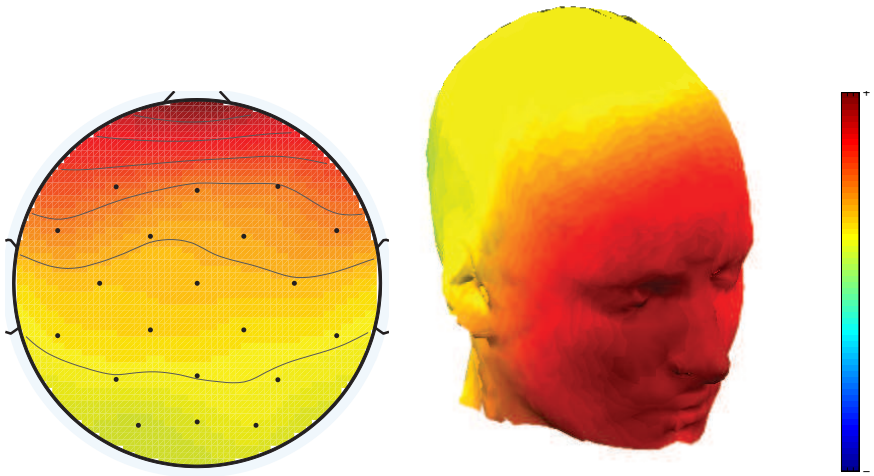


Figure 3.6: Strength of Channels in Independent Component 10, Created Using EEGLAB [2–4].

measurement to source association is known, the lth dipole measurement vector for the particle filter is simply $\mathbf{z}_{k,l} = \mathbf{A}_{k,l} + \mathbf{n}_{k,l}$, where $\mathbf{A}_{k,l}$ is the lth column of the mixture matrix $\mathbf{A}_k$, and $\mathbf{n}_{k,l}$ is Gaussian noise. Note that, in general a Gaussian model is not required as the particle filter allows for non-Gaussian noise. When features are incorporated, denoted as $\boldsymbol{\sigma}_{k,l}^T$, that discriminate neural activity from artifacts, we have $\mathbf{z}_{k,l} = [\mathbf{A}_{k,l}^T, \boldsymbol{\sigma}_{k,l}^T]^T + \mathbf{n}_{k,l}$. This gives a model probability density function (pdf) for neural activity. We denote the measurement vector pdf for brain activity as $p(\mathbf{z}_{k,l}|\mathbf{x}_{k,l})$, with $\mathbf{x}_{k,l} = \mathbf{r}_{k,l}$. This equation is used during the weight update in the particle filter, applying the forward model to solve the inverse problem. We assume that the dipoles remain in a local region for as long as they continue to exist. New dipoles may arise and old dipoles may vanish, but for a dipole to be considered as a single dipole moving, it should stay around a particular lobe of the brain. Thus, we apply a random walk model as the state transition equation. Thus, for each particle, $\mathbf{x}_{k,l}^{(n)} = \mathbf{x}_{k-1,l}^{(n)} + \mathbf{u}_{k,l}$, with $\mathbf{u}_k$ is assumed zero-mean Gaussian. The final state model

becomes

$$\mathbf{x}_{k,l} = \mathbf{x}_{k-1,l} + \mathbf{u}_{k-1,l}, \tag{3.12}$$

and our measurement model:

$$\mathbf{z}_{k,l} = \begin{bmatrix} \mathbf{A}_{k,l} \\ \boldsymbol{\sigma}_{k,l} \end{bmatrix} + \mathbf{n}_{k,l} = \begin{bmatrix} \mathbf{D}(\mathbf{x}_{k,l})\mathbf{q}_{k,l} \\ \boldsymbol{\sigma}_{k,l} \end{bmatrix} + \mathbf{n}_{k,l}\,. \tag{3.13}$$

This results in

$$p(\mathbf{z}_{k,l}|\mathbf{x}_{k,l}, \mathbf{q}_{k,l}) \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{D}(\mathbf{x}_{k,l})\mathbf{q}_{k,l} \\ \boldsymbol{\mu}_{\boldsymbol{\sigma}} \end{bmatrix}, \mathbf{C}_N \right), \tag{3.14}$$

where $\boldsymbol{\mu}_{\boldsymbol{\sigma}}$ is the model mean for the frequency discrimination features for targets, and $\mathbf{C}_N$ is the model noise covariance. For use in the PF, we approximate $p(\mathbf{z}_{k,l}|\mathbf{x}_{k,l})$ using with maximum likelihood estimation, as

$$p(\mathbf{z}_{k,l}|\mathbf{x}_{k,l}) \approx p(\mathbf{z}_{k,l}|\mathbf{x}_{k,l}, \hat{\mathbf{q}}_{k,l}) \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{D}(\mathbf{x}_{k,l})\hat{\mathbf{q}}_{k,l} \\ \boldsymbol{\mu}_{\boldsymbol{\sigma}} \end{bmatrix}, \mathbf{C}_N \right), \tag{3.15}$$

where $\hat{\mathbf{q}}_{k,l} = (\mathbf{D}(\mathbf{x}_{k,l})^T\mathbf{D}(\mathbf{x}_{k,l}))^{-1}\mathbf{D}(\mathbf{x}_{k,l})^T\mathbf{z}_{k,l}$ by least squares. These models are applied directly to the particle filter, probabilistic data association filter (PDAF) and PHDF in Algorithms 1, 2 and 3, respectively, in Chapter 2.

3.2.3 Tracking an Unknown Number of Sources

The particle filter can track neural activity sources for a fixed and known number of sources when the association of measurement to source is known. If this association is not known, the PDAF can be used to perform the tracking. However, it is often the case that the number of sources varies over time, which is not accounted for by the PDAF algorithm. When the number of sources varies with time, the PHDF can be used as it can estimate the number of sources at different time steps [22, 23,

36, 37, 85, 86]. Some results using the PHDF in our research group can be found in [22, 23, 36, 37, 115].

3.3 Artifact Suppression

3.3.1 Characterization of Artifacts

After applying ICA to EEG data, we obtain separate measurements, some of which originate from neural activity, and some of which originate from other non-neural electrical activity. The measurements that originate from non-neural electrical activity are artifacts that need to be suppressed before tracking neural sources. There are many kinds of artifact signals, and many different methods were designed to identify or suppress them [38–48]. Physiologic artifacts include muscle movements, breathing and heartbeat, while non-physiologic artifacts include the signal leaking from the power line, electrode popping, sweat on the electrode, or movement of the patient or bed. The artifact measurements can reduce the neural source estimation performance of the tracker unless methods are integrated with the tracker to account for the spurious artifact measurements. Some methods used for artifact suppression include ICA followed by pattern recognition [39, 49], wavelets [45], regression techniques [41], and using principle component analysis along with minimum norm estimation [40]. Noise canceling filters were used in real time in [47, 48] and blind source separation was applied in [46].

We considered artifact characteristics by inspection in order to obtain a model for the artifact distribution for use in the measurement model. Using the real dataset provided by EEGLAB [114], we developed a model for discriminating between neural activity and artifacts. The model depends on the amplitude parameter vector $\mathbf{s}_{k,l}$, though it is possible to incorporate the leadfield components in $\mathbf{A}_{k,l}$ as well. Our

chosen artifact discrimination statistic for the measurement model in Equation (3.13) is given by

$$\sigma_{k,l} = \frac{\int_4^{25} |\mathbf{S}_{k,l}(f)|^2 \mathrm{d}f}{\int_0^4 |\mathbf{S}_{k,l}(f)|^2 \mathrm{d}f + \int_{25}^{64} |\mathbf{S}_{k,l}(f)|^2 \mathrm{d}f}, \tag{3.16}$$

where $\mathbf{S}_{k,l}(f)$ is the Fourier transform of $\mathbf{s}_{k,l}(t)$. The model statistic is computed by approximating the integrals with sums, and it assumes that the neural source activity occurs between 4 Hz and 25 Hz in the frequency domain.

3.3.2 Suppression of Artifacts Based on Data Association

Our artifact suppression method is based on the use of the probabilistic data association filter (PDAF) after the EEG recordings are processing using ICA [3, 4]. The PDAF accounts for spurious measurements in its model by assuming measurements are not associated to sources, and it allows for some measurements formed by artifacts. Models for sources and artifacts are required, so we form an artifact model based on the frequency domain properties of the artifacts, as in the previous section. The PDAF uses the model to decide whether the measurements are more likely to be sources than artifacts before estimating the source locations. Note that our approach does not require any further pre-processing of the measurements after ICA, unlike other methods that require pre-processing to suppress artifact components before tracking.

The artifact model we considered in Equation (3.16) takes into consideration that the neural activity based measurements have high power in the alpha and beta frequency bands. We assume that the patient is awake, so there is not significant low frequency delta waves, which are strong during sleep [116]. We thus assume that strong power in very low frequencies is likely caused by eye artifacts, which have strong low frequency content. Using this frequency-based characterization for arti-

facts and neural activity, we can obtain estimates of the mean and variance of the discrimination statistic as well as the leadfield values for sources and artifacts needed for use in the PDAF model [4]. For the PDAF algorithm, described in Chapter 2, we model the artifacts as clutter distributed uniformly over the the measurement region; λ is the average number of artifacts and ϱ is the average density in the measurement space that is calculated using the volume of the measurement space where artifacts take on values, $\mathcal{V}$. In the EEG model, the measurements consist of a column vector of mixture coefficients $\mathbf{A}_{k,\text{artifact}}$ appended with an artifact discrimination statistic $\sigma_{k,\text{artifact}}$. Denote the range of values the mixture coefficients $a_{k,m,\text{artifact}}$ take on for M sensors as $a_{\text{range,artifact}}$, and the range of values the the discrimination statistic takes on for artifacts as $\sigma_{\text{range,artifact}}$ Then, the uniform volume artifact measurements appear within, $\mathcal{V} = (a_{\text{range,artifact}})^M \sigma_{\text{range,artifact}}$. Then, $\varrho = \frac{\lambda}{\mathcal{V}}$. Plugging ϱ into the PDAF allows for tracking neural sources while suppressing artifacts [4].

3.4 Tracking Performance for an Unknown Number of Neural Sources with Artifacts

We test this algorithm on synthetic data [4]. This data is sampled at 1,024 Hz. We place 2 dipole sources uniformly over the model head region, an upper hemisphere of radius 85 mm. Their position and orientation are unchanged over second long intervals, which is the duration of a particle filter time step. Their orientation is uniformly distributed over all possible angles at each second long set of 1,024 samples. Each position is updated every second with a 10 mm random walk, but prevented from leaving the hemisphere. Each source has an amplitude that varies in time at each sample.

Each of the synthetic sources and synthetic artifact signals are assumed to have two frequency components to make the sources temporally independent. One frequency is

within the range of neural activity frequencies, between theta and low beta frequencies (4 Hz to 25 Hz) [116]. Each source and artifact also has a frequency below 4 Hz or above 25 Hz up to 64 Hz. The frequency range can be different for each artifact and change over time; both frequencies stay the same over time for the sources.

To form the model pdf of the discrimination function, $\boldsymbol{\sigma}_{k,l}$, for sources and artifacts, we used the real EEG data provided on the website with EEGLAB [114]. We performed ICA over 240 seconds of data, and we used the scalp maps, the residual variance from the DIPFIT model provided by EEGLAB [117], and the estimated positions from DIPFIT to manually separate the ICA components into neural activity and artifacts. We estimated value for the discrimination function was 0.2 and 1.7 for artifacts, and between 2 and 4 for neural activity.

In constructing the synthetic data, the discrimination function value was drawn uniformly on the interval [2, 4] for sources, and from a normal distribution of mean 0.9 and variance 0.071 for artifacts. The frequency components are scaled appropriately to fit the value. New artifacts are drawn at each time step, and the number of artifacts is Poisson distributed with mean 2. The mixed measurements also included some Gaussian noise. Note that ICA decomposition adds additional error and is the primary cause of separated measurement noise.

To determine the model parameters, we drew 4,000 components this way and examined the signals obtained after ICA, comparing the true mixed signals to the ones extracted by ICA, to determine the approximate artifact density and a reasonable range for modeling the source measurement variance. For sources, the frequency function is modeled as a normal distribution with mean 3 and variance 0.11, so that the distribution falls on the interval [2,4]. The state model is a 10 mm random walk. The principal component analysis dimension reduction threshold, which determines the number of components ICA separates, was set based on test runs, so that the

cutoff was placed at the "elbow" of the plot, where the steepness decreases suddenly as the remaining eigenvectors extract noise. The probability of detection is set at 0.99.

We performed 100 Monte Carlo simulations of synthetic data with two sources, with different sources each time. We present a plot of a typical run for two dipole sources in Figure 3.7. In this plot, the hollow markers connected with solid lines are the true positions, and the cross-shaped marked markers connected by dashed lines are the estimates positions. The root mean-squared error (RMSE) for the source position across the Monte Carlo runs is provided in Figure 3.8. The estimation error falls off over time as the PDAF locks onto the sources, dropping below 8 mm, assuming a head of radius 85 mm. Note that is difficult to ensure that signals are properly constructed to be independent. The individual runs indicate that sources whose frequencies are harmonic with other sources present more of a challenge for ICA.

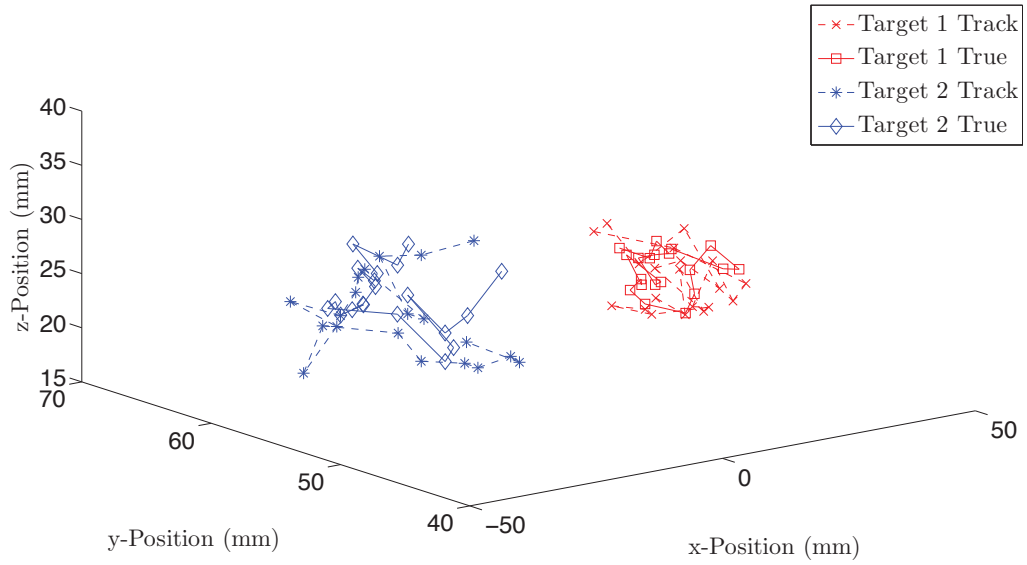


Figure 3.7: Typical Run of PDAF on Synthetic Data with Two Sources [4].

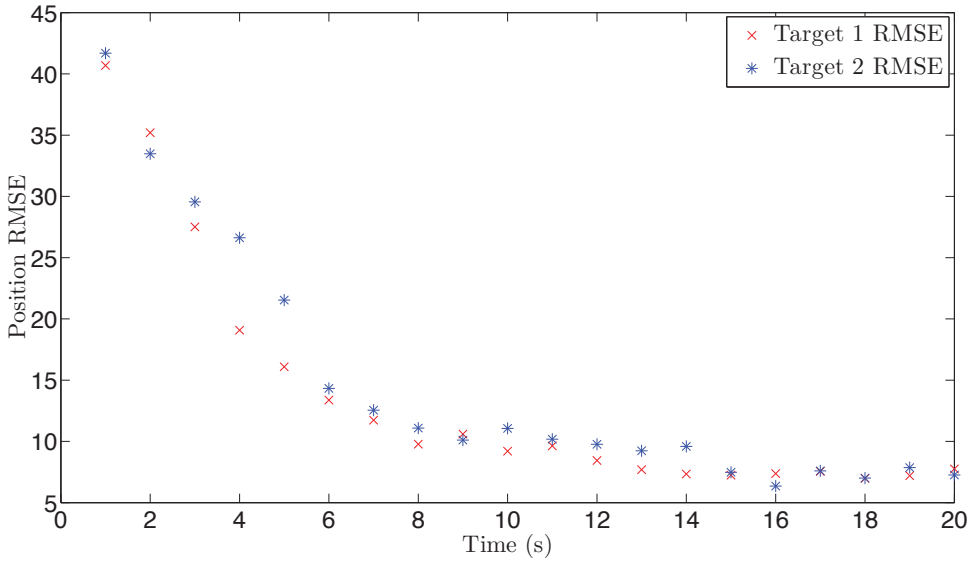


Figure 3.8: RMSE for 100 Monte Carlo Runs with Two Sources [4].

Chapter 4

ADAPTIVE CLUSTERING OF TASKS WITH NEUROSTIMULATION

4.1 Neurological Disorders and Neurostimulations

Recently, neurostimulation has been used to treat disorders such as Parkinson's disease and essential tremor. Patients with Parkinson's disease suffer from severe tremors as the disease progresses, slowly making movements more difficult. The primary symptoms listed on the Mayo Clinic website are: "tremor, slowed movement, rigid muscles, impaired posture and balance, loss of automatic movements, speech changes, and writing changes [6]." The tremor tends to show at rest, and it can early on take the form of rolling the thumb and first finger together. Shorter steps and dragging feet are common. Parkinson's disease patients can have reduced automatic movements, possibly blinking less or no longer swinging their arms while walking. Monotone voice or slurring of words can be a symptom, as well as loss of penmanship. The disease can also bring dementia, depression, difficulty swallowing, sleep disorders, bladder and digestive issues, loss of sense of smell, and other symptoms [6, 63, 65].

Parkinson's disease is caused by the death of neurons that produce dopamine. The lost dopamine directly causes the symptoms of Parkinson's disease. The exact reasons for the onset of Parkinson's disease are not certain, but genetics and the environment both play a role. It tends to affect men and usually develops at age 60 or older. It is incurable, but the symptoms can be managed with medication. However, as the disease advances, the medications can stop working as well as they worked when the disease began. This is when a treatment called deep brain stimulation (DBS) can be employed [6, 7, 53, 54].

The treatment of Parkinson's disease is very individualized, as treatments affect everyone differently. Dopamine replacement can be performed with a combination of levidopa and carbidopa. Over time, the symptoms return sooner after doses and patients can experience significant "off" times before it is time for the next dose. Some drugs can extend the duration of levidopa action when added to the treatment. Dopamine mimics can also be used to manage symptoms and amantadine can reduce tremors in some cases [6, 56].

There are also surgical techniques to manage the symptoms of Parkinson's disease. Older surgical techniques, such as pallidotomies and thalamotomies were replaced by deep brain stimulation (DBS) [7, 55]. The older techniques left permanent lesions in particular locations, as opposed to DBS, which is reversible and adjustable, characteristics important to managing a changing disease like Parkinson's disease [7]. Deep brain stimulation uses a surgical implant that is reminiscent of a pacemaker for the brain instead of the heart. Parkinson's disease symptoms were shown to improve after placing electrodes at particular points in the brain and applying certain waveforms electrically. Although DBS carries risks just like any other surgery, it is a powerful technique for certain cases of Parkinson's disease, such as those with severe medication side effects [6, 7, 53, 54].

Research on Parkinson's disease using electroencephalogram (EEG) measurements suggest possible changes in the signal power both in the beta band (13 - 30 Hz) and gamma band (30 - 100 Hz) frequencies [16, 28, 57–67]. In particular, significant fluctuations in beta power were observed during motor and language tasks in [67]. Repetitive motion tasks and tasks with both speech and motor caused significant increases in beta power, and these tasks can make the Parkinson's disease tremors more evident. These beta power changes seem to be related to the pathology of Parkinson's disease [16, 28, 57–67].

In the next section, we present some clustering methods that we apply to discriminate different tasks performed by Parkinson's disease patients in later sections.

4.2 Clustering Approaches

4.2.1 Gaussian Mixture Modeling

The Gaussian mixture model (GMM) is fundamentally a probabilistic model for measured data that considers that all possible measurement vectors $\mathbf{x}$ are obtained from a finite number of Gaussian distributions whose parameters need to be estimated. The multivariate Gaussian pdf over a k-dimensional space has two parameters, the mean, $\boldsymbol{\mu} \in \mathbb{R}^k$ and covariance, $\mathbf{C} \in \mathbb{R}^{k \times k}$, where $\mathbb{R}$ is the set of real numbers, and is given by

$$p(\mathbf{x}; \boldsymbol{\mu}, \mathbf{C}) = \frac{1}{(2\pi)^{(k/2)} \det(\mathbf{C})^{(1/2)}} \exp(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \mathbf{C}^{-1}(\mathbf{x} - \boldsymbol{\mu})), \tag{4.1}$$

where $\det(\mathbf{C})$ is the determinant of matrix $\mathbf{C}$. A Gaussian mixture model is just a weighted combination of Gaussian pdfs. If there are M components on the GMM, then we consider the indicator variable $z_m, m = 1, ..., M$, such that only one indicator variable is 1 and the others are 0. The weight of the mth Gaussian pdf is then given by $\alpha_m = \Pr(z_m = 1)$, where $\Pr(\cdot)$ denotes probability. As the GMM is also a pdf, all the weights add to 1, $\sum_{m=1}^{M} \alpha_m = 1$, and the pdf of the GMM can be written in the k-dimensional space using $\boldsymbol{\Theta} = \{\alpha_1, ..., \alpha_M, \boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_M, \mathbf{C}_1, ..., \mathbf{C}_M\}$, as

$$p(\mathbf{x}|\boldsymbol{\Theta}) = \sum_{m=1}^{M} \frac{\alpha_m}{(2\pi)^{(k/2)} \det(\mathbf{C}_m)^{(1/2)}} \exp(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_m)^T \mathbf{C}_m^{-1}(\mathbf{x} - \boldsymbol{\mu}_m)), \tag{4.2}$$

or more compactly as

$$p(\mathbf{x}|\boldsymbol{\Theta}) = \sum_{m=1}^{M} \alpha_m p(\mathbf{x}|z_m, \boldsymbol{\mu}_m, \mathbf{C}_m), \tag{4.3}$$

with

$$p(\mathbf{x}|z_m, \boldsymbol{\mu}_m, \mathbf{C}_m) = \frac{1}{(2\pi)^{(k/2)} \det(\mathbf{C}_m)^{(1/2)}} \exp(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_m)^T \mathbf{C}_m^{-1}(\mathbf{x} - \boldsymbol{\mu}_m)). \tag{4.4}$$

The GMM can be used as a classifier for different measurements $\mathbf{x}_n, n = 1, \ldots, N$, by calculating a classification weight for $\mathbf{x}_n$ for each of the M GMM components. The GMM construction does not make a hard decision until the very end, when the highest classification weight indicates in which class to assign the measurement. The weights can be found using Bayes' rule for each data point $\mathbf{x}_n$ and class m as:

$$w_{nm} = \Pr(z_{nm} = 1|\mathbf{x}_n, \mathbf{\Theta}) = \frac{p_m(\mathbf{x}_n|z_m, \boldsymbol{\mu}_m, \mathbf{C}_m)\alpha_m}{\sum_{\ell=1}^{M} p_\ell(\mathbf{x}_n|z_\ell, \boldsymbol{\mu}_\ell, \mathbf{C}_\ell)\alpha_\ell}. \tag{4.5}$$

Using the GMM for classification requires that the number of classes, M, is known. However, the exact type of each of the M classes is not required, thus leading to an unsupervised classifier that does not require training. The algorithm to find the GMM looks to maximize the probability of receiving the data set from the GMM pdf. This is done approximately using the expectation maximization (EM) algorithm that is provided in Algorithms 5 - 7. The stopping criteria of the EM is when the log-likelihood changes less than a certain threshold between iterations. The log-likelihood at iteration j is $l_j = \sum_{n=1}^{N} \log[p(\mathbf{x}_n|\Theta)]$

Note that the derivation here mirrors the one in [93].

4.2.2 Dirichlet Process Gaussian Mixture Models

As discussed in Section 4.2.1, the GMM algorithms requires knowledge of the number of classes when used for clustering the provided measurements. In many cases, however, the number of classes is unknown. As such, we employ the Dirichlet Process Gaussian Mixture Model (DP-GMM) [94–102]. Since the DP-GMM decides the number of classes itself, it can reclassify when new data is provided. This is important for our application as it can be used to monitor changes in the patient's brain activity. The Dirichlet process can be thought of as a distribution on distributions [97]. Draws from a Dirichlet process generate discrete distributions almost surely. Specifically, we

Algorithm 5 Expectation Maximization for Gaussian Mixture Model.

Initialization

$\{\mathbf{x}_1, ..., \mathbf{x}_N\}$ is the data to classify

There are M mixture components

$j = 0$

Initialization option 1:

Start with all initial w_{nm}^0

Enter while loop at M step

Initialization option 2:

Start with all initial $\alpha_m^0, \boldsymbol{\mu}_m^0, \mathbf{C}_m^0$

Enter while loop at E step

while $|l_j - l_{j-1}| \leq \epsilon$ **do**

Perform E step (Algorithm 6):

Perform M step (Algorithm 7):

Iteration:

$j = j + 1$

end while

Output GMM parameters at last iteration

Algorithm 6 Expectation (E) Step.

for $n = 1 : N$ **do**

for $m = 1 : M$ **do**

$w_{nm}^j = \frac{p_m^j(\mathbf{x}_n|z_m^j, \boldsymbol{\mu}_m^j, \mathbf{C}_m^j)\alpha_m^j}{\sum_{\ell=1}^M p_\ell^j(\mathbf{x}_n|z_\ell^j, \boldsymbol{\mu}_\ell^j, \mathbf{C}_\ell^j)\alpha_\ell^j}$

end for

end for

Algorithm 7 M Step.

for $m = 1 : M$ **do**

$N_m^j = \sum_{i=1}^{N} w_{nm}^j$

end for

for $m = 1 : M$ **do**

$\alpha_m^j = \frac{N_m^j}{N}$

end for

for $m = 1 : M$ **do**

$\boldsymbol{\mu}_m^j = \frac{1}{N_m^j} \sum_{n=1}^{N} w_{nm} \mathbf{x}_n$

end for

for $m = 1 : M$ **do**

$\mathbf{C}_m^j = \frac{1}{N_m^j} \sum_{n=1}^{N} w_{nm} (\mathbf{x}_n - \boldsymbol{\mu}_m)(\mathbf{x}_n - \boldsymbol{\mu}_m)^T$

end for

consider a Dirichlet process G as

$$G \sim \ DP(\alpha, G_0), \tag{4.6}$$

where α is called the concentration parameter and G_0 is some base distribution, which may be continuous. In clustering applications, this is a distribution on a vector of parameters. For the a DP-GMM, it is a distribution on the means and covariances for the mixture components. The mixture model has some distribution that is parameterized by values that are in the space that G_0 is defined on. A way to visualize the Dirichlet Process (DP) is by using the description of a Chinese restaurant process [97–99]. It proceeds as follows: there is a restaurant (the space that the base distribution is defined on) with tables (points in the space) for customers to sit at. Each new customer may join any table or start a new one. The probability of starting a new table is proportional to the concentration parameter, α, and the probability to

join an existing table is proportional to the number of customers already there. The process starts with the first customer selecting a table (a draw from G_0 is made, which is one of the discrete points in the distribution G). The second customer may join this table with probability $\frac{1}{1+\alpha}$ or select a new one with probability $\frac{\alpha}{1+\alpha}$. If the second customer joins the first table, then the probability for customer three to join the first table (a weight is added to the point) is $\frac{2}{2+\alpha}$, and to select a new table (a new point is drawn from G_0) is $\frac{\alpha}{2+\alpha}$. If the second customer formed Table 2, then the probability to join Table 1 is $\frac{1}{2+\alpha}$, to join Table 2 is $\frac{1}{2+\alpha}$, and to select a new table is $\frac{\alpha}{2+\alpha}$. This process continues, providing a clustering effect and a discrete distribution that resembles G_0. The cluster associations are drawn for the N measurements to construct mixture components. This construction makes G a discrete probability distribution for clusters: each point with non-zero probability in G gives the parameters for a component in the mixture model distribution, and the probability at that point is the weight of that distribution within the mixture model. A measurement drawn from a mixture model can be obtained by first drawing which mixture component it came from, and then the measurement is drawn from the distribution specified by the mixture component parameters. The parameters correspond to the means and covariances for the DP-GMM, so that the nth draw from G is

$$\boldsymbol{\theta}_n|G \sim \ G, n = 1, ..., N, \tag{4.7}$$

with $\boldsymbol{\theta}_n = [\boldsymbol{\mu}_n, \mathbf{C}_n]$, associated to each measurement and not unique, due to the discreteness of G.

The measurement set, $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N$ can be drawn from each parameter vector, which is also a cluster association due to the discreteness of G

$$\mathbf{x}_n|\boldsymbol{\theta}_n \sim \ \mathcal{N}(\boldsymbol{\mu}_n, \mathbf{C}_n), n = 1, ..., N. \tag{4.8}$$

Since an infinite mixture model can be described via the Dirichlet process, the challenge then becomes to estimate the mixture components and weights. To start with, we explicitly describe G, which has a simple stick-breaking construction:

$$G(\boldsymbol{\theta}) = \sum_{m=1}^{\infty} w_m \delta(\boldsymbol{\theta} - \boldsymbol{\theta}_m). \tag{4.9}$$

$$w_m \sim \; v_m \prod_{i=1}^{m-1} (1 - v_i), m = 1, ..., \infty. \tag{4.10}$$

$$v_i \sim \; \text{Beta}(1, \alpha), i = 1, ..., \infty. \tag{4.11}$$

$$\boldsymbol{\theta}_m \sim \; G_0, m = 1, ..., \infty. \tag{4.12}$$

with $w_m = \Pr(\boldsymbol{\theta} = \boldsymbol{\theta}_m)$, and all $\boldsymbol{\theta}_m$ are not associated to any measurement and unique, being drawn from the continuous G_0.

To make the construction more explicit, we eliminate G by introducing category variables, c_n, which associate the measurements $\mathbf{x}_n$ to parameter vectors $\boldsymbol{\theta}_m$. Each discrete point in G with non-zero probability is associated to a positive integer category variable. We designate that discrete distribution on the counting numbers as Discrete. This changes the final model of the data, $\mathbf{x}_n, n = 1, ..., N$ to:

$$\mathbf{x}_n | c_n \sim \; p(\mathbf{x}_n | \boldsymbol{\theta}_{c_n}), n = 1, ..., N. \tag{4.13}$$

$$c_n | \mathbf{w} \sim \; \text{Discrete}(\mathbf{w}), n = 1, ..., N. \tag{4.14}$$

$$w_m = v_m \prod_{i=1}^{m-1} (1 - v_i), m = 1, ..., \infty. \tag{4.15}$$

$$v_i \sim \text{ Beta}(1, \alpha), i = 1, ..., \infty. \tag{4.16}$$

$$\boldsymbol{\theta}_m \sim \ G_0, m = 1, ..., \infty. \tag{4.17}$$

This construction allows the data points to be assigned to the most likely mixture component. This way, the number of clusters does not have to be specified beforehand, since a possibly infinite number of clusters can be used. Additionally, only a finite number of clusters can have "large" weights, so for practical applications, the number of clusters can be reduced to achieve a certain level of error. If N is the number of data points and M is the number of clusters to truncate to, the truncation error is:

$$\epsilon = 4\, N \exp(-(M-1)/\alpha). \tag{4.18}$$

Note that the truncation limit should be higher than the number of clusters that are reasonable for the data set, so that there are still clusters available to allow fair movement through the space.

With a model established, we would like to estimate all of the parameter vectors $\boldsymbol{\theta}_m$ and associated weights w_m for $m = 1, ..., M$, where M is the truncation limit on the number of clusters, as well as each of the cluster associations for each data point, c_n for $n = 1, ..., N$, where N is the number of data points. Specifically, the posterior distribution, $p(\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_M, c_1, ..., c_N, w_1, ...w_M | \mathbf{x}_1, ..., \mathbf{x}_N)$, needs to be approximated. We can use the blocked Gibbs sampler [102, 118] to obtain samples from this distribution indirectly, under certain conditions. In order to do so, we start with some initial guess for all of the random variables involved, then sample certain blocks of random variables from their distribution conditioned on all of the other random variables and the data. A new sample is obtained after one full cycle of conditional

sampling. After a sufficient burn-in period, the samples will explore the posterior distribution. We can then use these samples to estimate all of the means, covariances, weights, and associations. One possible order is as follows: for iteration i,

$$\boldsymbol{\theta}_m^{(i)} \sim \; p(\boldsymbol{\theta}_m | c_1^{(i-1)}, ..., c_M^{(i-1)}, \mathbf{x}_1, ..., \mathbf{x}_N), m = 1, ..., M. \tag{4.19}$$

$$c_n^{(i)} \sim \; p(c_n | \boldsymbol{\theta}_1^{(i)}, ..., \boldsymbol{\theta}_M^{(i)}, c_1^{(i-1)}, ..., c_M^{(i-1)}, \mathbf{x}_1, ..., \mathbf{x}_N), n = 1, ..., N. \tag{4.20}$$

$$w_m^{(i)} \sim \; p(w_m | c_1^{(i)}, ..., c_M^{(i)}), m = 1, ..., M. \tag{4.21}$$

Some simplifications from conditional independence have been applied. By choosing conjugate priors, we can obtain the following:

$$p(\boldsymbol{\theta}_m | c_1, ..., c_M, \mathbf{x}_1, ..., \mathbf{x}_N) \propto G_0(\boldsymbol{\theta}_m) \prod_{n:c_n=m} p(\mathbf{x}_n | \boldsymbol{\theta}_m), m = 1, ..., M. \tag{4.22}$$

$$p(c_n | \boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_M, w_1, ..., w_M, \mathbf{x}_1, ..., \mathbf{x}_N) = \sum_{m=1}^{M} \left(w_m p(\mathbf{x}_n | \boldsymbol{\theta}_m) \right) \delta(c_n - m), \quad n = 1, ..., N. \tag{4.23}$$

$$p(v_m | c_1^{(i)}, ..., c_M^{(i)}) = \text{Beta}(1 + N_m^*, \alpha + \sum_{m'=m+1}^{M} N_{m'}^*), m = 1, ..., M. \tag{4.24}$$

$$w_m = v_m \prod_{j=1}^{m-1} (1 - v_j), m = 1, ..., M. \tag{4.25}$$

In these expressions, $n : c_n = m$ is the values of n such that $c_n = m$, N_m^* is the number of such values of n, and α is the concentration parameter. The pdf for the Beta distribution is: $p(x; a, b) = \frac{x^{(a-1)}(1-x)^{(b-1)}\Gamma(a+b)}{\Gamma(a)\Gamma(b)}$. The conjugate prior for means and covariances is the Normal-Wishart distribution, $\mathcal{NW}$. By using that as the prior for $\boldsymbol{\theta}$, the prior and posterior conditional distributions are:

$$G_0(\boldsymbol{\theta}) \cong \mathcal{NW}(\boldsymbol{\mu}, \boldsymbol{\Sigma}^{-1}; \boldsymbol{\mu}_{\mathcal{N}}, \tau_{\mathcal{N}}, \boldsymbol{\Psi}_{\mathcal{W}}, \nu_{\mathcal{W}}), \tag{4.26}$$

$$p(\boldsymbol{\theta}|c_1, ..., c_M, \mathbf{x}_1, ..., \mathbf{x}_N) \cong \mathcal{NW}(\boldsymbol{\mu}, \boldsymbol{\Sigma}^{-1}; \tilde{\boldsymbol{\mu}}_{\mathcal{N}}, \tilde{\tau}_{\mathcal{N}}, \tilde{\boldsymbol{\Psi}}_{\mathcal{W}}, \tilde{\nu}_{\mathcal{W}}). \tag{4.27}$$

Here, $\boldsymbol{\mu}_{\mathcal{N}}$ is a real vector of length N, $\tau_{\mathcal{N}}$ is positive and real, $\boldsymbol{\Psi}_{\mathcal{W}}$ is an $N \times N$ precision matrix, and $\nu_{\mathcal{W}}$ is the number of degrees of freedom, greater than $N - 1$. The equations for $\tilde{\boldsymbol{\mu}}_{\mathcal{N}}$, $\tilde{\tau}_{\mathcal{N}}$, $\tilde{\boldsymbol{\Psi}}_{\mathcal{W}}$, and $\tilde{\nu}_{\mathcal{W}}$ are:

$$\tilde{\boldsymbol{\mu}}_{\mathcal{N}} = \frac{\tau_{\mathcal{N}}\boldsymbol{\mu}_{\mathcal{N}} + N\boldsymbol{\mu}_{\mathbf{x}}}{\tau_{\mathcal{N}} + N}, \tag{4.28}$$

$$\tilde{\tau}_{\mathcal{N}} = \tau_{\mathcal{N}} + N, \tag{4.29}$$

$$\tilde{\boldsymbol{\Psi}}_{\mathcal{W}} = \boldsymbol{\Psi}_{\mathcal{W}} + \Sigma_{\mathbf{x}} + \frac{\tau_{\mathcal{N}} N}{\tau_{\mathcal{N}} + N}(\boldsymbol{\mu}_{\mathcal{N}} - \boldsymbol{\mu}_{\mathbf{x}})(\boldsymbol{\mu}_{\mathcal{N}} - \boldsymbol{\mu}_{\mathbf{x}})^T, \tag{4.30}$$

and

$$\tilde{\nu}_{\mathcal{W}} = \nu_{\mathcal{W}} + N, \tag{4.31}$$

where $\boldsymbol{\mu}_{\mathbf{x}}$ and $\Sigma_{\mathbf{x}}$ are the mean and covariance of the data, $\mathbf{x}_1, ..., \mathbf{x}_N$. Algorithm 8 describes the blocked Gibbs sampler algorithm, following [97].

Much of the notation is borrowed from [95–97]. A deeper discussion of the DP-GMM can be found in [100–102].

4.3 Clustering Behavior Tasks of Parkinson's Disease Patients

4.3.1 Neural Signals and Time-Frequency Features

As discussed in Section 4.1, Parkinson's disease patients that have undergone DBS may suffer from cognitive, speech, and balance side effects. By customizing DBS therapy to a patient's task, these stimulation side effects may arise only when they are non-detrimental to the patient's current goals. It is thus important to be able to use adaptive learning methods to identify different behavioral tasks of patients

Algorithm 8 Blocked Gibbs Sampler for Dirichlet Process Gaussian Mixture Model.

Inputs: Data $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ with each vector dimension k, Dirichlet process innovation parameter α, DP cluster truncation value M, Normal-Wishart hyper-parameters $\boldsymbol{\mu}_{\mathcal{N}}, \tau_{\mathcal{N}}, \boldsymbol{\Psi}_{\mathcal{W}}, \nu_{\mathcal{W}}$.

Outputs: L (specified) samples of $\{\boldsymbol{\mu}_m^{(i)}, \boldsymbol{\sigma}_m^{-1(i)}, c_m^{(i)}, w_m^{(i)}\}$ for $m = 1, ..., M$.

Loop for some duration of burn-in followed by useful samples as required, Gibbs iterations i=1,2,...

Update $\boldsymbol{\theta}_m^{(i)} = \{\boldsymbol{\mu}_m^{(i)}, \boldsymbol{\Sigma}_m^{-1(i)}\} \sim\ p(\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m^{-1} | \mathbf{c}^{(i-1)}, \mathbf{X}), m = 1, ..., M$:

▷ Let $\mathbf{X}_m = \{\mathbf{x}_n : c_n^{(i-1)} = m\}$ and $N_m = \text{cardinality}(\mathbf{X}_m), m = 1, ..., M$.

▷ For all $m = 1, ..., M$, compute:

$$\boldsymbol{\mu}_{\mathbf{x}_m} = \frac{1}{N_m} \sum_{n: c_n^{(i-1)} = m} \mathbf{x}_n \tag{4.32}$$

$$\boldsymbol{\Sigma}_{\mathbf{x}_m} = \frac{1}{N_m} \sum_{n: c_n^{(i-1)} = m} (\mathbf{x}_n - \boldsymbol{\mu}_{\mathbf{x}_m})^2 \tag{4.33}$$

$$\tilde{\boldsymbol{\mu}}_{\mathcal{N},m} = \frac{\tau_{\mathcal{N}} \boldsymbol{\mu}_{\mathcal{N}} + N_m \boldsymbol{\mu}_{\mathbf{x}}}{\tau_{\mathcal{N}} + N_m} \tag{4.34}$$

$$\tilde{\tau}_{\mathcal{N},m} = \tau_{\mathcal{N}} + N_m \tag{4.35}$$

$$\tilde{\boldsymbol{\Psi}}_{\mathcal{W},m} = \boldsymbol{\Psi}_{\mathcal{W}} + \Sigma_{\mathbf{x}_m} + \frac{\tau_{\mathcal{N}} N_m}{\tau_{\mathcal{N}} + N_m} (\boldsymbol{\mu}_{\mathcal{N}} - \boldsymbol{\mu}_{\mathbf{x}_m})(\boldsymbol{\mu}_{\mathcal{N}} - \boldsymbol{\mu}_{\mathbf{x}_m})^T \tag{4.36}$$

$$\tilde{\nu}_{\mathcal{W},m} = \nu_{\mathcal{W}} + N_m, \tag{4.37}$$

▷ Draw sample $\boldsymbol{\Sigma}_m^{-1(i)}$ from Wishart distribution: $\mathcal{W}(\boldsymbol{\Sigma}_m^{-1}; \tilde{\boldsymbol{\Psi}}_{\mathcal{W},m}, \tilde{\nu}_{\mathcal{W},m}$

▷ Draw sample $\boldsymbol{\mu}_m^{(i)}$ from Normal distribution: $\mathcal{N}(\boldsymbol{\mu}_m; \tilde{\boldsymbol{\mu}}_{\mathcal{N},m}, \frac{\boldsymbol{\Sigma}_m^{(i)}}{\tilde{\tau}_{\mathcal{N},m}})$

Update $c_n^{(i)} \sim\ p(c_n|\boldsymbol{\mu}^{(i)}, \boldsymbol{\Sigma}^{-1(i)}, \mathbf{w}^{(i-1)}, \mathbf{X}) n = 1, ..., N$:

▷ Let $q_{m,n} \cong w_m^{(i-1)} \mathcal{N}(\mathbf{x}_N; \boldsymbol{\mu}_m^{(i)}, \boldsymbol{\Sigma}_m^{(i)}), n = 1, ..., N$

▷ Normalize $q'_{m,n} = \frac{q_{m,n}}{\sum_{m=1}^{M} q_{m,n}}, m = 1, ..., M$ and $n = 1, ..., N$

▷ Draw sample $c_n^{(i)} \sim\ \sum_{m=1}^{M} q'_{m,n} \delta(c_n, m), n = 1, ..., N$

Update $w_m^{(i)} \sim\ p(w_m|\mathbf{c}^{(i)}), m = 1, ..., M$:

▷ Draw sample $v_m \sim\ \text{Beta}(1 + N_m^*, \alpha + \sum_{m'=m+1}^{M} N_{m'}^*)$,

where $N_m^* \cong \text{cardinality}(\{n : c_n^{(i)} = m\}), m = 1, ..., M$

▷ Evaluate $w_m^{(i)} = v_m \prod_{j=1}^{m-1}(1 - v_j), m = 1, ..., M$.

with Parkinson's disease. In particular, local field potential (LFP) signals, collected during DBS implantation surgeries when a patient is performing various tasks, can be clustered and used to monitor changes in the severity of the patient's disease.

LFP recordings are often performed with either invasive microelectrodes or DBS leads and reflect oscillatory activity within nuclei of the basal ganglia. As long data records are normally collected, relevant features are extracted and processed for clustering to reduce computational complexity. As these signals are time-varying in nature, task informative time-frequency based features can be extracted to provide distinct patterns for the different behavior tasks. Such features can be obtained using the matching pursuit decomposition (MPD) algorithm [119–121]. Using the MPD, a signal can be decomposed into a linear expansion of Gaussian basis functions that are selected from a redundant basic dictionary. The dictionary signals are time-shifted, frequency-shifted and scaled versions of an elementary Gaussian atom; these basis functions are chosen in order to best match the signal structure. The features ob-

tained from the MPD are the time-shift, frequency-shift, scale change and weight coefficient of each selected Gaussian in the signal expansion.

The MPD feature extraction algorithm is described as follows. We consider an MPD dictionary $\mathcal{D}$ consisting of the Gaussian signals $g(t;\mathbf{q})$ that are defined as

$$g(t;\mathbf{q}) = \frac{1}{(\pi a)^{1/4}} \exp\left(\frac{-(t-\tau)^2}{2a}\right) \exp(-j2\pi\nu)\,, \tag{4.38}$$

where $\mathbf{q} = [\tau\ \nu\ a]^T$, T denotes vector transpose, and τ, ν, and a are the time shift, frequency shift, and scale change parameters, respectively, of a basic Gaussian function located at the time-frequency origin. Using the iterative MPD, a continuous-time signal $x(t)$ can be decomposed after P iterations as [119]

$$x(t) = \sum_{p=1}^{P-1} \alpha_p\, g_p(t) + r_N(t)\,, \tag{4.39}$$

where α_p is the corresponding weight coefficient $r_P(t)$ is the remainder signal, and $g_p(t) = g(t;\mathbf{q}_p)$ is the selected Gaussian signal with $\mathbf{q}_p = [\tau_p\ \nu_p\ a_p]^T$. Using $r_1(t) = x(t)$ for the first iteration, the best matched Gaussian signal $g_p(t)$ at the pth iteration is selected such that it results in maximum correlation with the remainder signal. Specifically,

$$g_p(t) = \underset{g(t;\mathbf{q})\in\mathcal{D}}{\arg\max} \left| \int_{-\infty}^{\infty} r_p(t) g(t;\mathbf{q}) dt \right|. \tag{4.40}$$

The weight coefficient is thus given by

$$\alpha_p = \int_{-\infty}^{\infty} r_p(t) g_p(t) dt\,, \tag{4.41}$$

and the new remainder is $r_{p+1}(t) = r_p(t) - \alpha_p g_p(t)$. The algorithm can be stopped when the remainder has sufficiently small energy.

The resulting feature vectors resulting from the MPD decomposition of $x(t)$, assuming P MPD iterations, correspond to the 4-dimensional vector $F_p = [\alpha_p\ \mathbf{q}_p]^T = [\alpha_p\ \tau_p\ \nu_p\ a_p]^T$, for $p = 1, \ldots, P$.

4.3.2 Problem Formulation of Behavioral Tasks

We consider LFP signals, from Parkinson's disease patients. The signals correspond to tasks that describe four types of behaviors: simple motor task ($m = 1$), language ($m = 2$), language with motor task ($m = 3$), and language without motor task ($m = 4$) [1]. The LFP signal set obtained from I experiments is given by $\mathcal{S} = \{\mathbf{s}_i^1 \ \mathbf{s}_i^2 \ \mathbf{s}_i^3 \ \mathbf{s}_i^4\}$, where $\mathbf{s}_i^m = [s_i^m(1) \ \ldots \ s_i^m(K)]^T$, $i = 1, \ldots, I$, corresponds to the signal vector for the mth task, $m = 1, \ldots, 4$, and $s_i^m(k)$ is the discretized LFP signal from the mth task at time sample k, $k = 1, \ldots, K$. The extracted MPD features for the ith experiment is given by $\mathcal{F} = \{\mathbf{F}_i^1 \ \mathbf{F}_i^2 \ \mathbf{F}_i^3 \ \mathbf{F}_i^4\}$, where $\mathbf{F}_i^m = [F_{i,1}^m \ \ldots \ F_{i,P}^m]^T$ is the MPD feature matrix extracted from the LFP signal $\mathbf{s}_i^m$. The feature vector from the pth iteration, $p = 1, \ldots, P$, is $F_{i,p}^m = [\alpha_{(i,p)}^m \ \tau_{(i,p)}^m \ \nu_{(i,p)}^m \ a_{(i,p)}^m]^T$, corresponding to the weight coefficient, time shift, frequency shift, and scale change parameters, respectively. Using these extracted features for clustering, the mth cluster corresponds to the mth task.

4.4 Adaptive Clustering Using Experimental LFP Signals

We consider a Parkinsonian LFP signal set provided by the Colorado Neurological Institute [122]. The signal set was obtained from twelve Parkinson's disease patients. The signal segments associated with different behavioral tasks were labeled by physicians during data collection. Equal numbers of experiments are taken for each task, using a sampling rate of 4 kHz, collecting $K = 2,000$ data samples for each LFP signal. The LFP signals are labeled based on the task the patient is performing by the experimenter. As listed before, the tasks are: simple motor task ($m = 1$), language with motor task ($m = 3$), and language without motor task ($m = 4$) [1]. Note that Task 2 combines tasks 3 and 4.

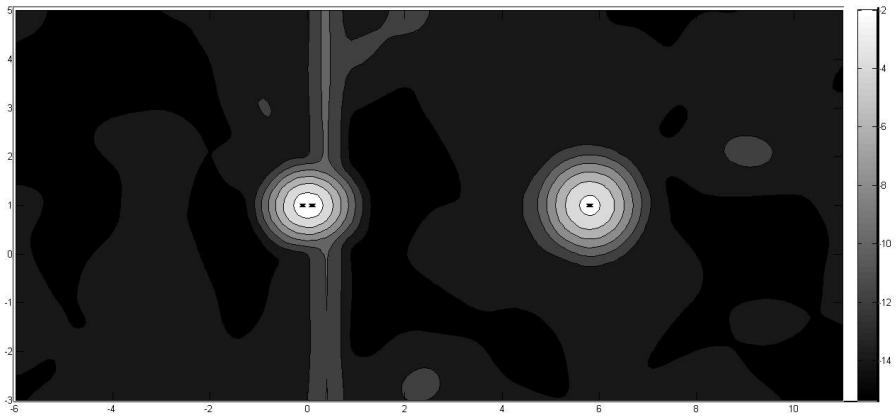
We performed adaptive clustering using the DP-GMM to classify Tasks 1, 3, and 4. The presence or absence of a motor component of a task is important for Parkinson's disease because fine motor control is a challenge for the patients. The data was collected with the DBS off, though obtained during DBS surgery. After feature extraction, we determined that the best feature vector resulted by using the weight coefficient and the time shift features, thus $F_{i,p}^{m} = [\alpha_{i,p}^{m} \ \tau_{i,p}^{m} \ ]$. Figures 4.1(a), 4.2(a), and 4.3(a) show the contour plots of the DP-GMM model, displaying the shapes of the DP-GMM classes, while Figures 4.1(b), 4.2(b), and 4.3(b) show the weight distribution of the resulting DP-GMM classes. Tables 4.1, 4.2, and 4.3 show the confusion matrices for the pairwise classifications. The parameters used in the DP-GMM were set as: innovation parameter 0.6, truncation error = $1e-2$, number of burn-in Gibbs iterations = 2000, and number of sampling Gibbs iterations = 1000.

	Pred. Class 1	Pred. Class 3
True Class 1	**0.92**	0.08
True Class 3	0.22	**0.78**

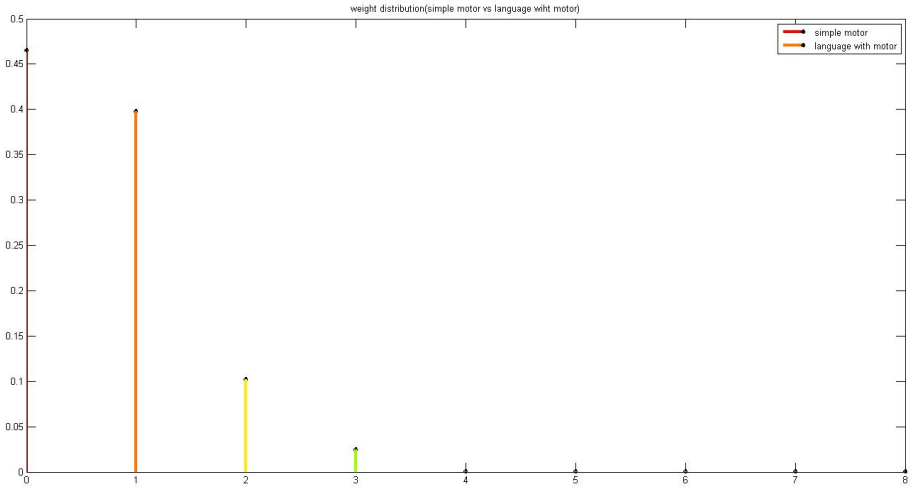
Table 4.1: Confusion Matrix, Simple Motor $m = 1$ vs. Language with Motor $m = 3$ [1].

	Pred. Class 1	Pred. Class 4
True Class 1	**0.84**	0.16
True Class 4	0.10	**0.90**

Table 4.2: Confusion Matrix, Simple Motor $m = 1$ vs. Language without Motor $m = 4$ [1].

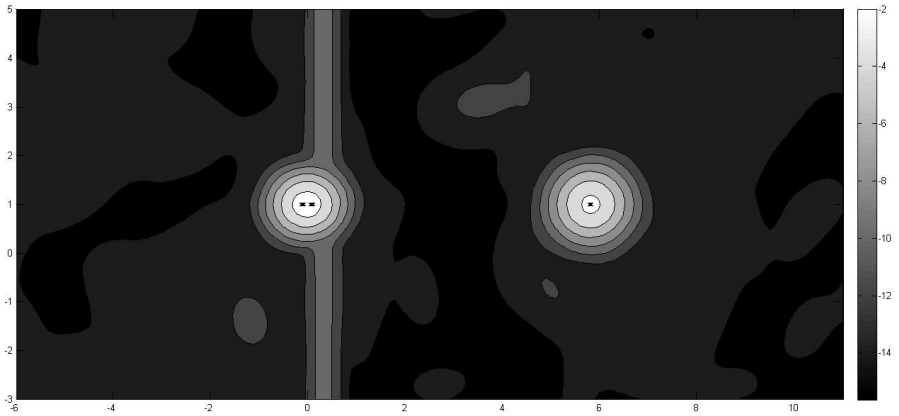


(a)

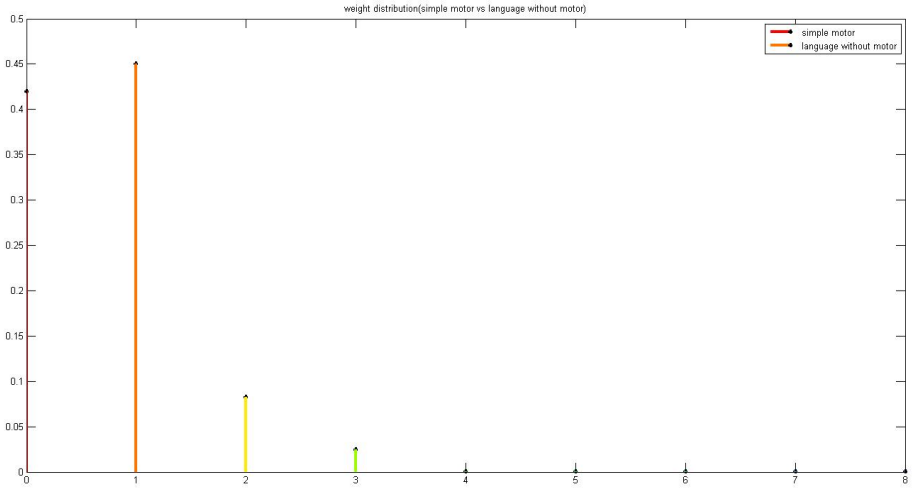


(b)

Figure 4.1: (a) Contour Plot of DP-GMM Output for $m = 1$ (Simple Motor Task) vs. $m = 3$ (Language with Motor Task). (b) Weight Distribution of the Two Classes [1].

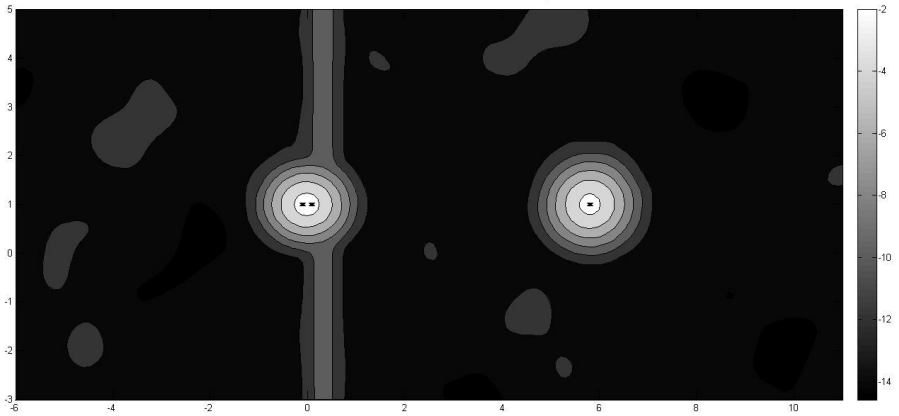


(a)

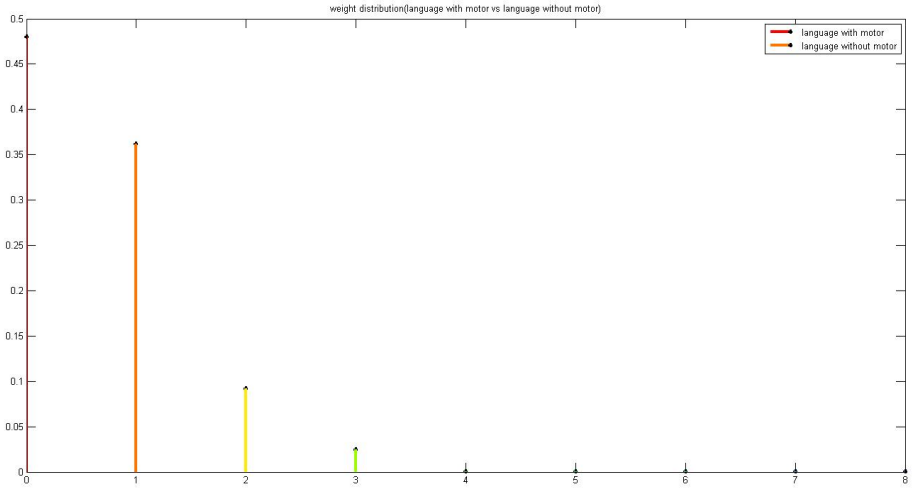


(b)

Figure 4.2: (a) Contour Plot of DP-GMM Output for $m = 1$ (Simple Motor Task) vs. $m = 4$ (Language without Motor Task). (b) Weight Distribution of the Two Classes [1].



(a)



(b)

Figure 4.3: (a) Contour Plot of DP-GMM Output for $m = 3$ (Language with Motor Task) vs. $m = 4$ (Language without Motor Task). (b) Weight Distribution of the Two Classes [1].

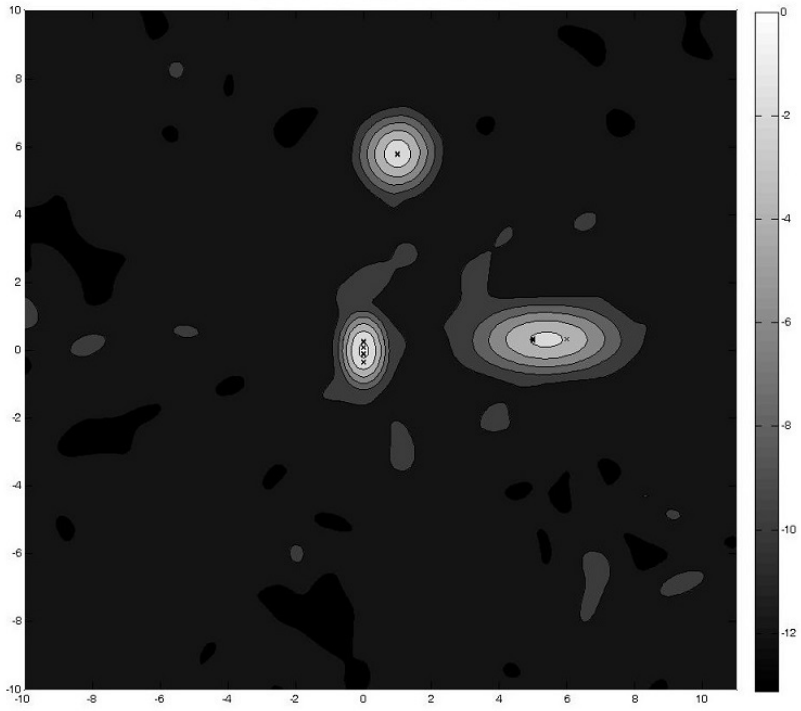
	Pred. Class 3	Pred. Class 4
True Class 3	**0.96**	0.04
True Class 4	0.28	**0.72**

Table 4.3: Confusion Matrix, Language with Motor $m = 3$ vs. Language without Motor $m = 4$ [1].

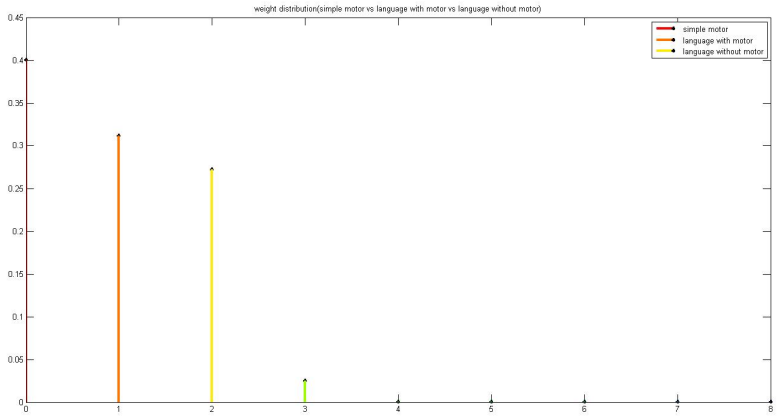
The contour plot for classifying all three classes is shown in Figure 4.4(a). The feature vectors resulting in the best classification, in this case, are $F_{i,p}^{m} = [\tau_{i,p}^{m} \ a_{i,p}^{m}]^{T}$, corresponding to time shift and scale change parameters. Figure 4.4(b) shows the weight distribution of the three classes. Table 4.4 is the confusion matrix for the three task classification.

	Pred. Class 1	Pred. Class 3	Pred. Class 4
True Class 1	**0.78**	0.11	0.11
True Class 3	0.035	**0.93**	0.035
True Class 4	0.095	0.095	**0.81**

Table 4.4: Confusion Matrix, Simple Motor $m = 1$ vs. Language with Motor $m = 3$ vs. Language without Motor m=4 [1].



(a)



(b)

Figure 4.4: (a) Contour Plot of DP-GMM Output for $m = 1$ (Simple Motor Task) vs. $m = 3$ (Language with Motor Task) vs. $m = 4$ (Language without Motor Task). (b) Weight Distribution of the Three Classes [1].

Chapter 5

ADAPTIVE CLUSTERING WITH NEUROSTIMULATION ARTIFACT SUPPRESSION

5.1 Structure of Deep Brain Stimulation Artifacts

The most common neurostimulation system currently available is deep brain stimulation (DBS), an advanced surgical technique that provides substantial relief of the motor signs of Parkinson's disease. DBS can be used to alleviate motor symptoms using high-frequency electrical stimulation when drug therapy is no longer sufficient [53, 54]. Similar to the majority of neurostimulation systems, clinicians are able to use one or a configuration of multiple electrodes to apply electrical stimulation to a small target area. The stimulation uses a periodic waveform with short pulses, which is characterized in the frequency domain by narrow spikes at the primary frequency (130 Hz) and its harmonics. DBS leads are inserted into the target area with the guidance of a stereotactic frame while patients are awake, allowing neurophysiological recording of brain areas and intraoperative monitoring of electrical stimulation side effects and clinical efficacy. New evidence demonstrates that neurostimulation increases firing rates and induces synchronization of neurons in downstream nuclei [123, 124]. This influence on communication between distant neuronal ensembles [125, 126] may disrupt cortico-cortical and cortico-subcortical oscillatory synchronization between connected brain regions, which form the basis of cognitive association, decision making, and learning. Changes of the effect caused by neurostimulation can determine whether variations to the stimulation waveform need to be made so that it continues to alleviate disease symptoms.

In order to be able to detect changes for monitoring the effect of DBS on Parkinson's disease patients, it is important to adequately process electroencephalograms (EEG) recordings obtained during DBS. However, DBS also causes artifacts in EEG that preclude meaningful neurophysiological activity from being quantified during stimulation. As the DBS artifact is generated entirely by the propagation of the DBS waveform through the head, and does not originate from brain activity, the artifact is independent of the patient paradigm and should be suppressed in order to improve processing performance.

The stimulation frequency and its harmonics can be removed with a lowpass filter as the EEG components at high frequencies are not necessary to the application [79]. Another frequency that must be suppressed using filtering is the the 60 Hz power line frequency. The DBS artifact suppression technique needs to take into consideration the narrow frequency domain spike nature of the artifacts. It must not, however, affect important EEG information at low frequencies. Specifically, it has been shown that the beta-band (13-30 Hz) and gamma-band (31-100 Hz) frequencies have critical information for Parkinson's disease patients [127]. In particular, subthalamic nucleus DBS has been shown to change the cortical profile of response inhibition in the beta-band for Parkinson's disease patients and also to reduce the magnitude of coupling between beta-phase and gamma-amplitude [53, 57, 128–130].

We demonstrate DBS signals, in time and frequency, using EEG recorded activity from patients who had been bilaterally implanted with a neurostimulation system; the data was provided by CNI [122]. Figure 5.1 shows the time-domain plot of a 0.2 second EEG recording segment with both DBS units active, without any processing. It can be observed that the EEG is masked by the periodic stimulation waveform. Figure 5.2 shows the Fourier transform (FT) of a longer segment of DBS data; the spikes can be clearly seen in the frequency domain. Figure 5.3 provides a zoomed

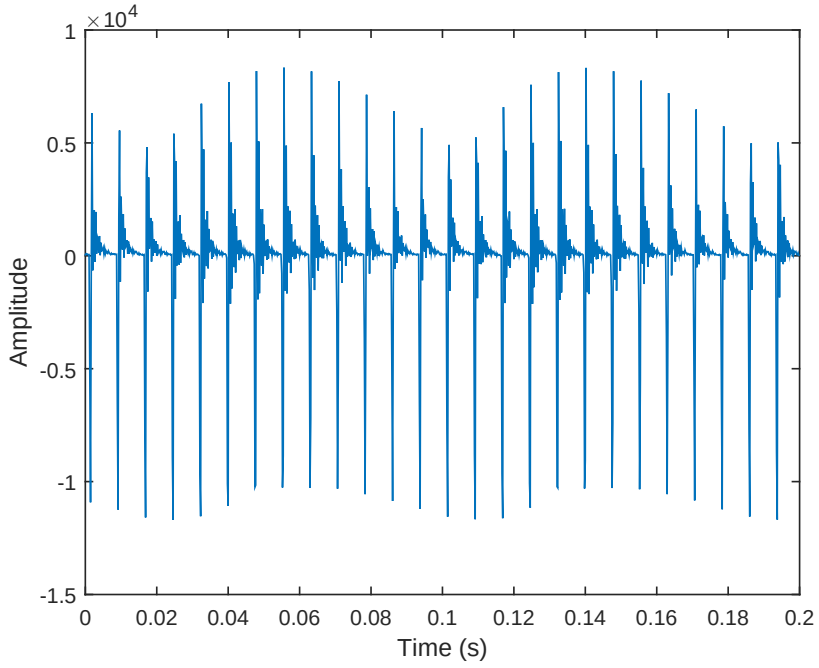


Figure 5.1: EEG Recording in the Time Domain, With Both DBS Sides Active.

version of Figure 5.2, showing only FT components up to 300 Hz. The spike around 12 Hz, and spikes offset from the DBS harmonics about 12 Hz, can be seen. The recording is then filtered using a lowpass filter with 90 Hz cut-off frequency; the FT of the processed signal is shown in Figure 5.4 and its corresponding time domain signal is shown in Figure 5.5. After filtering, the time domain signal looks more like EEG, although some large sinusoidal components, due to DBS artifacts, are clearly present. Figures 5.6 and 5.7 show the EEG signal, in frequency and time respectively, after it was processed using a Hampel filter (see next section) to suppress the DBS artifact. The time domain plot can be compared to Figure 5.8 that shows the EEG of the same data segment but recorded when the DBS was turned off.

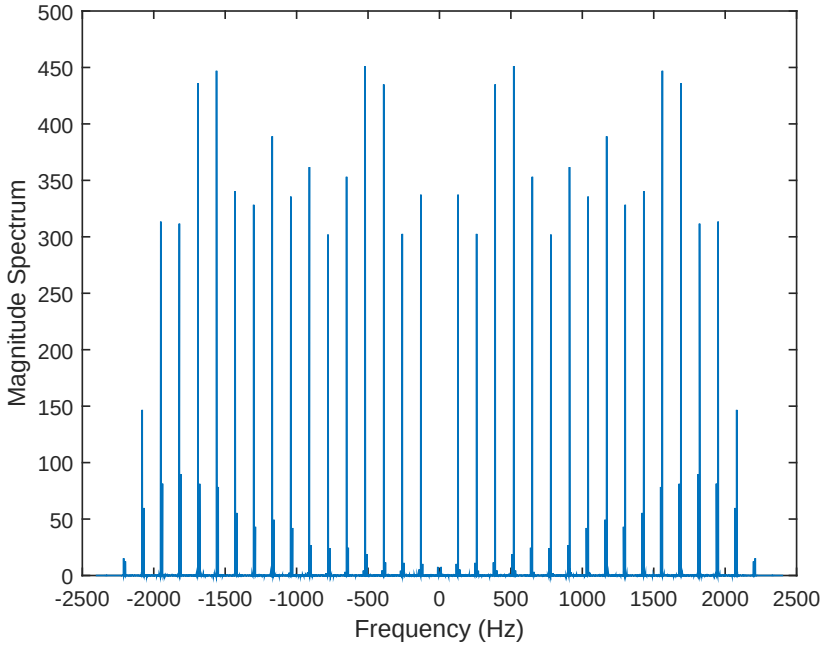


Figure 5.2: Fourier Transform (FT) of the Recording in Figure 5.1.

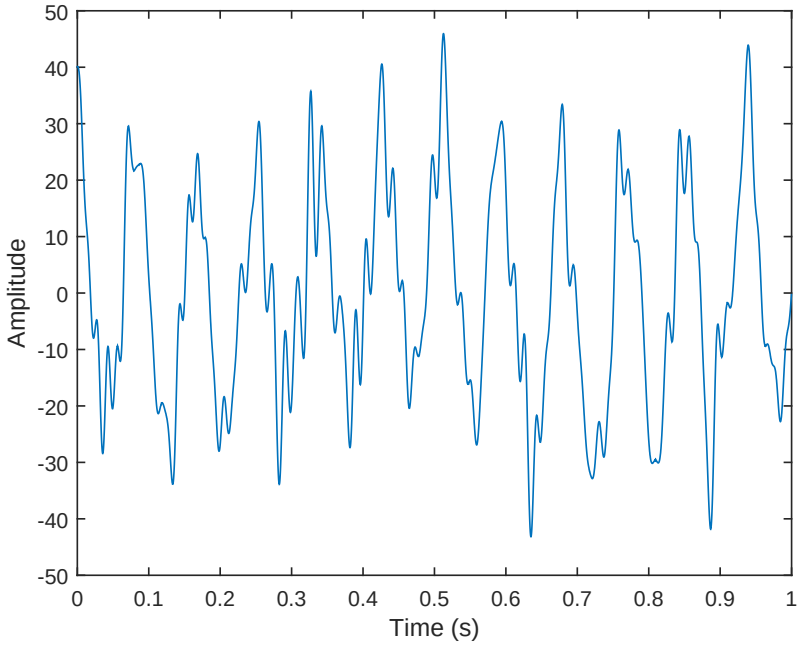


Figure 5.5: Recording in Figure 5.4 in the Time Domain.

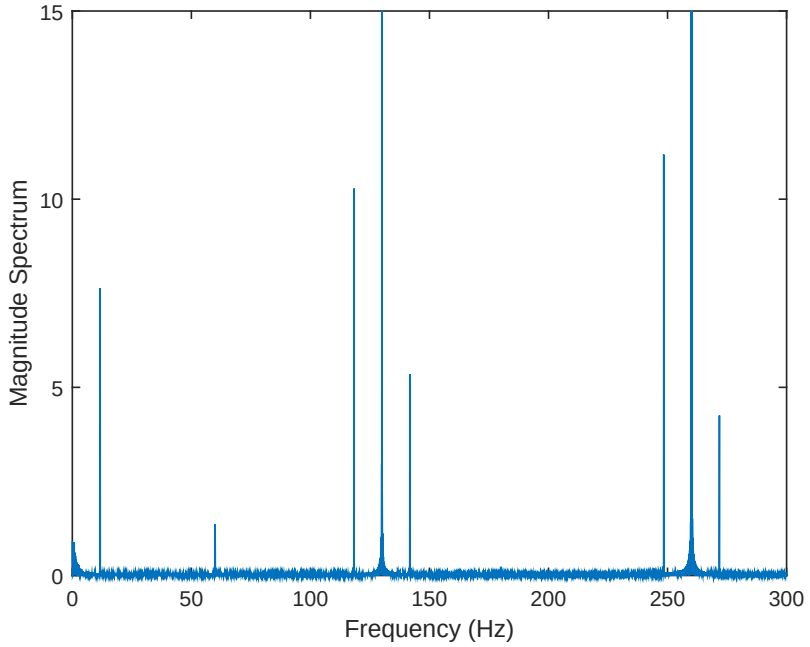


Figure 5.3: Zoomed Version of Figure 5.2, for Frequencies Up to 300 Hz.

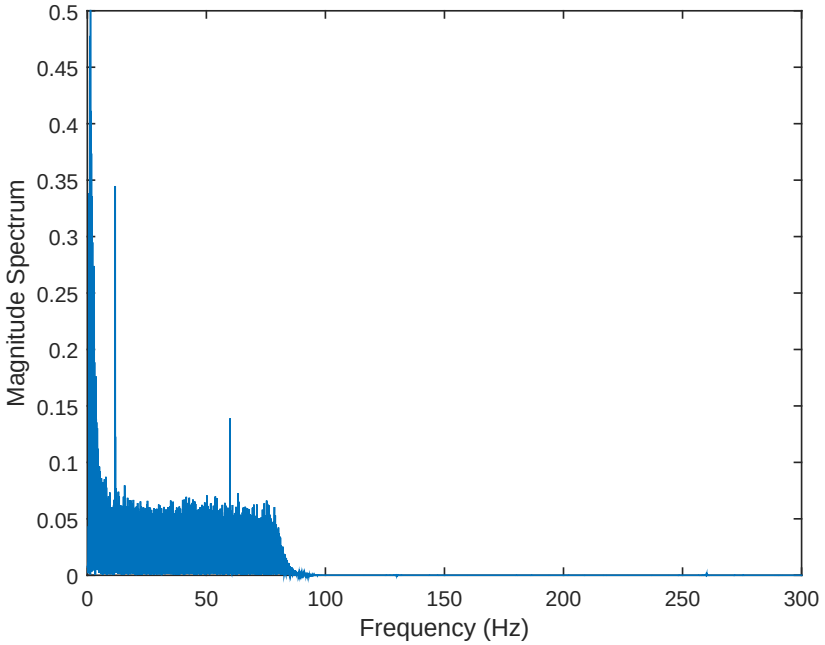


Figure 5.6: FT of the Signal in Figure 5.5 after Applying a Hampel Filter.

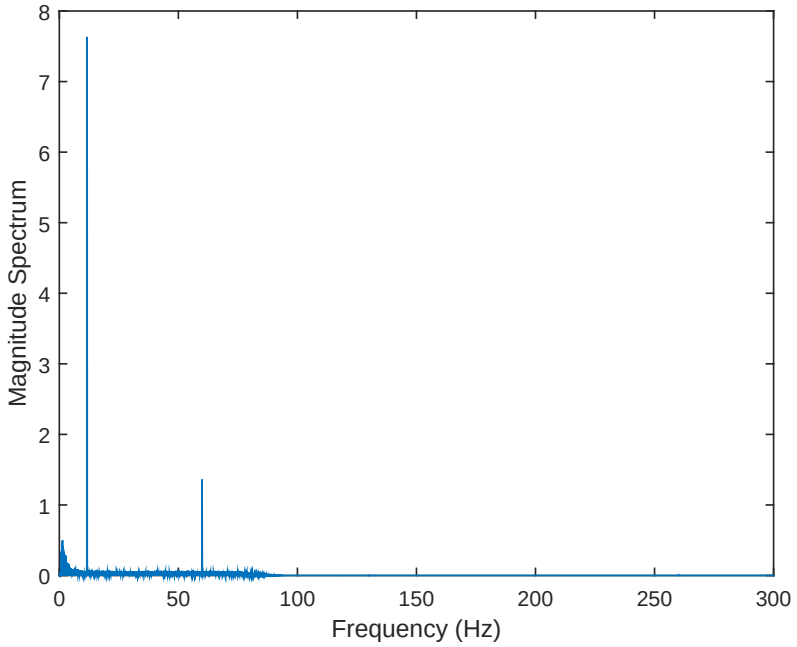


Figure 5.4: FT of Recording After Lowpass Filtering with 100 Hz Cut-off Frequency.

5.2 DBS Artifact Suppression

5.2.1 Hampel Filter

One approach for suppressing DBS artifacts is the use of the Hampel filter to process the EEG recordings [79]. The Hampel filter is a variant on the median filter that first examines a signal for outliers using a median absolute deviation metric. It is used to process data sets that have outliers, which is useful for DBS artifact suppression, as the frequency domain of the DBS artifacts consists of narrow spikes [79]. When outliers are found, those points are replaced by the median value of nearby points. If the Hampel filter is applied in the frequency domain, the narrow spikes can be removed, leaving behind neural activity unaffected in other frequency bins.

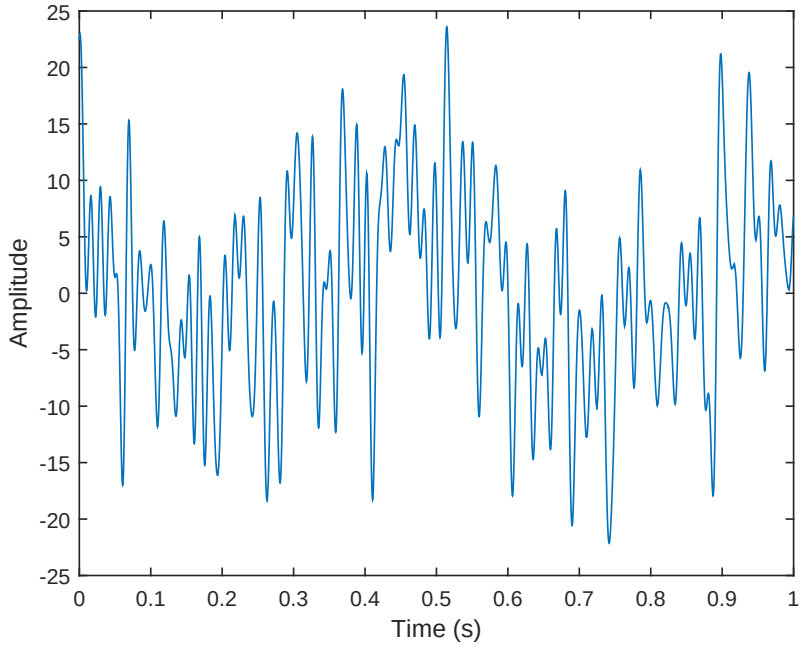


Figure 5.7: Recording in Figure 5.6 in the Time Domain.

In particular, the outlier detection is performed using the median and median absolute deviation. If the data sequence $\{X_0, X_1, ..., X_{N-1}\}$ is the input to the Hampel filter, then the test for identifying a point X_k as an outlier is $|X_k - X^*| > \beta\, B$, where $X^* = \text{median}\{X_{k-L}, X_{k-L+1}, \ldots, X_k, \ldots, X_{k+L-1}, X_{k+L}\}$, β and L are user-specified parameters, and $B = 1.4286\ \text{median}\{|X_{k-L} - X^*|, \ldots, |X_k - X^*|, \ldots, |X_{k+L} - X^*|\}$. The smaller the value of β, the more points are considered as outliers and L varies the number of points around a candidate outlier when computing the median. Note that the median filter uses $\beta = 0$. If a point is determined to be an outlier, that point is replaced with the median value X^*. This method is applied to the real and imaginary parts of the EEG signal in the frequency domain in order to remove the DBS spectral spikes.

Algorithm 9 Hampel Filter

Initialization

$x[n]$, $n = 0, 1, \ldots, N-1$, is the EEG data

X_k, $k = 0, 1, \ldots, N-1$, is the Fourier transform (FT) of the EEG data

L is user-defined

β is user-defined

for $i = 0 : N-1$ **do**

$X^* = \text{median}\{X_{k-L}, X_{k-L+1}, \ldots, X_k, \ldots, X_{k+L-1}, X_{k+L}\}$

(The FT sample is not used to compute the median if its index is outside the $\{0, 1, \ldots, N-1\}$ range)

$B = 1.4826 \text{ median}\{|X_{k-L} - X^*|, |X_{k-L+1} - X^*|, \ldots, |X_k - X^*|, \ldots, |X_{k+L-1} - X^*|, |X_{k+L} - X^*|\}$

if $|X_k - X^*| > \beta B$ **then**

$Y_k = X^*$

else

$Y_k = X_k$

end if

end for

Output filtered signal $\{Y_0, Y_1, \ldots, Y_{N-1}\}$

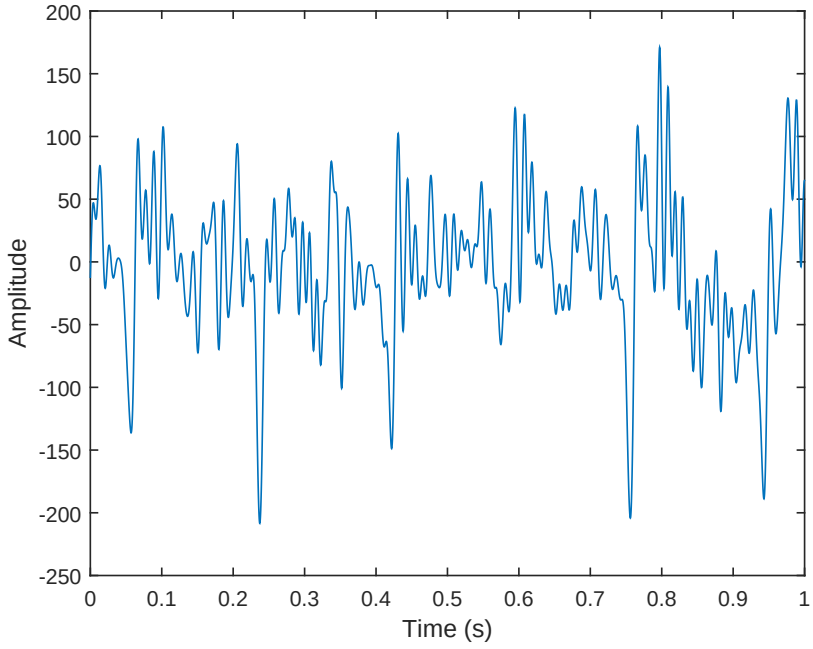


Figure 5.8: Short Segment of EEG With DBS Turned off.

5.2.2 Time-frequency Filtering

Another method for removing the DBS artifact proposed in literature is the matched time-frequency filter [80]. This method assumes that the DBS artifact is composed of a series of pure sinusoids. It thus searches, and then removes, for a single sinusoid whose frequency, amplitude and phase matches the frequency spike present in the DBS data [80]. The algorithm iteratively finds the frequency that best correlates a spike in the EEG signal spectrum. Specifically, first a sinusoid with frequency f_i is computed as $y[n] = \sin(2\pi f_i n/f_s)$, $n = 0, 1, \ldots, N-1$, where f_s is the sampling frequency and N is the number of samples.

At the jth iteration check set of frequencies F_j. For a particular test frequency $f_i \in F_j$, we optimize over the available time shifts to apply to the sinusoid in the matched filter using cross correlation. We check integer time shifts from $\left\lceil -\frac{1}{2}\frac{f_s}{f_i} \right\rceil$ to $\left\lceil \frac{1}{2}\frac{f_s}{f_i} \right\rceil$. Let

the EEG signal be $\mathbf{x} = [x_0, x_1, ..., x_{N-1}]$ and the zero padded EEG signal be $x[n]$. Let $y[n] = \sin[\frac{2\pi f_i n}{f_s}]$, and $(x \star y)[k] = \sum_{n=-\infty}^{\infty} x[n]y[n+k]$ since our signals are real. We then want to find a sinusoid of the form $y_i[n] = \mathrm{A}_i * \sin[\frac{2\pi f_i (n+\Delta k_i)}{f_s}]$. We repeatedly find the optimal frequency, time shift, and amplitude among the frequencies we test, then recenter our test around that frequency and check a smaller region in frequency around it, until we narrow in on a specific sinusoid to subtract out from the EEG signal.

5.2.3 Empirical Mode Decomposition

The empirical mode decomposition (EMD), also known as the Hilbert-Huang Transform, can be used to decompose a signal into a fixed number of signal components in time [90–92]. The signal components are called intrinsic mode functions (IMFs) and have unique instantaneous frequencies that are constrained to regions in frequency that change with time. The IMFs form a complete and nearly orthogonal basis for the original signal and they satisfy two unique properties: the envelope of the IMF averages to zero everywhere, and the number of zero crossings and number of extrema differ by at most one. The EMD algorithm extracts IMFs until the remainder signal has constant instantaneous frequency. The steps of the EMD, following [92], are provided in Algorithms 11 and 12. It is based on identifying, and subtracting out, the fastest oscillation (or highest nonlinearity in the time-frequency plane) present in the signal, and then iterating the residual signal.

As the EMD approach is data-driven and the IMFs are unique in the time-frequency plane, we use it to obtain time-frequency features that are different from those obtained using the matching pursuit decomposition (MPD), as presented in Chapter 4. The EMD can also be used as an approach to remove the DBS artifact.

Algorithm 10 Matched Filter Sinusoid Removal

Initialization

$x[n], \{n = 0, 1, ..., N-1\}$ is the EEG data

F_1 is a course initial set of test frequencies

for $j = 1 : J$ **do**

for $f_i \in F_j$ **do**

$y[n] = \sin[\frac{2\pi f_i n}{f_s}]$

$\Delta k_i = \text{argmax}(x \star y)[k] \left\{\text{s.t. } \left\lceil -\frac{1}{2}\frac{f_s}{f_i} \right\rceil \leq k \leq \left\lceil \frac{1}{2}\frac{f_s}{f_i} \right\rceil \right\}$

$\text{corr}_{max} = (x \star y)[\Delta k_i]$

$A_i = \frac{2\text{corr}_{max}}{N}$

end for

$i_{opt} = \text{argmax}\, A_i$

$A_{opt} = A_{i_{opt}}$

$\Delta k_{opt} = \Delta k_{i_{opt}}$

$f_{opt} = f_{i_{opt}}$

F_{j+1} is a finer set of frequencies centered around current estimate f_{opt}

end for

$y[n] = x[n] - A_{opt} \sin[\frac{2\pi f_{opt}(n+\Delta k_{opt})}{f_s}]$

Output filtered signal $y[n]$

Algorithm 11 Empirical Mode Decomposition

Initialization

$x(t)$ is the input signal

$k = 0$

$x_{-1}(t) = x(t)$

while $x_{k-1}(t) \neq 0$ and $x_{k-1}(t)$ does not have constant instantaneous frequency **do**

$\phi_k(t) = \text{SIFT}(x_{k-1}(t))$

$x_k(t) = x_{k-1}(t) - \phi_k(t)$

$k = k + 1$

end while

$\phi_k(t) = x_{k-1}(t)$

Output set of intrinsic mode functions $\{\phi_k(t)\}$

Algorithm 12 SIFT

Initialization

x_{k-1} is the input data to sift

$e(t) \neq 0$

while $e(t) \neq 0$ **do**

Find $\mathcal{U} = \{(t_p, u_p)\}$ set of all local maxima of x_{k-1}

Find $\mathcal{L} = \{(t_p, l_p)\}$ set of all local minima of x_{k-1}

$u(t) =$ cubic spline of $\mathcal{U}$

$l(t) =$ cubic spline of $\mathcal{L}$

$e(t) = \frac{u(t)+l(t)}{2}$

$r(t) = r(t) - e(t)$

end while

$\phi(t) = r(t)$

Output intrinsic mode function $\phi(t)$

However, we observed that the IMFs capture much of the brain activity at frequencies near to the frequencies where the DBS artifact can be found.

5.3 Adaptive Clustering with DBS Artifact Suppression

5.3.1 Description of Data from DBS System

With the ultimate goal being to monitor the DBS automatically to ensure the treatment is still working well, we look to classify what task a patient is performing to demonstrate that differences in EEG can be detected and classified algorithmically. We use 64 channel EEG data from a Parkinson's disease patient with DBS stimulators on both sides. Data is available from the patient performing six different tasks, with each side of the DBS system turned on or off, so that there is data for no DBS, left or right DBS only or both DBS sides on. The first two tasks are Button Force Left and Button Force Right, which require pressing a button with a specific force with either the left or right hand. The third and fourth task are Verbal Fluency tasks, either Spoken or Written. This involves giving the patient a letter or category, and the patient gives words verbally or written in the category or that start with the letter. The fifth task is Reading, reciting the months of the year, and the sixth task is Touch Pursuit, which is tracing on a touch screen. The tasks are listed in Table 5.1.

5.3.2 DBS Feature Extraction

In Chapter 4, we discussed the use of the MPD with a Gaussian dictionary to extract features from neurostimulation recordings for use in clustering behavioral tasks. The features obtained using the MPD are the amplitude, time shift, frequency shift and scale change of all the Gaussian signals used to expand the data. Another possible feature set can be obtained using a dictionary of IMFs obtained by expanding

Class #	Name of Task	Abbreviation
1	Button Force Left	BFL
2	Button Force Right	BFR
3	Verbal Fluency Spoken	VFS
4	Verbal Fluency Written	VFW
5	Reading	R
6	Touch Pursuit	TP

Table 5.1: Task Information

the data using the EMD. For the EMD based dictionary, we considered 500 time-domain EEG segments, both for the DBS system turned off and turned on. As IMFs with the highest nonlinear instantaneous frequency are extracted first, we use the first IMFs as the feature vectors.

5.3.3 DBS Artifact Suppression

We considered two methods for suppressing the DBS artifacts. The first method is the Hampel filter, which worked relatively well in removing frequency spikes present in the data. The second method is based on decomposing the data using the EMD into specific frequency bands and then removing the IMFs with peak frequencies present where the DBS artifact is expected. Note that the second method did not provide very good results as important frequency information near the artifact frequencies were also suppressed. An example of using the EMD for DBS artifact suppression is shown in time and frequency, respectively, in Figures 5.9 and 5.10.

When using the Hampel filter to suppress the DBS artifact, we applied it both on the real and imaginary parts of the frequency domain representation; this provided a

finely sampled set for clustering. As this approach was computationally time intensive, we first lowpass filtered the data using a 100 Hz cutoff frequency. The Hampel filter used a 301 point median operation, with β value 4. The lowpass filter applied was a Chebyshev II of order 14 with 80 dB stopband attenuation and a cutoff of 100 Hz. This was followed by a highpass Butterworth filter of order 4 at 1 Hz to remove the DC component, which was performed with forward-backward filtering.

For the EMD artifact suppression, the data was similarly lowpass and highpass filtered before applying the EMD. IMFs with high peaks the frequency spikes at 12 Hz and 60 Hz were first removed, before the remaining IMFs are combined to obtained the artifact-suppressed signal. Note that for robustness, we applied a slightly modified version of the EMD as in [91], with some more advanced maxima and minima detection based on interpolation as well as a sifting stop criteria based on the ratio of energy in the original signal versus the average of the IMF envelopes.

5.3.4 Adaptive Task Clustering

The aforementioned data processing resulted in six different data sets that were then used for task clustering. The data set (DS) description, based on the processing, is given as: MPD Gaussian features and Hampel filtering for artifact suppression (DS 1); MPD Gaussian features and EMD artifact suppression (DS 2); MPD Gaussian features and no artifact suppression (DS 3); IMF features and Hampel filtering for artifact suppression (DS 4); IMF features and EMD artifact suppression (DS 5); and IMF features and no artifact suppression (DS 6). The adaptive clustering was performed using a Gaussian mixture model (GMM), described in Chapter 4, with a fixed number of 2 classes. Each class consisted of 15 EEG segments, each of 1 second duration, initiated at the beginning of the patient response. We considered all pairwise task combinations from Table 5.1, Note that as the initialization of the GMM

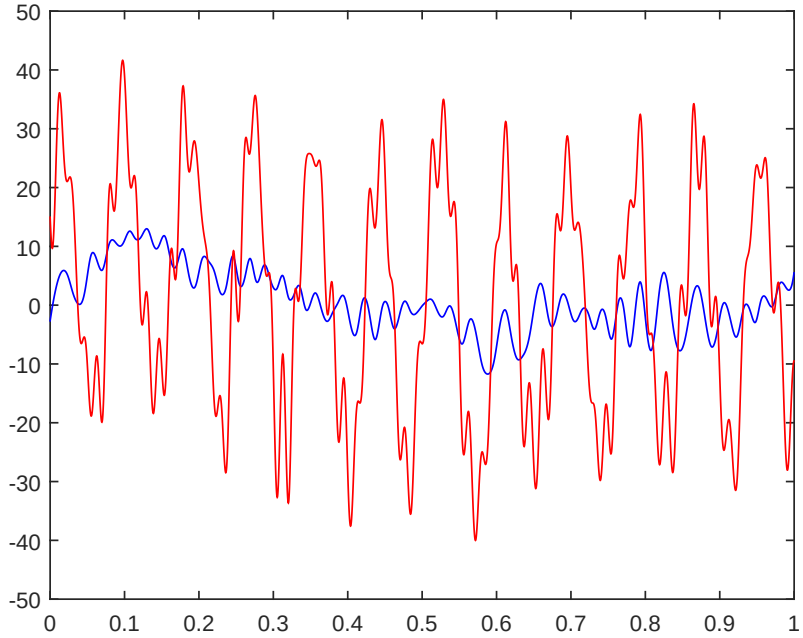


Figure 5.9: EEG Recording with DBS On (Red) and after EMD DBS Suppression (Blue), Vertical Axis: Amplitude, Horizontal Axis: Time (s).

presented a challenge, we compared the log-likelihood obtained from many different initializations, with the initial centers placed on all combinations of two points from the set, and then kept the GMM result with the best log-likelihood.

The clustering results for the six data set combinations are provided in Tables 5.2 and 5.3. Note that we observed a significant difference in the final results based on the selection of the features used as input to the GMM. The results reported in the two tables correspond to the selection of the best feature set. For example, when using the MPD, we did not always select all four Gaussian feature parameters for all MPD iterations.

The classification results are further demonstrated using two cases. The first case uses the Button Force Right (BFR) and Verbal Fluency Written (VFW) tasks

Classes		Suppression Method		
Class A	Class B	Gaussian, Hampel	Gaussian, EMD	Gaussian, None
1	2	63.33%	63.33%	60.00%
1	3	60.00%	73.33%	70.00%
1	4	66.67%	63.33%	63.33%
1	5	93.33%	83.33%	90.00%
1	6	56.67%	63.33%	60.00%
2	3	86.67%	63.33%	66.67%
2	4	80.00%	63.33%	73.33%
2	5	100.00%	93.33%	86.67%
2	6	60.00%	56.67%	63.33%
3	4	60.00%	60.00%	83.33%
3	5	70.00%	73.33%	80.00%
3	6	90.00%	73.33%	63.33%
4	5	66.67%	76.67%	90.00%
4	6	86.67%	60.00%	60.00%
5	6	96.67%	76.67%	86.67%
Ave.		75.78%	69.56%	73.11%

Table 5.2: Classification Results Using a Gaussian MPD Dictionary and Various DBS Suppression Methods, Best Case Features.

Classes		Suppression Method		
Class A	Class B	IMF, Hampel	IMF, EMD	IMF, None
1	2	63.33%	60.00%	66.67%
1	3	73.33%	70.00%	86.67%
1	4	76.67%	66.67%	83.33%
1	5	93.33%	90.00%	93.33%
1	6	56.67%	60.00%	70.00%
2	3	80.00%	80.00%	80.00%
2	4	70.00%	66.67%	63.33%
2	5	93.33%	90.00%	90.00%
2	6	63.33%	60.00%	60.00%
3	4	80.00%	66.67%	63.33%
3	5	66.67%	73.33%	73.33%
3	6	80.00%	66.67%	73.33%
4	5	80.00%	90.00%	93.33%
4	6	70.00%	70.00%	66.67%
5	6	80.00%	93.33%	93.33%
Ave.		75.11%	73.56%	77.11%

Table 5.3: Classification Results with IMF Dictionary and Various DBS Suppression Methods, Best Case Features.

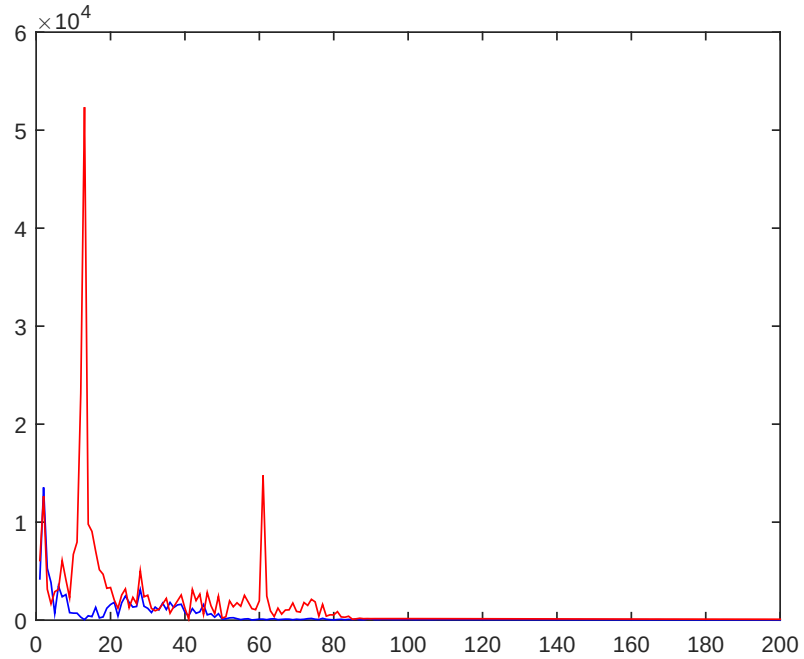


Figure 5.10: The Fourier Transform of the Signals in Figure 5.9, Vertical Axis: Magnitude Spectrum, Horizontal Axis: Frequency (Hz).

with the data processed using MPD Gaussian features and Hampel filtering artifact suppression (DS 1). Figures 5.11 and 5.12 demonstrate the GMM clusterering for the VFW task and BFR task, respectively, obtained using the magnitude and frequency shift MPD Gaussian features. In these figures, the BFR task features for each data point to classify are marked with white Os; white Xs are used for the VFW data points. The decision region used for clustering is shown in Figure 5.13. We can see the five misclassifications for BFR (two on top of each other) and one misclassification for VFW, resulting in an overall correct classification rate of 80%. The second case uses the Reading (R) task and the Touch Pursuit (TP) task with the data processed using EMD IMF features and Hampel filtering artifact suppression (DS 4). The three corresponding plots are provides in Figures 5.14, 5.15, and 5.16. The R task points

are marked with white Xs and the TP task points are marked with white Os. The best feature set selected consisted of the amplitudes of the first two IMF components.

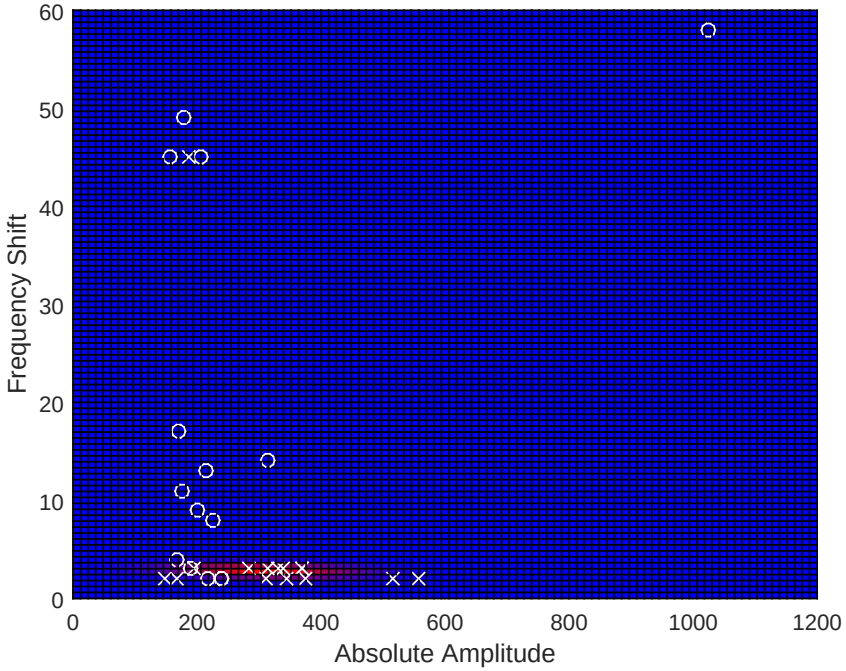


Figure 5.11: Gaussian Cluster for VFW Task.

5.4 Clustering Using Disease-Matched Features

As discussed at the beginning of this chapter, studies have shown that the beta frequency band ranging from 13 to 30 Hz and the gamma frequency band ranging from 31 to 100 Hz from EEG and LFP recordings provide critical information on the pathology of Parkinson's disease as well as on task performance [16, 28, 57–67, 127]. Tasks involving movement and word recognition have been shown to change the activity in the beta and gamma bands, as well as other bands, in characteristic ways that could be useful for classification. Using this prior information, we now

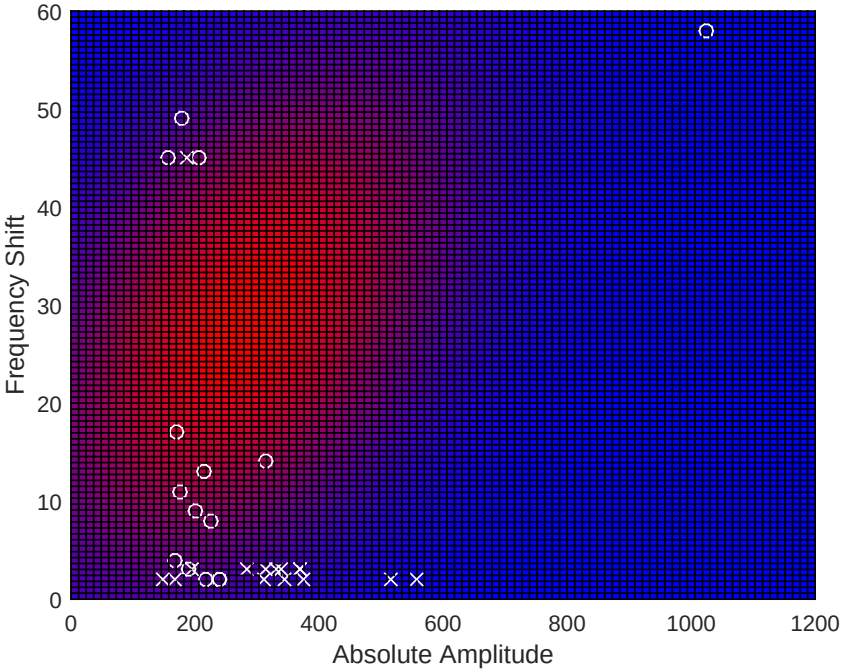


Figure 5.12: Gaussian Cluster for BFR Task.

select EEG recording features that concentrate in these specific frequency bands in order to increase classification performance.

We thus processed EEG recordings to consider five different cases, such that the resulting signals only contain components in specific frequency bands. All the recordings correspond to two behavioral tasks: Verbal Fluency Written (VFW) and Reading (R). In Case 1, the processing involved suppressing the DBS artifact near 12 Hz (when the DBS is turned on) and the power line artifact at 60 Hz using the time-frequency filtering approach. Case 2 consisted of frequencies in the delta band (frequencies below 4 Hz), beta band and gamma band, resulting from a 100 Hz lowpass filter and 4-13 Hz bandstop filter; Case 3 consisted of frequencies in the beta and gamma bands, resulting from a 100 Hz lowpass filter and a 13 Hz highpass filter; Case 4 consisted of frequencies in the beta band only, resulting from a 13-30 Hz bandpass filter; and

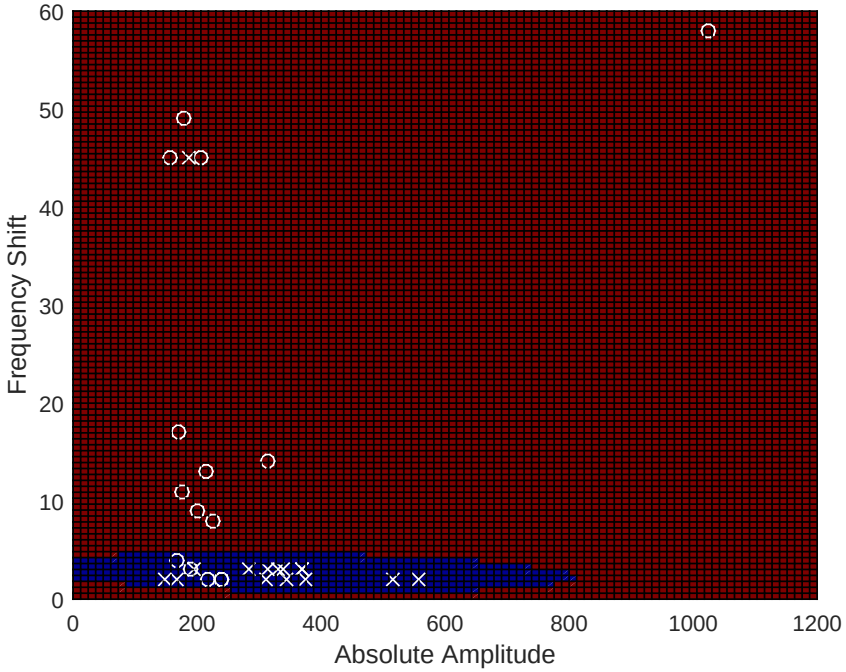


Figure 5.13: GMM Decision Boundary.

Case 5 consisted of frequencies only in the gamma band, resulting from a 30-100 Hz bandpass filter. Note that all filters used a 40th order Butterworth with half-power at the designated frequencies.

For this classification, we selected 15 EEG recordings from each task from the Cp2 electrode. This specific electrode was selected in order to minimize the effects of muscle and speaking artifacts in the EEG recordings. The Cp2 location is slightly back and to the right from the top of the head, which is far from the physical locations of the tongue tip and neck. The data is referenced to the mastoid, setting the EEG reference to the corner of the jaw to further reduce muscle effects. Also, half of the data segment was taken from before the patient starts to move, so only half of each signal is possibly corrupted by muscle and speaking artifacts.

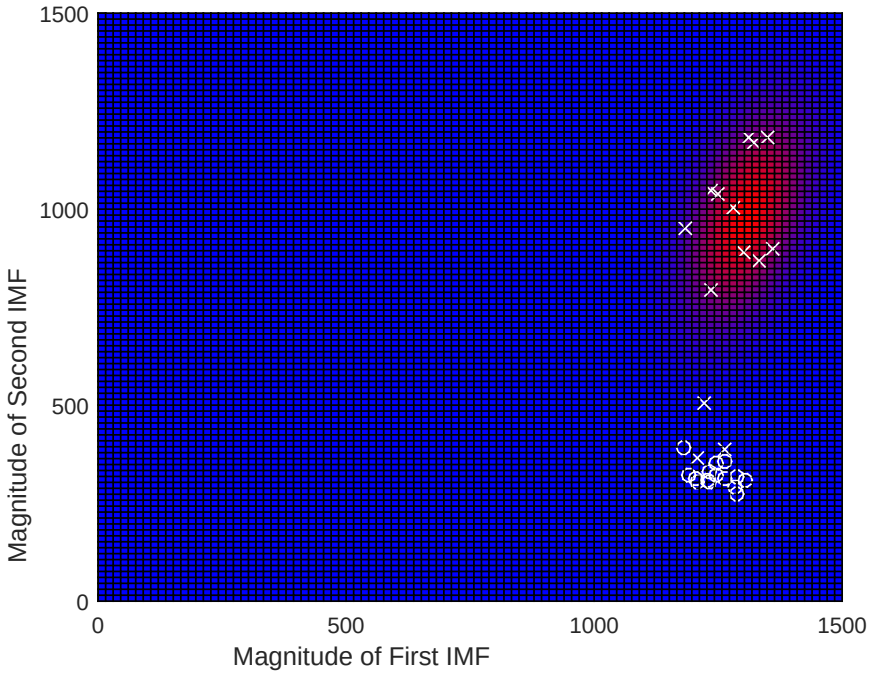


Figure 5.14: Gaussian Cluster for R Task.

For the clustering, we first obtained MPD time-frequency features and used them with a GMM with two classes representing two specific tasks. We considered classification results by varying the number of MPD iterations, for a maximum of 60 iterations. We provide results for the number of iterations that resulted in the highest overall correct classification rate; past that number, noise present in the data was decomposed and thus reduced classification performance. The GMM was initialized by using 100 replicates of the k-means++ initialization, using k-means to obtain a starting guess for the clusters.

When the DBS was turned on, we obtained the highest classification performance for Case 5 (only gamma band). The rate of correct classification remains between 80% and 83% (using 28 to 34 MPD iterations) before dropping off past 73%. When the DBS is turned off, Case 2 resulted in the highest classification performance (76%

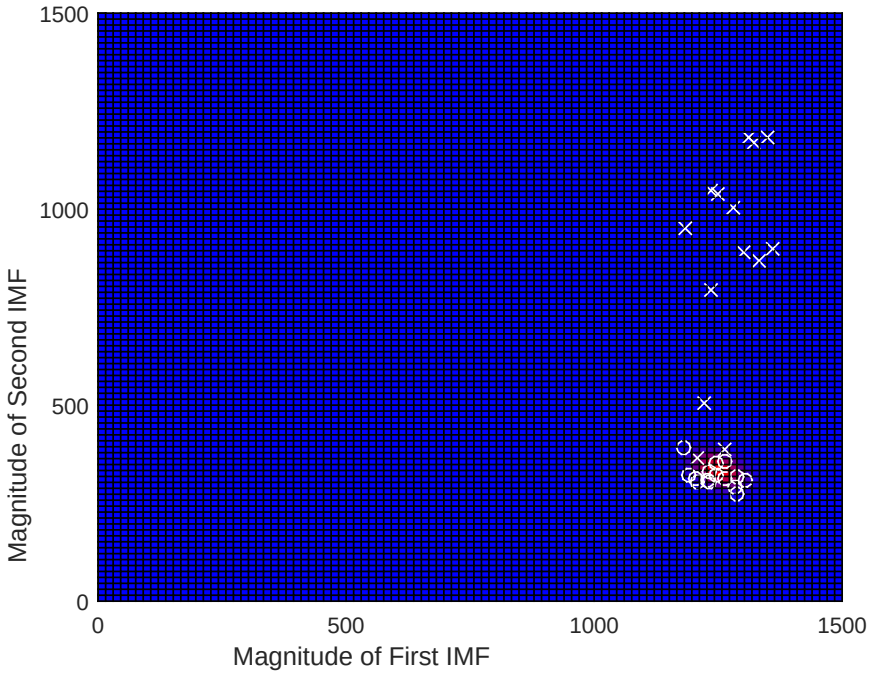


Figure 5.15: Gaussian Cluster for TP Task.

to 80% using 21 to 24 MPD iterations. The confusion matrices for Case 5 and DBS turned on are shown in Tables 5.4 and 5.5 (corresponding to range in performance). The confusion matrices for Case 2 and DBS turned off are shown in Tables 5.6 and 5.7.

In order to ensure that the classification results were obtained using representative information, we consider the signal energy captured by the MPD features. The following figures show the 95% confidence interval of the MPD residual energy, normalized to the energy of the initial signal, for the 15 signals in each class. Figure 5.17 shows the residual energy curve for Case 5 with DBS turned on, and Figure 5.18 shows the residual energy curve for Case 2 with the DBS turned off. As another example, Figure 5.19 shows the residual energy for Case 4 (only beta band) with the DBS turned on. As compared to Case 5 in Figure 5.17, even most of the signal is decomposed

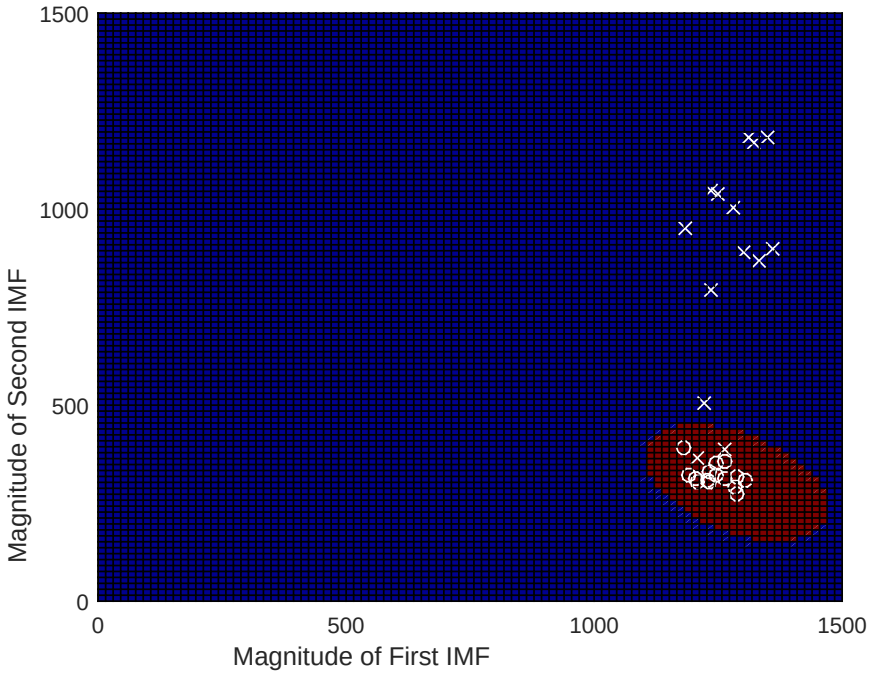
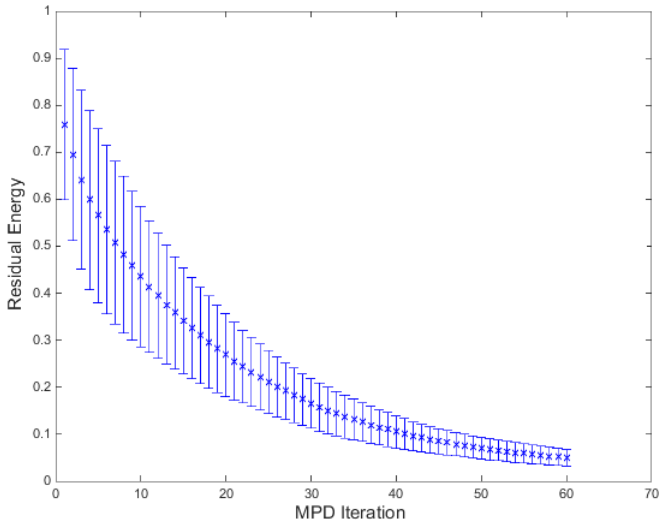


Figure 5.16: GMM Decision Boundary.

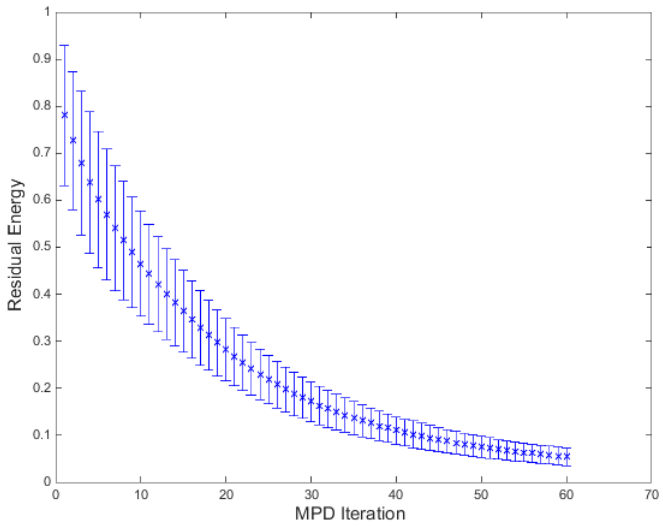
after a few iterations, the resulting features did not provide useful information for classification.

	Class VFW	Class R
True Class VFW	86.7%	13.3%
True Class R	20%	80%

Table 5.4: Confusion Matrix, Case 5 with DBS Turned on.

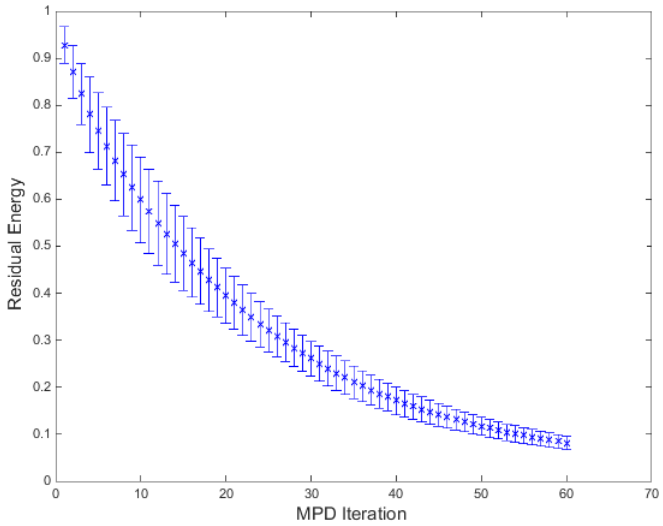


(a) Class VFW

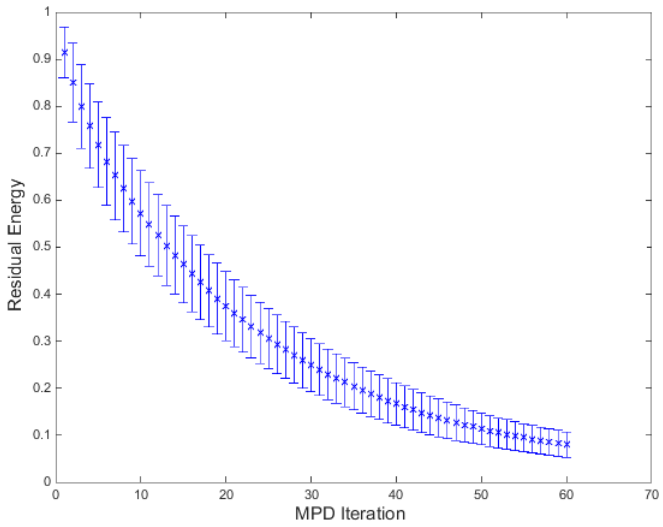


(b) Class R

Figure 5.17: Residual Energy (vertical axis) Versus Number of Iterations (horizontal axis), Case 5 with DBS Turned on: Classes (a) VFW and (b) R.

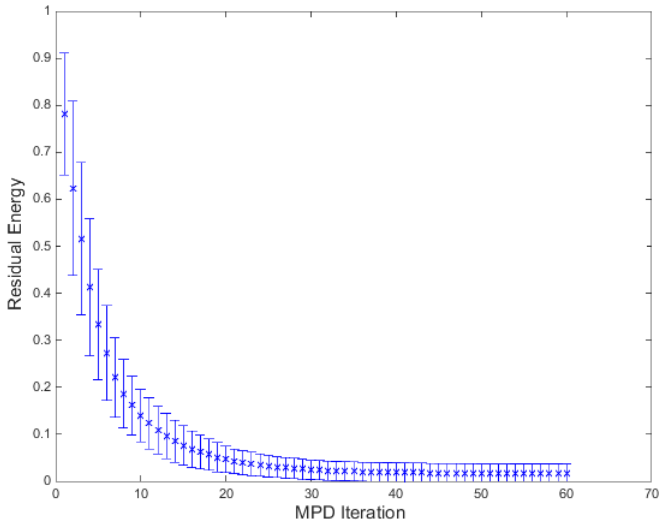


(a) Class VFW

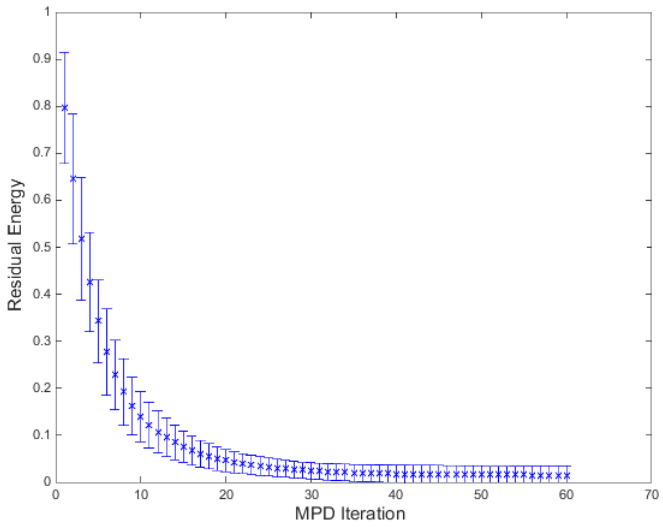


(b) Class R

Figure 5.18: Residual Energy (vertical axis) Versus Number of Iterations (horizontal axis), Case 2 with DBS Turned off: Classes (a) VFW and (b) R.



(a) Class VFW



(b) Class R

Figure 5.19: Residual Energy (vertical axis) Versus Number of Iterations (horizontal axis), Case 4 with DBS Turned on: Classes (a) VFW and (b) R.

	Class VFW	Class R
True Class VFW	80%	20%
True Class R	20%	80%

Table 5.5: Confusion Matrix, Case 5 with DBS Turned on.

	Class VFW	Class R
True Class VFW	93.3%	6.7%
True Class R	33.3%	66.7%

Table 5.6: Confusion Matrix, Case 2 with DBS Turned off.

	Class VFW	Class R
True Class VFW	86.7%	13.3%
True Class R	33.3%	66.7%

Table 5.7: Confusion Matrix, Case 2 with DBS Turned off.

Chapter 6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

Biomedical and biological signals provide very powerful information in medicine. We propose advanced signal processing methods that allow for the detection and treatment of some neurological diseases. These include methods in neuroscience that perform artifact suppression to track the location of brain activity and classify behavioral patient tasks for the purpose of improving the currently used deep brain stimulation treatment for Parkinson's disease. They also include methods in immune response that use time-frequency methods to process microarray data to determine what protein sequence from a possible pathogen generated antibodies.

6.1.1 Neural Stimulation and Activity Tracking

We use object tracking algorithms, originally based on radar applications, which account for spurious measurements to track the sources of neural activity in the brain. Independent component analysis (ICA) is used to separate the measurements from neural current sources and artifacts such as eye blinks and movements, the power line, muscle movements, heartbeats, and electrode popping. Our proposed method, the probabilistic data association filter, tracks neural sources, suppressing measurements that appear to originate from artifacts. This is done by looking at the power spectrum of the ICA separated measurements to discriminate neural activity from artifact. The source location error metric falls below 8 mm in an 85 mm radius

head region. Further methods of neural source tracking and artifact suppression have been explored by other members of the research group in [115, 131].

We also use the matching pursuit decomposition (MPD) to extract features for classification of electroencephalogram (EEG) recordings in Parkinson's disease patients for the purpose of assessing their deep brain stimulation (DBS) treatment. We look to detect differences in the brain rhythm as treatment progresses to see if the stimulation parameters need adjusting. To that end, we look to classify EEG recordings when the patient is performing different tasks to see if any changes in the patient's symptoms can be detected. The classifier is based on the Gaussian mixture model (GMM), an unsupervised clustering approach that does not need training. The features extracted by MPD describe the time-frequency characteristics of the EEG signal. We are sometimes able to achieve classification using particular features from just the first few atoms, but this likely will not provide a stable classification as it uses very little information about the signal. Thus, we perform classification using many MPD atoms that contain most of the energy in the signal. We are able to get 83% classification between a mental recitation task and writing task using about 30 MPD atoms comprising 80% of the energy in the signal within the relevant frequency bands examined. We chose the frequency bands to match up to the bands where brain activity has been found to occur in cognitive and behavioral tasks in the literature. When the DBS system is on, there is an artifact generated that can be suppressed with some lowpass filtering to remove the high frequency components of the artifact, leaving behind high spectral peaks that can be attenuated using with Hampel filtering in the frequency domain or a time-frequency matched filter in the time domain.

6.1.2 Pathogen Detection

Some additional work, separate from the neuroscience work in this thesis, is presented in Appendix A on pathogen detection based on peptide microarrays [121]. By using time-frequency matching pursuit methods, we were able to identify significant subsequences of monoclonal antibodies within peptide microarray data by determining which subsequences raised the fluorescence response higher for one monoclonal antibody than the others. Some antibodies only had partial matches, though this is consistent with antibodies having a specific active region that is not the full pathogen protein. The ability to identify protein sequences on a microarray that draw a strong antibody response can be used for pathogen identification by determining what protein sequences the antibodies respond to, and matching these up to pathogen proteins.

6.2 Future Work

For EEG source tracking, a metric for discriminating between neural activity and artifacts, more aligned with the physiological differences between neural sources and artifacts than a general frequency content energy trend, is important. It is difficult to construct synthetic data that both follows the model and is independent so that ICA can be applied, leading to ICA making errors in separation that challenge the tracking algorithm. Work towards this end has been undertaken by other members of the research group in [115, 131]. We will work toward suppressing more types of artifacts, as well as looking at additional experimental EEG data sets. We are also considering more involved models for the head, including patient specific magnetic resonance imaging (MRI) scan models.

The next step for the deep brain stimulation research is working with more data. The Colorado Neurological Institute (CNI) is reducing the complexity of the tasks

being performed, as it is difficult to extract information for the many tasks in our current data set, especially with the complexity of the tasks. Use of the Dirichlet process GMM (DP-GMM) instead of the GMM, so that the number of tasks does not need to be specified beforehand, is also important. This is because the ability of the DP-GMM to determine the number of classes allows for the addition of new data to past data. If the patient's brain activity has changed significantly since previous measurements, the DP-GMM will place the new activity in a new class instead of the already established class, allowing for further monitoring of the neurological treatment. Additional features provided to the DP-GMM would also increase its monitoring potential, including location and orientation features of the neural source activity. Incorporating the artifact suppression available while tracking could be used to help reduce possible muscle artifacts during tasks even further than the algorithms considered in this work. It is important to obtain measurements from more patients to generalize these results and examine the patient specificity of the time-frequency features. With more patients, the DBS artifact could be better characterized for suppression. Another idea is to improve the feature selection using beta distribution fitting and measuring the difference between the distributions, similarly to the method discussed in [97].

For pathogen detection, early results for identifying monoclonal antibodies on the peptide arrays are promising. New data with many monoclonals tested at the same time to minimize processing differences would be the next step, followed by testing immune responses of people with known diseases. Multiple patients with the same disease or monoclonal antibody target should be tested for patient specificity, as not all immune systems are the same. For patients with a known disease, estimated epitopes can be compared with protein databases to find proteins that drew the response. Finding multiple epitopes that map to pathogen proteins is strong evidence of a pa-

tient having a specific disease. The MPD algorithm can be improved by incorporating information about common substitutions and properties such as the hydrophobicity of amino acids. Matching could also be performed with three-dimensional protein information by mapping onto multidimensional Gaussians.

REFERENCES

[1] N. Zaker, A. Dutta, A. Maurer, J. Zhang, S. Hanrahan, A. Hebb, N. Kovvali, and A. Papandreou-Suppappola, "Adaptive learning of behavioral tasks for patients with Parkinson's disease using signals from deep brain stimulation," in *Asilomar Conference on Signals, Systems and Computers*, Nov. 2014.

[2] A. Delorme and S. Makeig, "EEGLAB: An open source toolbox for analysis of single-trial EEG dynamics," *Journal of Neuroscience Methods*, vol. 134, pp. 9–21, 2004.

[3] A. J. Maurer, "Electroencephalography and magnetoencephalography artifact suppression for improved neural activity estimation," Honors Thesis, Arizona State University, 2008.

[4] A. Maurer, L. Miao, J. J. Zhang, N. Kovvali, A. Papandreou-Suppappola, and C. Chakrabarti, "EEG/MEG artifact suppression for improved neural activity estimation," in *Asilomar Conference on Signals, Systems and Computers*, Nov. 2012, pp. 1646–1650.

[5] A. M. Foerschner, "The history of mental illness: From skull drills to happy pills," *Student Pulse*, vol. 2, 2010. [Online]. Available: http://www.studentpulse.com/a?id=283

[6] Mayo Clinic Staff. (2014, May) Parkinson's disease. Mayo Clinic. [Online]. Available: http://www.mayoclinic.org/diseases-conditions/parkinsons-disease/basics/definition/con-20028488

[7] Deep brain stimulation and surgical treatments. The Michael J. Fox Foundation for Parkinson's Research. [Online]. Available: https://www.michaeljfox.org/understanding-parkinsons/living-with-pd/topic.php?deep-brain-stimulation

[8] National Academies Press, "Grand challenge: How does the human brain work and produce mental activity?" Washington, D.C., 2008.

[9] E. S. Boyden, F. Zhang, E. Bamberg, G. Nagel, and K. Deisseroth, "Millisecond-timescale, genetically-targeted optical control of neural activity," *Nature Neuroscience*, vol. 8, pp. 1263–1268, 2005.

[10] R. C. Kelly, M. A. Smith, J. A. Samonds, A. Kohn, A. B. Bonds, J. A. Movshon, and T. S. Lee, "Comparison of recordings from microelectrode arrays and single electrodes in the visual cortex," *The Journal of Neuroscience*, vol. 27, pp. 261–264, 2007.

[11] S. Baillet, J. C. Mosher, and R. M. Leahy, "Electromagnetic brain mapping," *IEEE Signal Processing Magazine*, vol. 18, pp. 14–30, Nov. 2001.

[12] M. Hämäläinen, R. Hari, R. Ilmoniemi, J. Knuutila, and O. Lounasmaa, "Magnetoencephalography-theory, instrumentation, and applications to noninvasive studies of the working human brain," *Reviews of Modern Physics*, vol. 65, pp. 413–497, 1993.

[13] J. C. Mosher and R. M. Leahy, "Localization using recursively applied and projected (RAP) MUSIC," *IEEE Transactions on Signal Processing*, vol. 47, pp. 332–340, 1999.

[14] J. S. Ebersole, "Noninvasive localization of epileptogenic foci by EEG source modeling," *Epilepsia*, vol. 41, pp. 24–33, 2000.

[15] J. S. Ebersole and S. Hawes-Ebersole, "Clinical application of dipole models in the localization of epileptiform activity," *Journal of Clinical Neurophysioly*, vol. 24, pp. 120–129, 2007.

[16] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, "Brain-computer interfaces for communication and control," *Clinical Neurophysiology*, vol. 113, pp. 767–791, 2002.

[17] L. O. Bauer and V. M. Hesselbrock, "Lateral asymmetries in the frontal brain: Effects of depression and a family history of alcoholism in female adolescents," *Alcoholism: Clinical and Experimental Research*, vol. 26, Nov. 2002.

[18] T. Isotani, H. Tanaka, D. Lehmann, R. D. Pascual-Marqui, K. Kochi, N. Saito, T. Yagyu, T. Kinoshita, and K. Sasada, "Source localization of EEG activity during hypnotically induced anxiety and relaxation," *International Journal of Psychophysiology*, vol. 41, pp. 143–153, Jun. 2001.

[19] R. C. Kessler *et al.*, "Prevalence, severity, and comorbidity of 12-month DSM-IV disorders in the National Comorbidity Survey Replication," vol. 62, pp. 617–627, Jun. 2005.

[20] World Health Organization, *The global burden of disease; 2004 update*, 1st ed. Geneva, Switzerland: World Health Organization, 2008.

[21] E. Somersalo, "The inverse problem of magnetoencephalography: Source localization and the shape of a ball," *SIAM News*, vol. 40, no. 2, Mar. 2007.

[22] L. Miao, J. J. Zhang, C. Chakrabarti, and A. Papandreou-Suppappola, "Efficient Bayesian tracking of multiple sources of neural activity: Algorithms and real-time FPGA implementation," *IEEE Transactions on Signal Processing*, pp. 633–647, 2012.

[23] L. Miao, "Efficient bayesian tracking of multiple sources of neural activity: Algorithms and real-time FPGA implementation," Ph.D. dissertation, Arizona State University, 2013.

[24] J. C. Mosher, P. S. Lewis, and R. M. Leahy, "Multiple dipole modeling and localization from spatio-temporal MEG data," *IEEE Transactions on Biomedical Engineering*, vol. 39, pp. 541–557, Jun. 1992.

[25] B. V. Veen and K. Buckley, "Beamforming: A versatile approach to spatial filtering," *IEEE ASSP Magazine*, vol. 5, pp. 4–24, 1988.

[26] H. R. Mohseni, E. L. Wilding, and S. Sanei, "Sequential Monte Carlo techniques for EEG dipole placing and tracking," in *Sensor Array and Multichannel Signal Processing Workshop*, Jul. 2008, pp. 95–98.

[27] E. Somersalo, A. Voutilainen, and J. P. Kaipio, "Non-stationary magnetoencephalography by Bayesian filtering of dipole models," *Inverse Problems*, vol. 19, pp. 1047–1063, 2003.

[28] R. A. Andersen, S. Musallam, and B. Pesaran, "Selecting the signals for a brain machine interface," *Current Opinion in Neurobiology*, vol. 14, no. 6, pp. 720–726, 2004.

[29] A. Sorrentino, L. Parkkonen, and M. Piana, "Particle filters: A new method for reconstructing multiple current dipoles from MEG data," in *International Congress Series*, vol. 1300, 2007, pp. 173–176.

[30] C. Campi, A. Pascarella, A. Sorrentino, and M. Piana, "A Rao-Blackwellized particle filter for magnetoencephalography," *Inverse Problems and Imaging*, vol. 24, p. 025023, 2008.

[31] J. M. Antelis and J. Minguez, "EEG source localization based on dynamic Bayesian estimation techniques," *International Journal of Bioelectromagnetism*, vol. 11, pp. 179–184, 2009.

[32] H. R. Mohseni, F. Ghaderi, E. L. Wilding, and S. Sanei, "A beamforming particle filter for EEG dipole source localization," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 337–340.

[33] A. Pascarella, A. Sorrentino, C. Campi, and M. Piana, "Particle filtering, beamforming and multiple signal classification for the analysis of MEG time series: A comparison of algorithms," *Inverse Problems and Imaging*, vol. 4, pp. 169–190, Feb. 2010.

[34] A. Pascarella and A. Sorrentino, "Statistical approaches to the inverse problem," in *Magnetoencephalography*, E. W. Pang, Ed. InTech, Nov. 2011.

[35] R. D. Pascual-Marqui, C. M. Michel, and D. Lehmann, "Low resolution electromagnetic tomography: A new method for localizing electrical activity in the brain," *International Journal of Psychophysiology*, vol. 18, pp. 49–65, 1994.

[36] L. Miao, J. J. Zhang, C. Chakrabarti, and A. Papandreou-Suppappola, "Algorithm and parallel implementation of particle filtering and its use in waveform-agile sensing," *Journal of Signal Processing Systems*, vol. 65, pp. 1–17, Nov. 2011.

[37] L. Miao, J. J. Zhang, C. Chakrabarti, A. Papandreou-Suppappola, and N. Kovvali, "Real-time closed-loop tracking of an unknown number of neural sources using probability hypothesis density particle filtering," in *IEEE Workshop on Signal Processing Systems*, Beirut, Lebanon, Oct. 2011, pp. 367–372.

[38] A. Savelainen, "An introduction to EEG artifacts," Aalto University, Sweden, Tech. Rep. Mat-2.4108, 2010.

[39] Z. Xue, J. Li, S. Li, and B. Wan, "Using ICA to remove eye blink and power line artifacts in EEG," in *International Conference on Innovative Computing, Information and Control*, vol. 3, 2006, pp. 107–110.

[40] T. Liu and D. Yao, "Removal of the ocular artifacts from EEG data using a cascaded spatio-temporal processing," *Computer Methods and Programs in Biomedicine*, vol. 83, pp. 95–103, 2006.

[41] A. M. Kladosa, C. Papadelisb, C. Braunb, and D. P. Bamidis, "REG-ICA: A hybrid methodology combining blind source separation and regression techniques for the rejection of ocular artifacts," *Biomedical Signal Processing and Control*, vol. 6, pp. 291–300, 2011.

[42] D. Mantini, R. Franciotti, G. L. Romani, and V. Pizzella, "Improving MEG source localizations: An automated method for complete artifact removal based on independent component analysis," *NeuroImage*, vol. 40, pp. 160–173, 2008.

[43] M. van de Velde, G. van Erp, and P. J. Cluitmans, "Detection of muscle artefact in the normal human awake EEG," *Electroencephalogy and Clinical Neurophysiology*, vol. 107, pp. 149–158, Aug. 1998.

[44] M. Medvedovsky, S. Taulu, R. Bikmullina, and R. Paetau, "Artifact and head movement compensation in MEG," *Neurology, Neurophysiology, and Neuroscience*, p. 4 pages, 2007.

[45] N. P. Castellanos and V. A. Makarov, "Recovering EEG brain signals: Artifact suppression with wavelet enhanced independent component analysis," *Journal of Neuroscience Methods*, vol. 158, pp. 300–312, 2006.

[46] F. Shayegh and A. Erfanian, "Real-time ocular artifacts suppression from EEG signals using an unsupervised adaptive blind source separation," in *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2006, pp. 5269–5272.

[47] A. Erfanian and B. Mahmoudi, "Real-time ocular artifacts suppression using recurrent neural network for EEG-based brain computer interface," *Medical & Biological Engineering & Computing*, vol. 43, pp. 296–305.

[48] R. Abächerli, C. Pasquier, F. Odille, M. Kraemer, J. J. Schmid, and J. Felblinger, "Suppression of MR gradient artefacts on electrophysiological signals based on an adaptive real-time filter with LMS coefficient updates," *Magnetic Resonance Materials in Physics, Biology and Medicine*, vol. 18, pp. 41–50.

[49] A. Delorme and S. Makeig. (2012, Mar.) EEGLAB tutorial outline chapter 9: Decomposing data using ICA. Schwartz Center for Computational Neuroscience. [Online]. Available: http://sccn.ucsd.edu/wiki/Chapter_09:_Decomposing_Data_Using_ICA

[50] S. Eljamel and K. Slavin, Eds., *Neurostimulation: Principles and Practice*. John Wiley & Sons, 2013.

[51] A. O. Hebb and G. A. Ojemann, "The thalamus and language revisited," *Brain and Language*, vol. 126, no. 1, pp. 99–108, Jul. 2013.

[52] R. Q. Quiroga and S. F. Panzeri, *Principles of Neural Coding*. CRC Press, 2013.

[53] A. O. Hebb, J. J. Zhang, M. H. Mahoor, C. Tsiokos, C. Matlack, H. J. Chiseck, and N. Pouratian, "Creating the feedback loop: Closed-loop neurostimulation," *Neurosurgery Clinics of North America*, vol. 25, pp. 187–204, Jan. 2014.

[54] Q. Li, Z.-M. Qian, G. W. Arbuthnott, Y. Ke, and W.-H. Yung, "Cortical effects of deep brain stimulation: Implications for pathogenesis and treatment of Parkinson disease," *JAMA Neurology*, vol. 71, no. 1, pp. 100–103, 2014.

[55] Surgical treatment options. National Parkinson Foundation. [Online]. Available: http://www.parkinson.org/understanding-parkinsons/treatment/surgery-treatment-options/

[56] Parkinson's disease medications. The Michael J. Fox Foundation for Parkinson's Research. [Online]. Available: https://www.michaeljfox.org/understanding-parkinsons/living-with-pd/topic.php?medication

[57] N. Swann, H. Poizner, M. Houser, S. Gould, I. Greenhouse, W. Cai, J. Strunk, J. George, and A. R. Aron, "Deep brain stimulation of the subthalamic nucleus alters the cortical profile of response inhibition in the beta frequency band: A scalp EEG study in Parkinson's disease," *Journal of Neuroscience*, vol. 31, no. 15, pp. 5721–5729, Apr. 2011.

[58] G. Pfurtscheller and F. H. Lopes da Silva, "Event-related EEG/MEG synchronization and desynchronization: Basic principles," *Clin Neurophysiol*, vol. 110, no. 11, pp. 1842–1857, Nov. 1999.

[59] S. Fitzgibbon, K. Pope, L. Mackenzie, C. Clark, and J. Willoughby, "Cognitive tasks augment gamma EEG power," *Clinical Neurophysiology*, vol. 115, no. 8, pp. 1802 – 1809, 2004.

[60] C. M. Krause, P. Korpilahti, B. Pörn, J. Jäntti, and H. A. Lang, "Automatic auditory word perception as measured by 40 Hz EEG responses," *Electroencephalography and Clinical Neurophysiology*, vol. 107, no. 2, pp. 84 – 87, 1998.

[61] S. Little and P. Brown, "What brain signals are suitable for feedback control of deep brain stimulation in Parkinson's disease?" *Annals of the New York Academy of Sciences*, vol. 1265, pp. 9–24, Aug. 2012.

[62] L. M. F. Doyle, A. A. Kühn, M. Hariz, A. Kupsch, G.-H. Schneider, and P. Brown, "Levodopa-induced modulation of subthalamic beta oscillations during self-paced movements in patients with Parkinson's disease," *European Journal of Neuroscience*, vol. 21, no. 5, pp. 1403–1412, 2005.

[63] C. Hammond, H. Bergman, and P. Brown, "Pathological synchronization in Parkinsons disease: Networks, models and treatments," *Trends Neuroscience*, pp. 357–364, 2007.

[64] A. A. Kühn, L. Doyle, A. Pogosyan, K. Yarrow, A. Kupsch, G.-H. Schneider, M. I. Hariz, T. Trottenberg, and P. Brown, "Modulation of beta oscillations in the subthalamic area during motor imagery in Parkinson's disease," *Brain*, vol. 129, pp. 695–706, 2005.

[65] A. A. Kühn, A. Tsui, T. Aziz, N. Ray, C. Brücke, A. Kupsch, G. H. Schneider, and P. Brown, "Pathological synchronisation in the subthalamic nucleus of patients with parkinsons disease relates to both bradykinesia and rigidity," *Experimental Neurology*, pp. 380–387, 2009.

[66] K. J. Miller, T. J. Abel, A. O. Hebb, and J. G. Ojemann, "Rapid online language mapping with electrocorticography," *Neurosurgury: Pediatrics*, pp. 482–490, 2011.

[67] A. O. Hebb, F. Darvas, and K. J. Miller, "Transient and state modulation of beta power in human subthalamic nucleus during speech production and finger movement," *Neuroscience*, pp. 218–233, 2012.

[68] M. I. Hariz, S. Rehncrona, N. P. Quinn, J. D. Speelman, and C. Wensing, "Multicenter study on deep brain stimulation in Parkinson's disease: An independent assessment of reported adverse events at 4 years," *Movement Disorders*, vol. 23, no. 3, pp. 416–421, Feb. 2008.

[69] M. K. York, E. A. Wilde, R. Simpson, and J. J. Jankovic, "Relationship between neuropsychological outcome and DBS surgical trajectory and electrode location," *Journal of the Neurological Sciences*, vol. 287, no. 1, pp. 159–171, 2009.

[70] D. F. Marshall, A. M. Strutt, A. E. Williams, R. K. Simpson, J. Jankovic, and M. K. York, "Alternating verbal fluency performance following bilateral subthalamic nucleus deep brain stimulation for Parkinson's disease," *European Journal of Neurology*, vol. 19, no. 12, pp. 1525–1531, 2012.

[71] A. Anzak, L. Gaynor, M. Beigi, P. Limousin, M. Hariz, L. Zrinzo, T. Foltynie, P. Brown, and M. Jahanshahi, "A gamma band specific role of the subthalamic nucleus in switching during verbal fluency tasks in Parkinson's disease," *Experimental Neurology*, vol. 232, no. 2, pp. 136–142, Dec. 2011.

[72] A. Mikos, D. Bowers, A. M. Noecker, C. C. McIntyre, M. Won, A. Chaturvedi, K. D. Foote, and M. S. Okun, "Patient-specific analysis of the relationship between the volume of tissue activated during DBS and verbal fluency," *Neuroimage*, vol. 54, pp. 238–246, Jan. 2011.

[73] G. Mirabella, S. Iaconelli, N. Modugno, G. Giannini, F. Lena, and G. Cantore, "Stimulation of subthalamic nuclei restores a near normal planning strategy in Parkinson's patients," *PLoS One*, vol. 8, no. 5, p. e62793, 2013.

[74] W. Wu, S. Nagarajan, and Z. Chen, "Bayesian machine learning: EEG/MEG signal processing measurements," *IEEE Magazine in Signal Processing*, vol. 33, pp. 14–36, 2016.

[75] S. E. Ross, "The effects of deep brain stimulation in the ventral pallidum and the central nucleus of the amygdala on food consumption, motivation, and palatability," Ph.D. dissertation, University of Michigan, 2013.

[76] D. A. Wagenaar and S. M. Potter, "Real-time multi-channel stimulus artifact suppression by local curve fitting," *Neuroscience Methods*, vol. 120, pp. 113–20, Oct. 2002.

[77] A. Santillan-Guzman, U. Heute, M. Muthuraman, U. Stephani, and A. Galka, "DBS artifact suppression using a time-frequency domain filter," in *International Conference of the IEEE Engineering in Medicine and Biology Society*, Jul. 2013, pp. 4815–4818.

[78] C. Waddell, J. Pratt, B. Porr, and S. Ewing, "Deep brain stimulation artifact removal through under-sampling and cubic-spline interpolation," in *International Congress on Image and Signal Processing*, Oct. 2009, pp. 1–5.

[79] D. P. Allen, E. L. Stegemller, C. Zadikoff, J. M. Rosenow, and C. D. Mackinnon, "Suppression of deep brain stimulation artifacts from the electroencephalogram by frequency-domain hampel filtering," *Clinical Neurophysiology*, vol. 121, pp. 1227–1232, Aug. 2010.

[80] Y. Sun, F. Farzan, L. G. Dominguez, M. S. Barr, P. Giacobbe, A. M. Lozano, W. Wong, and Z. J. Daskalakis, "A novel method for removal of deep brain stimulation artifact from electroencephalography," *Journal of Neuroscience Methods*, vol. 237, pp. 33–40, 2014.

[81] Y. Bar-Shalom, F. Daum, and J. Huang, "The probabilistic data association filter," *IEEE Control Systems Magazine*, vol. 29, pp. 82–100, Dec. 2009.

[82] S. P. Sira, "Waveform-agile sensing for target tracking and detection in clutter," Ph.D. dissertation, Arizona State University, 2007.

[83] S. P. Sira, A. Papandreou-Suppappola, and D. Morrell, "Dynamic configuration of time-varying waveforms for agile sensing and tracking in clutter," *IEEE Transactions on Signal Processing*, vol. 55, no. 7, pp. 3207–3217, Jul. 2007.

[84] J. Vermaak, S. Godsill, and P. Perez, "Monte Carlo filtering for multi-target tracking and data association," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, pp. 309–332, Jan. 2005.

[85] R. Mahler, "Multi-target Bayes filtering via first-order multi-target moments," *IEEE Transactions on Aerospace and Electronic System*, vol. 39, pp. 1152–1178, 2003.

[86] O. Erdinc, P. Willett, and Y. Bar-Shalom, "Probability hypothesis density filter for multitarget multisensor tracking," in *International Conference on Information Fusion*, vol. 1, July 2005.

[87] B. N. Vo, S. Singh, and A. Doucet, "Sequential Monte Carlo implementation of the PHD filter for multi-target tracking," in *International Conference of Information Fusion*, vol. 2, 2003, pp. 792–799.

[88] M. Tobias and A. D. Lanterman, "Probability hypothesis density-based multitarget tracking with bistatic range and doppler observations," *IEEE Radar, Sonar and Navigation*, vol. 152, pp. 195–205, 2005.

[89] D. E. Clark and J. Bell, "Bayesian multiple target tracking in forward scan sonar images using the PHD filter," *IEEE Radar, Sonar and Navigation*, vol. 152, pp. 327–334, 2005.

[90] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, and H. H. Liu, "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis," in *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 454, no. 1971, 1998, pp. 903–995.

[91] R. T. Rato, M. D. Ortigueira, and A. G. Batista, "On the HHT, its problems, and some solutions," *Mechanical Systems and Signal Processing*, vol. 22, no. 6, pp. 1374 – 1394, 2008, Special Issue: Mechatronics.

[92] S. Sandoval and P. L. De Leon, "Theory of the Hilbert spectrum," Arizona State University and New Mexico State University, Tech. Rep. [Online]. Available: http://www.public.asu.edu/~spsandov/Sandoval2015Theory.pdf

[93] P. Smyth, "The EM algorithm for Gaussian mixtures," Univ. of California, Irvine, Lecture Notes, CS274. [Online]. Available: http://www.ics.uci.edu/~smyth/courses/cs274/notes/EMnotes.pdf

[94] D. Görür and C. Edward Rasmussen, "Dirichlet process gaussian mixture models: Choice of the base distribution," *Journal of Computer Science and Technology*, vol. 25, no. 4, pp. 653–664, 2010.

[95] A. Malin, J. J. Zhang, B. Chakraborty, N. Kovvali, A. Papandreou-Suppappola, S. Johnston, and P. Stafford, "Adaptive learning of immunosignaturing peptide array features for biothreat detection and classification," in *Asilomar Conference on Signals, Systems and Computers*, 2011, pp. 1883–1887.

[96] A. Malin, N. Kovvali, A. Papandreou-Suppappola, J. J. Zhang, S. Johnston, and P. Stafford, "Beta process based adaptive learning of immunosignaturing peptide-antibody factors," in *Asilomar Conference on Signals, Systems and Computers*, 2012, pp. 1651–1655.

[97] A. Malin, "Adaptive learning and unsupervised clustering of immune responses using microarray random sequence peptides," Ph.D. dissertation, Arizona State University, 2013.

[98] Z. S. Qin, "Clustering microarray gene expression data using weighted Chinese restaurant process," *Bioinformatics*, vol. 22, no. 16, pp. 1988–1997, Aug. 2006.

[99] M. Medvedovic and S. Sivaganesan, "Bayesian infinite mixture model based clustering of gene expression profiles," *Bioinformatics*, vol. 18, no. 9, pp. 1194–1206, Sep 2002.

[100] D. Young, "An overview of mixture models," *Statistics Surveys*, pp. 1–24, 2008.

[101] J. Sethuraman, "A constructive definition of dirichlet priors," *Statistica Sinica*, vol. 4, pp. 639–650, 1994.

[102] H. Ishwaran and L. F. James, "Gibbs sampling methods for stick-breaking priors," *Journal of the American Statistical Association*, vol. 96, no. 453, pp. 161–173, 2001.

[103] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.

[104] S. Kay, *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*, 1st ed. Upper Saddle River, New Jersey: Prentice Hall, 1993.

[105] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, pp. 174–188, Feb. 2002.

[106] G. L. Smith, S. F. Schmidt, and L. A. McGee, "Application of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle," Tech. Rep. [Online]. Available: https://archive.org/details/nasa_techdoc_19620006857

[107] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," in *Signal Processing, Sensor Fusion, and Target Recognition VI*, I. Kadar, Ed., vol. 3068, Jul. 1997.

[108] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, "A consistent metric for performance evaluation of multi-object filters," *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3447–3457, Aug. 2008.

[109] J. Hoffman and R. Mahler, "Multitarget miss distance via optimal assignment," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 34, no. 3, pp. 327–336, 2004.

[110] J. C. Mosher, R. M. Leahy, and P. S. Lewis, "EEG and MEG: Forward solutions for inverse methods," *IEEE Transactions on Biomedical Engineering*, vol. 146, pp. 245–259, Mar. 1999.

[111] J. Sarvas, "Basic mathematical and electromagnetic concepts of the biomagnetic inverse problems," *Physics in Medicine and Biology*, vol. 32, pp. 11–22, 1987.

[112] S. Makeig. (2012, Mar.) Frequently asked questions about ICA applied to EEG and MEG data. [Online]. Available: http://sccn.ucsd.edu/~scott/tutorial/icafaq.html

[113] T. Jung, S. Makeig, C. Humphries, T. Lee, M. J. McKeown, V. Iragui, and T. J. Sejnowski, "Removing electroencephalographic artifacts by blind source separation," *Psychophysiology*, vol. 37, pp. 163–178, Mar. 2000.

[114] A. Delorme and S. Makeig. (2012, Mar.) EEGLAB tutorial outline chapter 1: Loading data in EEGLAB. Schwartz Center for Computational Neuroscience. [Online]. Available: http://sccn.ucsd.edu/wiki/Chapter_01:_Loading_Data_in_EEGLAB

[115] F. J. Solis, A. J. Maurer, J. Jiang, and A. Papandreou-Suppappola, "Adaptive EEG artifact suppression using Gaussian mixture modeling," in *Asilomar Conference on Signals, Systems and Computers*, Nov. 2015, pp. 607–611.

[116] M. Teplan, "Fundamentals of EEG measurement," *Measurement Science Review*, vol. 2, pp. 1–11, 2002.

[117] A. Delorme and S. Makeig. (2012, Mar.) EEGLAB tutorial outline a08: DIPFIT. Schwartz Center for Computational Neuroscience. [Online]. Available: http://sccn.ucsd.edu/wiki/A08:_DIPFIT

[118] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, *Markov chain Monte Carlo in practice*. London: Chapman & Hall, 1996.

[119] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, pp. 3397–3415, 1993.

[120] D. Chakraborty, N. Kovvali, J. Wei, A. Papandreou-Suppappola, D. Cochran, and A. Chattopadhyay, "Damage classification structural health monitoring in bolted structures using time-frequency techniques," *Journal of Intelligent Material Systems and Structures*, vol. 20, pp. 1289–1305, 2009.

[121] B. O'Donnell, A. Maurer, and A. Papandreou-Suppappola, "Biosequence time-frequency processing: Pathogen detection and identification," in *Excursions in Harmonic Analysis, Volume 3*, R. Balan, M. Begué, J. Benedetto, W. Czaja, and K. Okoudjou, Eds. Birkhäuser Basel, 2015, ch. 3, pp. 65–85.

[122] The Colorado Neurological Institute. [Online]. Available: www.thecni.org/

[123] C. Beurrier, B. Bioulac, J. Audin, and C. Hammond, "High-frequency stimulation produces a transient blockade of voltage-gated currents in subthalamic neurons," *Journal of Neurophysiology*, vol. 85, no. 4, pp. 1351–1356, 2001.

[124] T. Hashimoto, C. M. Elder, M. S. Okun, S. K. Patrick, and J. L. Vitek, "Stimulation of the subthalamic nucleus changes the firing pattern of pallidal neurons," *Journal of Neuroscience*, vol. 23, no. 5, pp. 1916–1923, Mar. 2003.

[125] J. Fell and N. Axmacher, "The role of phase synchronization in memory processes," *Nature Reviews Neuroscience*, vol. 12, no. 2, pp. 105–118, 2011.

[126] J. Fell, G. Widman, B. Rehberg, C. E. Elger, and G. Fernandez, "Human mediotemporal EEG characteristics during propofol anesthesia," *Biological Cybernetics*, vol. 92, no. 2, pp. 92–100, 2005.

[127] G. Deuschl and A. Eisen, Eds., *Recommendations for the Practice of Clinical Neurophysiology: Guidelines of the International Federation of Clinical Physiology*. Elsevier Science, 1999.

[128] Y. Zhang, Y. Chen, S. L. Bressler, and M. Ding, "Response preparation and inhibition: The role of the cortical sensorimotor beta rhythm," *Neuroscience*, vol. 156, no. 1, pp. 238–246, 2008.

[129] J. Lopez-Azcarate, M. Tainta, M. C. Rodriguez-Oroz, M. Valencia, R. Gonzalez, J. Guridi, J. Iriarte, J. A. Obeso, J. Artieda, and M. Aleg, "Coupling between beta and high-frequency activity in the human subthalamic nucleus may be a pathophysiological mechanism in Parkinson's disease," *Neurobiology of Disease*, vol. 30, pp. 6667–6677, 2010.

[130] A. I. Yang, N. Vanegas, C. Lungu, and K. A. Zaghloul, "Beta-coupled high-frequency activity and beta-locked neuronal spiking in the subthalamic nucleus of parkinson's disease," *The Journal of Neuroscience : The Official Journal of the Society for Neuroscience*, vol. 34, no. 38, pp. 12 816–12 827, 2014.

[131] J. Jiang, "Multiple neural artifacts suppression using Gaussian mixture modeling and probability hypothesis density filtering," Master's thesis, Arizona State University, 2014.

APPENDIX A

BIOSEQUENCE TIME-FREQUENCY PROCESSING: PATHOGEN DETECTION AND IDENTIFICATION

A.1 Description of Work in Appendix

This appendix contains a published chapter, referenced as [121] above, in the references section for the main chapters. This work is part of collaborative research on biological signal processing, and it is not related to the biomedical signal processing work presented in Chapters 1-5.

A.2 Abstract

Diagnostic information obtained from antibodies binding to random peptide sequences is now feasible using immunosignaturing, a recently developed microarray technology. The success of this technology is highly dependent upon the use of advanced algorithms to analyze the random sequence peptide arrays and to process variations in antibody profiles to discriminate between pathogens. This work presents the use of time-frequency signal processing methods for immunosignaturing. In particular, highly-localized waveforms and their parameters are used to uniquely map random peptide sequences and their properties in the time-frequency plane. Advanced time-frequency signal processing techniques are then applied for estimating antigenic determinants or epitope candidates for detecting and identifying potential pathogens.

A.3 Introduction

A.3.1 Signal Processing of Biological Sequences and Challenges

The area of bioinformatics is mainly involved with the management of biological information using computer technology and statistics. Signal processing for molecular biology, on the other hand, encompasses the development of algorithms and methodologies for extracting, processing and interpreting information from biological

sequences [1-6]. Intelligent use of signal processing algorithms can provide invaluable insight into the structure, functioning and evolution of biological systems. For example, complex assays to determine functional activities of analytes or peptide chips to manifest key residues for protein binding can provide a wealth of information on underlying biological systems. However, in each of these cases, appropriately designed processing is required to robustly extract the most relevant information. Images of array fluorescence are enhanced to improve the estimation of gene reactivity, while gene expression classification performance is increased by including biological and experimental variability in the algorithm design [4].

Genomics and proteomics, in general terms, study the functions and structures of genomes and proteomes, respectively. Genomes, which are genetic material of organisms encoded in deoxyribonucleic acid (DNA) or ribonucleic acid (RNA), and proteomes, which are expressed proteins in given organisms, provide discrete information, represented in sequences of unique elements [7,8]. More specifically, DNA are bio-molecules that are represented as letter sequences of precise orderings of four nucleobases; the different orderings correspond to patterns that influence the formation and development of different organisms. Similarly, proteins are bio-molecules represented as sequences of unique orderings of twenty linked amino acids, with each amino acid represented by a letter of the alphabet. DNA and protein sequence analysis requires significant processing of the discrete gene orderings in order to identify intrinsic common features or find gene variations such as mutations [9,10]. One genome analysis application is gene sequence periodicity as regions of genetic repetition have been shown to correlate with functionally important genes [11,12]. Gene periodicity has been analyzed using spectral methods [13-16]; such methods have also been used to estimate variations in base pair frequencies between organisms as they can indicate phylogenic origin from the species genome. Time-frequency signal process-

ing methods such as wavelet transforms have also been used in gene sequencing such as to characterize long range correlations or identify irregularities in DNA sequences [14,17,18].

Signal processing methods have also been used for sequence alignment, or arranging sequences to identify regions of similarity due to functional, structural, or evolutionary relationships between the sequences [19,20]. As thousands of organisms have been sequenced completely, and many more have been partially sequenced, searching for these similarities requires a vast number of computations. There are many algorithms designed to perform these searches including dynamic programming algorithms such as Smith-Waterman and BLAST, correlation based methods, Bayesian approaches, and time-frequency (TF) based methods [10, 21-28]. Computational alignment tools based on dynamic programming such as the Smith-Waterman algorithm is guaranteed to find all similarity matches, but it runs slowly [21]. Other tools, such as BLAST [22,23], are widely made available for database similarity searching as they were developed to provide a fast approach of approximating the complete alignment found by dynamic programming algorithms. BLAST runs very quickly, around an order of magnitude faster than the complete alignment algorithms, and finds most significant alignments under most circumstances. However, it tends to miss alignments for queries with repetitive segments. Correlation based methods map DNA or amino acid sequences to real or complex numbered sequences and use sequence correlation to achieve a match in similarity [26]. The algorithm can be implemented fast using the fast Fourier transform; however, errors increase when aligning sequences of longer lengths. We have recently developed a TF based method that first uniquely maps sequences to highly-localized Gaussian waveforms in the TF plane, and then uses the matching pursuit decomposition (MPD) algorithm to perform alignment [28-

30]. The alignment approach is compared to other approaches and shown to perform well with repetitive segments in real time without pre-processing.

In addition to gene sequencing, microarray analysis has also played a significant role in the extraction and interpretation of genomic information. Microarrays can provide measurements of expression levels of large numbers of genes. For example, peptide microarrays have been used to study binding properties and functionality of different types of protein-protein interactions and provide insight into specific pathogens [31-35]. Peptide microarrays are a relative new application for biological signal processing. The technology to create assays using single peptide chains has been around for a while in the form of the enzyme linked immunosorbent assay (ELISA) [36]. In recent years, as the cost of printing many peptide clusters onto a single substrate has been dropping, tens or hundreds of thousands of peptide clusters can be reasonably printed on a single array. In addition to be able to construct large scale peptide arrays to detect specific diseases, another important aspect is the robust interpretation and analysis of the extracted data in order to establish relationships between peptide sequences and binding strengths. Some recent analysis approaches include support vector machine (SVM) modeling methods [37], computational alignment approaches [38], and statistical tools such as t-test and analysis of variance linear regression [39-41].

A.3.2 Signal Processing Challenges: Random-Sequence Microarrays

The recently developed immunosignaturing technology uses microarrays with random-sequence peptides to associate antibodies to a pathogen or infectious agent, in a patient's blood sample [32, 42-49]. The immunosignatures can potentially provide pre-symptomatic diagnosis for infectious diseases [35,44,48]. The large number of peptide sequences on each microarray, and the attraction of the ability to diagnose as many

pathological ailments as possible, renders a challenging problem in signal processing. This is further complicated by the fact that, in general, training data is not available. Current processing methods include statistical tests [45] and supervised classification and learning methods such as support vector machines [43,48]. Recently, we have developed adaptive learning methodologies for unsupervised clustering integrated with immunosignature feature extraction approaches [50-52].

This work develops new algorithms for analyzing and processing random peptide sequences in the TF plane in order to recognize pathogens from variations in antibody profiles without any prior information. Given immunosignaturing random-sequence peptide microarray data for an individual, the task is to detect and identify the binding sites of antibodies for target antigens. These binding sites, or linear epitopes, are short continuous sub-sequences of the peptide sequence that correspond to the part of an antigen that is recognized by the antibodies [32,53,54]. Detecting which peptides bind to which antibodies by identifying the corresponding antibody sub-sequence binding sites using immunosignaturing data is very useful as one dataset contains can localized information on multiple pathogens [52]. As a result, the detection and identification algorithms can be used to characterize antibody specificity for the molecular recognition of the immune system or for deciphering molecular mechanisms for various diseases.

A.4 Random Sequence Peptide Microarrays

Immunosignaturing is a microarray-based technology that uses random peptide sequences to provide a comprehensive profiling of a person's antibodies [42,44,55]. It has been shown that a person's antibody profile, about 109 different antibodies in the blood at a given time, is a sensitive indicator of the person's health status. Part of the body's response to a foreign pathogen is to create antibodies which identify and aid in

the destruction of that pathogen. Pathogen detection and determination is possible due to a uniquely identifying amino acid sequence on its exterior called an antigen. The antibodies created in response to the pathogen are designed to only bind to that specific antigen sequence, or one that is very similar. As the antibodies amplify when the host is exposed to an infectious agent, the amplified antibody response enables monitoring a disease upon infectivity. Rather than trying to identify an antibody by designing a microarray specific to a pathogen, the concept of immunosignaturing is to identify an entire immune response. This is achieved by printing an array with many different random peptides, so that small subsets of peptide sequences are similar enough to antigen sequences of specific pathogen antibodies to bind to them.

The immunosignaturing technology has been developed by the director and researchers of the Center for Innovations in Medicine (CIM) at the Biodesign Institute at Arizona State University [56]. In particular, the random-peptide microarray data used for algorithm demonstration in this work was provided by CIM. Information on the technology, such as a description of the equipment, arrays, and a technological overview, can be found at `http://www.immunosignature.com`. The immunosignaturing technology currently employs slides spotted with peptides, resulting in microarrays with 330,000 peptide sequences (330k chip). The peptides sequences are 20 amino acids long and a random number generator is used to generate the specific peptide sequences. Other than cysteine that is used as the C-terminal amino acid, all natural amino acids are included in the peptide sequence generation. As a result, the peptide sequences are random and not related to any naturally occurring peptide sequence; however, the sequence on each spot on the slide is known. This ensures that the peptide array is not designed to monitor one specific disease or a set of diseases. Array peptides are designed to fluoresce in proportion to an antibody binding strength when light of a specific frequency is shined on them. Peptides with attached

antibodies are expected to fluoresce more brightly than the those not related to the antigen. Multiple identical peptides are printed within a predefined circular area on the array; after the sample is given sufficient time to bind, the array is washed and then illuminated. An image of the fluorescing array is taken, and the median fluorescence value of the pixels in each circular area is calculated and recorded. The resulting data used for analysis is the peptide sequence of amino acids and its corresponding median fluorescence value at each array spot. Using the data from the whole array, the problem is to detect the highly fluorescing peptides and identify the corresponding underlying pathogens.

This is not a simple detection and identification problem; processing can be complicated by the fact that there are additional macromolecules in blood samples that can also bind to peptides due to hydrogen bonding, electrostatic interactions and van der Waals forces [57]. The concept of adding a large number of random peptides on the array is novel as more pathogens can be detected on a single patient. However, the large number of sequences to process also increases the number of sequences that are close in structure to more than one pathogen's antigen. Antibodies bind with enough variability that trends across multiple peptides must be used. However, a significant difficulty in finding these trends is that only a sub-sequence of the peptide which binds to the antibody is responsible for the binding, and within that sub-sequence there can be one or two peptides which have little or no effect on the binding strength. Determining which peptide sub-sequences are responsible for that binding must be determined using multiple peptides with similar sub-sequences [44,45].

A.5 Time-Frequency Processing of Peptide Sequences

The novel signal processing algorithms presented in this work aim to improve pathogen detection and identification performance when using immunosignaturing

random peptide sequences. Toward this end, advanced signal processing methodologies are exploited to first map amino acid sequences to unique and highly TF localized waveforms and then use matched TF representations to identify specific peptide subsequences.

A.5.1 Mapping Peptide Sequences to Time-frequency Waveforms

The biosequence-to-waveform mapping considered must provide a unique waveform in the TF plane for each peptide sequence. When deciding on appropriate waveforms to use in the mapping, the waveform parameters and properties must be selected to ensure uniqueness in peptide representation and robustness in matched correlation-based processing, respectively. Following the scheme we adopted in [28]. Gaussian waveforms are selected for mapping as they are the most localized waveforms in both time and frequency [58]. A basic Gaussian waveform $g(t)$ is first obtained as

$$g(t) = \frac{1}{(\pi\sigma^2)^{1/4}}\, e^{-t^2/(2\sigma^2)}, \quad t \in (-T_g/2, T_g/2)\,, \tag{A.1}$$

with unit energy and centered at the origin in the TF plane. The parameter σ^2 affects the waveform's duration T_g and spread in frequency. When this waveform is time-shifted by nT and frequency-shifted by kF,

$$g_{n,k}(t) = g(t - nT)\, e^{j2\pi kF(t-nT)}, \quad t \in (nT - T_g/2, nT + T_g/2) \tag{A.2}$$

for integer n and k, the resulting Gaussian waveform is highly-localized at the TF point (nT, kF). Note that the time shift step $T > T_g$ and the frequency shift step F, and thus σ^2 in (A.1), are chosen to ensure that the spacing between the time-freqency shifted Gaussian waveforms is compact and the waveforms are non-overlapping.

For the biosequence-to-waveform mapping, the time shift and frequency shift are used to uniquely represent properties of the amino acids in the peptide sequence. Each

of the twenty possible different amino acids in a peptide sequence can be characterized by a unique one-letter code, as shown in the first two columns of Table A.1. For the mapping, twenty possible frequency shifts kF, $k = 1, \ldots, 20$, in Equation (A.2) are used to represent the twenty different types of amino acids, as shown in the third column of Table A.1. The position of the amino acid in the peptide sequence is mapped to the time shift parameter nT in (A.2). Considering a peptide sequence of length $N = 20$ amino acids, N time shifts are needed to represent the peptide sequence; the number of time shifts is the same as the length of the sequence. A TF representation of all possible Gaussian waveforms needed to map peptide sequences of length $N = 10$ amino acids is demonstrated in Figure A.1(a).

Considering a peptide sequence $p[n] = \alpha_n$, $n = 1, \ldots, N$, of N amino acids $\alpha_1\, \alpha_2 \ldots \alpha_{N-1}\, \alpha_N$, the mapping function $f[\{\alpha_n\}] = k$ is used to identify the one-letter code representing the amino acid α_n and its corresponding frequency shift kF from Table A.1. Note that the range of the mapping function $f[\{\cdot\}]$ is the set of positive integers, $k = 1, \ldots, 20$; the domain of the function consists of the one-letter codes from Table A.1. Using this mapping function, the resulting waveform that is used to map peptide sequence $p[n]$ is given by

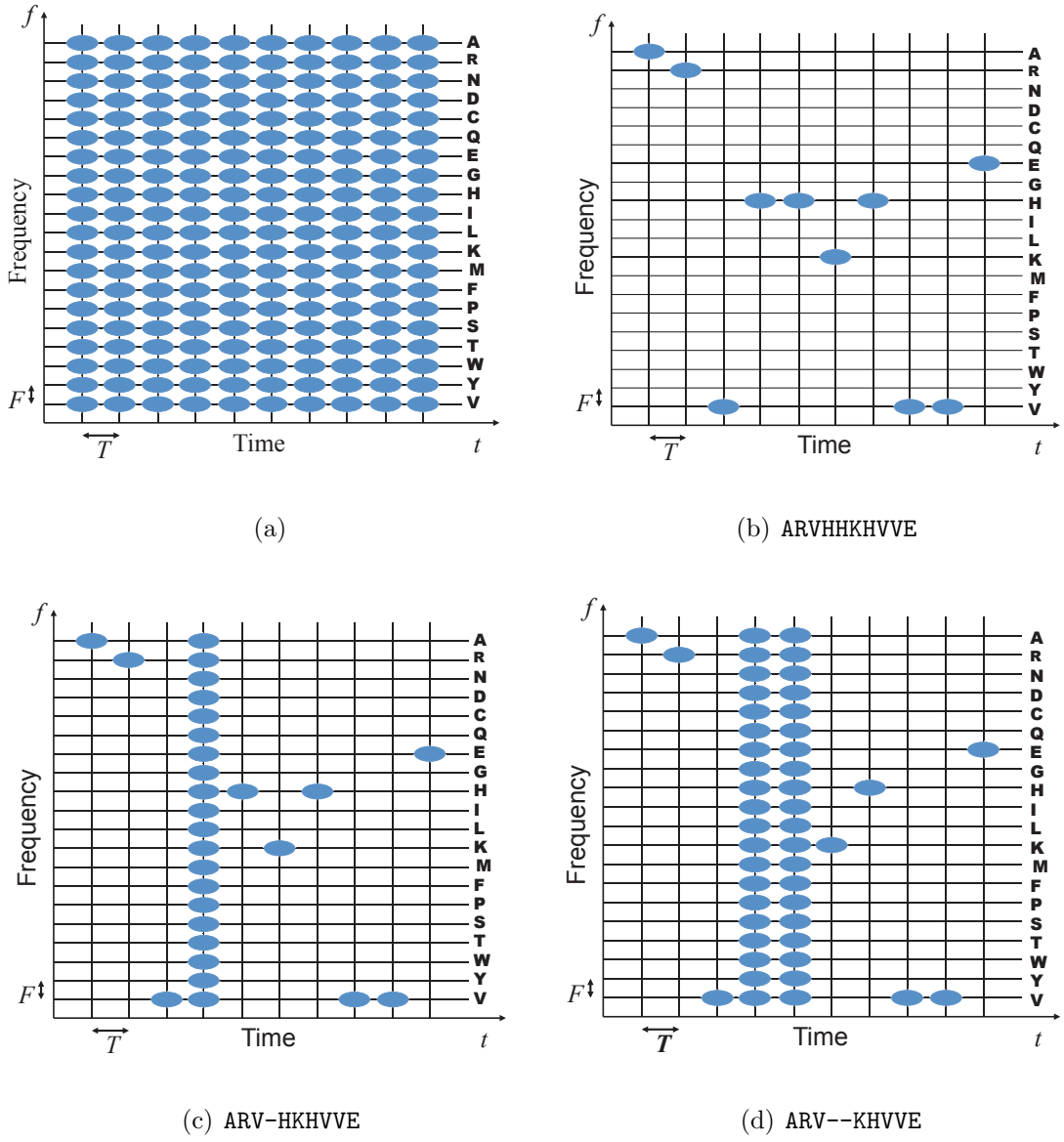
$$g_{\text{pept}}(t) = \sum_{n=1}^{N} g_{n,f[\{\alpha_n\}]}(t; p) = \sum_{n=1}^{N} g(t - nT)\, e^{j2\pi f[\{\alpha_n\}]F(t-nT)}\,. \tag{A.3}$$

The duration of the overall waveform $g_{\text{pept}}(t)$ $N\,T + T_g$.

An example of a peptide sequence of length $N = 10$ is given by `ARVHHKHVVE`; its corresponding TF representation is shown in Figure A.1(b). The waveform in (A.3) used to map this sequence is a linear combination of 10 TF-shifted Gaussian waveforms. Ten unique time shifts are used in the mapping; the frequency shifts are not unique since the same amino acid can occur multiple times in a peptide sequence. It follows that there is only one Gaussian waveform at each time shift but (possibly)

Amino Acid	**One-letter Code**	**Mapped Frequency**
Alanine	A	20 F
Arginine	R	19 F
Asparagine	N	18 F
Aspartic acid	D	17 F
Cysteine	C	16 F
Glutamic acid	E	15 F
Glutamine	Q	14 F
Glycine	G	13 F
Histidine	H	12 F
Isoleucine	I	11 F
Leucine	L	10 F
Lysine	K	9 F
Methionine	M	8 F
Phenylalanine	F	7 F
Proline	P	6 F
Serine	S	5 F
Threonine	T	4 F
Tryptophan	W	3 F
Tyrosine	Y	2 F
Valine	V	F

Table A.1: Frequency Mapping of Twenty Amino Acids.



(a)

(b) ARVHHKHVVE

(c) ARV-HKHVVE

(d) ARV--KHVVE

Figure A.1: Time-Frequency Representation of Gaussian Mapped Waveforms (a) for Peptide Sequences of 10 Amino Acids in Length; (b) for Amino Acid Sequence ARVHHKHVVE; (c) for the Same Sequence with Any Substitution in the 4th Amino Acid position, ARV-HKHVVE; (d) for the Same Sequence with Any Substitutions in the 4th and 5th Amino Acid Positions, ARV--KHVVE.

multiple Gaussian waveforms at different frequency shifts; a peptide sequences does not necessarily consist of all possible 20 amino acids. For the example of the length 10 peptide sequence ARVHHKHVVE in Figure A.1(b), and using the mapping in (A.3) and the information from Table A.1, $\alpha_1\,\alpha_2\,\alpha_3\,\alpha_4\,\alpha_5\,\alpha_6\,\alpha_7\,\alpha_8\,\alpha_9\,\alpha_{10} = \texttt{ARVHHKHVVE}$ and $f[\{\alpha_1\}]$=20, $f[\{\alpha_2\}]$=19, $f[\{\alpha_3\}]$=$f[\{\alpha_8\}]$=$f[\{\alpha_9\}]$=1, $f[\{\alpha_4\}]$=$f[\{\alpha_5\}]$=$f[\{\alpha_7\}]$=12, $f[\{\alpha_6\}]$=9, and $f[\{\alpha_{10}\}]$= 15. Specifically, the three histidine (H) amino acids in the sequence are represented in Figure A.1(b) by the three Gaussian waveforms at the same frequency shift $12\,F$ and different time shifts, $4\,T$, $5\,T$, and $7\,T$, respectively.

A.5.2 Processing Waveforms of Mapped Peptide Sequences

The peptide sequence mapping in Equation (A.3) results in a linear combination of non-overlapping Gaussian signals in the TF plane. A linear epitope or small continuous segment of the peptide sequence can be used as an antigenic determinant to a pathogen's antibodies. Identifying epitopes can be seen as searching for potential sub-sequences that are either repeated very often or are frequently repeated with significant binding strength on the microarray. After waveform mapping, the detection and identification problem of epitopes or repeated sub-sequences over a large number of peptide sequences on a microarray becomes an estimation problem of the matched Gaussian waveform parameters representing the amino acids in the epitope. As a result, epitope waveform parameter (EpiWP) estimation can be performed using matched signal expansion algorithms, such as the MPD [59]. Specifically, identifying repetitions in the sequences maps to estimating matched parameters in the waveforms.

The MPD is an iterative algorithm that can decompose a waveform into a linear combination of weighted dictionary waveforms. The dictionary waveforms are formed by TF shifting a basis waveform that is selected to be well-matched to the analysis

waveform. The MPD can be applied to the EpiWP estimation problem using the Gaussian waveform in (A.1) as the dictionary basis signal. Then the epitope mapped waveform to be decomposed and the MPD dictionary waveforms are in the form of A.3. In particular, mapped epitope candidate (MEpiC) waveforms are formed by considering possible amino acid sub-sequences from the peptide sequences. Using the MPD to decompose an MEpiC waveform results in a small set of MPD features that uniquely characterize the MEpiC waveforms; those features are then searched over all mapped peptide sequences on the microarray. A suitably derived metric for the number of peptide sequences identified to have the matched MEpiC waveform can then be used to indicate whether the epitope candidate could be related to an epitope of the antibodies of a particular pathogen.

Assuming an epitope of length L, the MEpiC waveform $g_{\text{epit}}(t)$ is given by Equation (A.3) with N replaced by L, the length of the epitope. [1] At each iteration, the MPD identifies a single TF shifted Gaussian waveform from the MEpiC waveform. This is accomplished by finding the best match between each of the mapped amino acids forming the MEpiC waveform $g_{\text{epit}}(t)$ and possible mapped amino acids $g_{\text{pept},n,f[\{\alpha_n\}]}(t)$ forming the peptide waveform. The MPD requires L iterations to find a match of the MEpiC waveform within the mapped peptide waveforms. At the start of the MPD algorithm, the best matched dictionary waveform between the MEpiC waveform and the mapped peptide amino acid waveforms is obtained as

$$g^{(1)}_{n_1,f[\{\alpha_{n_1}\}]}(t) = \underset{n}{\text{argmax}} \int g_{\text{epit}}(t)\; g_{\text{pept},n,f[\{\alpha_n\}]}(t)\; dt\,, \tag{A.4}$$

where $g^{(1)}_{n_1,f[\{\alpha_{n_1}\}]}(t)$ is a Gaussian waveform centered at time shift $n_1\,T$ and frequency shift $f[\{\alpha_{n_1}\}]\,F$, and n_1 is the value of n that yields the maximum correlation value in

[1] Note that the same notation, α_n, is used to denote amino acids in peptide sequences and amino acids in epitope sequences, which are sub-sequences of the peptide sequences. The specific type of sequence, peptide or epitope, is differentiated, when needed, using the notation $g_{\text{pept}}(t)$ and $g_{\text{epit}}(t)$, respectively.

(A.4) after the first iteration. At the ℓth iteration, $\ell = 2, \ldots, L$, the residual MEpiC waveform is given by

$$r_{\text{epit}}^{(\ell)}(t) = g_{\text{epit}}(t) - \sum_{m=1}^{\ell-1} g_{n_m, f[\{\alpha_{n_m}\}]}^{(m)}(t)\,.$$

The best matched dictionary waveform between the residual MEpiC waveform and the mapped peptide waveform is given by

$$g_{n_\ell, f[\{\alpha_{n_\ell}\}]}^{(\ell)}(t) = \underset{n}{\text{argmax}} \int r_{\text{epit}}^{(\ell)}(t)\; g_{\text{pept},n,f[\{\alpha_n\}]}(t)\; dt\,. \tag{A.5}$$

The discrete value n_ℓ is the sequence position index n that yields the maximum correlation value in (A.5) at the ℓth iteration. Note that there are no correlation coefficients to consider in the expansion as the Gaussian waveforms are normalized to have unit energy. The algorithm iteratively continues until L iterations, when there are no more matches left between the MEpiC waveform and the mapped peptide waveform. After L iterations, the decomposed mapped peptide waveform is given by

$$\tilde{g}(t) = \sum_{\ell=1}^{L} g_{n_\ell, f[\{\alpha_{n_\ell}\}]}^{(\ell)}(t) + r_{\text{epit}}^{(L+1)}(t)\,. \tag{A.6}$$

The matched MEpiC waveform components are given by the summation term in the right-hand side of Equation (A.6); the unmatched ones are in the residue $r_{\text{epit}}^{(L+1)}(t)$. The Gaussian waveform matching can then be used to obtain an epitope identification metric in terms of the energy of the decomposed Gaussian waveform components. The metric, for a candidate epitope, is given by

$$\text{s}_{\text{epit}} = \int \Big| \sum_{\ell=1}^{L} g_{n_\ell, f[\{\alpha_{n_\ell}\}]}^{(\ell)}(t) \Big|^2 dt \tag{A.7}$$

Note that each mismatch between the MEpiC waveform and the matched peptide waveform decreases the matching metric by one as the energy of the decomposed term also decreases by one.

The MPD algorithm can also be used for matching MEpiC waveforms which model biologically relevant substitutions. The matching is performed using the same MPD algorithm with a modification to the MEpiC waveforms. This is demonstrated in Figures A.1(c) and A.1(d) for the length 10 sequence `ARVHHKHVVE` represented in the TF plane in Figure A.1(b). In Figure A.1(c), the same sequence is considered but with a substitution allowed by any amino acid in the 4th position. The effect on the MEpiC waveform is to include a Gaussian waveform at each frequency shift at the 4th position (or time shift); this implies that any mapped peptide waveform is matched to the MEpiC waveform at the 4th time shift. The same sequence but with two substitutions in the 4th and 5th amino acid positions is demonstrated in Figure A.1(d).

A.6 Epitope Waveform Parameter Estimation

The epitope waveform parameter estimation algorithm consists of three main steps. During the first step, the candidate epitope and peptide amino acid sequences are mapped to Gaussian waveforms, following the discussion in Section A.5.1. During the second step, the peptide sequences are down selected by first pre-processing the peptide sequences, and then applying some selection criteria and thresholding; the reduced number of peptides after selection are the ones most likely to have been bound to by antibodies. The third step performs the epitope waveform parameter estimation using the MPD-based matching approach discussed in Section A.5.2. The steps are summarized in Figure A.2.

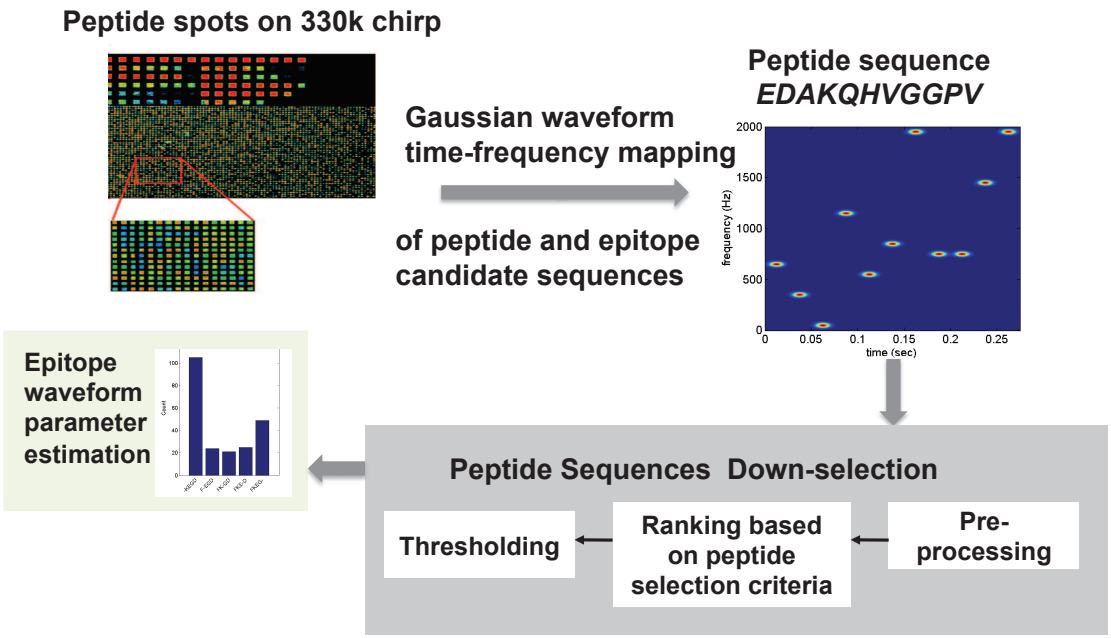


Figure A.2: Block Diagram Depicting the Algorithm for Epitope Waveform Parameter (EpiWP) Estimation.

A.6.1 Peptide Selection Method

Pre-processing. Peptide array data from individual disease samples are median normalized to account for the different binding times required. Some samples require longer time to bind fully to the array before the sample solution is rinsed off.

Ranking Based on Peptide Selection Criteria. As the number of microarray random peptide sequences, M_p, is very large for efficient processing, the peptide sequences need to be ranked according to some peptide selection criteria, and then a selected smaller number of peptides, M_s, can be used as input to the EpiWP estimation algorithm. One peptide selection criterion is based on fluorescent intensity levels; the peptides with the highest fluorescent intensity levels, or levels above some

background fluorescent intensity threshold, are selected as they correspond to the peptides that bind to the antibodies. For some datasets, it is possible for antibodies to bind weakly to peptides that do not have the highest fluorescence values. As a result, a different peptide selection criterion is needed for these datasets. The criterion is based on finding correlations or dependence between multiple datasets; the fluorescent intensity levels from multiple datasets can be compared in order to select peptides with high fluorescence values relative to the comparison data.

The second peptide selection criterion is applied using Pearson's correlation coefficient between the fluorescent intensity levels of the array peptides and a binary indicator vector. Assuming D microarray datasets for comparison, with M_p peptides per microarray dataset, the correlation coefficient for the mth peptide, $m = 1, \ldots, M_p$, at microarray $\tilde{d}$, is computed as

$$r_{\tilde{d},m} = \frac{\sum_{d=1}^{D} \left(\mathrm{fl}_{d,m} - \bar{\mathrm{fl}}_m\right)\left(b_{\tilde{d},d} - (1/D)\right)}{\left(\sum_{d=1}^{D} \left(\mathrm{fl}_{d,m} - \bar{\mathrm{fl}}_m\right)^2\right)^{1/2} \left(\sum_{d=1}^{D} \left(b_{\tilde{d},d} - (1/D)\right)^2\right)^{1/2}} \tag{A.8}$$

where $\mathrm{fl}_{d,m}$ is the fluorescent intensity of the mth peptide of the dth array,

$$\bar{\mathrm{fl}}_m = \frac{1}{D} \sum_{d=1}^{D} \mathrm{fl}_{d,m} \,,$$

is the fluorescence sample mean of the mth peptide across all D microarray datasets, and $b_{\tilde{d},d}$ is 1 if $\tilde{d} = d$ and 0 otherwise,

One example of a monoclonal antibody for which a different peptide selection criterion can give different estimation results is `2C11`. For this monoclonal antibody, using the fluorescent intensity peptide selection criterion demonstrates that the antibodies bind weakly to the peptides relative to background binding and cannot be detected at any threshold. In particular, there are many peptides with high fluorescent intensity that do not bind to the `2C11` antibody. This is illustrated in Figure A.3(a), where

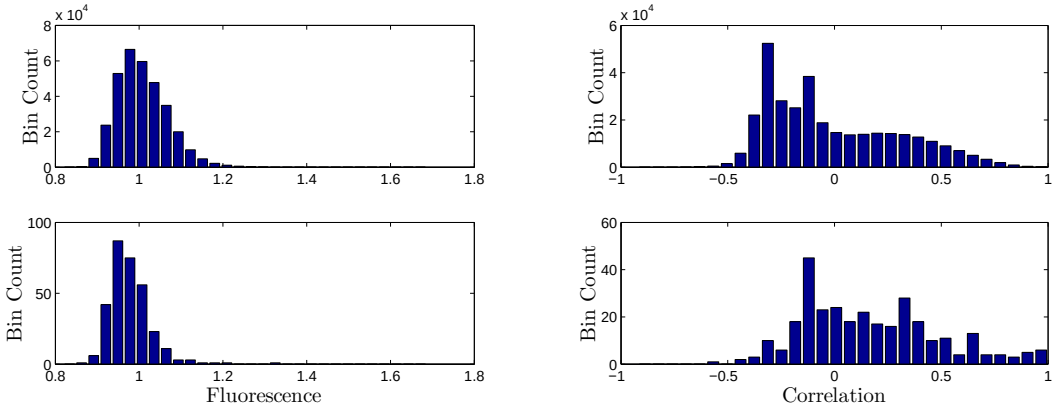


Figure A.3: Histogram Plots for the Monoclonal Antibody `2C11` Using Values of (a) Fluorescence and (b) Correlation. Top Plots Process All Peptide Sequences; Bottom Plots Process Only Peptides with the Exact Epitope Sub-Sequence.

the figure on the top is a histogram of all of the 330k fluorescent intensity levels on the array, while the figure on the bottom is a histogram of just the fluorescent intensity levels of peptides which contain a sub-sequence of the `2C11` epitope. For this dataset, as there are peptides with higher fluorescent intensities than most of the peptides with the mAb epitope, the fluorescent intensity selection criterion fails to provide correct epitope estimates. On the other hand, if the correlation value peptide selection criterion is applied, an improved epitope estimation performance. This is demonstrated in Figure A.3(b), where many of the peptides with the largest correlation values are also the peptides which contain epitope sub-sequences. Note that an epitope sub-sequence is four or more contiguous amino acids from the epitope and that the fluorescent intensity levels in Figure A.3(a) were logarithmically transformed to improve visualization.

Thresholding. Depending on the peptide selection criterion, thresholding is used to keep $M_s \ll M_p$ peptide sequences as input to the epitope estimation. A background fluorescent intensity threshold is used with the fluorescent intensity criterion. With the correlation coefficient criterion, the correlation coefficient values $r_{\tilde{d},m}$ of the mth peptide, $m = 1, \ldots, M_p$, on the $\tilde{d}$ array in (A.8) are first ranked in descending order and then compared to some threshold.

A.6.2 Epitope Estimation Algorigthms

The epitope candidate sequences are derived from the remaining Ppeptide array sequences obtained after applying the selection method in Section A.6.1. There are three different methods considered for epitope waveform parameter (EpiWP) estimation, resulting in the detection and identification of the epitope candidate sequences. The epitope candidate sequences for the EpiWP-1 estimation method include all possible sub-sequences of length L adopted from the peptide microarray sequences. The epitope candidate sequences for the EpiWP-2 estimation method include all sub-sequences of length L from the peptide array sequences, together with the sub-sequences formed by allowing for a single amino acid substitution (by any other type of amino acid). The epitope candidate sequences for the EpiWP-3 estimation method include all sub-sequences of length L from the peptide array sequences, together with the sub-sequences formed by allowing for two adjacent amino acid substitutions. Note that the allowable substitutions in EpiWP-2 and EpiWP-3 are only possible any amino acid in a sub-sequence that is not at the amino-terminus or N-terminus (start of an amino acid chain) or the C-terminus (end of amino acid chain).

The main steps of the estimation algorithm are summarized in Algorithm 13. Using the down-selected M_s peptide sequences, the MPD is used to compare the peptide sequences to M_e epitope candidate sequences. The overall matching score

uses the metric in (A.7) and the number of peptide sequences that include each of the M_e epitope candidate sequences. Algorithm 14 finds the maximum match between two sequences with dissimilar lengths. The two sequences are peptide array sequence of length N and the epitope candidate sequence of length L, where $N > L$. The algorithm first maps both sequences to Gaussian waveforms and then uses the MPD to perform the matching. The number of maximum matches found using Algorithm 13 is recorded, and epitope candidate sequences are sorted in descending order according to the number of peptides they were found in. The epitope candidate sequences that occur most frequently are the top epitope estimates.

Algorithm 13 Matches between Peptides and Epitope Candidate Sequences.

for $i = 1$ to M_s **do**

  for $j = 1$ to M_e **do**

    Run Algorithm 14 on the ith peptide and jth epitope candidate sequences

    Record the number of maximum matches for each epitope candidate sequence

  end for

end for

Algorithm 14 Maximum Match between Two Sequences.

for $n = 1$ to $N - L + 1$ **do**

  Map peptide sequence $p[m]$, $m = n, \ldots, n + L - 1$, onto TF waveforms $g_{\text{pept}}(t)$

  Map epitope candidate sequence $e[l]$, $l = 1, \ldots, L$, onto TF waveforms $g_{\text{epit}}(t)$

  Perform MPD using $g_{\text{pept}}(t)$ and $g_{\text{epit}}(t)$ to find score s_{epit}

end for

A.6.3 Evaluation of Epitope Estimation

In order to evaluate the performance of the random-sequence peptide microarray with the EpiWP estimation method for identifying antibody epitopes, eight monoclonal antibodies (mAbs) were acquired. The mAbs used have known epitopes that were used to probe the microarray. Monoclonal antibodies are used in the evaluation, instead of blood samples from patients, as the mAbs bind to a single linear epitope selected for high specificity for the antigen [44, 60-63]. On the contrary, epitopes for most diseases are not known; even if the epitope for a single strain of a disease is known, it may not be known for the specific strain of the analyzing sample. The mAb random-sequence peptide microarray data were provided by CIM [56]; each microarray sample consists of 330,000 peptide sequences (330k chip). Although this is a large number of sequences on the array, only a small percentage of the sequences bind to different mAbs.

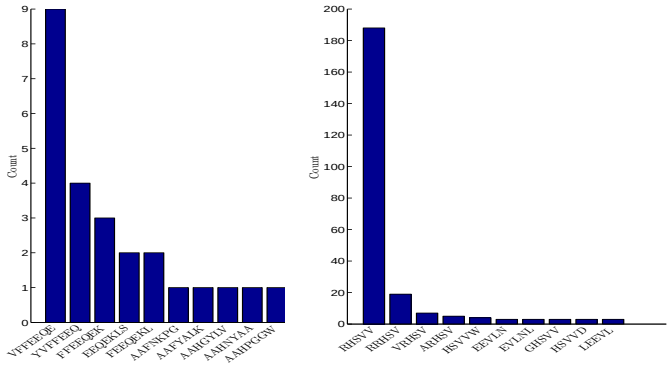
Table A.2 provides a list of the eight mAbs used to demonstrate epitope waveform parameter estimation. The known epitope of each mAb is provided in the third column of this table. The last column provides the estimated epitopes with varying lengths. The EpiWP estimation method performed well for all but the monoclonal antibody `ab8` epitope. Based on this result, the mAb epitope estimation performance is about 88% accurate.

For most of the monoclonal antibody samples, the EpiWP-1 estimation method performs well in estimating the true epitope. Two examples of this are for estimating the epitopes of mAbs `2C11` and `ab1`. The true epitope of the monoclonal antibody `2C11` is `NAHYYVFFEEQE`; using EpiWP-1 finds `YVFFEEQE` as the epitope. Similarly, the epitope for the monoclonal antibody `ab1` is `RHSVV`; EpiWP-1 estimates epitope `RHSVV`. The top results for these two epitopes are shown in Figures A.4(a) and A.4(b). In

Antibody	Full Epitope	Estimated Epitope
2C11	NAHYYVFFEEQE	YVFFEEQE
4C1	QAFDSH	AFDSH
A10	EEDFRV	EDFRV
Ab-1	RHSVV	RHSVV
Ab-8	SDLWKL	-
DMLA	AALEKD	ALEKD
FLAG	DYKDDDDK	KDGD
HA	YPYDVPDYA	YDAPE

Table A.2: Epitope Estimates for Eight Monoclonal Antibodies.

some of monoclonal antibodies, an obvious substitution but not the exact epitope is found. This is demonstrated in Figures A.5(a) and A.5(b) for the monoclonal antibody HA with true epitope YPYDVPDYA. The EpiWP-1 estimation method results in candidate epitopes YDAPE and PYDAP. Allowing one amino acid substitutions as in estimation method EpiWP-2, the candidate epitopes are -YDAP and YDAP-. One way to interpret this is that the Y in YDAPE is not required for binding, nor is the final P in PYDAP; while the Y in YDAPE is part of the true epitope, the second P in PYDAP is not.

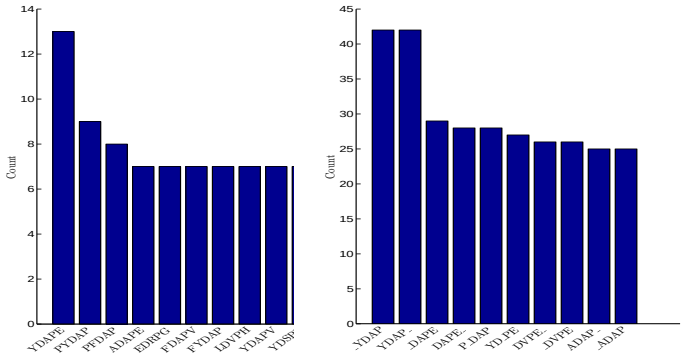


(a) (b)

Figure A.4: Top Epitope Estimates Using Estimation Method EpiWP-1 for Antibody (a) `2C11` (True Epitope `NAHYYVFFEEQE`) and (b) `ab1` (True Epitope `RHSVV`).

A.6.4 Comparison with Existing Epitope Identification Methods

Existing sequence alignment approaches [10, 21-26, 64] can potentially be used for the epitope estimation problem, in order to find similarities between peptide and epitope sequences. However, most of these approaches were optimized for very long amino acid sequencers and not for short-length peptide sequences [65]. An approach for finding a motif or pattern among the peptides is a direct sequence analysis approach that compares peptide sequences to epitope sequences based on their primary structure. This was demonstrated in [65] using data obtained using phage display technology; the scoring used for this approach is similarity between the sequences. Other approaches use pattern graphs, combinatorics for motif finding, exhaustive length and substitution analysis, and optimization methods to find motifs by maximizing scoring functions [66-74]. A most recent statistical based approach arranges peptides in position specific scoring matrices and computes their mean value for each



(a) (b)

Figure A.5: Top Epitope Estimates for Antibody `HA` (True eEpitope `YPYDVPDYA`) Using Estimation Method (a) EpiWP-1 and EpiWP-2.

position; a threshold value is then used to identify positions where the mean differs significantly [75].

Directly applied to immunosignaturing, a method called GuiTope was presented in [47] for mapping random-sequence peptides to protein sequences. The method is based on using a scoring matrix and a local alignment approach that compares similarity results using a score threshold. Using GuiTope, monoclonal antibody epitopes were estimated with about 74% to 81% accuracy.

A.7 Efficient Implementation of Epitope Estimation

The aforementioned pathogen detection and identification methods need to be repeated tens to hundreds of thousands of times to scan through all necessary peptide sequences when estimating a single epitope. Therefore, for this method to be useful, it is very important to decrease the runtime of the epitope estimation algorithm. When implemented, the algorithm spends most of its time computing the multiplication

in Equation (A.5). The inner product computational step involves sample multiplication and summation of all products. Reducing the number of multiplications can drastically decrease the algorithm's runtime, as discussed next.

Reducing Number of Multiplications. To increase code efficiency, the time domain waveforms described in Section A.5.1 can be constructed by selecting relevant parameters T, F, and σ^2 so that the Gaussian waveforms are close together in TF but are non-overlapping. While the Gaussian waveforms are theoretically non-zero across all time, setting $T = 3\sigma^2$ and fixing the time-bandwidth product to be $T\,F = 1$, is sufficient for the accuracy required in this application. The resulting Gaussian waveforms can also be sampled at Nyquist to minimize the number of samples needed to uniquely represent each frequency shift.

Frequency Domain Implementation of Epitope Estimation. Even after taking steps to reduce the number of time domain multiplications, it is still more efficient to represent the waveforms in the frequency domain, where each waveform is sampled once at the location of all the frequency shifts. Because the Gaussian waveforms in the dictionary are non-overlapping, each of the frequency domain samples will either be a 1 or a 0.

Eliminating all Multiplications. For the EpiWP-1 estimation method, the multiplications in Equation (A.5) can be eliminated simply by counting the number of Gaussian waveforms, in each epitope amino acid and peptide amino acid waveform pairs, that occur at the same TF location. When matched in the TF plane, a maximum matching score is obtained when all waveform pairs share the same TF support. The frequency domain implementation can still be used as it uses the smallest number of samples.

A.8 Conclusions

This work presented advanced signal processing approaches to analyze immunosignature biosequences. Immunosignaturing technology uses random sequence peptide microarrays to assess health status by associating antibodies from a biological sample to immune responses. The immunosignature processing requires the detection and identification of antibody epitopes from the microarray peptide sequences to discriminate between pathogens and diagnose diseases. This is achieved by first mapping characteristics of peptide and epitope sequences to parameters of highly-localized Gaussian waveforms in the time-frequency plane. After down-selecting the large number of sequences from a microarray, time-frequency based matching methods are used to estimate epitope candidates corresponding to specific pathogens. The performance of the novel epitope detection and identification method is demonstrated using eight monoclonal antibodies. The candidate sequences that resulted in a stronger response for one antibody over the others corresponded well with the actual epitope sequences that generated the monoclonal antibodies.

Acknowledgements

The authors gratefully acknowledge Dr. Stephen A. Johnston, Dr. Phillip Stafford and Joshua Richer (Director, Associate Professor Research, and Graduate Research Assistant, respectively, of the Center for Innovations in Medicine, Biodesign Institute, Arizona State University) for providing the immunosignaturing data used in this work and for valuable discussions on the epitope estimation algorithm. This work was supported by the Defense Threat Agency Reduction Contract HDTRA1-12-C-0058.

References

1. Anastassiou D. Genomic signal processing. IEEE Signal Process Mag. 2001:8-20.

2. Zhang XY, Chen F, Zhang Y-T, Agner SC, AkayM, Lu Z-H,WayeMMY, Tsui SK-W. Signal processing techniques in genomic engineering. Proc IEEE. 2002; 90(12):1822-33.

3. Vaidyanathan PP. Genomics and proteomics: a signal processors tour. IEEE Circuits Syst Mag. 2004:6-29.

4. Dougherty ER, Datta A, Sima C. Research issues in genomic signal processing. IEEE Signal Process Mag. 2005; 22(6):46-68.

5. Aydin Z, Altunbasak Y. A signal processing application in genomic research: protein secondary structure prediction. IEEE Signal Process Mag. 2006:128-31.

6. Schonfeld D, Goutsias J, Shmulevich I, Tabus I, Tewfik AH. Introduction to the issue on genomic and proteomic signal processing. IEEE J Sel Top Signal Process. 2008; 2.

7. Rowen L, Mahairas G, Hood L. Sequencing the human genome. Science. 1997; 278(5338):605-7.

8. Schuster SC. Next-generation sequencing transforms todays biology. Nature Methods. 2008; 5:16-8.

9. Durbin RM, Eddy SR, Krogh A, Mitchison G. Biological sequence analysis: probabilistic models of proteins and nucleic acids. Cambridge University Press; 1998.

10. DAvenio G, Grigioni M, Orefici G, Creti R. SWIFT (sequence-wide investigation with Fourier transform): a software tool for identifying proteins of a given class from the unan notated genome sequence. Bioinformatics. 2005; 21(13):2943-9.

11. Herzel H, Große I. Correlations in DNA sequences: the role of protein coding segments. Phys Rev E. 1997; 55(1):800-10.

12. Herzel H, Trifonov EN, Weiss O, Große I. Interpreting correlations in biosequences. Phys A Stat Mech Appl. 1998; 249(1):449-59.

13. Anastassiou D. Frequency-domain analysis of biomolecular sequences. Bioinformatics. 2000; 16(12):1073-81.

14. Dodin G, Vandergheynst P, Levoir P, Cordier C, Marcourt L. Fourier and wavelet transform analysis, a tool for visualizing regular patterns in DNA sequences. J Theor Biol 2000:323-6.

15. Berger JA, Mitra SK, Astola J. Power spectrum analysis for DNA sequences. IEEE Trans Signal Process. 2003; 2:29-32.

16. Sussillo D, Kundaje A, Anastassiou D. Spectrogram analysis of genomes. EURASIP J Adv Signal Process. 2004; 2004(1):29-42.

17. Altaiski M, Mornev O, Polozov R. Wavelet analysis of DNA sequences. Genet Anal Biomol Eng. 1996; 165-8.

18. Meng T, Soliman AT, Shyu M, Yang Y, Chen S, Iyengar SS, Yordy JS, Iyengar P. Wavelet analysis in current cancer genome research: a survey. IEEE/ACM Trans Comput Biol Bioinform. 2013; 10(6):1442-59.

19. Rockwood AL, Crockett DK, Oliphant JR, Elenitoba-Johnson KSJ. Sequence alignment by cross-correlation. J Biomol Tech. 2005; 16:453-8.

20. Rosenberg MS, Editor. Sequence alignment: methods, models, concepts, and strategies. University of California Press; 2009.

21. Smith TF, Waterman MS. Identification of common molecular subsequences. J Mol Biol. 1981; 147(1):195-7.

22. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. J Mol Biol. 1990; 215(3):403-10.

23. Kent WJ. BLATThe BLAST-like alignment tool. Genome Res. 2002; 12(4):656-64.

24. Zhu J, Liu JS, Lawrence CE. Bayesian adaptive sequence alignment algorithms. Bioinformatics. 1998; 14(1):25-39.

25. Wang W, Johnson DH. Computing linear transforms of symbolic signals. IEEE Trans Signal Process. 2002; 50(3):628-34.

26. Brodzik AK. A comparative study of cross-correlation methods for alignment of DNA sequences containing repetitive patterns. European Signal Processing Conference, 2005.

27. Brodzik AK. Phase-only filtering for the masses (of DNA data): a new approach to DNA sequence alignment. IEEE Trans Signal Process. 2006.

28. Ravichandran L, Papandreou-Suppappola A, Spanias A, Lacroix Z, Legendre C. Waveform mapping and time-frequency processing of DNA and protein sequences. IEEE Trans Signal Process. 2011; 59(9):4210-24.

29. Ravichandran L. Waveform mapping and time-frequency processing of biological sequences and structures, Ph.D., Arizona State University, Tempe, Arizona, 2011.

30. ODonnell B, Maurer A, Papandreou-Suppappola A.Waveform processing for protein multialignment by mapping locational, structural and functional attributes. Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, November 2013.

31. Delehanty JB, Ligler FS. A microarray immunoassay for simultaneous detection of proteins and bacteria. Anal Chem. 2002; 74(21):5681-7.

32. Reineke U, Ivascu C, Schlief M, Landgraf C, Gericke S, Zahn G, Herzel H, Volkmer-Engert R, Schneider-Mergener J. Identification of distinct antibody epitopes and mimotopes from a peptide array of 5520 randomly generated sequences. J Immunol Methods. 2002; 267(1):37-51.

33. Duburcq X, Olivier C, Malingue F, Desmet R, Bouzidi A, Zhou F, Auriault C, Gras-Masse H, Melnyk O. Peptide-protein microarrays for the simultaneous detection of pathogen infections. Bioconjugate Chem. 2004; 15(2):307-16.

34. Breitling F, Nesterov A, Stadler V, Felgenhauer T, Bischoff FR. High-density peptide arrays. Mol BioSyst. 2009; 5(3):224-34.

35. Price JV, Tangsombatvisit S, Xu G, Yu J, Levy D, Baechler EC, Gozani O, Varma M, Utz PJ, Liu CL. On silico peptide microarrays for high-resolution mapping of antibody epitopes and diverse protein-protein interactions. Nature Med. 2012; 18(9):1434-40.

36. Lequin RM. Enzyme immunoassay (EIA)/enzyme-linked immunosorbent assay (ELISA). Clin Chem. 2005; 51(12):2415-8.

37. Chen G, Zuo Z, Zhu Q, Hong A, Zhou X, Gao X, Li T. Qualitative and quantitative analysis of peptide microarray binding experiments using SVM-PEPARRAY. Methods Mol Biol. 2009; 570:403-11.

38. Renard BY, Lower M, Kuhne Y, Reimer U, Rothermel A, Tureci O, Castle JC, Sahin U. rapmad: robust analysis of peptide microarray data. BMC Bioinform. 2011; 12(324):1-10.

39. Wolfinger RD, Gibson G, Wolfinger ED, Bennett L, Hamadeh H, Bushel P, Afshari C, Paules RS. Assessing gene significance from cDNA microarray expression data via mixed models. J Comput Biol. 2001; 8(6):625-37.

40. Haan JR, Bauerschmidt S, van Schaik R, Piek E, Buydens L, Wehrens R. Robust ANOVA for microarray data. Chemom Intell Lab Syst. 2009; 98:38-44.

41. Hirakawa A, Hamada C, Yoshimura I. Sample size calculation for a regularized t-statistic in microarray experiments. Stat Probab Lett. 2011; 81:870-5.

42. Legutki JB, Magee DM, Stafford P, Johnston SA. A general method for characterization of humoral immunity induced by a vaccine or infection. Vaccine. 2010; 28(28):4529-37.

43. Restrepo L, Stafford P, Magee DM, Johnston SA. Application of immunosignatures to the assessment of Alzheimers disease. Am Neurol Assoc. 2011; 70:286-95.

44. Halperin RF, Stafford P, Johnston SA. Exploring antibody recognition of sequence space through random-sequence peptide microarrays. Mol Cell Proteomics. 2011; 10(3).

45. Brown JR, Stafford P, Johnston SA, Dinu V. Statistical methods for analyzing immunosignatures. BMC Bioinform. 2011; 12(349):1-15.

46. Stafford P, Halperin R, Legutki JB, Magee DM, Galgiani J, Johnston SA. Physical characterization of the immunosignaturing effect. Mol Cell Proteomics. 2012; 11(4):011 593-1-14.

47. Halperin RF, Stafford P, Emery JS, Navalkar KA, Johnston SA. GuiTope: an application for mapping random-sequence peptides to protein sequences. BMC Bioinform. 2012; 13(1):1.

48. Kukreja M, Johnston SA, Stafford P. Comparative study of classification algorithms for immunosignaturing data. BMC Bioinform. 2012; 13(1):139-52.

49. Legutki JB, Johnston SA. Immunosignatures can predict vaccine efficacy. Proc Natl Acad Sci. 2013:18614-9.

50. Malin A, Kovvali N, Zhang JJ, Chakraborty B, Papandreou-Suppappola A, Johnston SA, Stafford P. Adaptive learning of immunosignaturing peptide array features for biothreat detection and classification. Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, November 2011, pp. 1883-7.

51. Malin A, Kovvali N, Papandreou-Suppappola A, Zhang JJ, Johnston SA, Stafford P. Beta process based adaptive learning of immunosignaturing peptide-antibody factors. Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, November 2012, pp. 1651-5.

52. Malin A, Kovvali N, Papandreou-Suppappola A. Adaptive learning of immunosignaturing features for multi-disease pathologies. Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, November 2013.

53. Mierendorf RC, Hammer B, Novy RE. A method for the rapid identification of epitopes and other functional peptide domains. Mol Diagn Infect Dis. 1997; 13:107-23.

54. Yua X, Owensa GP, Gilden DH. Rapid and efficient identification of epitopes/ mimotopes from random peptide libraries. J Immunol Methods. 2006; 316:67-74.

55. Sykes KF, Legutki JB, Stafford P. Immunosignaturing: a critical review. Trends Biotechnol. 2013; 31:45-51.

56. (Online). Available: http://www.biodesign.asu.edu/research/research-centers/innovations-inmedicine/

57. Emery JS. Computational modeling of peptide-protein binding, Ph.D., Arizona State University, Tempe, Arizona, 2010.

58. Cohen L. Time-frequency analysis. Prentice-Hall; 1995.

59. Mallat SG, Zhang Z. Matching pursuits with time-frequency dictionaries. IEEE Trans Signal Process. 1993; 41:3397-415.

60. Petersen G, Song D, Hugle-Dorr B, Oldenburg I, Bautz EK. Mapping of linear epitopes recognized by monoclonal antibodies with gene-fragment phage display libraries. Mol & Gen Genet. 1995; 249(4):425-31.

61. Goding JW. Monoclonal antibodies: principles and practice. 3rd ed. Academic Press; 1996.

62. Yip YL,Ward RL. Epitope discovery using monoclonal antibodies and phage peptide libraries. Comb Chem High Throughput Screen. 1999; 2(3):125-38.

63. Clementi N, Mancini N, Castelli M, Clementi M, Burioni R. Characterization of epitopes recognized by monoclonal antibodies: experimental approaches supported by freely accessible bioinformatic tools. Drug Discov Today. 2012:1-8.

64. Randic M, Zupan J, Vikic-Topic D, Plavsic D. A novel unexpected use of a graphical representation of DNA: graphical alignment of DNA sequences. Elsevier Chem Phys Lett. 2006; 431:375-9.

65. Mandava S,Makowski L, Devarapalli S, Uzubell J, Rodi DJ. RELICa bioinformatics server for combinatorial peptide analysis and identification of protein-ligand interaction site. Proteomics. 2004; 4:1439-60.

66. I. Jonassen. Efficient discovery of conserved patterns using a pattern graph. Comput Appl Biosci. 1997; 13:509-22.

67. Rigoutsos I, Floratos A. Combinatorial pattern discovery in biological sequences: the TEIRE-SIAS algorithm. Bioinformatics. 1998; 14:55-67.

68. Alix AJ. Predictive estimation of protein linear epitopes by using the program PEOPLE. Vaccine. 1999; 18:311-4.

69. Bastas G, Sompuram SR, Pierce B, Vani K, Bogen SA. Bioinformatic requirements for protein database searching using predicted epitopes from disease-associated antibodies. Mol Cell Proteomics. 2008; 7(2):247-56.

70. Buus S, Rockberg J, Forsstrom B, Nilsson P, Uhlen M, Schafer-Nielsen C. High-resolution mapping of linear antibody epitopes using ultrahigh-density peptide microarrays. Mol Cell Proteomics. 2012; 11:1790-1800.

71. Bailey TL, Williams N, Misleh C, Li WW. MEME: discovering and analyzing DNA and protein sequence motifs. Nucl Acids Res. 2006; 34:369-73.

72. Bailey TL, Elkan C. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. Proceedings International Conference on Intelligent Systems for Molecular Biology; ISMB. International Conference on Intelligent Systems for Molecular Biology, vol. 2, pp. 28-36; 1994.

73. Andreatta M, Schafer-Nielsen C, Lund O, Buus S, Nielsen M. NNAlign: a web-based prediction method allowing non-expert end-user discovery of sequence motifs in quantitative peptide data. PLoS ONE. 2011; 6(11):1-11.

74. Meskin N, Nounou H, Nounou M, Datta A. Parameter estimation of biological phenomena: an unscented Kalman filter approach. IEEE/ACM Trans Comput Biol Bioinform. 2013; 10(2):537-43.

75. Hansen LB, Buus S, Schafer-Nielsen C. Identification and mapping of linear antibody epitopes in human serum albumin using high-density peptide arrays. PLoS ONE. 2013; 8(7):1-10.

APPENDIX B

PERMISSIONS

Permission has been granted by co-authors on "Biosequence Time-frequency Processing: Pathogen Detection and Identification" to include in this dissertation.