Multiple Neural Artifacts Suppression Using Gaussian Mixture Modeling and Probability Hypothesis Density Filtering

by

Jiewei Jiang

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved July 2014 by the
Graduate Supervisory Committee:

Antonia Papandreou-Suppappola, Chair
Daniel Bliss
Chaitali Chakrabarti

ARIZONA STATE UNIVERSITY

August 2014

ABSTRACT

Neural activity tracking using electroencephalography (EEG) and magnetoencephalography (MEG) brain scanning methods has been widely used in the field of neuroscience to provide insight into the nervous system. However, the tracking accuracy depends on the presence of artifacts in the EEG/MEG recordings. Artifacts include any signals that do not originate from neural activity, including physiological artifacts such as eye movement and non-physiological activity caused by the environment.

This work proposes an integrated method for simultaneously tracking multiple neural sources using the probability hypothesis density particle filter (PPHDF) and reducing the effect of artifacts using feature extraction and stochastic modeling. Unique time-frequency features are first extracted using matching pursuit decomposition for both neural activity and artifact signals.

The features are used to model probability density functions for each signal type using Gaussian mixture modeling for use in the PPHDF neural tracking algorithm. The probability density function of the artifacts provides information to the tracking algorithm that can help reduce the probability of incorrectly estimating the dynamically varying number of current dipole sources and their corresponding neural activity localization parameters. Simulation results demonstrate the effectiveness of the proposed algorithm in increasing the tracking accuracy performance for multiple dipole sources using recordings that have been contaminated by artifacts.

ACKNOWLEDGMENTS

I would like to thank my advisor, Prof. Antonia Papandreou-Suppappola, for providing me with the opportunity to study with her. Without her guidance, knowledge and support, this work would have been possible. I would particularly like to thank her for her patience and assistance in guiding the research project and giving thoughts and ideas at each step of the work. It has been such a pleasure and honor to be one of her students and learn from her. Her passion in signal processing as well as life attitude inspired me from the past to the future. I will always be grateful for her constant encouragement and understanding during my whole research time.

I would also like to thank Prof. Chaitali Chakrabarti and Prof. Danial Bliss for their willingness to serve in my graduate committee, and assist my thesis with their valuable insights and suggestions.

Special thanks also go to Dr. Narayan Kovvali for his thoughtful suggestions and ideas to overcome the difficulties in the research. My appreciation goes to all my SPAS lab mates, and in particular to Alexander Maurer, Brian O'Donnel, John Kota and Meng Zhou who helped me throughout my work. I cherish the time we spent discussing projects and laughing together.

Thanks to my friends, Rick and Mary, and many others, their friendship and advices is the source to gain energy. And I would like to show my greatest appreciation to my parents, who give me unconditional love and support to keep me going.

Finally, thank you all for your love and patience that make things happen.

TABLE OF CONTENTS

Page

CHAPTER Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

1.1 Background and Motivation

Neuroscience is an interdisciplinary research field that encompasses techniques devoted to a better understanding of the human brain [1]. Using various methods to measure brain activity has helped to improve our basic understanding of the brain mechanisms of cognitive processes, and more importantly, to develop better characterization of neurological disorders that impair normal function. Among the available neuroimaging techniques, the electroencephalography (EEG) and magnetoencephalography (MEG) methods are extensively used in neuroscience as they have many advantages over other techniques [2]. EEG sensors, in particular, can be easily moved and they are tolerant to patient movement. Both the EEG and MEG techniques have very high temporal resolutions, in the order of milliseconds, and involve noninvasive procedures [1]. EEG and MEG sensors measure external electromagnetic signals that can then be processed to identify and localize neural electrical activity. The electromagnetic signals are formed when information is transferred between neurons in the brain. The neurons are excitable cells whose resting state is characterized by a cross-membrane voltage difference. The signal transfer between neurons occurs in a chemical process performed by neurotransmitters over the synaptic gap, and the resulting postsynaptic potential can be modeled as a current dipole [3].

Although the EEG/MEG measurements have high temporal resolution, they are characterized by low spatial sensitivity. As a result, accurate processing is required to solve the EEG/MEG inverse problem, which is the problem of using EEG/MEG signals to estimate localization information on the current dipoles, and on their associated neural activity [4-6]. Tracking electrical neural activity by estimating current dipole information is very important for understanding the nervous system. It can help diagnose and treat neurological disorders such as epilepsy, multiple sclerosis and Alzheimer's disease as well as brain disorders such as tumors and stroke [7].

Many approaches have been proposed to solve the EEG/MEG inverse problems including local autoregressive average (LAURA) method [8], the recursively applied and projected multiple-signal classification (RAP-MUSIC) method [8-9], beamforming [10] and various Bayesian techniques [11-16]. Kalman filtering in [13] was used to estimate dipole parameters, however this filtering method cannot be effectively applied to systems that cannot be described by linear models. For nonlinear and/or non-Gaussian systems, the sequential Bayesian method can result in higher estimation performance than the Kalman filter. As the EEG/MEG measurement models are highly nonlinear, the particle filter outperformed the Kalman filter when used to estimate dipole parameters [12-14]. In [17], multiple particle filters were used to track the parameters of a known number of current sources in real time. When the number of dipoles is unknown, as in most realistic scenarios, both the number of dipoles and parameters of dipoles were estimated using the probability hypothesis density filtering (PHDF) implemented using particle filters [17].

Although the methodologies for extracting brain activity from neural recordings have improved over the years, a major concern that still remains in neural tracking is the presence of artifact signals that often corrupt the recordings. Artifacts are irregularly occurring signal that do not originate from brain activity. They can include patient movement, normal heart electrical activity, muscle and eye movement, or equipment and environmental clutter. As artifacts can affect the neural activity estimation accuracy, their detention, identification and removal from the EEG/MEG measurements are very important and necessary as their performance accuracy can affect the neural localization results [18]. Different methods for identifying and suppressing artifact signals from EEG/MEG recordings have been proposed in the literature. One of these methods uses independent component analysis (ICA), which is a statistical approach for separating a signal into its individual components based on the assumption that the components are statistically independent according to the kurtosis property of their amplitude distributions over time [19-21]. Brain activity and artifacts signals can be assumed independent as they are caused from different physiological and/or anatomical processes.

In [22], it was shown that simply using ICA for artifact suppression could lead to removing important activity signals that were falsely classified as artifact; the proposed improved method integrated ICA with wavelet thresholding (wICA). In [23], a cascaded spatio-temporal processing method was used to remove ocular (or eye related) contaminated artifacts. The empirical mode decomposition (EMD) data adaptive filtering technique was used in [24] to separate ocular artifacts from EEG recordings, and it was shown to outperform wavelet thresholding. In [25], blind source separation was first applied to EEG recordings before applying the EMD to recover neural components that leaked into artifact components. In [26], blind source separation is integrated with the recursive least squares algorithm to suppress ocular artifacts based on the amplitude relationship between the two different types of signals. This hybrid method was compared to the use of other regression algorithms as well as to the wICA approach; the comparison considered both the performance of the algorithms in removing artifacts as well as the distortion caused by the algorithm to the brain activity using time and frequency analysis. In [27], ocular artifacts are suppressed using second order blind identification methods and muscle artifacts are suppressed using canonical correlation analysis (CCA) methods; these methods appear as part of an automatic artifact removal (AAR) toolbox in EEGLAB [28]. Note that EEGLAB is an interactive Matlab toolbox for processing continuous and event-related EEG and MEG recordings. In [29], the wavelet transform was applied with selected frequency bands of neural signals to detect and remove artifacts; this approach was compared with and shown to outperform artifact removal algorithms based on the ICA, wICA, EMD and CCA methods.

The aforementioned methods of artifact suppression involved a two-step procedure to achieve neural tracking: an artifact suppression algorithm was first applied to provide neural recordings with a higher artifact-to-signal ratio and then a suitable neural tracking algorithm would need to be selected to estimate the current dipole parameters from the artifact-suppressed signals. The approach proposed in [7, 30] performed both steps simultaneously. In particular, the ICA recording components, without first undergoing artifact identification and suppression, were directly applied as input to a multiple dipole tracking algorithm. The artifacts were modeled as clutter drawn from a point Poisson process uniformly distributed in a region of the measurement space, and characteristic frequency-domain features of the artifacts were used to obtain a model of their probability density function. The particle filter and the probabilistic data association filter (PDAF) [31-32] were used to estimate the parameters of multiple dipoles in the presence of artifact clutter. Note that the PDAF is a data association method that was applied in order to compute the posterior probabilities of all possible association measurements and find the measurements that most likely originated from neural activities. This approach performed well in estimating the parameters of neural activity based on a known but small number of current dipoles and/or artifacts. As the number of dipoles and/or artifacts increased, however, the method did not always converge to provide estimates of the parameters of more than a few dipoles, as data association is very computationally intensive. Another problem with this method is the characterization of the probability density function of the artifacts based on observed frequency domain features. This feature selection approach was not shown to be robust as it necessitates pre-processing of recordings with known artifacts to observe and extract specific frequency bands for each different type of artifact.

1.2 Proposed Thesis Work

In this work, we propose an integrated one-step algorithm of neural tracking while suppressing artifacts, in a similar fashion to the algorithm in [30]. Our proposed algorithm aims to improve the neural tracking performance for an increased and unknown number of current dipoles as well as to increase the effect of artifact suppression by incorporating better-matched artifact features.

In particular, we avoid the use of data association methods by using the probability hypothesis density filter (PHDF) as in [17]. The PHDF is an approach that is based on random finite sets, and as a result, it does not require each measurement component to be associated to the corresponding neural source it originated from [16, 33]. It can be implemented using particle filtering and is less computationally intensive than data association as its computational complexity is linearly proportional to both the number of current dipoles and measurements.

In our approach, we also model artifacts as clutter drawn from a point Poisson process uniformly distributed within measurement space. However, we improve the statistical characterization of the artifact signals for use in the particle filter implementation by selecting well-matched features for different types of artifacts. These features do not need to be observed but can be directly obtained by pre-processing the ICA recording components. The feature selection is performed in two parts. We first use the matching pursuit decomposition (MPD) algorithm [34-35] to extract discriminatory deterministic features for both neural and artifact signals in the time-frequency plane. The MPD features are then used as input to a Gaussian mixture model (GMM) clustering algorithm [36-38] to provide estimates of the parameters of two unique probability density functions. These are the functions needed for the neural activity and artifact signals in the PHDF particle filtering neural tracking algorithm. This approach provides for a more accurate statistical representation needed to separate artifacts from neural activity.

1.3 Thesis Organization

This thesis is organized as follows. In Chapter 2, we discuss the measurement model for neural dipole sources and the problem of measurement contamination by the presence of artifact signals. In Chapter 3, we provide a summary on particle filtering for estimating time-varying parameters, and we discuss the PHDF and particle filter (PHDF-PF) implementation. In Chapter 4, we present our new approach of obtaining marched characterizations of the artifacts using GMMs with time-frequency features, and we provide our overall PHDF-PF tracking algorithm with the improved artifact characterizations. In chapter 5, we demonstrate the improved performance of our proposed algorithm using both simulated and real neural measurements. Our overall conclusions and future work plans are provided in Chapter 6.

CHAPTER 2

MODELS OF NEURAL SOURCES AND THE PRESENCE OF ARTIFACTS

2.1 Dipole Source Model

Neuroscience is a broad research area that studies the nervous system, including the mechanisms of how neurons process signals electrochemically. As many studies have shown, groups of neurons encode information with electrical signals and then transmit the information by synapses to other group of neurons. The chemical postsynaptic potential created when an electromagnetic signal is transferred between groups of neurons can be modeled as a localized current dipole [7]. The electrical fields produced can be recorded from the scalp as electroencephalography (EEG) signals whereas the magnetic fields produced can be recorded at a short distance from the scalp as magneto encephalography (MEG) signals.

Current dipoles are thus models for populations of neurons. Using physical-based source models for current dipoles and measurements of EEG/MEG can provide information on the location and orientation of neural activity. This can be very useful, for example, during brain surgery for patients with neurological disorders. Obtaining this information on neural source location and orientation is called the inverse problem [1] [6]. In order to solve the EEG/MEG inverse problem, a person's head is assumed to consist of nested concentric spheres of constant conductivity [2]. Based on current dipole models, the primary current I_k at time k can be represented in terms of N_d current dipoles as [39]

$$I_k(\mathbf{r}) = \sum_{j=1}^{N_d} \mathbf{m}_{k,j}\delta(\mathbf{r} - \mathbf{r}_{k,j}) \tag{2.1}$$

where $\mathbf{r}_{k,j} = [r_{k,j}^{(x)}\ r_{k,j}^{(y)}\ r_{k,j}^{(z)}]^T$ and $\mathbf{m}_{k,j} = [m_{k,j}^{(x)}\ m_{k,j}^{(y)}\ m_{k,j}^{(z)}]^T$ are location and moment vectors of the jth current dipole, respectively. $\delta(\mathbf{r} - \mathbf{r}_{k,j})$ is the Dirac delta function, The moment of the dipole is given by

$$\mathbf{m}_{k,j} = \mathbf{q}_{k,j} s_{k,j} \tag{2.2}$$

where $\mathbf{q}_{k,j} = [q_{k,j}^{(x)}, q_{k,j}^{(y)}, q_{k,j}^{(z)}]^T$ and $s_{k,j}$ are the orientation vector and amplitude of the dipole respectively. The MEG/EEG signals acquired by M sensors can be expressed as

$$\mathbf{z}_k = h(\mathbf{x}_k) + \boldsymbol{\eta}_k = \mathbf{A}_k s_k + \boldsymbol{\eta}_k \tag{2.3}$$

and $\mathbf{x}_{k,j} = [\mathbf{r}_{k,j}^T, \mathbf{q}_{k,j}^T, s_{k,j}]^T$ is the 7-D parameter vector corresponding to 3-D locations, 3-D orientations and amplitude. We assume the dipole sources are independent from each other and statistically independent of noise $\boldsymbol{\eta}_k$, also the noise components $\boldsymbol{\eta}_{k,m}$ $m = 1,2,...M$ are mutually independent. $\mathbf{A}_k$ is the lead-field matrix that depends on the location, orientation and amplitude of jth dipole. Specify each element of lead-field matrix $a_{k,m,j}$, $j = 1,2,..., N_d, m = 1,2,.., M$, it is given by [4]

$$\begin{aligned} a_{k,m,j} = {} & \frac{1}{4\pi\sigma}\cos(\alpha_{k,j})[\frac{2}{d_{k,m,j}^3}\left(|\mathbf{r}_{k,j}|\cos\left(\gamma_{k,m,j}\right) - r\right) + \left(d_{k,m,j}|\mathbf{r}_{k,j}|\right)^{-1} - \left(r_{k,m,j}|\mathbf{r}_{k,j}|\right)^{-1}] \\ & + \frac{1}{4\pi\sigma}\cos(\alpha_{k,j})\sin(\beta_{k,j})\sin(\gamma_{k,m,j})[\frac{2r}{d_{k,m,j}^3} + \frac{d_{k,m,j} + r}{rd_{k,m,j} - \left(r_{k,m,j}|\mathbf{r}_{k,j}| + d_{k,m,j}\right)}] \end{aligned} \tag{2.4}$$

here r is the radius of the head model, $d_{k,m,j}$ is the distance between the jth dipole source and the mth sensor and the vector pointing to the ith dipole location, $\alpha_{k,j}$ is the angle between the jth dipole orientation and the vector pointing to the jth dipole location, $\beta_{k,j}$ is the angle between the plane performed by the jth dipole and the origin, σ is the head tissue conductivity constant. In Equation (2.4),

$$\left|\mathbf{r}_{k,j}\right| = [\left(r_{k,j}^{(x)}\right)^2 + \left(r_{k,j}^{(y)}\right)^2 + \left(r_{k,j}^{(z)}\right)^2] \tag{2.5}$$

the following figure shows the equivalent current dipole model for EEG localization for the jth dipole source and mth EEG sensor.

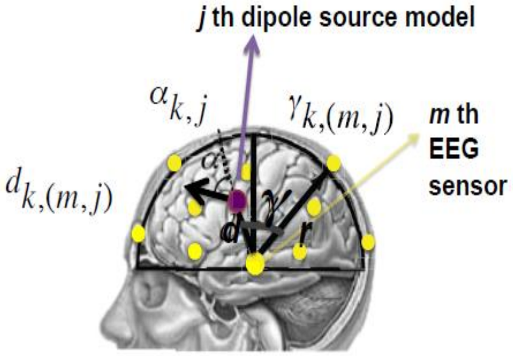

Figure 1 Equivalent Current Dipole Model for EEG Localization for the jth Dipole and mth EEG Sensor

2.2 State Space Representation of Neural Tracking System

The neural activity tracking problem is the estimation of the location and moment vector parameters of the current dipole model sources. These parameters vary dynamically, and they are related to the EEG/MEG measurements as described in Equation (2.3). This equation provides the measurement model of the state space system that uniquely characterizes neural dipole source tracking. In an overall state space dynamic system representation, as there is no physiological time source evolution model, the state model is assumed to follow a random walk model [11]. The dynamic state model for the jth EEG current dipole at time step k is given by:

$$\mathbf{x}_{k,j} = f(\mathbf{x}_{k-1,j}) + v_{k-1} = \mathbf{x}_{k-1,j} + \mathbf{v}_{k-1} \tag{2.6}$$

where $\mathbf{x}_{k,j} = [\mathbf{r}_{k,j}^T, \mathbf{q}_{k,j}^T, s_{k,j}]^T$ is a 7-D state vector, $\mathbf{r}_{k,j}, \mathbf{q}_{k,j}$ are the position vector and orientation vector of the jth EEG current dipole at time step k respectively. $s_{k,j}$ is amplitude factor and $\mathbf{v}_{k-1}$ is the error model random vector.

2.3 Artifacts in Neural Recordings

Although EEG is designed to record the cerebral activities, it also record unavoidable activities from artifacts. EEG artifacts are usually divided into two categories based on the cause. Physiological artifacts are from patients, includes muscle, tongue, eye, skin, heart movement and/or respiration during recording the brain activity. Extra-physiological artifacts can caused by environment or/and equipment, like electrodes popping, alternating current, surrounding movements and interference in the environment.

Among all the above artifacts, the most common artifact is eye blinking artifact since it is a spontaneous and natural behavior and is hard to be controlled in the recording procedure. In clinical experiment, eye blinking artifact appears as a sharp transients (see figure 4).

EEG waveforms are generally grouped into bands according to their frequency. Alpha waves lie in the range of 8-15 Hz, frequency of Beta waves is in 16-31 Hz, Theta waves' range is from 4 to 7 Hz and Delta waves is between 0 Hz and 4 Hz. Frequency is a key characteristic to classify the normal EEG and abnormal EEG [39]. Delta rhythmic activity is normally located frontally in adults and posteriorly in children, and it frequently appears in babies, or adults who are in deep sleep. Deep sleep is referred to as slow-wave sleep, since the EEG activity is synchronized, producing slow waves with frequency of less than 1 Hz and a relatively high amplitude. It consists of stages three and four of non-rapid eye movement sleep [45].

CHAPTER 3

NEURAL ACTIVITY ESTIMATION

The neural tracking problem is described in Chapter 2 as the problem of estimating the location and moment state parameters of the current dipole model sources. The state space representation of the neural tracking system is described in Equations (2.3) and (2.6). This chapter discusses Bayesian estimation methods for estimating the state of single and multiple dipole source models.

3.1 Bayesian Estimation

Recursive Bayesian estimation [40] is a general probabilistic approach for estimating the unknown probability density function over time of the hidden state of a dynamic system. The dynamic system model is, in general, described by two mathematical models. The measurement model relates the observed data to the hidden system states that need to be estimated, the state model describes how state parameters change over time a. The state space representation of a dynamic system is given by:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}) + \mathbf{v}_{k-1} \tag{3.1}$$

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{u}_k \tag{3.2}$$

where $\mathbf{x}_k$ is a vector of unknown state parameters at time step k, $f(\cdot)$ is a general function that describes the state evolution, and $\mathbf{v}_{k-1}$ is a modeling error vector that characterizes the random error process between the model used and the actual model. In (3.2), $\mathbf{z}_k$ is the vector of measurements at time k, $h(\cdot)$ is a function that defines the relationship between the measurement and the state, and $\mathbf{u}_k$ is the measurement noise vector. The tracking problem is to recursively estimate $\mathbf{x}_k$ from a set of measurements $\mathbf{z}_{1:k} = \{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_k\}$.

Recursive Bayesian filtering is based on the concept of feedback control. The system predicts the process state at some time and then observes feedback in the form of noisy measurements to update the state. The two main components of the algorithm are: prediction and update.

During prediction step, and assuming that the probability density function $p(\mathbf{x}_{k-1} | \mathbf{z}_{1:k-1})$ at time $k-1$ is obtained recursively from the assumed initial density $p(\mathbf{x}_0 | \mathbf{z}_0) = p(\mathbf{x}_0)$, then the system model is used with the Chapman-Kolmogorov equation [41] to obtain the prior probability density function $p(\mathbf{x}_k | \mathbf{z}_{1:k-1})$ at time k as:

$$p(\mathbf{x}_k | \mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k | \mathbf{x}_{k-1})\ p(\mathbf{x}_k | \mathbf{z}_{1:k-1})\ d\mathbf{x}_{k-1} \tag{3.3}$$

where $p(\mathbf{x}_k | \mathbf{x}_{k-1})$ is the state prior density function. During the update step, the new available measurement $\mathbf{z}_k$ is used to update the predicted state using Bayes' rule [41] to obtain the posterior density function:

$$p(\mathbf{x}_k | \mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k | \mathbf{x}_k)\ p(\mathbf{x}_k | \mathbf{z}_{1:k-1})}{\int_{-\infty}^{\infty} p(\mathbf{z}_k | \mathbf{x}_k)\ p(\mathbf{x}_k | \mathbf{z}_{1:k-1})\ d\mathbf{x}_k} \tag{3.4}$$

where $p(\mathbf{z}_k | \mathbf{x}_k)$ is the measurement likelihood function.

If the functions $f(\cdot)$ and $h(\cdot)$ in Equations (3.1) and (3.2) are linear, $\mathbf{v}_k$ and $\mathbf{u}_k$ are Gaussian, then the Kalman filter [42] can be used to estimate the state analytically. For systems described by nonlinear equations, modifications of Kalman filter such as the extended Kalman filter and the unscented Kalman filter [43] yield higher estimation performance than the Kalman filter. However, for highly nonlinear systems, such as the EEG/MEG neural measurement models, traditional Bayesian methods perform poorly and require the use of sequential Monte Carlo methods such as particle filtering [12, 44].

3.2 Particle Filtering Estimation

Particle filtering estimates the posterior density of the unknown state of a dynamic system implementing the recursive Bayes filter [44]. The basic idea behind particle filtering is to estimate the posterior density of the unknown state parameters using a set of particles and associated weights. The technique has been successfully used in many applications including visual tracking, quality control in the semiconductor industry, positioning systems, radar tracking and missile guidance.

The Particle filter uses a set of particles to represent samples from a posterior distribution, and each particle has a weight associated with it, the joint posterior probability density function of the state at time step k, and thus an estimate of the state can be approximated by this set of N random weighted samples.

The probability density function at time k can be described as

$$p(\mathbf{x}_k \mid \mathbf{z}_k) = \sum_{n=1}^{N} w_k^{(n)} \delta(\mathbf{x}_k - \mathbf{x}_k^{(n)}) \tag{3.5}$$

where $w_k^{(n)}$ is the weight of the nth particle at time k and $\delta(\cdot)$ is the Dirac delta function. The state estimate is obtained directly from (3.5) as

$$\hat{\mathbf{x}}_k = \int \mathbf{x}_k \, p(\mathbf{x}_k \mid \mathbf{z}_k) d\mathbf{x}_k = \sum_{n=1}^{N} w_k^{(n)} \mathbf{x}_k^{(n)} \tag{3.6}$$

The sequential importance resampling particle filter (SIR-PF) is a very commonly used particle filtering algorithm. An importance density function $q(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{z}_k)$ is chosen to minimize the variance of the weights, and to sample the posterior probabilistic density function. The SIR-PF has three main steps:

1) Particle generation

The particles $\mathbf{x}_k^{(n)}$ are drawn from an importance density function $q(\mathbf{x}_k \mid \mathbf{x}_{k-1}^{(n)}, \mathbf{z}_{1:k})$, where $\mathbf{z}_{1:k} = \{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_k\}$.

2) Weights calculation

The importance density function is chosen as the prior density function $q(\mathbf{x}_k \mid \mathbf{x}_{k-1}^{(n)}, \mathbf{z}_{1:k}) = p(\mathbf{x}_k \mid \mathbf{x}_{k-1}^{(n)})$, and the corresponding weights are obtained as

$$w_k^{(n)} \propto w_{k-1}^{(n)} \frac{p(\mathbf{z}_k \mid \mathbf{x}_k^{(n)}) p(\mathbf{x}_k^{(n)} \mid \mathbf{x}_{k-1}^{(n)})}{q(\mathbf{x}_k^{(n)} \mid \mathbf{x}_{k-1}^{(n)}, \mathbf{z}_{1:k})} = w_{k-1}^{(n)} \; p(\mathbf{z}_k \mid \mathbf{x}_k^{(n)}) \qquad (3.7)$$

the weights are normalized and summed to one.

$$\sum_{n=1}^{N} w_k^{(n)} = 1 \qquad (3.8)$$

3) Resampling

The particles are resampled to avoid particle degeneracy, which occurs when most weights are very small and thus provide a poor representation of the posterior probability density function [44]. This step eliminates particles with low weights and replicates particles with high weights, to make sure sufficient particles are used at each time step and present probability density function well.

The main steps of the particle filter algorithm for state estimation are described in Algorithm 1.

Algorithm 1 SIR Particle Filter algorithm

```
• Initialization at time k=0
Draw $N$ particles from a uniform distribution
Compute the initial weights $w_0^{(n)} = 1/N$
• Prediction step
For $k = 1:K$
    For $n = 1:N$
Predict for each particle $\mathbf{x}_k^{(n)} = \mathbf{x}_{k-1}^{(n)} + \mathbf{u}_{k-1}^{(n)}$ and keep weights $w_{k\text{-}1}^{(n)}$
• Update step
Update corresponding weights $w_k^{(n)} = w_{k-1}^{(n)} p(\mathbf{z}_k \mid \mathbf{x}_k^{(n)})$
Normalize weights $\sum_{i=1}^{N} w_k^{(n)} = 1$
•Resampling step [44]
```

3.3 Probability Hypothesis Density Filtering for Multiple Object Tracking

In many real applications, the number of objects whose dynamic parameters need to be estimated also changes with time. In neural activity tracking, as the current dipoles model neural sources, the number of dipoles at time step k depend on the number of new dipoles tracked at time step k, the number of dipoles that are no longer present at time step k but were present at time step $k-1$, and the number of dipoles that were present at time step $k-1$ and remain present at time step k. The neural recordings may also be contaminated by a varying number of artifact signals. Their presence at time step k can be considered as clutter, resulting in reducing the probability of dipole detection as well as increasing the probability of false mistaking artifact signals for neural activity signals. For such realistic scenarios with an unknown varying number of dipoles, it is important to be able to associate each sensor measurement component to its corresponding dipole or determine that it originated from artifacts.

As it is known that EEG measurements can contain both neural activity and artifacts components, estimating brain activity requires knowledge of which measurement most likely originated from neural activity. Although data association provides a solution, it is computationally intensive and requires that the number of dipoles is fixed at each time step, it also result in poor estimation performance when the dipoles are close to each other. In order to avoid the measurement association problem, the probability hypothesis density filter (PHDF) is a filter that can recursively estimate the number and states of objects in environments of nonzero probability of detection [45-46]. The PHDF has been applied in many practical problems, such as multi-beam forward-looking sonar images

[47], simultaneous localization and mapping (SLAM) [48], and multi-target visual tracking [49].

3.3.1 Probability Hypothesis Density Filtering Formulation

The PHDF is based on a recursion propagating the first moment of the posterior density function of the multiple object state and uses random finite sets (RFS). For the neural tracking application, for example, due to the complexity of brain activity, current dipole sources may randomly appear or disappear, which causes the number of dipoles to vary a each time step. The number of measurements can also change, especially since some of the recordings can correspond to artifacts. It is also not known a *prior* which dipole source has generated a given measurement, assuming that dipoles generate activity independently. RFS are thus appropriate to use for representing the dipole source model states and the measurements. An RFS is defined as a finite set-valued random vector whose elements can be characterized by a discrete probability distribution and a family of joint probability density functions [50]. When used with neural tracking, the cardinality of the multi-object state RFS is the random number of dipoles sources, and the unknown state of each object is represented by each RFS element.

Using the RFS formulation and assuming that at time step k, there are N_k dipole source models and M_k measurements, then the multiple dipole sources state RFS and multiple sensor measurement RFS are given respectively by:

$$\mathbf{X}_{k,j} = \{\mathbf{x}_{k,j}^{1}, \mathbf{x}_{k,j}^{2}, ..., \mathbf{x}_{k,j}^{N_k}\} \tag{3.9}$$

$$\mathbf{Z}_{k,m} = \{\mathbf{z}_{k,m}^{1}, \mathbf{z}_{k,m}^{2}, ..., \mathbf{z}_{k,m}^{M_k}\} \tag{3.10}$$

where $\mathbf{x}_{k,j} = [\mathbf{r}_{k,j}^T, \mathbf{q}_{k,j}^T, s_{k,j}]^T$ is the state vector of jth dipole source in Equation (2.6), and $\mathbf{z}_{k,m}$ is the measurement for the mth sensor in Equation (2.3). Note that the measurements can also include artifact signals.

In order to formulate the PHDF for multi-object estimation, assume that $\mathbf{x}_k$ is a vector element of the RFS $\mathbf{X}_k$ in (3.9) at time k and $\mathbf{z}_k$ is a vector element of the RFS $\mathbf{Z}_k$ in (3.10), multi-target states and sensor measurements can be represented as random finite sets. Given the source state vector $\mathbf{x}_{k-1}$ at time $k-1$, the state random finite set $\mathbf{x}_k$ is formed by combining the state vectors from sources still present from the previous time step, from sources that spawned from sources from previous time steps $\mathbf{x}_{k|k-1}^{spn}$ and from new sources at the present time step $\mathbf{x}_k^{new}$. Due to the possible presence of artifacts, the measurements can be due to both the source and clutter or due to only clutter. It is assumed that the clutter RFS and the source measurement RFS are independent, and that the source measurements RFS are mutually independent.

In the PHDF algorithm, the predicted posterior density function $p(\mathbf{x}_k \mid \mathbf{z}_{k-1})$ can be completely characterized by the corresponding intensity function $\lambda(\mathbf{x}_k \mid \mathbf{z}_{k-1})$. This is because integrating the intensity function over a specified region provides the expected number of sources present in that region. Also, the locations of the peaks of the intensity function provide estimates of the parameters of the sources in the region. Thus, given the posterior intensity $\lambda(\mathbf{x}_{k-1} \mid \mathbf{z}_{k-1})$ at time step $k-1$, the predicted intensity is obtained as

$$\begin{aligned}\lambda(\mathbf{x}_k \mid \mathbf{z}_{k-1}) = \int [\mathrm{P}_{\mathrm{k|k\text{-}1}}(\mathbf{x}_{k-1})\ p(\mathbf{x}_k \mid \mathbf{x}_{k-1}) + \lambda^{spn}(\mathbf{x}_k \mid \mathbf{z}_{k-1})]\, \lambda(\mathbf{x}_{k-1} \mid \mathbf{z}_{k-1})\, d\mathbf{x}_{k-1} \\ + \lambda^{new}(\mathbf{x}_k \mid \mathbf{z}_k)\end{aligned} \qquad (3.11)$$

where $\lambda^{sp}(\mathbf{x}_k \mid \mathbf{z}_{k-1})$ is the intensity function of the sources spawned from the existed sources at the previous time step $k-1$, $\lambda^{new}(\mathbf{x}_k \mid \mathbf{z}_k)$ is the intensity functions of the new sources at the current time step k, and $\mathrm{P}_{\mathrm{k|k\text{-}1}}(\mathbf{x}_{k-1})$ is the probability that a source that was present at time step $k-1$ is still present at time step k.

The posterior intensity function is updated given the information from the prediction step. And it given by

$$\begin{aligned}\lambda(\mathbf{x}_k \mid \mathbf{z}_k) = (1-\mathrm{P}_k^D(\mathbf{x}_k))\lambda(\mathbf{x}_k \mid \mathbf{z}_{k-1}) + \\ \sum_{Z_k} \frac{\mathrm{P}_k^D(\mathbf{x}_k)\, p(\mathbf{z}_k \mid \mathbf{x}_k)\lambda(\mathbf{x}_k \mid \mathbf{z}_{k-1})}{\lambda^{clutter}(\mathbf{z}_k) + \int \mathrm{P}_k^D(\tilde{\mathbf{x}}_k)\, p(\mathbf{z}_k \mid \tilde{\mathbf{x}}_k)\lambda(\tilde{\mathbf{x}}_k \mid \mathbf{z}_{k-1})d\tilde{\mathbf{x}}_k}\end{aligned} \tag{3.12}$$

where $\lambda^{clutter}(\mathbf{z}_k)$ is the clutter intensity function in the measurement space and $\mathrm{P}_k^D(\mathbf{x}_k)$ is the probability of detecting a source at time k. Here we assume that the likelihood function $p(\mathbf{z}_k \mid \mathbf{x}_k)$ is Gaussian likelihood with mean $h(\mathbf{x}_k)$ and covariance Σ_k, M is sensor number.

$$p(\mathbf{z}_k \mid \mathbf{x}_k) = \frac{1}{\left(\sqrt{2\pi}\right)^{M/2} \left|\Sigma\right|^{1/2}} \exp\{-\frac{1}{2}[\mathbf{z}_k - h(\mathbf{x}_k)]^T \Sigma^{-1}[\mathbf{z}_k - h(\mathbf{x}_k)]\} \tag{3.13}$$

3.3.2 Probability Hypothesis Density Filter Implementation using Particle Filtering

In our work, the PHDF is implemented using particle filtering (PF) [51] which is applicable to nonlinear system models as in the neural tracking problem.

Similar to particle filtering, the particle probability hypothesis density filter (PPHDF) has the three main steps of prediction, update and resampling. It is a recursive algorithm whose prior intensity function is estimated and then the posterior intensity function is updated based on the prior intensity function. The recursion requires an initialization of the intensity function at time step $k = 0$.

At time step ($k-1$), the intensity function $\lambda(\mathbf{x}_{k-1} \mid \mathbf{z}_{k-1})$ can be approximated by T_{k-1} particles $\mathbf{x}_{k-1}^{(n)}$ and their corresponding weights $w_{k-1}^{(n)}, n = 1,2,..,T_{k-1}$:

$$\lambda(\mathbf{x}_{k-1} \mid \mathbf{z}_{k-1}) \approx \sum_{n=1}^{T_{k-1}} w_{k-1}^{(n)}\ \delta(\mathbf{x}_{k-1} - \mathbf{x}_{k-1}^{(n)}) + \lambda^{new}(\mathbf{x}_k \mid \mathbf{z}_k) \tag{3.14}$$

We assume that $\{\mathbf{x}_k^{(n)}\}_{n=1}^{T_{k-1}}$ are drawn from the proposal density $q_k(\cdot \mid \mathbf{x}_{k-1}^{(n)}, \mathbf{z}_k)$ and $\{\mathbf{x}_k^{(n)}\}_{i=T_{k-1}+1}^{T_{k-1}+J_k}$ are drawn from another density $p_k(\cdot \mid \mathbf{z}_k)$,

$$\mathbf{x}_k^{(n)} \sim \begin{cases} q_k(\cdot \mid \mathbf{x}_{k-1}^{(n)}, \mathbf{z}_k), n = 1,..,T_{k-1} \\ p_k(\cdot \mid \mathbf{z}_k), i = T_{k-1}+1,...,T_{k-1}+J_k \end{cases} \tag{3.15}$$

T_{k-1} particles are used to approximate the surviving targets from time step $k-1$ to time step k, J_k particles are for new born targets RFS at time step k. Then the predicted intensity function is

$$\lambda(\mathbf{x}_k \mid \mathbf{z}_{k-1}) \approx \sum_{n=1}^{T_{k-1}+J_k} w_{k-1}^{(n)}\ \delta(\mathbf{x}_k - \mathbf{x}_k^{(n)}) \tag{3.16}$$

where

$$w_{k|k-1}^{(n)} = \begin{cases} \dfrac{\phi_{k|k-1}(\mathbf{x}_k^{(n)}, \mathbf{x}_{k-1}^{(n)}) w_{k-1}^{(n)}}{q_k(\mathbf{x}_k^{(i)} \mid \mathbf{x}_{k-1}^{(i)}, \mathbf{z}_k)}, n = 1,..,T_{k-1} \\ \dfrac{\lambda^{birth}(\mathbf{x}_k^{(n)} \mid \mathbf{z}_k)}{J_k \cdot p_k(\mathbf{x}_k^{(n)} \mid \mathbf{z}_k)}, n = T_{k-1}+1,...,T_{k-1}+J_k \end{cases} \tag{3.17}$$

In the update step, the particle weights are changed to:

$$w_k^{(n)} = (1 - \mathrm{P}_k^D(\mathbf{x}_k)) w_{k|k-1}^{(n)} + \sum_{\mathbf{z}_k \in \mathbf{z}_{1:k}} \frac{\mathrm{P}_k^D(\mathbf{x}_k^{(n)}) p(\mathbf{z}_k \mid \mathbf{x}_k^{(n)}) w_{k|k-1}^{(n)}}{\lambda^{clutter}(\mathbf{z}_k) + C_k(\mathbf{z}_k)} \tag{3.18}$$

where

$$C_k(\mathbf{z}_k) = \sum_{n=1}^{T_{k-1}+J_k} \mathrm{P}_k^D(\mathbf{x}_k^{(n)})\, p(\mathbf{z}_k \mid \mathbf{x}_k^{(n)}) w_{k|k-1}^{(n)} \tag{3.19}$$

$\lambda^{clutter}(\mathbf{z}_k)$ is the clutter intensity function in the measurement space and $\mathrm{P}_k^D(\mathbf{x}_k)$ is the probability of a source to be detected at time k, and. $p(\mathbf{z}_k \mid \mathbf{x}_k)$ is Gaussian likelihood in Equation in (3.15).

Similarly, resampling is used to avoid the particle degeneracy by eliminating low weights particles and replicating high weights particles, and make particles focus on important regions of the intensity function.

The particle filter implementation of the PHDF is robust and computationally possible when compare to existing multiple objects tracking techniques, and it has been applied successfully in radar tracking and sonar image [47].

Algorithm 2 Probability Hypothesis Density Particle Filtering

```
• Initialization  $\lambda(\mathbf{x}_0)$,  $T_0$,  $w_0 = 1/T_0$,
• Step 1 prediction step
For  $n = 1,2,...T_{k-1}$, sample and compute weights
$\mathbf{x}_k^{(n)} \sim q_k(\cdot\,|\,\mathbf{x}_{k-1}^{(n)},\mathbf{z}_k)$,  $q_k(\cdot)$  is the Gaussian in (3.15)
$w_{k|k-1}^{(n)} = \frac{\phi_{k|k-1}(\mathbf{x}_k^{(n)},\mathbf{x}_{k-1}^{(n)})w_{k-1}^{(n)}}{q_k(\mathbf{x}_k^{(n)}\,|\,\mathbf{x}_{k-1}^{(n)},\mathbf{z}_k)}$
For  $n = T_{k-1}+1,...,T_{k-1}+J_k$, sample and compute weights
$\mathbf{x}_k^{(n)} \sim p_k(\cdot\,|\,\mathbf{z}_k)$,
$w_{k|k-1}^{(n)} = \frac{\lambda^{new}(\mathbf{x}_k^{(n)}\,|\,\mathbf{z}_k)}{J_k \cdot p_k(\mathbf{x}_k^{(n)}\,|\,\mathbf{z}_k)}$
• Step 2 update step
For  $\mathbf{z}_k \in \mathbf{z}_{1:k}$, compute
$C_k(\mathbf{z}_k) = \sum_{n=1}^{T_{k-1}+J_k} \mathrm{P}_k^D(\mathbf{x}_k^{(n)})\,p(\mathbf{z}_k\,|\,\mathbf{x}_k^{(n)})w_{k|k-1}^{(n)}$
For  $n = 1,2,...T_{k-1}+J_k$, update weights
$w_k^{(n)} = (1-\mathrm{P}_k^D(\mathbf{x}_k))w_{k|k-1}^{(n)} + \sum_{\mathbf{z}_k\in\mathbf{z}_k} \frac{\mathrm{P}_k^D(\mathbf{x}_k^{(n)})\,p(\mathbf{z}_k\,|\,\mathbf{x}_k^{(n)})w_{k|k-1}^{(n)}}{\lambda^{clutter}(\mathbf{z}_k)+C_k(\mathbf{z}_k)}$
• Step 3 resampling step [44]
Compute the total mass  $R_{k|k} = \sum_{n=1}^{T_{k-1}+J_k} w_k^{(n)}$
Resample  $\{w_k^{(n)}/\hat{R}_{k|k},\mathbf{x}_k^{(n)}\}_{n=1}^{T_{k-1}+J_k}$  to get  $\{w_k^{(n)}/\hat{R}_{k|k},\mathbf{x}_k^{(n)}\}_{n=1}^{T_k}$
```

CHAPTER 4

ARTIFACT FEATURE CHARACTERIZATION WITH MULTIPLE NEURAL SOURCE TRACKING

4.1 Proposed Algorithm of Neural Sources Tracking with Stochastic Artifact Modeling

In our work, we propose an integrated algorithm for simultaneously tracking multiple neural sources using the probability hypothesis density particle filter (PPHDF), discussed in Section 3.4, and suppressing artifact signals by estimating their statistical representations using time-frequency methods and Gaussian mixture modeling. For the rest of the thesis, we refer to our proposed algorithm as NEST-SAM (or Neural Sources Tracking with Stochastic Artifact Modeling).

The main steps of the NEST-SAM algorithm are depicted in Figure 4.1. EEG recordings from multiple sensors or channels are first preprocessed by performing highpass filtering to remove linear trends in the data and by extracting specific events of interest [52]. Independent component analysis (ICA) is then used to separate the channel recordings into independent components, with each component corresponding to a single measurement. The different signal components correspond to either dipole source model or artifacts signals. We extract unique features for each type of signal using the matching pursuit decomposition time-frequency (MPD-TF) algorithm. The resulting feature vectors are used as input to the Gaussian mixture model (GMM) algorithm that results in two clusters with unique probability density functions for each type of signal. These density functions are then used in the PPHDF algorithm that estimates the number of dipole source models and their parameters at each time step; the PPHDF incorporated the estimated artifact information as clutter to improve the performance accuracy of the overall tracking algorithm.

The different steps of the NEST-SAM algorithm are discussed in detail in the rest of the chapter.

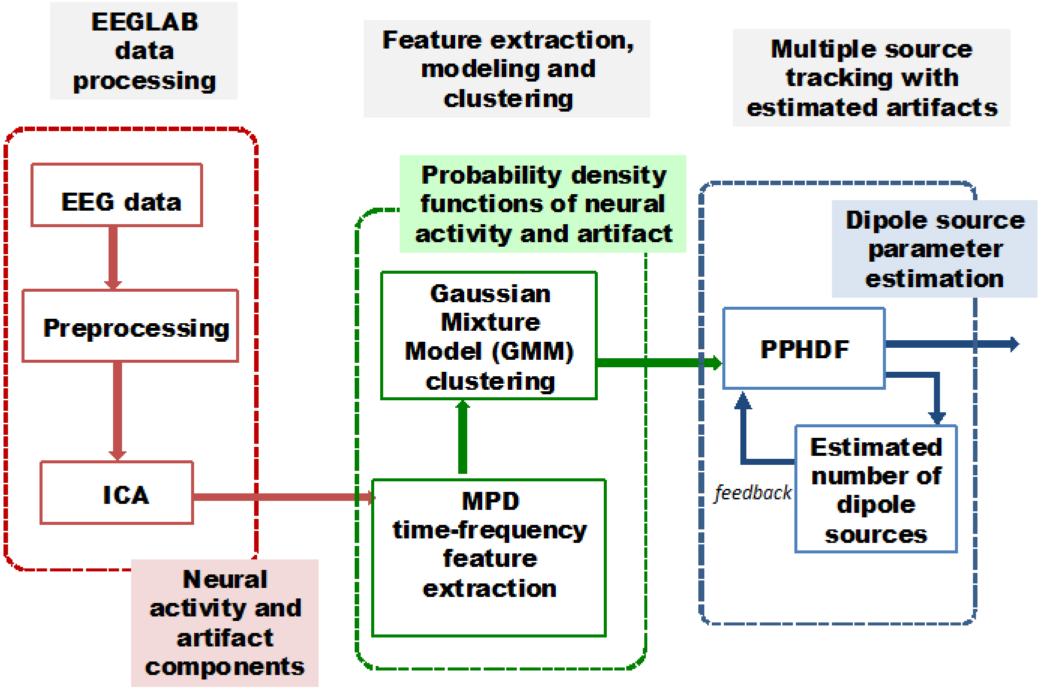
EEGLAB data processing
Feature extraction, modeling and clustering
Multiple source tracking with estimated artifacts
EEG data
Preprocessing
ICA
Neural activity and artifact components
Probability density functions of neural activity and artifact
Gaussian Mixture Model (GMM) clustering
MPD time-frequency feature extraction
Dipole source parameter estimation
PPHDF
Estimated number of dipole sources
feedback

Figure 2 Block Diagram of the Proposed Neural Sources Tracking with Stochastic Artifact Modeling (NEST-SAM) algorithm.

4.2 Independent Component Analysis

The problem in neural activity estimation is that EEG data consists of electrical potentials from brain activities that can be contaminated by artifacts. All current dipole sources contribute to EEG measurements. In order to associate each component to one current dipole, the EEG measurements must be decomposed into independent components using ICA.

As a computational and statistical signal processing technique, the general idea of ICA is to search for a linear transformation of sets of random variables, measurements or signals that minimize the statistical dependence between their components. A well-known linear transformation method is principal component analysis (PCA) [55], which can only impose independence up to second order and defines directions that are orthogonal. ICA is the extension of PCA, and its goal is to decompose a multivariate signal into independent non-Gaussian signal components are statistically independent. Theoretically using ICA assumes that the EEG sources do not have a Gaussian distribution, since in the case of Gaussian statistics, the ICA model can only be estimated up to an orthogonal transformation..

If we assume that the t random measurement vector $\mathbf{m} = [m_1\, m_2\, ...\, m_t]^T$ is a linear mixture of p random source signal vector $\mathbf{s} = [s_1\, s_2\, ...\, s_p]^T$, the ICA can be expressed as

$$\mathbf{z} = \mathbf{As} \tag{4.1}$$

where $\mathbf{A}$ is the $t \times p$ mixing matrix of full rank. The source signals are assumed independent with non-Gaussian distributions (or just one component is Gaussian) and cannot be observed directly. ICA is used to estimate both the mixing matrix and $\mathbf{s}$ using the measurement $\mathbf{z}$. If we assume that $\mathbf{z}$ and $\mathbf{s}$ have the same dimensionality, then we can rewrite Equation (4.1) as :

$$\hat{s} = \mathbf{A}^{-1}\mathbf{m} = \mathbf{Wm} \tag{4.2}$$

where $\mathbf{W}$ is called the unmixed matrix or weight matrix. ICA can now estimate $\hat{\mathbf{s}}$ as an approximation to the source signal $\mathbf{s}$ by estimating the weight matrix which goes from source signal space to observed space. The independent components are optimal when $\mathbf{W}$ is an accurate estimate.

4.3 Time-frequency Feature Extraction using the Matching Pursuit Decomposition

Analysis of EEG mostly relies on visual inspection which is rather very subjective and hardly allows any statistical analysis or standardization. One of the methods to quantify the information of EEG is Fourier analysis. The Fourier transform provides the spectral part of the signal, but no information on the time localization of the frequency components. In order to analyze the time-varying EEG signals, the time and frequency domain characteristics must be considered jointly. Time-frequency analysis [35] is an effective solution for EEG signals.

The matching pursuit decomposition (MPD) is a popular technique for sparse signal representation [21]. The general idea behind the MPD algorithm is the decomposition of any signal into a linear expansion of waveforms that are selected from a redundant dictionary of functions, to find the best matches of the signal time-frequency structures. Decomposition of signals over both time and frequency domain is widely used in signal processing and harmonic analysis. For these dictionaries of time-frequency atoms, the MPD is an adaptive transform that decomposes any function into a sum of complex time-frequency atoms that can best match its residues. As Gaussian signals are the most localized signals in both time and frequency, a Gaussian dictionary is the optimal choice.

The MPD is a time-frequency based technique that decomposes a signal into highly localized time-frequency atoms and can provide a highly concentrated time-frequency representation. This is done by performing an exhaustive search over all the Gaussian time-frequency atoms in the dictionary [33]. Suppose we have a real signal $y(t)$, then the steps of the MPD algorithm are as follows.

With $r_0(t) = y(t)$, at the ith iteration, $i = 0, 1, \dots, N-1$, the projection of the residue $r_i(t)$ onto every dictionary element $g^{(d)}(t) \in D$ is computed to obtain

$$\kappa_i^{(d)} = \left\langle r_i, g^{(d)} \right\rangle = \int_{-\infty}^{\infty} r_i(t)\, g^{(d)*}(t)\, dt \tag{4.3}$$

The selected dictionary atom $g_i(t)$ is the one that maximizes the magnitude of the projection,

$$g_i(t) = \arg\max_{g^{(d)}(t) \in D} \left| \kappa_i^{(d)} \right| \tag{4.4}$$

The corresponding expansion coefficient is

$$a_i = < r_i, g_i > = \int_{-\infty}^{\infty} r_i(t)\, g_i(t)\, dt \tag{4.5}$$

The residue at the ith and $(i+1)$th iterations are related as $r_{i+1}(t) = r_i(t) - a_i g_i(t)$. Thus after N MPD iterations, the residue is given by

$$r_N(t) = r_{N-1}(t) - a_{N-1} g_{N-1}(t) = s(t) - \sum_{i=0}^{N-1} a_i\; g_i(t) \tag{4.6}$$

$g(t)$ is generated by scaling, translating and modulating a simple Gaussian window function centered at origin. With $\eta \neq 0$, and any initial phase $\varphi \in [0, 2\pi]$, $g(t)$ is defined as

$$g(t) = \frac{K}{\sqrt{s}} \varphi\left(\frac{t - \tau}{s} \right) \cos(\eta t + \varphi) \tag{4.7}$$

here the coefficient constant K is the adjust factor to keep $\|g\ (t)\| = 1$, s is scale, η is frequency modulating and τ is time delay.

Although the matching pursuit decomposition is nonlinear, the energy conversation is maintained to guarantee its convergence. In our simulation, we use conventional Gabor (Gaussian) dictionary since it is the most concentrated signal in the time-frequency plane. The Gaussian atom belongs to the dictionary is given by

$$g(t) = \frac{1}{\sqrt{s}} \varphi(\frac{t-\tau}{s}) \exp(j\eta t) \tag{4.8}$$

where

$$\varphi(t) = 2^{\frac{1}{4}} e^{\pi t^2} \tag{4.9}$$

and Wigner distribution of a Gaussian atom is

$$Wy(t,w) = 2\exp(\frac{-2\pi(t-\tau)^2}{s^2})\exp(-\frac{w^2}{2\pi}) \tag{4.10}$$

MPD iterations will stop when the signal energy residue reaches certain threshold, and based on the dictionary, we obtain the 4-D feature $\zeta = (K\ s\ \eta\ \tau)$ from input EEG signals, here K is coefficient, s is scale, η is frequency modulation and τ is time delay.

Algorithm 3 MPD based on Gaussian Atom Time-frequency dictionary

```
• Initialize $K$, dictionary size $N_a$, $N_f$, residue $R$=input EEG signal, energy $E_n=|R|^2$
• Create Gaussian atom time-frequency dictionary
      For i = 1 : $N_a$
         For j = 1 : $N_f$
            $dictionary = \exp(-(t(i)^2 s(i)^2)) \cdot \cos(2\pi f(j)t(i))$
            normalize the dictionary
          End for
      End for
• MPD Iteration
  For i = 1: $K$
     Compute inner product for each atom, $P = < R, dictionary >$
     Find the dictionary atoms for maximum P
     Update residue R and energy $E=R^2/E_n$
     i=i+1;
     if E <= $E_n$End if
  End for
      Input: EEG signals
      Output: characteristic factors
      Feature vector = [coefficient $K$, scale $s$, time-shift $\tau$, frequency-shift $\eta$]
```

4.4 Gaussian Mixture Model (GMM)

We use MPD feature vectors to uniquely represent and differentiate between neural activity and artifacts. As this information needed to be integrated with the PPHDF, we use Gaussian Mixture Modeling (GMM) [36-38] to obtain stochastic representations of the MPD features. GMM is a simple and effective method to represent the probability density functions of the characteristic feature vectors. This method is commonly used as a parametric model of the probability distribution of continuous measurements or features of biometric system. The basic idea of using Gaussian mixtures is to model the unknown PDFs as a linear combination of several weighted Gaussian component densities. Assume we have N single Gaussians distributions with each own mean μ_k and variance σ_{k} , $k = 1, 2, ,...N$. and a given random signal could be approximated by these N Gaussians as follows:

$$p(x) = \sum_{k=1}^{N} \pi_k \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left(\frac{(x-\mu_k)^2}{2\sigma_k^2} \right) \tag{4.14}$$

here π_k is the kth weight for the kth Gaussian component and all the weights sum up to one. In our work, the input of the GMM is the characteristic vector, which contains coefficient K, scale a, time-shift τ and frequency-shift ω, thus the Gaussian Mixture model for the these deterministic features is

$$p(\mathbf{x} \mid \boldsymbol{\mu}_k, \Sigma_k) = \sum_{k=1}^{N} \pi_k \frac{1}{\sqrt{2\pi|\Sigma_k|}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)\Sigma_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k) \right) \tag{4.15}$$

where $\mathbf{x} = [K\ a\ \tau\ \eta]$, π_k is the kth weight for kth Gaussian component, $\boldsymbol{\mu}_k$ is the mean vector and Σ_k is the covariance matrix of input features. Each component is a multivariate Gaussian.

Given training vectors and a GMM configuration, we wish to estimate the parameters of the GMM, which in some sense best matches the distribution of the training feature vectors [36]. By far there are several techniques available to estimate the parameters, but the most popular and commonly used is the maximum-likelihood (ML) estimation, and ML parameter estimates can be obtained iteratively using a special case of the expectation-maximization (EM) algorithm [37]. The EM algorithm is an iterative method to expect better parameters based on the old estimates and to maximize the result. The iteration stops once some certain converge threshold is achieved. Each iteration has expectation (E)-step and maximization (M)-step.

Assume each cluster has responsibilities [38] for each data point, responsibilities assign data point to the corresponding clusters. In the E-step, we estimate distributions of the hidden variables given the data point :

$$\gamma_{\mathrm{k}}^{(n)} = \frac{\pi_k g(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \Sigma_k)}{\sum_{j=1}^{N} \pi_j g(\mathbf{x}_n \mid \boldsymbol{\mu}_j, \Sigma_j)} \tag{4.16}$$

where $\gamma_k^{(n)}$ stands for responsibility for nth data point corresponds to kth cluster, π_k is the weights for kth cluster, $g(\cdot)$ is Gaussian distribution for kth cluster. The estimated parameters are grouped by calculating the probabilities from the equation (4.13).

In the M-step, each cluster's parameters is computed to match the responsible data points, and accordingly, the weighted means and variances for kth cluster are updated, and they will be used as input for the next iteration.

$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^{T} \gamma_k^n \mathbf{x}_n}{N_k} \tag{4.17}$$

$$\Sigma_k = \frac{\sum_{n=1}^{T} \gamma_k^n (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T}{N_k} \tag{4.18}$$

where N_k is the total responsibility of kth cluster, $N_k = \sum_{n=1}^{T} \gamma_k^n$ and weights for each cluster is updated as :

$$\pi_k = \frac{N_k}{\sum_{k=1}^{K} N_k} \tag{4.19}$$

With the mean and covariance matrix, we can model the probability density functions for the two clusters representing neural activity and artifacts and then apply them in the PPHDF algorithm. The steps of the EM algorithm for GMM are provided in Algorithm 4.

4.5 Probability Distribution of Artifacts

Due to the received measurements are either from dipole targets or artifacts, the way to model artifacts is another significant factor in multiple targets tracking. In target tracking applications, the presence of clutter or false alarm is often modeled as Poisson distribution. Similarly, we model the presence of artifact measurements using Poisson distribution with average rate ρ, which provides a measure of the probability of the number of artifacts present at a certain time step, and we assume the number of clutter per scan is independent of the time. Then the discrete probability distribution for t artifact measurements is given as:

$$\psi(t) = \frac{\rho^t}{(t)!} e^{-\rho} \tag{4.20}$$

Algorithm 4 EM algorithm for Gaussian Mixture Model

- Initialization of parameters π_k^1, μ_k^1, Σ_k^1,
- E step: compute the responsibilities

$$\gamma_{\mathrm{k}}^{(n)} = \frac{\pi_k g(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \Sigma_k)}{\sum_{j=1}^{N} \pi_j g(\mathbf{x}_n \mid \boldsymbol{\mu}_j, \Sigma_j)}, k = 1,2,\ldots,K$$

- M step: compute each cluster's parameters to match the responsible data points.

Add up total responsibility of cluster k

$$N_k = \sum_{n=1}^{T} \gamma_k^n$$

Compute the weights for each cluster

$$\pi_k = \frac{\boldsymbol{N}_k}{\sum_{k=1}^{K} \boldsymbol{N}_k}$$

Weighted mean

$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^{T} \gamma_k^n \mathbf{x}_n}{\boldsymbol{N}_k}$$

Covariance matrix

$$\Sigma_k = \frac{\sum_{n=1}^{T} \gamma_k^n (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T}{N_k}$$

•Repeat EM steps until convergence.

CHAPTER 5

DIPOLE TRACKING USING PHDF IN ARTIFACT ENVIRONMENT

5.1 EEGLAB Software and EEG Recordings

EEGLAB is an open source toolbox for MATLAB environment that is used for processing collections of single-trial and/or averaged EEG data of any number of channels [52-53]. It stores data, acquisition parameters, channel locations, epochs and events in a single structure which can be accessed directly from the MATLAB command line. There are various types of functions available in this toolbox such as multi-trial data visualization, data processing, independent component analysis (ICA) and time-frequency decompositions.

We download the data from EEGLAB and extract a desired number of time steps or epochs for processing data into certain number of epochs, and run ICA on these source signals. First, we preprocess the real EEG data and use independent component analysis (ICA) to separate the channel recordings into independent components. These steps are computed directly within the EEGLAB software package [52, 54].For our simulation, EEG data was taken from 32 sensors over the human scalp, where the sensors are distributed as shown in Figure 3. We selected dataset segments centered on the presentation of a square stimulus, from one second prior to presentation to two seconds after presentation. Figure 4 shows the channel raw signals collected directly from sensors.

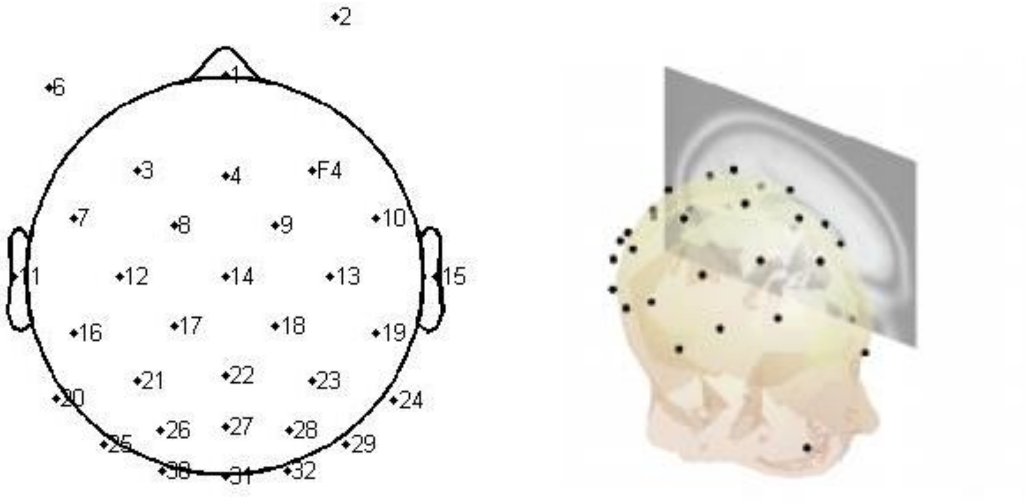

Figure 3 EEG Sensor Locations, Created Using EEGLAB.

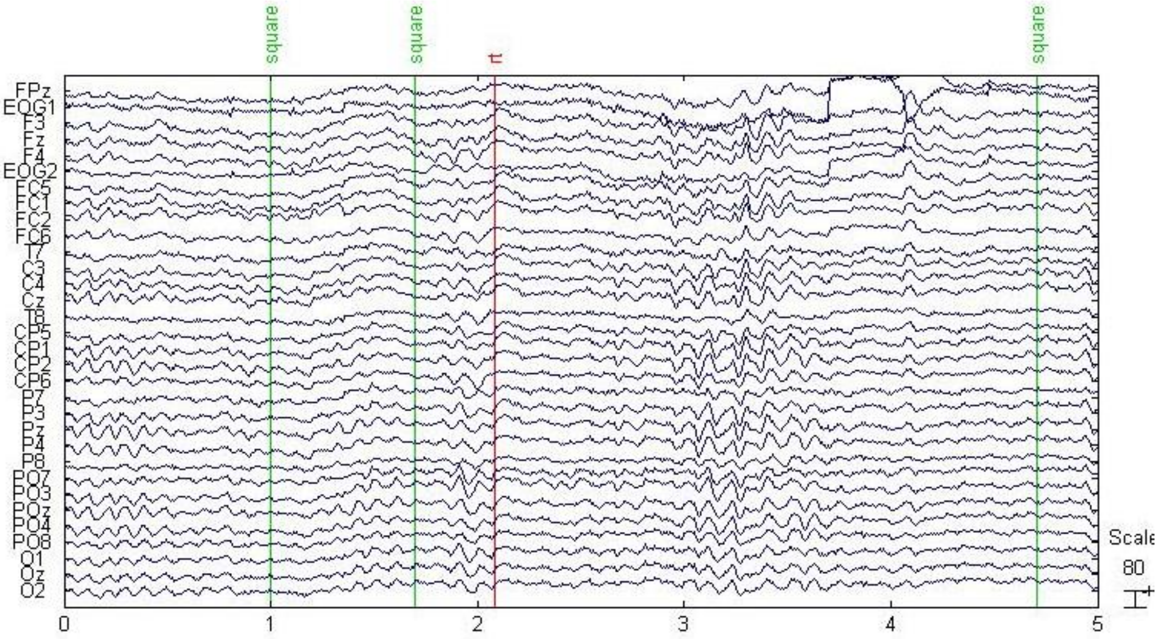

Figure 4 EEG Signals Segment from 32 Sensors.

In our work, we used the EEGLAB software package which contains an automated version “*runica*” of infomax ICA decomposition to get the independent components. Figure 5 shows the 32 separated independent components by using ICA in

EEGLAB toolbox. The toolbox also allows users to select up to 20 available ICA algorithms [27]. Each one of the obtained independent components can be seen as an independent measurement that we can use to obtain the dipole source and artifacts characteristic parameters.

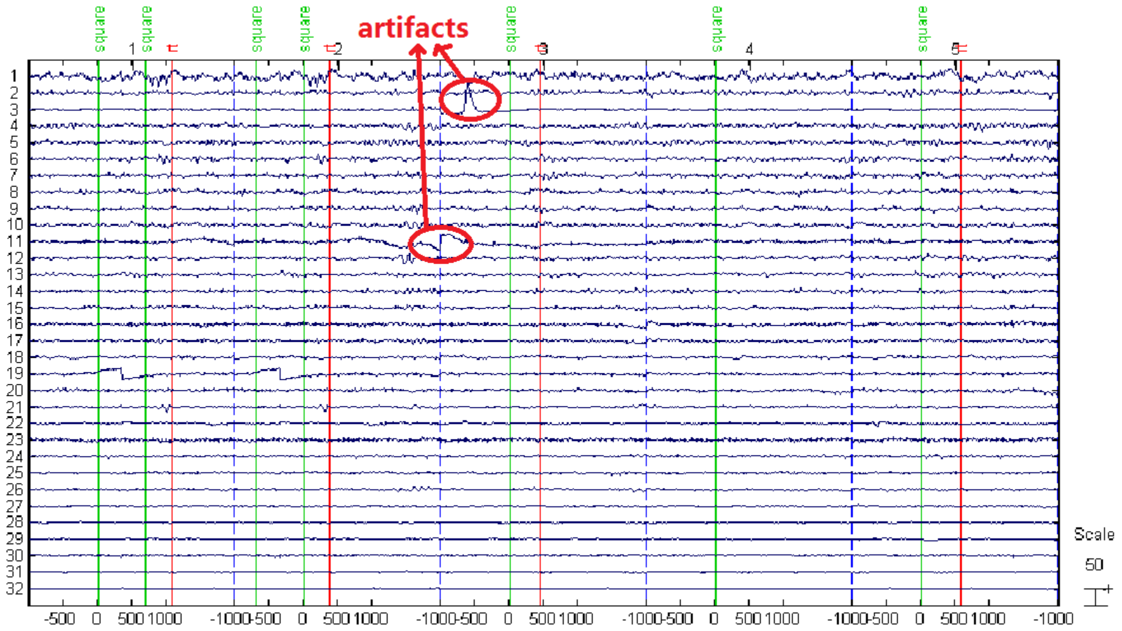

Figure 5 Separated EEG Components after ICA from -1s to 2s

5.2 Simulation Set Up

In order to demonstrate the effectiveness and performance of the proposed algorithm, we provide MATLAB simulations for multiple dipole current sources tracking in the clutter environment. Block diagram in Figure 2 illustrates the whole procedure of simulation for the proposed approach and will be detailed in the following sub sections. There are three main parts in the proposed algorithm for multiple targets tracking with PHDF-PF and artifacts suppression. The first part includes preprocessing the EEG data with low pass filter and whitening, separation of the original EEG data into independent components by using ICA decomposition, this part is done with the EEGLAB toolbox. The second part focuses on analysis of discriminative features for neural activity and artifacts, we obtained amplitude, dilation time-shift and frequency shift features of components by using the MPD with Gaussian time-frequency atom dictionary, and we used the features to model the PDFs for artifacts and neural activity with Gaussian mixture model. The last part is to track the multiple dipoles with PHDF, PDF of neural activity and PDF of artifact is applied in the PHDF to suppress artifacts, we implemented PHDF with a set of weighted particles, which is used to represent the intensity function, and update the posterior intensity function by updating the assigned weights of each particle.

In our work, we modeled the head as a hemisphere with radius of 85mm, x and y axis ranged from -85mm to 85mm while z is from 0 to 85mm. The EEG data are sampled at 128Hz. The state space model for the dipoles parameters $\mathbf{x}_i^k = [(\mathbf{r}_i^k)^T, (\mathbf{q}_i^k)^T, s_i^k]^T$ to be estimated is a random walk model:

$$\mathbf{x}_k(t) = \mathbf{x}_{k-1}(t) + \varphi(\mu, \sigma) \qquad (5.1)$$

$\varphi(\mu, \sigma)$ is the Gaussian distribution with mean 0 and variance 5 mm for position parameters.

Measurement noise $\mathbf{u}_k$ in equation (3.10) is Gaussian with 0 mean and variance 10^{-9}. We apply 1000 particles for each dipole and initialize these particles with uniform distribution. Each dipole source has a probability $\mathrm{P}_{\mathrm{k|k\text{-}1}}(x_{k-1})$= 0.9 to survive from the previous time step, and a probability $\mathrm{P}_k^D(x_k)$ = 0.95 to be detected at the measurement space. The number of dipoles is three and the number of artifacts at each time step is modeled to have a Poisson distribution with average rate 1. By given the ground truth, we performed 100 Monte Carlo simulations with synthetic data for two tracking scenarios, which are with artifacts and without artifacts. We evaluated of the tracking result using root mean-squared error (RMSE) analysis.

5.3 Independent Components Separation

5.3.1 Synthetic Data Generation

In our simulations, synthetic data is reconstructed based on the real EEG data, by given EEG data the ground truth, we can estimate the tracking performance. The block diagram below shows the synthetic data generation procedure.

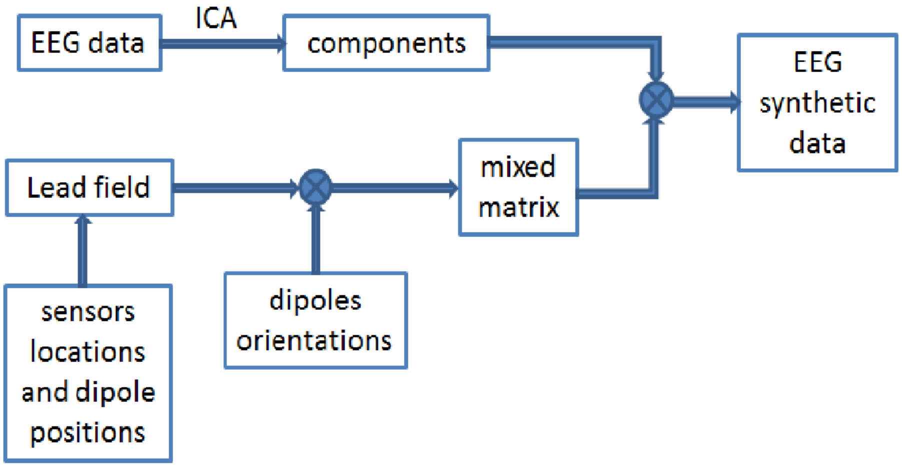

Figure 6 Block Diagram for Synthetic Data Reconstruction

5.3.2 Eigenvalue Threshold Selection

EEGLAB toolbox is a powerful toolbox which integrated many signal processing techniques. We use *'runica'* to separate EEG data into independent components. ICA provides 32 independent components (see Figure 4) which correspond to 32 sensors that are placed over the scalp. According to the clinical experience and proof, we can tell that several components are distinguished from others, for example, in the 61st and 62nd interval, on the 3rd channel, there is a sharp spike. This spike is from sensor 3 which is placed near the frontal part of scalp, this indicates the presence of eye artifact. Component 11 is considered as lateral eye movement artifact which shows a bump in signal amplitude ends at a different value than when it started. These artifacts appear in different intervals out of total 80 intervals. Components with smooth fluctuation in all 80 intervals like component 17, component 20 and component 27 are considered as the normal neural activity.

ICA algorithm integrated in EEGLAB works constantly for 32 channel signals. In order to separate mixed signals well and reduce the dimension of EEG data, the threshold based eigenvalue decomposition can be used to obtain independent components by setting a threshold to choose the leading eigenvalues instead of all of them. Eigenvalue selection method is significant since it determines the number of independent components and the reconstruction error of ICA. Figure 7 shows all 32 eigenvalue amplitudes of EEG covariance matrix in the simulated dataset, there are four leading eigenvalues in this plot. The threshold of eigenvalues in our simulation is set to be 1×10^{-10} in case of more than four leading eigenvalues.

Figure 8 gives an example of components separation with ICA in simulation at a single time step, the isolated five components consist of two neural activity components and three blinking artifact components. ICA is the first significant step in the proposed method to ensure artifact components and normal EEG component are separated independently and correctly.

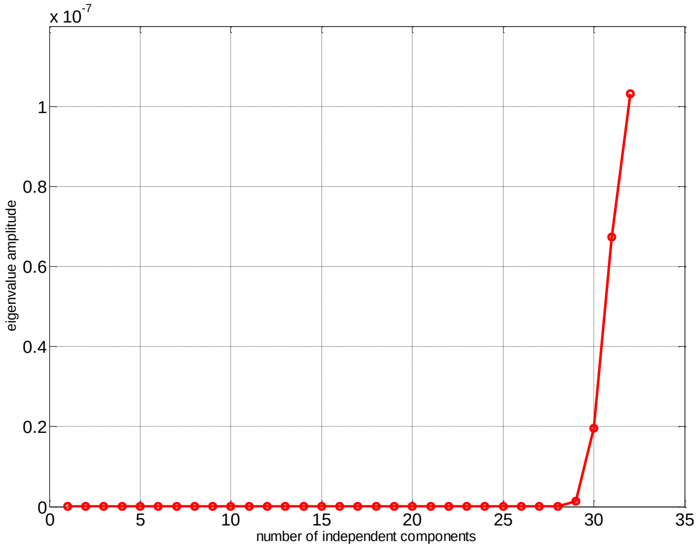

Figure 7 Eigenvalue Plot of EEG Covariance Matrix

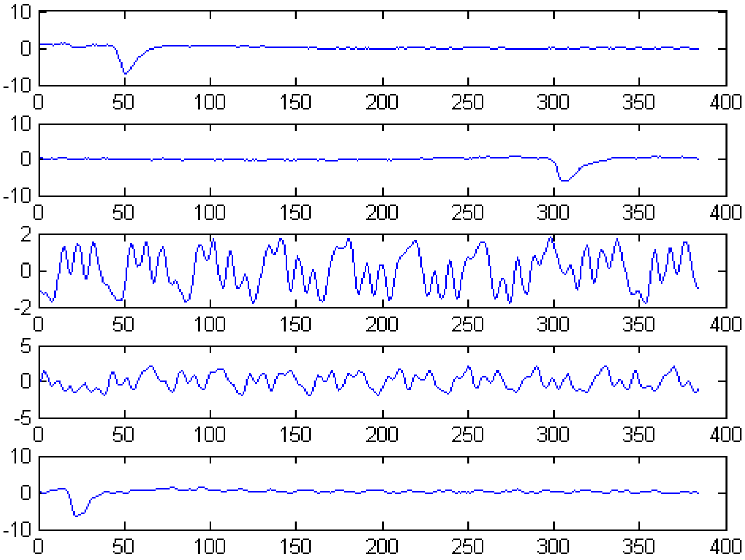

Figure 8 Separated Components of Synthetic Data

5.4 Features Extraction and Analysis

MPD with Gaussian atom time-frequency dictionary is used to optimize the approximation of signal by decomposing into different Gaussian components. According to Equation (4.5), Gaussian dictionary is decided by four parameters: coefficient constant (amplitude of the Gaussian), time shift, frequency shift and scale (dilation of the Gaussian). Frequency range is chosen from 0-30 Hz on the basis of brain activity frequency characteristic discussed in section 2.3. Time shift parameter can be decided by 3-sigma rule of normal distribution, shown in Figure 9. Three standard deviations σ account for probability of over 99%. To fit a single wave in the normal neural component signal with a Gaussian, 6σ corresponds to time interval of the single wave, as shown in Figure 10. Then the scale s can be determined with knowledge of standard deviation σ.

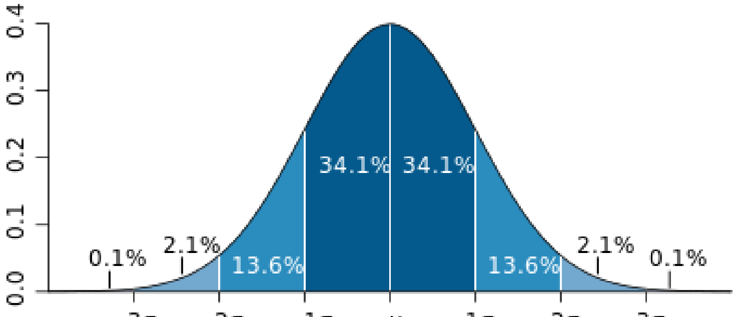

Figure 9 Three-Sigma Rule

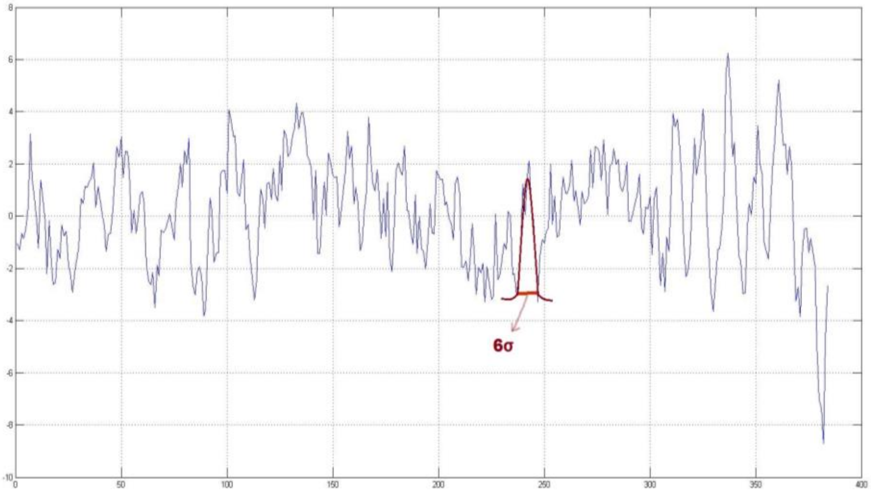

Figure 10 Single Gaussian Approximation of Neural Activity

$$\left.\begin{array}{c} s^2 = \left(\dfrac{1}{\sqrt{2}\sigma}\right)^2 \\ t = 6\sigma \end{array}\right\} \sim \left\{\begin{array}{l} s_{\max} = \dfrac{1}{6\sqrt{2}t_{\min}} \\ s_{\min} = \dfrac{1}{6\sqrt{2}t_{\max}} \end{array}\right. \tag{5.2}$$

With the same threshold (say 90%) to keep the signal energy, MPD iterations of artifacts and normal activity are different. This means 90 percent of artifact signal is decomposed into several Gaussian components, while the normal activity signal is decomposed into many more.

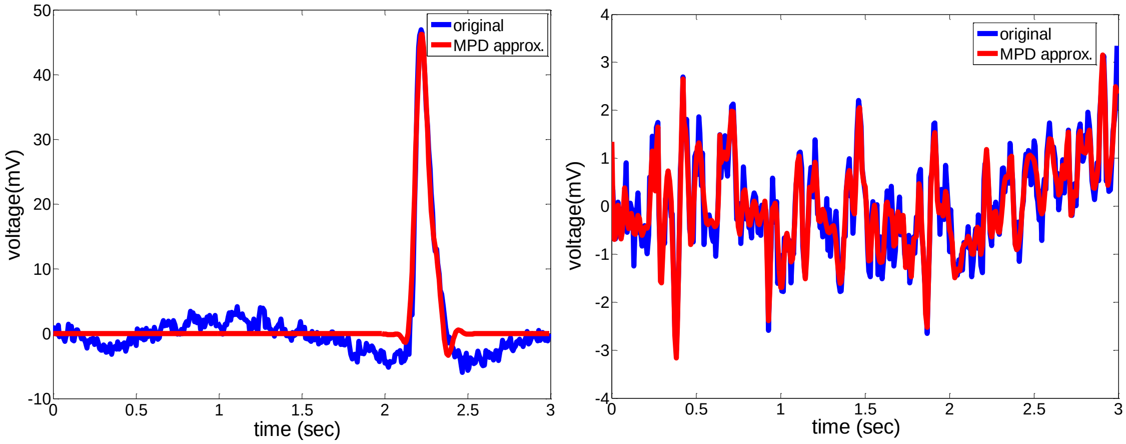

Figure 11 MPD Approximation for Blinking Artifact and Normal Activity

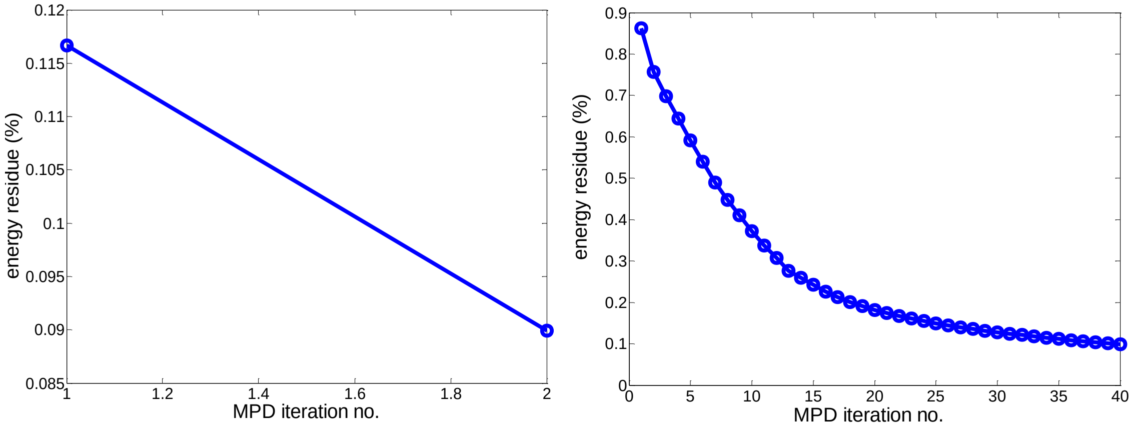

Figure 12 Energy Residue vs MPD Iteration Times

Figure 11 shows the MPD approximation for blinking artifact and normal activity signal, the blue curve is the original signal and the red curve is MPD approximation with different Gaussians. The sharp spike in blinking signal is a discriminative feature which indicates dominant energy is centered at this spike. Figure 12 shows the iteration numbers of MPD needed for blinking artifact and normal neural activity are different. To keep the same energy, Only 2 Gaussian components are sufficient to represent the envelope of blinking artifact, while for normal activity signal, 40 Gaussians are required. This difference is shown clearer in time-frequency plane.

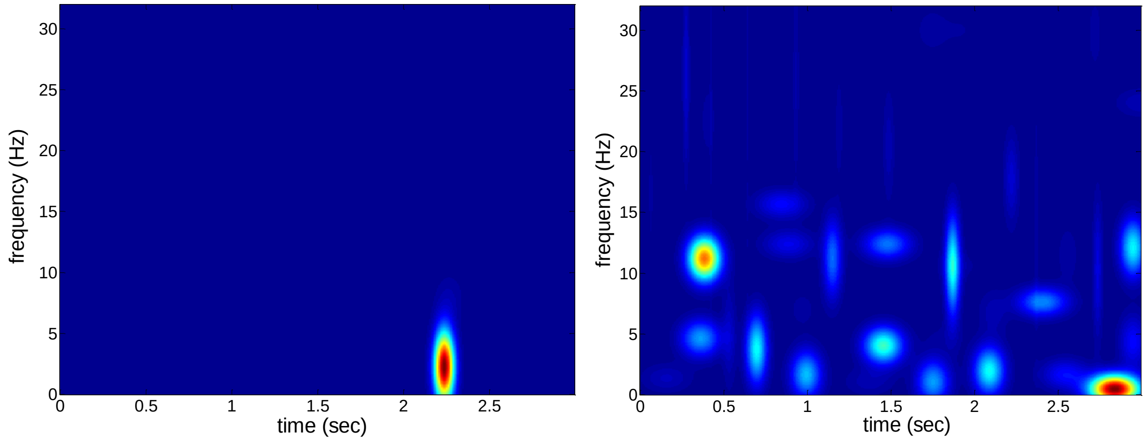

Figure 13 Cross-term Free Time-Frequency Representation for Blinking Artifact and Normal Neural Activity

Figure 13 shows the cross-term free time-frequency representations for blinking artifact and normal neural activity respectively. It can be seen that Gaussian components for blinking artifact are more centered near the 4Hz in frequency band, and Gaussian components of normal neural activity are more spread out in time- frequency plane, from 0 to 20 Hz in frequency domain and 0 to 3 seconds in time domain.

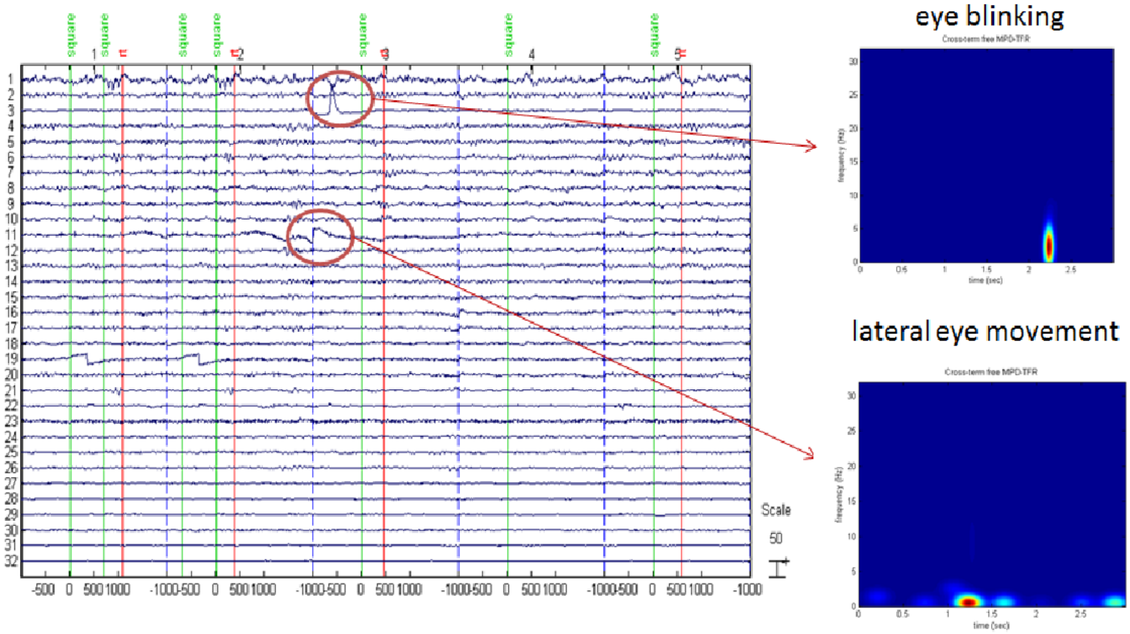

Figure 14 Two Different Artifacts in Time-Frequency Plane

Figure 14 gives two examples of artifacts: eye blinking artifact and lateral eye movement artifact and their cross-term free time-frequency representations. Eye blinking artifact is centered near 4 Hz in frequency domain while the lateral eye movement artifact is centered near 2 Hz, and variances of Gaussian components for blinking artifact are smaller than for eye movement artifact.

However, among the four features we obtained from Equation (4.7), time delay τ feature does not provide useful information since the blinking of eyes occurs randomly, thus we keep the rest three features in the PDFs modeling. Other problem we have met is the features overlap. When both artifacts signals and normal neural signals are decomposed into many Gaussian components, MPD features of artifacts have overlap with features of normal neural signals. Figure 15 shows the features overlap problem. Features overlap makes it hard to tell the difference between artifacts and normal neural activity, thus we consider the iteration number as priority instead of energy reservation. With one-time iteration in MPD, we can separate the blinking artifact from normal neural activity since the sharp spike of artifact is distinguished from other waves.

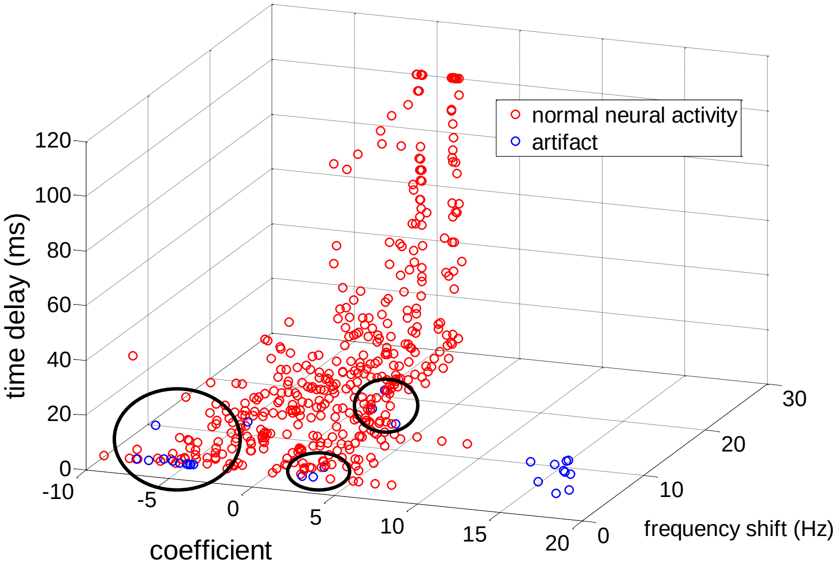

Figure 15 Features Overlap

When modeling the PDFs for normal neural activity and artifact with Gaussian mixture model, there is no information on which probability density function corresponds to which cluster, thus we use data training to classify the probability density functions. Table 1 shows the partial three features of blinking artifact and neural activity. Among the three features, there are slightly differences in the scale and frequency shift between normal neural signal and artifacts, but the coefficient difference between two groups is the most obvious, coefficient of artifacts is about two times bigger than normal neural signal's. Based on these differences, we trained the data obtained from the Gaussian mixture modeling and assigned the probability density functions to the corresponding groups.

	Coefficient(absolute value)		Scale		Frequency-shift(Hz)	
	neural	artifact	neural	artifact	neural	artifact
1	9.1212	17.6993	5.0000	11.7677	0.5000	3.1818
2	7.0870	17.2432	11.7677	14.1367	0.7980	2.8838
3	7.7798	16.2559	5.0000	7.2157	2.2879	2.8838
4	6.0193	18.3960	9.2148	18.0532	3.1818	1.6919
5	7.2870	18.3053	5.0000	14.1367	0.5000	1.6919
6	8.2628	18.4356	5.0000	18.0532	0.5000	1.9899
7	9.7250	17.6513	7.2157	5.3152	8.8434	1.9899
8	6.1678	18.4649	19.1914	7.2157	5.8636	1.9899
9	7.7604	18.4176	5.0000	12.5096	10.0354	2.2879
10	7.0952	15.4458	5.0000	12.5096	0.5000	3.7778

Table 1Three Features of Normal Neural Activity and Artifacts

5.5 Dipoles Estimation Results with PHD-PF and Artifacts Suppression

To estimate three dipoles as well as suppress artifacts, we combined the PDF of artifacts with clutter intensity function in PHDF, and combined probability density functions of normal neural signal with likelihood function in PHDF.

Figure 16-18 show the particles motion towards dipole sources at different time step. It can be seen that position estimates at the first step is inaccurate, this is because the particles are uniformly distributed. After updating the weights, particles start to lock on the dipoles at time step 2, and with more time steps, the position estimates are getting closer to dipoles true positions. Figure 19 shows a typical run of three dipoles tracking result with 20 time steps, the continuous red, green and cyan-blue curves are true motion locus of three dipoles, and the sparse blue stars are the position estimations with applying K-means clustering algorithm on weighted particles.

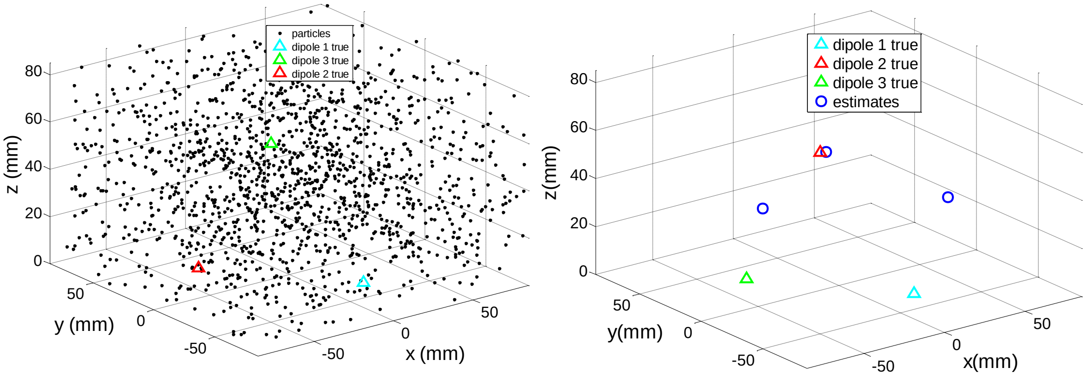

Figure 16 Particles Distribution and Estimate Result at time k=1

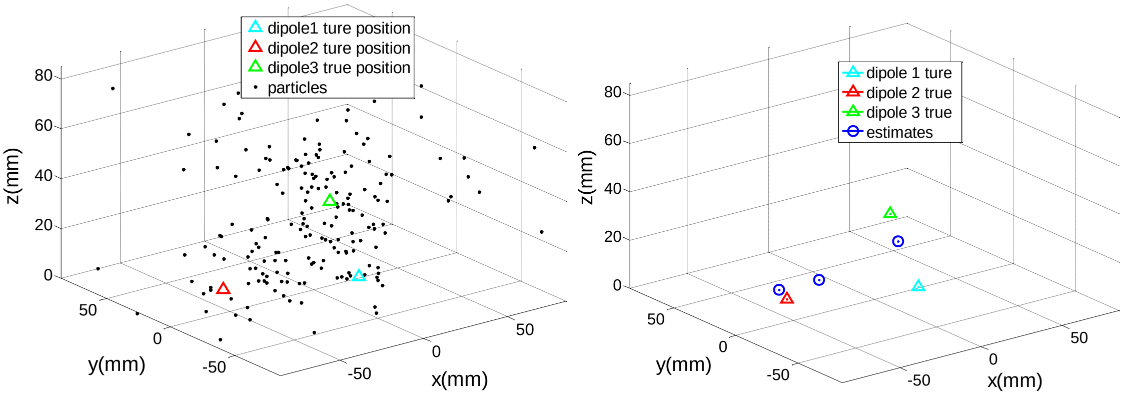

Figure 17 Particles Distribution and Estimate Result at time k = 2

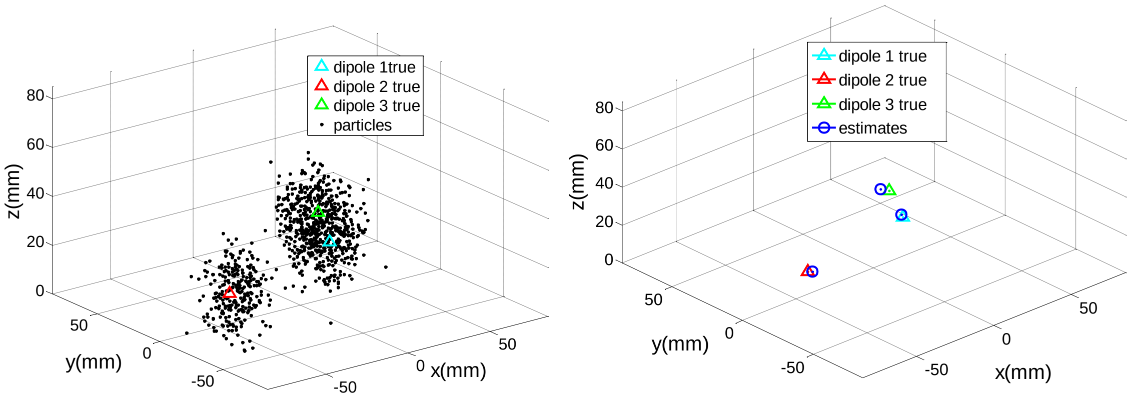

Figure 18 Particles Distribution and Estimate Result at time k= 7

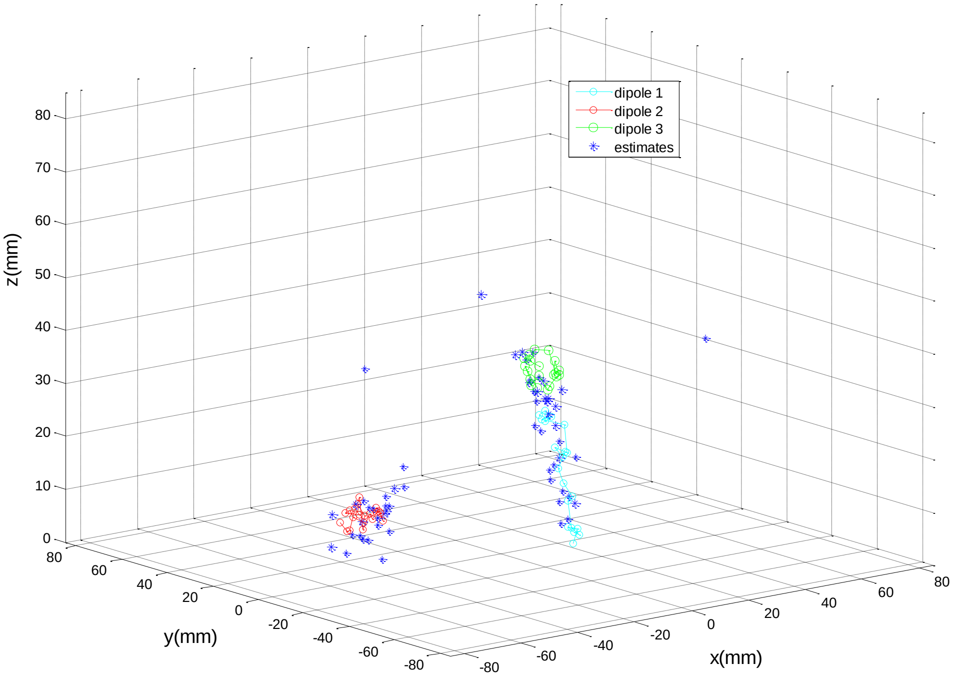

Figure 19 Three Targets Tracking using PHDF-PF

As number of artifacts at each step is time varying, it affects the dipoles estimation result. In order to further investigate the performance of the PHDF, we estimated the number of dipoles at 20 steps based on PHDF algorithm. Table 2 and Figure 20 provide the information of artifacts number in a single run, and the estimated number of dipoles in 20 time steps. The estimated number of dipoles with 20 time steps is shown in Figure 21, which is the average of 100 Monte Carlo (MC) simulations.

Time step k	1	2	3	4	5	6	7	8	9	10
Clutter number	0	2	0	1	2	0	1	0	1	0
Estimated number	3	4	3	3	3	3	3	3	3	3
Time step k	11	12	13	14	15	16	17	18	19	20
Clutter number	0	1	0	2	3	0	1	1	1	2
Estimated number	3	3	3	4	4	3	3	3	3	4

Table 2 Artifacts Number and Estimated Diploes Number in a typical run

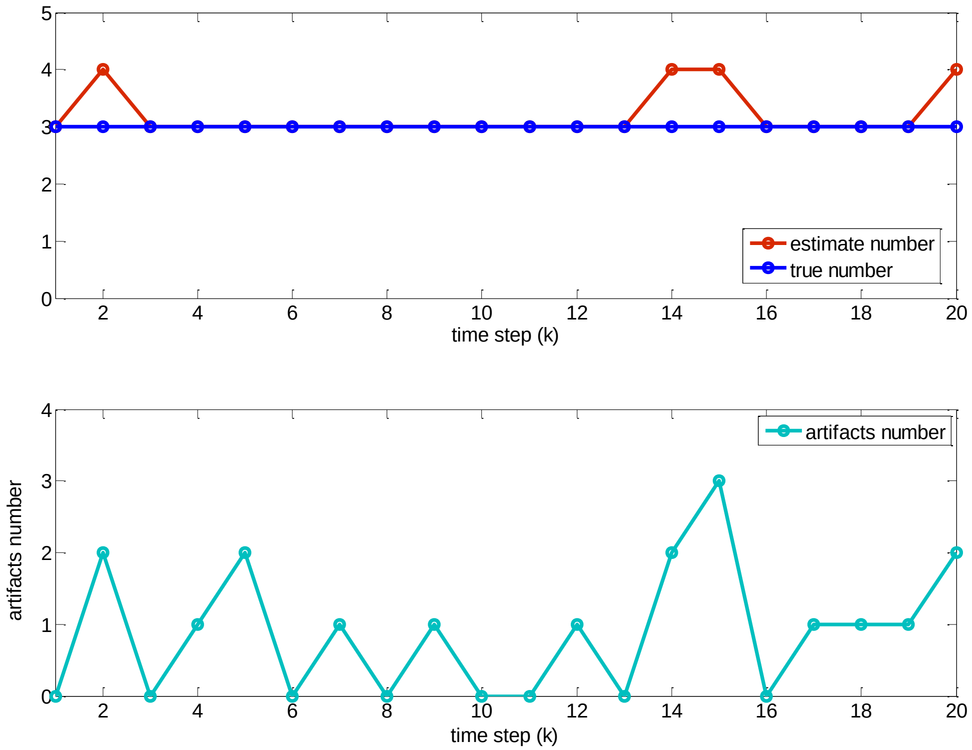

Figure 20 Estimated Dipoles Number and Artifacts Number in a typical run with PPHDF

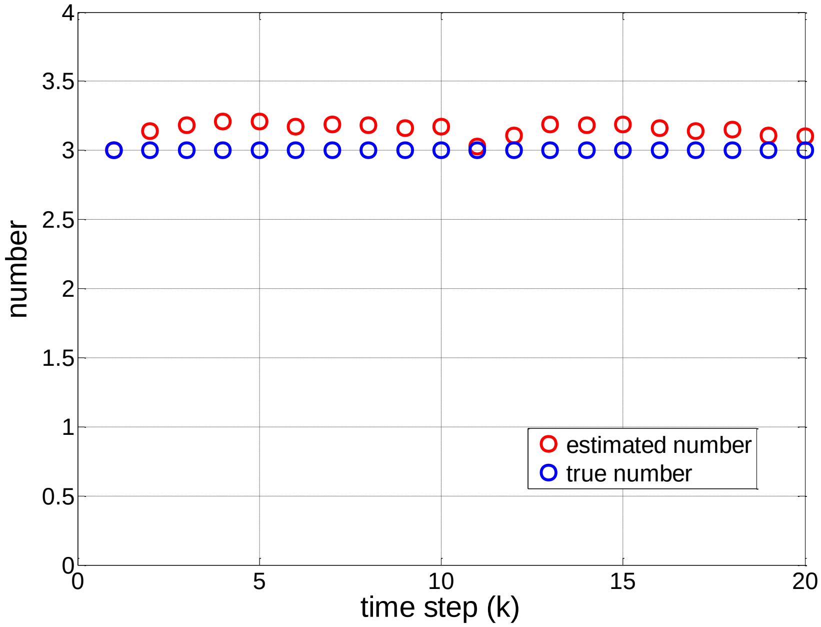

Figure 21 Estimated Average Dipoles Number of 100 MC Simulations with PPHDF

RMSE is the Root-Mean-Square Error that is a frequently used in measuring difference between the actual observed values and the predicted values. It is a commonly used measure of estimation accuracy. RMSE in the simulation is described as:

$$RMSE = \sqrt{\frac{\sum_{k=1}^{K} \frac{1}{N} (\mathbf{r} - \hat{\mathbf{r}})^T (\mathbf{r} - \hat{\mathbf{r}})}{K}} \tag{5.3}$$

$$|\mathbf{r}| = \sqrt{(r_x)^2 + (r_y)^2 + (r_z)^2} \tag{5.4}$$

K is time step and N is sensor number. Table 3 and Figure 23 show the position RMSE of three dipoles with 100 MC simulations, this is the case in the absence of artifacts when tracking dipoles.

Time step(k)	Dipole1 RMSE(mm)	Dipole2 RMSE(mm)	Dipole3 RMSE(mm)
1	26.888	28.326	27.814
2	6.402	10.196	10.837
3	5.735	5.312	5.702
4	4.596	4.522	4.575
5	5.153	4.487	3.880
6	4.118	5.079	4.759
7	4.781	3.974	5.227
8	4.745	4.524	4.207
9	5.019	4.806	4.461
10	4.773	4.785	4.408
11	4.547	4.216	4.672
12	4.420	4.732	4.814
13	4.208	5.138	4.313
14	5.070	5.842	4.147
15	4.891	5.449	4.923
16	4.579	4.266	4.589
17	4.830	5.374	4.977
18	4.546	4.542	4.016
19	4.726	4.093	4.184
20	5.060	4.607	4.093

Table 3 Position RMSE of 100 MC Simulations with Artifacts

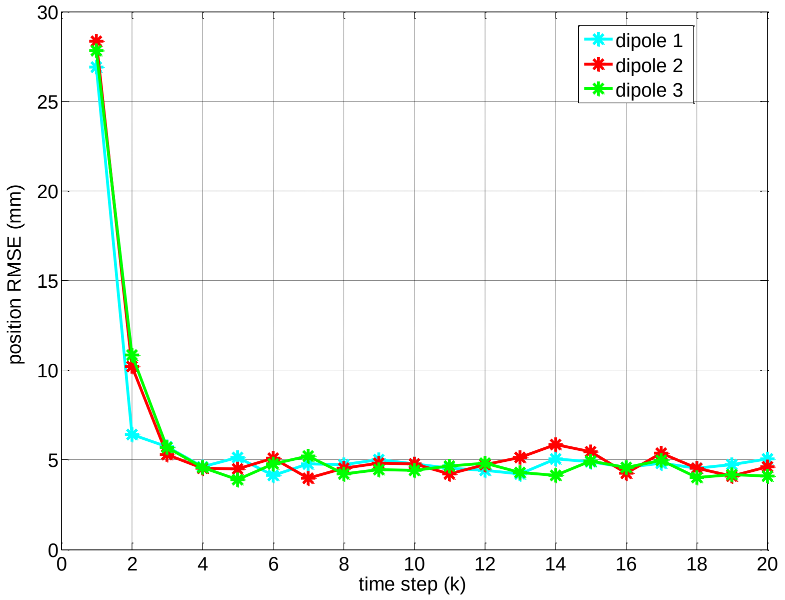

Figure 22 Position RMSE of 100 MC Simulations with Artifacts

Time step(k)	Dipole1 RMSE(mm)	Dipole2 RMSE(mm)	Dipole3 RMSE(mm)
1	28.122	27.090	28.962
2	7.035	10.103	12.130
3	5.362	4.547	4.302
4	4.403	3.777	3.844
5	3.938	3.413	3.953
6	3.842	4.008	4.036
7	4.694	3.965	4.485
8	4.034	4.023	3.896
9	4.077	4.800	3.976
10	4.029	3.964	4.108
11	4.799	3.807	4.788
12	4.591	4.224	3.821
13	4.096	4.109	4.205
14	4.378	4.029	4.453
15	4.258	4.277	4.703
16	3.754	4.013	3.961
17	4.476	3.967	4.421
18	4.231	3.705	3.810
19	4.408	3.689	5.007
20	3.532	4.083	4.796

Table 4 Position RMSE of 100 MC Simulations without Artifacts

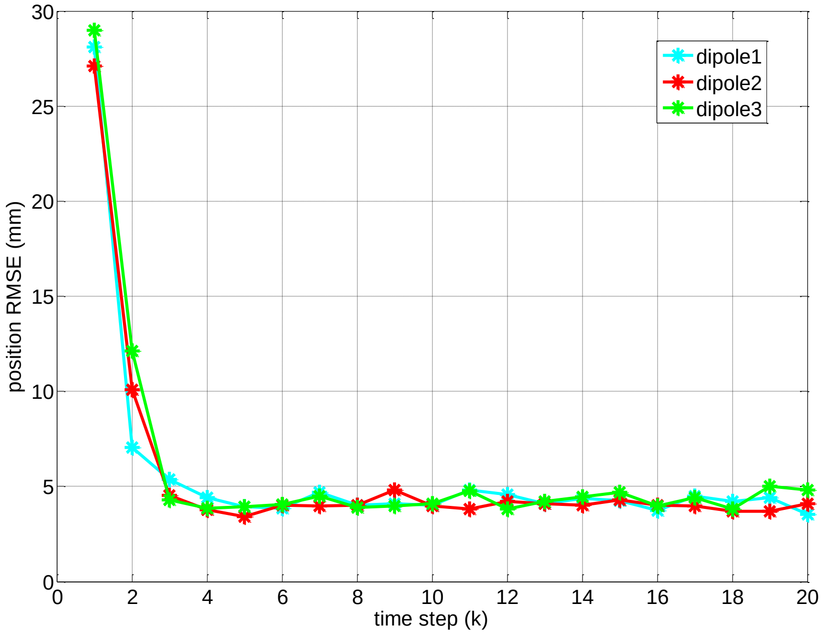

Figure 23 Position RMSE of 100 MC Simulations without Artifacts

In order to compare the tracking result with the non-artifact scenario, we provide the position RMSE of 100 MC simulations without artifacts in Table 4 and Figure 23. Plot in Figure 23 and Figure 23 both show a rapid drop from time step 1 to time step 2. The large position RMSE in the first step is caused by uniformly distributed particles. As particles spread out all over the scalp, the K-means algorithm randomly estimate three clusters as the position estimate. From step 4, the RMSE curves in both figures have slightly changes and their fluctuation is in a small range near 5mm. From Table 3 and Table 4, we can see there is a small RMSE difference between scenario with artifacts and scenario without artifact, tracking with artifacts has a RMSE of about 5mm, while without artifacts, the tracking result has a RMSE of near 4mm. Compare to the radius of the head in our model, 85mm, the RMSE 5mm is relatively small. The error would be from the

EEG dataset since the synthetic data is reconstructed from EEG dataset, and we have no knowledge of the measurement noise on this dataset.

Overall, the results from applying PHDF to neural source localization are encouraging. It proved the effectiveness of proposed method to suppress the artifacts in neural activity with characteristic analysis, and realize the multiple dipoles tracking with presence of artifacts.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this thesis, we proposed an integrated method of stochastic modeling artifacts present in EEG/MEG recordings and tracking neural activity. In order to optimize the tracking results, we used the matching pursuit decomposition (MPD) algorithm to decompose signals into different Gaussian components and extract time-frequency based features. These unique features for both neural activity and artifacts are then used as input to the Gaussian mixture modeling (GMM) algorithm to estimate corresponding probability density functions. These functions are incorporated in the probability hypothesis density particle filter (PPHDF) to help reduce the probability of falsely using measurements from artifacts to estimate localization information on the current dipole sources.

Simulation results demonstrated the effectiveness of our proposed neural sources tracking with stochastic artifact modeling (NEST-SAM) algorithm. Using NEST-SAM, we demonstrated that our algorithm improves the estimation of tracking three dipoles compare to only using independent component analysis for artifact suppression when a varying of artifacts is present in the recordings at each time step. In particular, with 100 MC simulations, the estimated average number of dipoles using PPHDF algorithm at each step is around 3.2, which is close to the true number of three dipole sources. We also demonstrated that our neural dipole tracking accuracy only slightly decreased from when PPHDF is used with data without any artifacts. Specifically, the root mean-squared error

(RMSE) of the location of the dipoles was 5mm, note that we assumed that the human brain model has a radius of 85mm.

6.2 Future Work

Some ideas for future work include the following:

1. In our simulations, we use the real EEG data to generate synthetic EEG data in order to obtain ground truth, for comparison, the tracking estimates can be improved by estimating the covariance of the actual measurement noise in the real EEG data.
2. We only analyzed ocular artifacts presents in EEG recordings. Future work will be extended to include more kinds of artifacts with our approach in more complex situations.
3. In our work, we only evaluate accuracy performance using the RMSE metric. However, using the PPHDF algorithm, we also estimate the correct number of dipole sources at each time step. A better matched PPHDF metric to use in future work in order to include this estimation result is the Optimal Sub Pattern Assignment (OSPA).
4. We applied PPHDF instead of data association for multiple-dipole tracking since it is computational intensive. In future work, we need to include a performance comparison between the two methods.

REFERENCES

[1] E. Somersalo, “The inverse problem of magnetoencephalography: Source localization and the shape of a ball,” *SIAM News*, vol. 40, no. 2, March 2007.

[2] S. Baillet, J. C. Mosher, and R. M. Leahy, “Electromagnetic brain mapping,” *IEEE Signal Processing Magazine*, vol. 18, pp. 14-30, November 2001.

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

[4] J. C. Mosher, R. M. Leahy, and P. S. Lewis, “EEG and MEG: Forward solutions for inverse methods,” *IEEE Transactions in Biomedical Engineering*, vol. 46, pp. 245-259, 1999.

[5] S. Baillet, L. Garnero, “A Bayesian approach to introducing anatomo-functional priors in the EEG/MEG inverse problem,” *IEEE Transactions on Biomedical Engineering*, vol. 44, no. 5, pp. 374-385.

[6] A. Pascarella and A. Sorrentino, “Statistical approaches to the inverse problem,” *Magnetoencephalography*, E. W. Pang, Ed. InTech, November 2011.

[7] N. Maurits, *From Neurology to Methodology and Back: An Introduction to Clinical Neuroengineering*, Springer, 2012.

[8] Roberta Grech, Tracey Cassar, Joseph Muscat, Kenneth P Camilleri, Simon G Fabri, Michalis Zervakis, Petros Xanthopoulos, Vangelis Sakkalis and Bart Vanrumste, “Review on solving the inverse problem in EEG source analysis,” *Journal of Nueroengineering and Rehabilitation*, vol. 5, no.1, pp.25, 2008.

[9] J. C. Mosher and R. M. Leahy, “Source localization using recursively applied and projected (RAP) MUSIC,” *IEEE Transactions on Signal Processing*, vol. 47, pp. 332-340, 1999.

[10] B. D. Van Veen and K. Buckley, “Beamforming: A versatile approach to spatial filtering,” *IEEE ASSP Magazine*, vol. 5, pp. 4-24, 1988.

[11] 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.

[12] 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, pp. 173-176, 2007.

[13] 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.

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

[15] C. Campi, A. Pascarella, A. Sorrentino, and M. Piana, "A Rao-Blackwellized particle filter for magnetoencephalography," *Inverse Problems and Imaging*, vol. 24, pp.15, 2008.

[16] R. Mahler. "Multi-target Bayes filtering via first-order multi-target moments," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1152-1178, 2003.

[17] L. Miao, J. J. Zhang, C. Chakrabarti, A. Papandreou-Suppappola, "Efficient Bayesian tracking of multiple sources of neural activity: Algorithms and real-time FPGA implementation," *IEEE Transactions on Signal Processing*, vol. 61, pp. 633-647, February, 2013.

[18] S. Romero, M. A. Mañanas, M. J. Barbanoj, "Ocular reduction in EEG signals based on adaptive filtering, regression and blind source separation," *Annals of biomedical engineering*, vol.37, no.1, pp. 176-191, 2009.

[19] R. Vigário,V. Jousmäki, M. Hämäläinen, R. Hari, E. Oja, "Independent component analysis for identification of artifacts in Magnetoencephalographic recordings," *Advances in Neural Information Processing Systems*, pp. 229-235, 1997.

[20] R. Vigário, J. Särelä, V. Jousmäki, M. Hämäläinen, E. Oja, "Independent component approach to the analysis of EEG and MEG recordings," *IEEE Transactions on Biomedical Engineering*, vol. 47, no. 5, pp. 589-593, 2000.

[21] A. Delorme, T. Sejnowski and S. Makeig, "Enhanced detection of artifacts in EEG data using higher-order statistics and independent component analysis," *NeuroImage,* vol. 34, no. 4, pp. 1443-1449, 2007.

[22] 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.

[23] 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.

[24] K. Islam Molla, R. Islam, T. Tanaka, T. M. Rutkowski, "Artifact suppression from EEG signals using data adaptive time domain filtering," *Neurocomputing,* vol.97, pp. 297-308, 2012.

[25] H. Zeng, A. Song, R. Yan, H. Qin, "EOG artifact correction from EEG recording using stationary subspace analysis and empirical mode decomposition," Sensors, vol. 13, pp. 14839-14859, 2013.

[26] M. A. Klados, C. Papadelis, C. Braun, and P. D. 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.

[27] R. Ferdousy, A. I. Choudhory, M. S. Islam, M. A. Rab, M. E. H. Chowdhory, "Electrooculographic and electromyographic artifacts removal from EEG," International Conference on Chemical, Biological and Environmental Engineering, pp. 163-167, November 2010.

[28] EEGLAB: An open source environment for electrophysiological signal processing (http://sccn.ucsd.edu/eeglab/)

[29] M. K. Islam, A. Rastegarnia, A. T. Nguyen, Z. Yang, "Artifact characterization and removal for *in vivo* neural recording," *Journal of Neuroscience Methods,* vol. 224, pp. 110-123, 2014.

[30] A. Maurer, L. Miao, J. J. Zhang, N. Kovvali, A. Papandreou-Suppappola, C. Chakrabarti. "EEG/MEG Artifact Suppression for Improved Neural Activity Estiation," *Signals, Systems and Computers (ASILOMAR), 2012 Conference Record of the Forty Sixth Asilomar Conference on,* pp. 1646-1650, 2012.

[31] Y. Bar-Shalom, E. Tse, "Tracking in a cluttered environment with probabilistic data association," *Automatica*, vol. 11, no.5, pp. 451-460, 1975.

[32] E Tse, Y Bar-Shalom, "Sonar tracking of multiple targets using joint probabilistic data association," *IEEE Journal of Oceanic Engineering*, vol.8, no. 3, pp.173-184.

[33] R. Mahler, *Statistical Multisource Multitarget Information Fusion*, Norwood, MA: Artech House 2007.

[34] S.G. Mallat, Z Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no.12, pp. 3397-3415, 1993.

[35] L. Cohen, *Time-Frequency Analysis*, ISSN 1050-2769.

[36] D Reynolds, *Gaussian Mixture Models*, ISBN 978-0-387-73003-5, 2009.

[37] J. A. Bilmes. "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models," International Computer Science Institute, TR-97-021, Berkeley CA, 1998.

[38] D. J. C. MacKay, *Information theory, inference and learning algorithms*, Cambridge University Press 2003.

[39] K. Uutela, M. Hämäläinen, R. Salmelin, "Global optimization in the localization of neuromagnetic sources," *IEEE Transactions on Biomedical Engineering*, vol. 45, pp. 716-723, 1998.

[40] N. Bergman, "Recursive Bayesian Estimation," Linköping Studies in Science and Technology, Desertations, No. 579, Sweden, 1999.

[41] N. J. Gordon, D. J. Salmon, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEEE Proceedings in Radar and Signal Processing*, vol. 140, pp. 107–113, 1992.

[42] G. Welch, G. Bishop, "An introduction to the Kalman filter", available http://www.cs.unc.edu, UNC-Chapel Hill, TR95-041, November 2000.

[43] E. A. Wan, R. Van der Merwe, "The unscented Kalman filter for nonlinear estimation," *Adaptive Systems for Signal Processing, Communications, and Control Symposium,* pp. 153-158, 2000.

[44] M. 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, no. 2, pp. 174–188, 2002.

[45] O. Erdinc, P. Willett, Y. Bar-Shalom, “Probability hypothesis density filter for multitarget multisensor tracking,” *Information Fusion, 2005 8th International Conference on*, vol.1, July 2005.

[46] H. Sidenbladh, “Multi-target particle filtering for the probability hypothesis density,” Swedish Defence Research Agency, SE–172 90, 2003.

[47] D. Clark, I. T. Ruiz, Y. Petillot, J. Bell, “Particle PHD filter multiple target tracking in sonar image,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 1, pp. 409-416, 2007.

[48] J Mullane, B. N. Vo,M. D. Adams and W. S. Wijesoma, “Random set formulation for Bayesian SLAM,” *In 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems Acropolis Convention Center*, Nice, France, Sept. 22-26, pp.1043-1049, 2008.

[49] E. Maggio, M. Taj, and A. Cavallaro, “Efficient multitarget visual tracking using random finite sets,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol.18, no.8, pp:1016-1027, 2008.

[50] D. Daley and D. Vere-Jones, *An Introduction to the Theory of Point Process*, G. Berlin, Ed. Springer Verlag, 1988.

[51] D. E. Clark, K. Panta, and B. N. Vo, “The GM-PHD filter multiple target tracker,” *Proceedings of 9th International Conference on Information Fusion*, pp.1–8, 2006.

[52] A. Delorme and S. Makeig, (2012, March) EEGLAB tutorial outline chapter 1: Loading data in EEGLAB. [Online]. Available: http://sccn.ucsd.edu/wiki/Chapter 01: Loading Data in EEGLAB.

[53] A. Delorme, S. Makeig, “EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis,” *Journal of Neuroscience Methods*, vol.134, no.1, pp.9-21, 2004.

[54] EEGLAB: Open source Matlab Toolbox for EEG anaylsis, http://sccn.ucsd.edu/eeglab/.

[55] A Hyvärinen, J Karhunen, E Oja, “Independent Component Analysis: Algorithms and Applications,” *Neural Networks*, vol.13, no.4 & no.5, pp. 411-430, 2000.