

# Multi-source Neural Activity Estimation and Sensor Scheduling: Algorithms and Hardware Implementation

Lifeng Miao · Stefanos Michael ·  
Narayan Kovvali · Chaitali Chakrabarti ·  
Antonia Papandreou-Suppappola

the date of receipt and acceptance should be inserted later

**Abstract** Electroencephalography (EEG) and magnetoencephalography (MEG) measurements are used to localize neural activity by solving the electromagnetic inverse problem. In this paper, we propose a new approach based on the particle filter implementation of the probability hypothesis density filter (PF-PHDF) to automatically estimate the unknown number of time-varying neural dipole sources and their parameters using EEG/MEG measurements. We also propose an efficient sensor scheduling algorithm to adaptively configure EEG/MEG sensors at each time step to reduce total power consumption. We demonstrate the improved performance of the proposed algorithms using simulated neural activity data. We map the algorithms onto a Xilinx Virtex-5 field-programmable gate array (FPGA) platform and show that it only takes 10 ms to process 100 data samples using 6,400 particles. Thus, the proposed system can support real-time processing of an EEG/MEG neural activity system with a sampling rate of up to 10 kHz.

**Keywords** Probability hypothesis density filter · particle filter · neural activity · dipole source modeling · sensor scheduling · power consumption · parallel architecture · FPGA implementation

---

This work was supported by the National Science Foundation under Grant No. 0830799.

The real-time estimation of an unknown number of neural sources was discussed in our 2011 IEEE Workshop on Signal Processing Systems paper [1]. In addition, this work presents the probability hypothesis density filter implemented using particle filtering with a new pre-whitening processing algorithm (Section 4); a sensor scheduling algorithm (Section 5) and its hardware implementation (Section 6.2); performance results to demonstrate sensor scheduling for simulated EEG data (Section 7.1); new hardware implementation results with sensor scheduling (Section 7.2); and scalability analysis (Section 7.3).

School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ  
E-mail: lifeng.miao@asu.edu

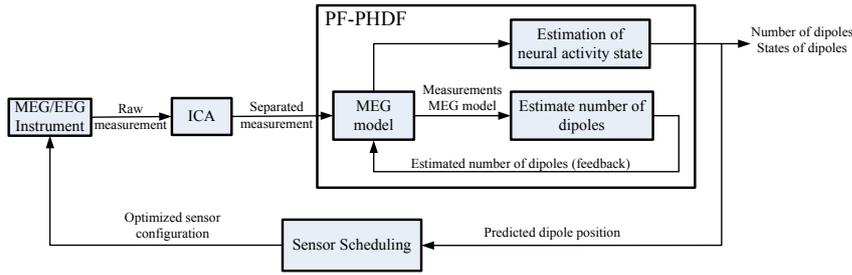
## 1 Introduction

Electroencephalography (EEG) and magnetoencephalography (MEG) are neurophysiological diagnostic tools that are used to measure external electromagnetic signals generated by the brain's neural activity [2]. MEG/EEG measurements can be used to localize neural activity and can lead to new treatments of illnesses such as stroke, epilepsy, and Parkinson's disease. Although MEG/EEG yield high temporal resolution (in the order of tens of ms), for superior diagnostics, high spatial resolution is also required. This can only be achieved if the neural sources could be accurately localized from the measurements, a problem that is referred to as the MEG/EEG inverse problem. Furthermore, since current dipole models have been successfully used to determine the brain's electrical current distribution [3, 4], the inverse problem can be simplified estimating the parameters of localized current dipole neural sources from MEG/EEG measurements.

Different methods have been applied to solve the inverse problem [4–13]. More recently, Bayesian methods such as Kalman filtering [14, 15] and particle filtering (PF) [16] have been used to estimate the parameters (location, orientation, and amplitude) of the multiple dipole sources [15, 17, 18]. The Rao-Blackwellized PF and the beamforming PF were used in [8, 9] to further improve estimation performance. Note that these methods assume that neural activity can be represented by a fixed and known number of current dipole sources. However, in realistic scenarios, the number of neural dipole sources varies with time and has to be estimated together with their parameters. An approach to dynamically estimate the number of dipoles and their neural activity was proposed in [11], based on modeling the number of neural sources at each time step using a random finite set (RFS) [19]. However, this approach requires a large number of particles (in the order of 100,000) to achieve good estimation performance, and as a result it is not amenable to real-time implementation.

In this paper, we propose a new approach to estimate both the unknown number of neural dipole sources and their parameters from EEG/MEG data. The approach is based on the PF implementation of the probability hypothesis density filter (PHDF), as shown in Figure 1. The PF-PHDF is used to estimate the state of multiple objects [20]. However, it cannot be directly used for solving the EEG/MEG inverse problem. We first separate the measurements into independent components corresponding to different neural sources [21], and then we use the PF-PHDF to estimate the unknown number of dipoles at each time step in a closed-loop configuration followed by the estimation of their parameters. The closed-loop approach simplifies the dual estimation to a known-number of dipole sources problem; it thus decreases the number of required particles and reduces the computational complexity. For real-time implementation, we propose using window-based processing along with threshold-based eigenvalue distilling to reduce the computational complexity and speed up processing.

We also consider a wireless ambulatory and portable (wearable) EEG system which allows monitoring of a patient over long periods of time [22, 23]. We propose a sensor scheduling algorithm which identifies a subset of sensors to be powered on, thereby significantly reducing the power consumption of the portable EEG system. The algorithm is based on adaptively configuring the sensors used to collect the EEG measurements at each time step using the minimum predicted mean-squared error (MSE) or maximum signal-to-noise ratio (SNR) as the performance metric.



**Fig. 1** Block diagram of proposed closed-loop neural dipole source estimation system.

Optimization is performed globally by searching over all available sensors. We show that the proposed sensor scheduling algorithm significantly reduces the number of sensors required with minimum estimation performance degradation. For example, using only 15 out of 32 sensors, the root MSE (RMSE) in dipole location estimation only slightly increases from 6.28 mm to 6.41 mm.

The proposed PF-PHDF and sensor scheduling algorithms are implemented on a Xilinx Virtex-5 FPGA platform. For the 4-processor architecture, the processing time to process 100 data samples using 6,400 particles is shown to be 10 ms, which makes real-time neural activity tracking feasible for EEG instrumentation with sampling rates up to 10 kHz.

The rest of the paper is organized as follows. In Section 2, we present the neural activity dipole model for the EEG/MEG inverse problem, and we review the PF-PHDF in Section 3. In Section 4, we develop a new algorithm for jointly estimating the unknown number of dipole sources and their parameters using PF-PHDF. In Section 5, we develop a new sensor scheduling algorithm to reduce the number of active sensors. Hardware implementations of the proposed algorithms are provided in Section 6. We present numerical simulation results, together with algorithmic and hardware performance results, in Section 7.

## 2 Neural Activity Dipole Source Model

The MEG/EEG source localization problem is based on the current dipole source models as well as the assumption that the head consists of nested concentric spheres of constant conductivity [5, 15, 17, 18, 24]. According to this model, the primary current  $I_k(\mathbf{r}) = \sum_{j=1}^{N_d} \mathbf{m}_{k,j} \delta(\mathbf{r} - \mathbf{r}_{k,j})$  at time  $k$  can be represented in terms of  $N_d$  current dipoles where  $\mathbf{r}_{k,j} = [r_{k,j}^{(x)} \ r_{k,j}^{(y)} \ r_{k,j}^{(z)}]^\top$  and  $\mathbf{m}_{k,j} = [m_{k,j}^{(x)} \ m_{k,j}^{(y)} \ m_{k,j}^{(z)}]^\top$  denote the three-dimensional (3-D) location and moment vectors, respectively, in Cartesian coordinates, of the  $j$ th current dipole at time  $k$ ,  $j = 1, \dots, N_d$ ,  $k = 1, \dots, K$ . The moment of the  $j$ th dipole is given by  $\mathbf{m}_{k,j} = \mathbf{q}_{k,j} s_{k,j}$ , where  $\mathbf{q}_{k,j} = [q_{k,j}^{(x)} \ q_{k,j}^{(y)} \ q_{k,j}^{(z)}]^\top$  is the orientation vector and  $s_{k,j}$  is the amplitude of the dipole. The measured MEG/EEG signals acquired by  $M$  sensors can be represented as [24]

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{n}_k = \mathbf{A}_k \mathbf{s}_k + \mathbf{n}_k, \quad (1)$$

where  $\mathbf{z}_k = [z_{k,1} \ z_{k,2} \ \dots \ z_{k,M}]^\top$  and  $\mathbf{n}_k = [n_{k,1} \ n_{k,2} \ \dots \ n_{k,M}]^\top$  are the  $M \times 1$  signal and measurement noise vectors, respectively. The source parameter state vector

for  $N_d$  dipoles is  $\mathbf{x}_k = [\mathbf{x}_{k,1}^\top \mathbf{x}_{k,2}^\top \dots \mathbf{x}_{k,N_d}^\top]^\top$ , where  $\mathbf{x}_{k,j} = [\mathbf{r}_{k,j}^\top \mathbf{q}_{k,j}^\top s_{k,j}]^\top$  is the 7-D parameter vector consisting of the 3-D location  $\mathbf{r}_{k,j}$ , 3-D orientation  $\mathbf{q}_{k,j}$ , and 1-D amplitude  $s_{k,j}$  of the  $j$ th dipole source at time  $k$ . We assume that the dipole sources are mutually independent of each other and statistically independent of the noise  $\mathbf{n}_k$ , and also that the noise components  $n_{k,m}$ ,  $m = 1, \dots, M$ , are mutually independent as in [24]. Also in Equation (1),  $\mathbf{s}_k = [s_{k,1} \ s_{k,2} \ \dots \ s_{k,N_d}]^\top$  is the amplitude vector corresponding to  $N_d$  dipoles and  $\mathbf{A}_k$  is the  $M \times N_d$  lead-field matrix that depends on the  $j$ th dipole location  $\mathbf{r}_{k,j}$  and orientation  $\mathbf{q}_{k,j}$ . Note that MEG/EEG systems have different lead-field matrices. Specifically, the  $(m, j)$ th element  $a_{k,m,j}$ ,  $j = 1, \dots, N_d$ ,  $m = 1, \dots, M$ , of the lead-field matrix  $\mathbf{A}_k$  at time  $k$  of an EEG model is given by [25]

$$a_{k,m,j} = \frac{\cos(\theta_{k,j})}{4\pi\sigma} \left[ \frac{2 \left( |\mathbf{r}_{k,j}| \cos(\gamma_{k,m,j}) - h \right)}{d_{k,m,j}^3} + \frac{1}{d_{k,m,j} |\mathbf{r}_{k,j}|} - \frac{1}{h |\mathbf{r}_{k,j}|} \right] + \frac{\sin(\theta_{k,j}) \cos(\beta_{k,j}) \sin(\gamma_{k,m,j})}{4\pi\sigma} \left[ \frac{2h}{d_{k,m,j}^3} + \frac{d_{k,m,j} + h}{h d_{k,m,j} (h - |\mathbf{r}_{k,j}| + d_{k,m,j})} \right] \quad (2)$$

In the dipole source model of Equation (2),  $h$  is the radius of the head model,  $d_{k,m,j}$  is the distance between the  $j$ th dipole source and the  $m$ th sensor,  $\gamma_{k,m,j}$  is the angle between the vector pointing to the  $m$ th sensor and the vector pointing to the  $j$ th dipole location,  $\theta_{k,j}$  is the angle between the  $j$ th dipole orientation and the vector pointing to the  $j$ th dipole location,  $\beta_{k,j}$  is the angle between the plane formed by the  $j$ th dipole and the origin, and  $\sigma$  is the head tissue conductivity constant. Also,  $|\mathbf{r}_{k,j}| = [(r_{k,j}^{(x)})^2 + (r_{k,j}^{(y)})^2 + (r_{k,j}^{(z)})^2]^{1/2}$ .

For an MEG system, the  $(m, j)$ th lead-field matrix element for the  $m$ th sensor, with position  $\mathbf{r}_m$  and orientation  $\mathbf{q}_m$ , is given by [25]

$$a_{k,m,j} = \left[ \frac{\mu_0}{4\pi g^2(\mathbf{r}_{k,j}, \mathbf{r}_m)} \mathbf{r}_{k,j} \times \left( g(\mathbf{r}_{k,j}, \mathbf{r}_m) \mathbf{q}_m - \mathbf{f}^\top(\mathbf{r}_{k,j}, \mathbf{r}_m) \mathbf{q}_m \mathbf{r}_m \right) \right]^\top \mathbf{q}_{k,j},$$

where  $(\mathbf{a} \times \mathbf{b})$  denotes the cross product between vectors  $\mathbf{a}$  and  $\mathbf{b}$ . The scalar  $g(\mathbf{r}_{k,j}, \mathbf{r}_m)$  and the vector  $\mathbf{f}(\mathbf{r}_{k,j}, \mathbf{r}_m)$  are defined

$$g(\mathbf{r}_{k,j}, \mathbf{r}_m) = d_{k,m,j} \left( d_{k,m,j} |\mathbf{r}_m| + |\mathbf{r}_m|^2 - \mathbf{r}_{k,j}^\top \mathbf{r}_m \right),$$

$$\mathbf{f}(\mathbf{r}_{k,j}, \mathbf{r}_m) = \left( \frac{d_{k,m,j}^2}{|\mathbf{r}_m|} + \eta_{k,m,j} + 2d_{k,m,j} + 2|\mathbf{r}_m| \right) \mathbf{r}_m - \left( d_{k,m,j} + 2|\mathbf{r}_m| + \eta_{k,m,j} \right) \mathbf{r}_{k,j},$$

where  $\eta_{k,m,j} = (\mathbf{r}_{k,j} - \mathbf{r}_m)^\top \mathbf{r}_m / d_{k,m,j}$  and  $\mu_0$  is the permittivity of free space.

The MEG/EEG inverse problem is to estimate the dynamic parameters,  $\mathbf{s}_{k,j}$ ,  $\mathbf{r}_{k,j}$  and  $\mathbf{q}_{k,j}$ , of the  $N_d$  dipole sources at time  $k$  using the MEG/EEG signals  $\mathbf{z}_k$  in Equation (1) up to time  $k$ .

### 3 PHD Filter and Its Particle Implementation

#### 3.1 Probability Hypothesis Density Filtering

The PHDF is an efficient approach for solving the multiple object tracking problem. It can be used to dynamically estimate the parameters or states of an unknown number of objects based on noisy observations. The multiple dipole source RFS probability hypothesis density or intensity function can be used to represent the source first moment, which in turn can be used to characterize the multiple dipole source posterior density. Thus, integrating the intensity function over a given region  $\mathcal{R}$  provides the expected number of objects present in region  $\mathcal{R}$ . Also, the locations of the peaks of the intensity function provide estimates of the parameters of the objects in the region  $\mathcal{R}$  [19].

In a multi-object tracking formulation, the state and measurement at time  $k$  are modeled as RFS [19, 26]  $\mathbf{X}_k = \{\mathbf{x}_{k,1}, \dots, \mathbf{x}_{k,N_k}\}$  and  $\mathbf{Z}_k = \{z_{k,1}, \dots, z_{k,M_k}\}$ , where  $\mathbf{x}_{k,i}$  is the state vector of the  $i$ th single object,  $N_k$  is the number of objects at time  $k$ ;  $z_{k,m}$  is the  $m$ th measurement in (1) and  $M_k$  is the number of measurements at time  $k$ . The PHDF can recursively approximate the intensity function of multiple objects using prediction and updating steps [20]:

$$\zeta(\mathbf{x}_k | \mathbf{Z}_{1:k-1}) = \zeta(\mathbf{x}_k^{\text{new}} | \mathbf{Z}_{1:k}) + \int \left[ \Pr_{k|k-1}(\tilde{\mathbf{x}}_{k-1}) p(\mathbf{x}_k | \tilde{\mathbf{x}}_{k-1}) + \zeta(\mathbf{x}_k^{\text{spn}} | \mathbf{Z}_{1:k-1}) \right] \zeta(\tilde{\mathbf{x}}_{k-1} | \mathbf{Z}_{1:k-1}) d\tilde{\mathbf{x}}_{k-1} \quad (3)$$

where  $\mathbf{x}_k^{\text{new}} \in \mathbf{X}_k^{\text{new}}$ ,  $\mathbf{x}_k^{\text{spn}} \in \mathbf{X}_k^{\text{spn}}$ , and  $\Pr_{k|k-1}(\mathbf{x}_{k-1})$  is the probability that a dipole source that was present at time step  $(k-1)$  will still be present at time step  $k$ . The posterior intensity is given by

$$\zeta(\mathbf{x}_k | \mathbf{Z}_{1:k}) = (1 - \Pr_k^{\text{det}}(\mathbf{x}_k)) \zeta(\mathbf{x}_k | \mathbf{Z}_{1:k-1}) + \sum_{\mathbf{z}_k \in \mathbf{Z}_{1:k}} \frac{\Pr_k^{\text{det}}(\mathbf{x}_k) p(\mathbf{z}_k | \mathbf{x}_k) \zeta(\mathbf{x}_k | \mathbf{Z}_{1:k-1})}{\zeta(\mathbf{z}_k^{\text{clt}}) + \int \Pr_k^{\text{det}}(\tilde{\mathbf{x}}_k) p(\mathbf{z}_k | \tilde{\mathbf{x}}_k) \zeta(\tilde{\mathbf{x}}_k | \mathbf{Z}_{1:k-1}) d\tilde{\mathbf{x}}_k} \quad (4)$$

where  $\Pr_k^{\text{det}}(\mathbf{x}_k)$  is the probability of detecting a dipole source at time step  $k$ . As the received multiple measurement RFS  $\mathbf{Z}_k$  can also include clutter,  $\mathbf{z}_k^{\text{clt}}$ , due to possible false alarms,  $\zeta(\mathbf{z}_k^{\text{clt}})$  is used to denote the clutter intensity; it is assumed that the clutter RFS is independent of the dipole source measurement RFS and that the dipole source measurement RFS elements are mutually independent.

#### 3.2 PHDF Particle Filtering Implementation

The PHDF prediction and update equations in (3) and (4) involve multiple integrals that do not have computationally tractable closed form expressions, even for the simple linear Gaussian dynamic case. One possible implementation of the PHDF is using particle filtering [20]. Specifically, assuming that the intensity function  $\zeta(\mathbf{x}_{k-1} | \mathbf{Z}_{1:k-1})$  at time step  $(k-1)$  can be approximated by a set of  $N_p$  particles  $\mathbf{x}_{k-1}^{(\ell)}$  and their corresponding weights  $w_{k-1}^{(\ell)}$ ,  $\ell = 1, \dots, N_p$ ,

$$\zeta(\mathbf{x}_{k-1} | \mathbf{Z}_{1:k-1}) = \sum_{\ell=1}^{N_p} w_{k-1}^{(\ell)} \delta(\mathbf{x}_{k-1} - \mathbf{x}_{k-1}^{(\ell)}). \quad (5)$$

where  $\delta(\cdot)$  is the Dirac delta function, then substituting (5) into (3), we obtain

$$\zeta(\mathbf{x}_k|\mathbf{Z}_{1:k-1}) = \zeta(\mathbf{x}_k^{\text{new}}|\mathbf{Z}_{1:k}) + \sum_{\ell=1}^{N_p} w_{k-1}^{(\ell)} \left[ \text{Pr}_{k|k-1}(\mathbf{x}_{k-1}^{(\ell)}) p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(\ell)}) + \zeta(\mathbf{x}_k^{\text{spn}}|\mathbf{Z}_{1:k-1}) \right] \quad (6)$$

A particle approximation of  $\zeta(\mathbf{x}_k|\mathbf{Z}_{1:k-1})$  can be obtained by applying importance sampling to each term in (6). Specifically, the samples  $\mathbf{x}_k^{(\ell)}$ ,  $\ell = 1, \dots, N_p$ , are drawn from the importance density function  $q_k(\mathbf{x}_k^{(\ell)}|\mathbf{x}_{k-1}^{(\ell)}, \mathbf{Z}_k) = p(\mathbf{x}_k^{(\ell)}|\mathbf{x}_{k-1}^{(\ell)})$ , and the samples  $\mathbf{x}_k^{(\ell)}$ ,  $\ell = N_p + 1, \dots, (N_p + N_q)$  are drawn from the importance intensity function  $\xi_k(\mathbf{x}_k^{(\ell)}|\mathbf{Z}_k) = \zeta(\mathbf{x}_k^{\text{new}}|\mathbf{Z}_{1:k})$ . Then, the prior intensity function  $\zeta(\mathbf{x}_k|\mathbf{Z}_{1:k-1})$  can be approximated by particles  $\mathbf{x}_k^{(\ell)}$  and their weights  $w_{k|k-1}^{(\ell)}$ ,  $\ell = 1, 2, \dots, (N_p + N_q)$  as

$$\zeta(\mathbf{x}_k|\mathbf{Z}_{1:k-1}) = \sum_{\ell=1}^{N_p+N_q} w_{k|k-1}^{(\ell)} \delta(\mathbf{x}_k - \mathbf{x}_k^{(\ell)}) \quad (7)$$

where

$$w_{k|k-1}^{(\ell)} = \begin{cases} w_{k-1}^{(\ell)} \left( \text{Pr}_{k|k-1}(\mathbf{x}_{k-1}^{(\ell)}) + \frac{\zeta(\mathbf{x}_k^{\text{spn}}|\mathbf{Z}_{1:k-1})}{p(\mathbf{x}_k^{(\ell)}|\mathbf{x}_{k-1}^{(\ell)})} \right), & \ell = 1, \dots, N_p \\ 1/N_q, & \ell = N_p + 1, \dots, (N_p + N_q) \end{cases} \quad (8)$$

and  $N_q$  is the additional number of particles for the new dipole sources at time step  $k$ .

Substituting (7) into (4), we obtain the particle approximation of the posterior intensity function as

$$\zeta(\mathbf{x}_k|\mathbf{Z}_{1:k}) = \sum_{\ell=1}^{N_p+N_q} w_k^{(\ell)} \delta(\mathbf{x}_k - \mathbf{x}_k^{(\ell)}), \quad (9)$$

where

$$w_k^{(\ell)} = w_{k|k-1}^{(\ell)} \left( 1 - \text{Pr}_k^{\text{det}}(\mathbf{x}_k^{(\ell)}) + \sum_{\mathbf{Z}_k \in \mathbf{Z}_{1:k}} \frac{\text{Pr}_k^{\text{det}}(\mathbf{x}_k^{(\ell)}) p(\mathbf{Z}_k|\mathbf{x}_k^{(\ell)})}{\zeta(\mathbf{Z}_k^{\text{clt}}) + C_k(\mathbf{Z}_k)} \right) \quad (10)$$

and

$$C_k(\mathbf{Z}_k) = \sum_{\ell=1}^{N_p+N_q} w_{k|k-1}^{(\ell)} \text{Pr}_k^{\text{det}}(\mathbf{x}_k^{(\ell)}) p(\mathbf{Z}_k|\mathbf{x}_k^{(\ell)}).$$

Based on (3) and (4), a particle approximation of the posterior intensity  $\zeta(\mathbf{x}_k|\mathbf{Z}_{1:k})$  can be obtained at time step  $k$  from its particle approximation at the previous time step  $k-1$ . Using this particle approximation, both the number of sources in a given region and the sources' parameters can be estimated. The PF-PHDF method can solve the MEG/EEG inverse problem even when the number of dipoles is unknown. It assumes that the  $m$ th measurement  $z_{k,m}$  is generated from a single-target or false alarm. However, as the MEG/EEG measurements  $\mathbf{z}_k$  in (1) are due to contributions from multiple dipoles, we next consider an approach that first decomposes the MEG/EEG data before estimating the unknown number of dipoles using the PF-PHDF.

## 4 Tracking Neural Activity Using PF-PHDF

The PF-PHDF is robust and computationally inexpensive compared to existing multiple target tracking techniques and has been successfully used in radar and sonar tracking problems [27, 28]. However, there are significant challenges in applying PF-PHDF to solve the EEG/MEG inverse problem. In radar and sonar systems, each measurement is generated from a single-target or false alarm. But in the EEG/MEG model described in Section 2, each sensor measurement is due to the contributions of all the dipoles. Thus, the raw EEG/MEG data must be decomposed initially before it can be applied to the PF-PHDF.

In this section, we propose an algorithm based on the PF-PHDF to estimate both the number of neural dipoles and their parameters. First, we apply an efficient pre-whitening technique to the raw EEG/MEG data to reduce the size of the measurement matrix and simplify the computation. Next, we use a fast independent component analysis (ICA) algorithm to decompose the pre-whitened data. Finally, we use the decomposed EEG/MEG data as input to the PF-PHDF to estimate the number of dipoles and their parameters.

### 4.1 Efficient Pre-whitening Technique

The problem of decomposing the EEG/MEG measurement can be simplified by performing a preliminary pre-whitening of the EEG/MEG measurement  $\mathbf{z}_k = [z_{k,1} \ z_{k,2} \ \dots \ z_{k,M}]^T$ . The observed EEG/MEG data is linearly transformed to a vector  $\mathbf{v}_k = \mathbf{U}\mathbf{z}_k = \mathbf{U}\mathbf{A}_k\mathbf{s}_k$ , where  $\mathbf{U}$  is a linear transformation matrix, whose elements are mutually uncorrelated and with unit variance,  $E[\mathbf{v}_k\mathbf{v}_k^T] = \mathbf{I}$  [21]; here,  $\mathbf{I}$  is the identity matrix. An example of such a linear transformation can be obtained by eigenvalue decomposition  $\mathbf{U} = \Lambda^{-1/2}\boldsymbol{\varphi}^T$ , where  $\Lambda$  is a diagonal matrix whose elements are the eigenvalues of  $E[\mathbf{Z}\mathbf{Z}^T]$  and the columns of  $\boldsymbol{\varphi}$  are the corresponding eigenvectors. In particular, assuming  $M$  EEG/MEG measurement sensors, the first step is to find the eigenvalues of the  $M \times M$  covariance matrix  $\boldsymbol{\Sigma}_{\text{cov}} = E[\mathbf{Z}\mathbf{Z}^T]$ .

There are many algorithms to calculate eigenvectors from a covariance matrix, but when the size  $M > 5$ , most of them cannot be easily mapped into an efficient VLSI architecture. For EEG/MEG systems, the number of sensors is typically between 30 and 150. In this case, the challenge is to find an efficient approach to calculate the eigenvalues of a large matrix. In our previous work [1], we used a channel decomposition method to solve this problem. The channels were divided into several groups and ICA was applied to each group. The method resulted in some amount of loss in tracking performance. In addition, in spite of the reduction in computational cost, this step was still the bottleneck of our implementation.

In this paper we take a different approach, and assume that only a small set of patches of the human brain are activated at a time [7, 10, 24]. Under this assumption, the number of dipoles  $N_d$  is much smaller than the number of sensors  $M$ ,  $N_d \ll M$ . Thus, instead of calculating all the eigenvalues and eigenvectors, we only need to find the leading eigenvalues and eigenvectors corresponding to the  $N_d$  active dipoles. Here we use the eigenvector distilling algorithm [29] to find the leading eigenvalues and eigenvectors. As the number of dipoles  $N_d$  is unknown, the number of eigenvalues and eigenvectors to be distilled is unknown as well. We use

an eigenvalue threshold  $V_{thr}$  to pick the number of eigenvalues to be computed. The influence of this threshold on algorithm performance is analyzed in Section 7.

#### 4.2 Measurement Decomposition

After the pre-processing of the raw data, a new data vector  $\mathbf{v}_k = \mathbf{U}\mathbf{z}_k = \mathbf{U}\mathbf{A}_k\mathbf{s}_k$  with reduced dimensionality and noise power is obtained. Since distinct neural sources are mutually independent [24], the ICA technique can be used to decompose the pre-processed EEG/MEG data. The FastICA algorithm [21] identifies independent components in data using a fixed-point iteration procedure. Using FastICA, the estimated independent sources  $\hat{\mathbf{s}}_k = [\hat{s}_{k,1}, \dots, \hat{s}_{k,J}]^T$  can be obtained as  $\hat{\mathbf{s}}_k = \mathbf{W}^T\mathbf{v}_k$ , where  $\mathbf{W} = \mathbf{U}\mathbf{A}_k$  is called the demixing matrix which can be calculated using FastICA and  $J$  is the number of independent components. The estimated mixing matrix  $\hat{\mathbf{A}}_k = [\hat{\mathbf{a}}_{k,1}^T \dots \hat{\mathbf{a}}_{k,J}^T]$  can be obtained as  $\hat{\mathbf{A}}_k = \mathbf{U}^{-1}\mathbf{W}$ , where  $\hat{\mathbf{a}}_{k,j}$  is the  $j$ th column of  $\hat{\mathbf{A}}_k$ .  $\hat{\mathbf{a}}_{k,j}$  is then used to obtain  $\hat{\mathbf{z}}_{k,j}$ , the EEG/MEG signal contributed by the  $j$ th individual source as  $\hat{\mathbf{z}}_{k,j} = \hat{\mathbf{a}}_{k,j}\hat{s}_{k,j}$ . Thus this method can be used to decompose the mixed EEG/MEG measurement  $\mathbf{z}_k$  to a set of unmixed measurement vectors  $\hat{\mathbf{z}}_k = [\hat{z}_{k,1}, \dots, \hat{z}_{k,J}]$  where each new measurement  $\hat{z}_{k,j}$  corresponds to a single dipole or false alarm. These decomposed EEG/MEG measurements are then used as the input of the PF-PHDF to find the number of dipole sources and their states.

#### 4.3 Multi-dipole Estimation Using PF-PHDF

In order to formulate the PHDF for the dipole source estimation problem, we first need to describe an RFS model for the time evolution of the multiple dipole source state and an RFS model for the measurements. Specifically, given the dipole source state RFS  $\mathbf{X}_{k-1}$  at time step  $k-1$ , the dipole source state RFS  $\mathbf{X}_k$  is formed by combining (a) dipole sources still present from the previous time step,  $\mathbf{X}_{k|k-1}^{\text{prev}}$ , (b) dipole sources that are new at the present time step,  $\mathbf{X}_k^{\text{new}}$ , and (c) dipole sources spawning from sources from the previous time step,  $\mathbf{X}_{k|k-1}^{\text{spn}}$ . The new measurement RFS  $\hat{\mathbf{Z}}_k = \{\hat{\mathbf{z}}_{k,1}, \hat{\mathbf{z}}_{k,2}, \dots, \hat{\mathbf{z}}_{k,J}\}^T$  is the decomposed EEG/MEG signal after preprocessing. The state-space model for a single dipole source is given by

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \boldsymbol{\nu}_k, \quad (11)$$

$$\hat{\mathbf{z}}_k = h(\mathbf{x}_k) + \boldsymbol{\xi}_k, \quad (12)$$

where  $\mathbf{x}_k \in \mathbf{X}_k$  and  $\hat{\mathbf{z}}_k \in \hat{\mathbf{Z}}_k$ .  $\boldsymbol{\nu}_k$  and  $\boldsymbol{\xi}_k$  are the processing noise and measurement noise, respectively. Then the steps of PF-PHDF for multiple dipole sources tracking problem can be represented as follows.

##### Algorithm PF-PHDF:

**Step 1-Initialization:** Draw particles  $x_0^{(\ell)}, \ell = 1, \dots, N_0$  from the initial intensity  $\zeta(\mathbf{x}_0|\hat{\mathbf{Z}}_0)$ , where  $N_0 = N_{d_0} \times N$ ,  $N_{d_0}$  is the initial number of dipoles and  $N$  is the number of particles used for each dipole. Assign particle weights  $w_0^{(\ell)} =$

$N_{d_0}/N_0, \ell = 1, \dots, N_0$ .

**Step 2-Prediction:** For existing dipole sources, sample  $\tilde{x}_k^{(\ell)}$ ,  $\ell = 1, \dots, N_{k-1}$  from an importance density  $q_k(\cdot|x_{k-1}^{(\ell)}, \hat{\mathbf{Z}}_k)$  and evaluate the predicted weights

$$\tilde{w}_{k|k-1}^{(\ell)} = \frac{\left( \Pr_{k|k-1}(\mathbf{x}_{k-1}^{(\ell)}) p(\mathbf{x}_k^{(\ell)}|\mathbf{x}_{k-1}^{(\ell)}) \right)}{q_k(\mathbf{x}_k^{(\ell)}|\mathbf{x}_{k-1}^{(\ell)}, \hat{\mathbf{Z}}_k)} w_{k-1}^{(\ell)}.$$

Here, we assume no dipole spawning. For convenience, we choose the importance density  $q_k(\cdot|x_{k-1}^{(\ell)}, \hat{\mathbf{Z}}_k)$  to be the transition probability density  $p(\cdot|x_{k-1}^{(\ell)})$ . As a result,  $\tilde{w}_{k|k-1}^{(\ell)} = \Pr_{k|k-1}(\mathbf{x}_{k-1}^{(\ell)}) w_{k-1}^{(\ell)}$ . For the newborn dipole, sample  $\tilde{x}_k^{(\ell)}$ ,  $\ell = N_{k-1} + 1, \dots, N_{k-1} + N_{ext}$  from another importance density  $\kappa_k(\cdot|\hat{\mathbf{Z}}_k)$ . Compute the weights for the newborn particles

$$\tilde{w}_{k|k-1}^{(\ell)} = \frac{1}{N_{ext}} \frac{\zeta(\tilde{\mathbf{x}}_k^{new}|\hat{\mathbf{Z}}_k)}{\kappa_k(\tilde{\mathbf{x}}_k^{(\ell)}|\hat{\mathbf{Z}}_k)}.$$

Here, we choose  $\kappa_k(\cdot|\hat{\mathbf{Z}}_k)$  equal to  $\frac{1}{a} \zeta(\tilde{\mathbf{x}}_k^{new}|\hat{\mathbf{Z}}_k)$ , thus  $\tilde{w}_{k|k-1}^{(\ell)} = \frac{a}{N_{ext}}$ , where  $a$  is a constant.

**Step 3-Update:** Set  $R_k = N_{k-1} + N_{ext}$  and update the weights  $\tilde{w}_k^{(\ell)}$ ,  $\ell = 1, \dots, R_k$  based on Equation (4) as

$$\tilde{w}_k^{(\ell)} = [1 - \Pr_k^{\det}(\mathbf{x}_k^{(\ell)}) + \sum_{j=1}^J \frac{\Pr_k^{\det}(\mathbf{x}_k^{(\ell)}) p(\hat{\mathbf{z}}_{k,j}|\mathbf{x}_k^{(\ell)})}{\zeta(\mathbf{z}_k^{\text{clt}}) + C_k(\hat{\mathbf{z}}_{k,j})}] \tilde{w}_{k|k-1}^{(\ell)}, \quad (13)$$

$$C_k(\hat{\mathbf{z}}_{k,j}) = \sum_{\ell=1}^{R_k} \Pr_k^{\det}(\mathbf{x}_k^{(\ell)}) p(\hat{\mathbf{z}}_{k,j}|\mathbf{x}_k^{(\ell)}) \tilde{w}_{k|k-1}^{(\ell)}, \quad (14)$$

where  $p(\hat{\mathbf{z}}_k|\mathbf{x}_k)$  is the likelihood and  $\Pr_k^{\det}(\mathbf{x}_k)$  is the probability of state detection. In this paper, we assume no clutter and set  $\zeta(\mathbf{z}_k^{\text{clt}}) = 0$ .

**Step 4-Resampling:** Estimate the number of dipoles  $\hat{N}_{d_k} = \sum_{\ell=1}^{R_k} \tilde{w}_k^{(\ell)}$  and resample  $\{\tilde{x}_k^{(\ell)}, \tilde{w}_k^{(\ell)}\}$ ,  $\ell = 1, \dots, R_k$  to get  $\{x_k^{(\ell)}, w_k^{(\ell)}\}$ ,  $\ell = 1, \dots, N_k$ , where  $N_k = N \cdot \text{round}(\hat{N}_{d_k})$  and  $\text{round}(a)$  denotes the nearest integer to  $a$ . After resampling, each of the new particles has weight  $\hat{N}_{d_k}/N_k$ .

**Step 5-Estimation of the dipole state:** Cluster the resampled particles and estimate the state parameters. The clustering is performed in a 3-dimensional space using the k-means clustering algorithm.

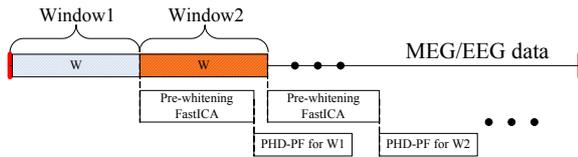
In this algorithm, the number of particles changes over time and is proportional to the number of dipoles, i.e., at time  $k$ ,  $N_k \propto \hat{N}_{d_k}$ . Unlike standard PF, there is a summation among sub-measurements  $\hat{\mathbf{z}}_{k,j}$ ,  $j = 1, \dots, J$  when updating the weights in Step 3. These independent sub-measurements correspond to the individual sources. After Step 3, the posterior intensity  $\zeta(\mathbf{x}_k|\hat{\mathbf{Z}}_k)$  at time  $k$  is approximated by a set of weighted particles  $\{\tilde{x}_k^{(\ell)}, \tilde{w}_k^{(\ell)}\}$ ,  $\ell = 1, \dots, R_k$  which contains all the available information about the dipoles. For example, the number of dipoles

can be obtained by computing the integral of the posterior intensity, which is equal to the summation of the weights  $\sum_{\ell=1}^{R_k} \tilde{w}_k^{(\ell)}$  and the estimate of dipole parameters can be found from the peaks of the intensity. In the resampling step, the new weights  $w_k^{(\ell)}$  are not normalized to 1, but sum to  $\hat{N}_{d_k}$ , the estimated number of dipoles.

#### 4.4 Window Processing

Most of the existing processing methods process EEG/MEG data off-line after all the data has been collected [30]. In order to process the data on-line, we window the EEG/MEG data, and apply the proposed algorithm to the data in each window as shown in Figure 2. In the first step, we acquire the EEG/MEG data in window 1 and store the data in memory. Then, while obtaining data in window 2, we pre-whiten and run FastICA on the data from window 1.

The window length  $L_w$  is an important parameter because the processing time and tracking accuracy both depend on  $L_w$ . If  $L_w$  is small, the computations take less time. However, the data in a short window is not able to statistically represent the neural activity, and as a result, the estimation error is larger as shown later in Table 3. In addition, in this scheme, the window length has to be larger than the execution time for pre-whitening and FastICA as shown in Figure 2.



**Fig. 2** Window processing of EEG/MEG data.

## 5 Sensor scheduling

In this section, we propose a sensor scheduling algorithm that enables optimum use of sensor resources for neural activity tracking. Such a method can lead to good tracking performance with fewer number of active sensors. Since the power consumption of each EEG sensor is about 10 mW [31, 32], sensor scheduling can reduce the total power consumption, making wearable EEG devices common place. Sensor scheduling is a closed-loop feedback optimization procedure that allows adaptive selection of the sensors to be used for obtaining measurements at each time-step in order to optimize the cost function of interest. Here we consider two cost functions: (a) the predicted mean-squared error (PMSE) in estimation (which is to be minimized), and (b) the signal-to-noise ratio (SNR) of the measurements (which is to be maximized), and optimize them with respect to different sensor configurations.

### 5.1 Sensor scheduling based on minimization of PMSE

Minimum the PMSE has been used as a cost function for sensor scheduling [33, 34]. The PMSE in dipole state estimation at time  $k$  can be written as

$$\begin{aligned} \mathcal{J}^P(\mathbf{y}_k) &= E_{\mathbf{x}_k, \mathbf{z}_k} [(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T] \\ &= \iint (\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T p(\mathbf{z}_k | \mathbf{x}_k, \mathbf{y}_k) p(\mathbf{x}_k | \mathbf{z}_{k-1}, \mathbf{y}_{k-1}) d\mathbf{x}_k d\mathbf{z}_k, \end{aligned} \quad (15)$$

where  $\mathbf{x}_k$  denotes the dipole state at time  $k$  and  $\mathbf{y}_k$  is the sensor configuration at time  $k$ . Here, we consider the sensor configuration problem with  $M$  sensors. The sensor configuration vector  $\mathbf{y}_k = [y_{k,1} \ y_{k,2} \ \dots \ y_{k,M}]^T$  is comprised of binary values  $y_{k,m} \in \{0, 1\}$ , with  $y_{k,m} = 1$  indicating that the  $m$ th sensor is selected and  $y_{k,m} = 0$  indicating that the  $m$ th sensor is not selected. The measurement  $\mathbf{z}_k$  is related to  $\mathbf{y}_k$  and  $\mathbf{z}_k(\mathbf{y}_k) = \mathbf{z}_{k,\mathcal{M}}$ , where  $\mathcal{M} = \{m : y_{k,m} = 1\}$ .  $\hat{\mathbf{x}}_k$  is the estimate of  $\mathbf{x}_k$  at time  $k$  computed using the measurement  $\mathbf{z}_k$ . Using Monte Carlo integration,  $\mathcal{J}^P(\mathbf{y}_k)$  can be approximated as

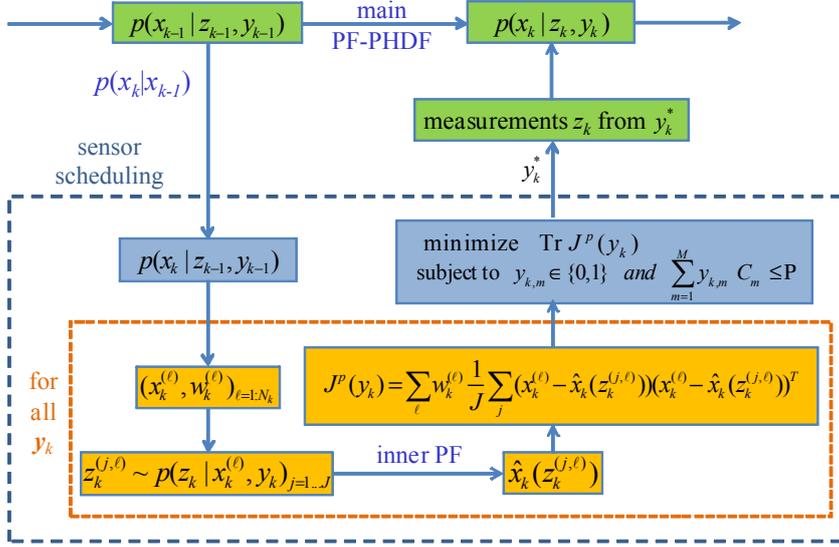
$$\mathcal{J}^P(\mathbf{y}_k) \approx \sum_{\ell=1}^{N_k} w_k^{(\ell)} \frac{1}{J} \sum_{j=1}^J (\mathbf{x}_k^{(\ell)} - \hat{\mathbf{x}}_k(\mathbf{z}_k^{(j,\ell)})) (\mathbf{x}_k^{(\ell)} - \hat{\mathbf{x}}_k(\mathbf{z}_k^{(j,\ell)}))^T, \quad (16)$$

where  $\mathbf{x}_k^{(\ell)}$  and  $w_k^{(\ell)}$ ,  $\ell = 1, \dots, N_k$ , are the particles and weights in the PF representation of the predicted state distribution  $p(\mathbf{x}_k | \mathbf{z}_{k-1}, \mathbf{y}_{k-1})$ ,  $\mathbf{z}_k^{(j,\ell)} \sim p(\mathbf{z}_k | \mathbf{x}_k^{(\ell)}, \mathbf{y}_k)$ ,  $j = 1, \dots, J$ , are Independent and identically distributed samples drawn from the measurement model, and  $\hat{\mathbf{x}}_k(\mathbf{z}_k^{(j,\ell)})$  is the estimate of  $\mathbf{x}_k$  computed using the predicted measurement  $\mathbf{z}_k^{(j,\ell)}$  with a secondary PF. The approximation error is small for large  $N_k$  and  $J$ . The PMSE  $\mathcal{J}^P(\mathbf{y}_k)$  can now be optimized by searching through all sensor configurations. We impose a constraint to limit the total power consumption to  $\mathcal{P}$  as shown below

$$\sum_{m=1}^M y_{k,m} C_m \leq \mathcal{P}, \quad (17)$$

where  $C_m$  is the power consumption of the  $m$ th sensor. A block diagram of the dipole state estimation and sensor scheduling method is shown in Figure 3. The key steps of the sensor scheduling algorithm, performed at each time  $k$ , are as follows:

1. **State prediction:** Compute the distribution  $p(\mathbf{x}_k | \mathbf{z}_{k-1}, \mathbf{y}_{k-1})$  of the predicted state at time  $k$  by propagating the PF-PHDF posterior distribution  $p(\mathbf{x}_{k-1} | \mathbf{z}_{k-1}, \mathbf{y}_{k-1})$  at time  $k-1$  using the state evolution model  $p(\mathbf{x}_k | \mathbf{x}_{k-1})$ .
2. **Measurement sampling:** For each particle  $\mathbf{x}_k^{(\ell)}$  and sensor configuration  $\mathbf{y}_k$ , draw predicted measurements  $\mathbf{z}_k^{(j,\ell)}$  from the measurement distribution  $p(\mathbf{z}_k | \mathbf{x}_k^{(\ell)}, \mathbf{y}_k)$ .
3. **Inner estimation:** For each measurement  $\mathbf{z}_k^{(j,\ell)}$ , estimate the state  $\hat{\mathbf{x}}_k(\mathbf{z}_k^{(j,\ell)})$  using a secondary PF.
4. **Error calculation:** Calculate the PMSE  $\mathcal{J}^P(\mathbf{y}_k)$  using the approximation in Equation (16).
5. **Optimization:** Find the optimum sensor configuration  $\mathbf{y}_k^*$  which minimizes the PMSE under the power constraint in Equation (17).



**Fig. 3** Block diagram of the state estimation and sensor optimization method.

The scheduled sensor configuration  $\mathbf{y}_k^*$  is then used to obtain the measurement  $\mathbf{z}_k(\mathbf{y}_k^*)$  at time  $k$  and estimate the state  $\hat{\mathbf{x}}_k$  using the PF-PHDF algorithm.

For  $M$  sensors, there can be  $R = 2^M$  possible sensor configurations, making the optimization problem very difficult to solve directly unless  $M$  is small (e.g.  $M = 10$ ). For large  $M$ , the current best methods for solving such problems include branch and bound methods [35] (which provide global optima at the expense of computational cost) or other convex relaxation based techniques [36] (which are fast but only provide approximate solutions). Even so, sensor scheduling based on the direct computation and optimization of the PMSE remains computationally demanding and not suitable for real-time implementation.

## 5.2 Sensor scheduling with Maximum SNR

We now describe a sensor scheduling algorithm that relies on maximization of the signal-to-noise ratio (SNR) of the measured sensor data. This approach can potentially lead to significant reduction in the number of required sensors for accurate tracking performance, with a computational complexity that is much lower than the minimum PMSE based sensor scheduling method discussed earlier.

The proposed method is as follows. Suppose that at some time step  $k$  the parameters of the dipole sources  $(\mathbf{r}_k, \mathbf{q}_k, \mathbf{s}_k)$  are given. From Equations (1) and (2),

the SNR of the measurement from the  $m$ th sensor can be represented by

$$\text{SNR}_{k,m} = \frac{(\mathbf{A}_{k,m} \mathbf{s}_k)^2}{\sigma_n^2} = \frac{(\sum_{j=1}^{N_d} a_{k,m,j} s_{k,j})^2}{\sigma_n^2} \propto \left( \sum_{j=1}^{N_d} a_{k,m,j} s_{k,j} \right)^2, \quad (18)$$

where  $\mathbf{A}_{k,m}$  is the  $m$ th row of the gain matrix,  $\sigma_n^2$  is the variance of the measurement noise and  $N_d$  is the number of dipole sources. After ICA, the measurement is decomposed to independent components and each component corresponds to an individual dipole source. The SNR of the  $m$ th sensor for the  $j$ th dipole source can thus be assumed to depend on

$$\text{SNR}_{k,m,j} \propto (a_{k,m,j} s_{k,j})^2, \quad (19)$$

where  $a_{k,m,j}$  is a function of the distance  $d_{k,m,j}$  between the  $m$ th sensor and  $j$ th dipole source. Figure 4 shows the relationship between  $|a_{k,m,j}|$  and  $d_{k,m,j}$ . From Figure 4 we can see that as  $d_{k,m,j}$  increases  $|a_{k,m,j}|$  decreases and so does the corresponding SNR. Since the MSE is expected to be decrease with increasing SNR measurements, using sensors with smaller  $d_{k,m,j}$  can provide better neural tracking performance. The sensors are therefore scheduled by adaptively selecting the ones with smallest  $d_{k,m,j}$  to form the final sensor configuration used for obtaining the measurements. The steps of the sensor scheduling method, performed at each time  $k$ , are as follows:

1. **State prediction:** Calculate the predicted state of the dipole source  $\tilde{\mathbf{x}}_k$  at time  $k$  using the particles  $x_{k-1}^{(\ell)}$  and weights  $w_{k-1}^{(\ell)}$  at time  $k-1$  based on the state model

$$\tilde{\mathbf{x}}_k = \sum_{\ell=1}^{N_k} p_{x_{k-1}|x_k}(x_{k-1}^{(\ell)}) w_{k-1}^{(\ell)},$$

where  $N_k$  is the number of particles and  $p_{x_{k-1}|x_k}(\cdot)$  is the state updating equation. Extract the predicted dipole location  $\tilde{\mathbf{r}}_k$  from  $\tilde{\mathbf{x}}_k$ .

2. **Distance calculation:** For each sensor, calculate the distance between the sensor location  $\mathbf{r}_m$  and the predicted dipole source location  $\tilde{\mathbf{r}}_k$  as

$$\tilde{d}_{k,m} = |\mathbf{r}_m - \tilde{\mathbf{r}}_k|, \quad m = 1, \dots, M,$$

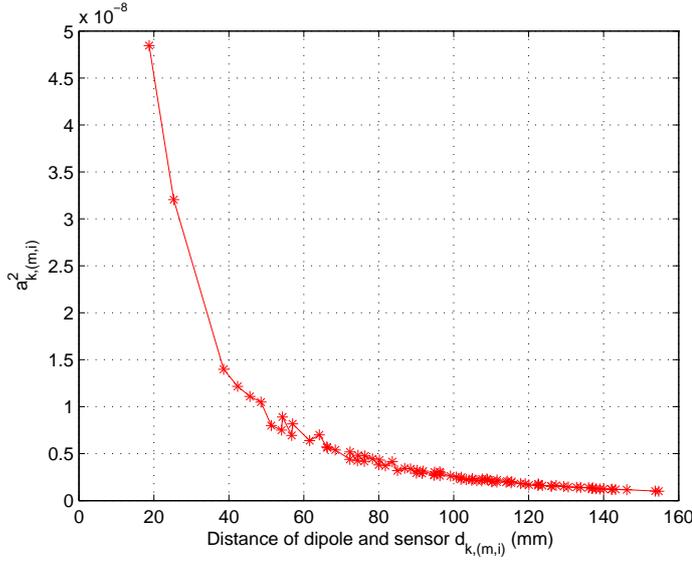
where  $|\cdot|$  denotes Euclidean distance.

3. **Optimization:** Sort the sensors in increasing order of  $\tilde{d}_{k,m}$ . Choose the first  $\mathfrak{N}_s$  sensors to estimate the dipole states, where  $\mathfrak{N}_s$  is the number of sensors to be used depending on power constraint in Equation 17.

As before, the scheduled sensor configuration  $\mathbf{y}_k^*$  is then used to obtain the measurement  $\mathbf{z}_k(\mathbf{y}_k^*)$  at time  $k$  and estimate the state  $\hat{\mathbf{x}}_k$  using the PF-PHDF algorithm.

## 6 Hardware Implementation

In this section, we propose efficient hardware architectures for the PF-PHDF and sensor scheduling algorithm described in Section 4.3 and Section 5.2, respectively. Pre-whitening and FastICA in Section 4 can be implemented using the architecture in [29, 37] and are not described here.



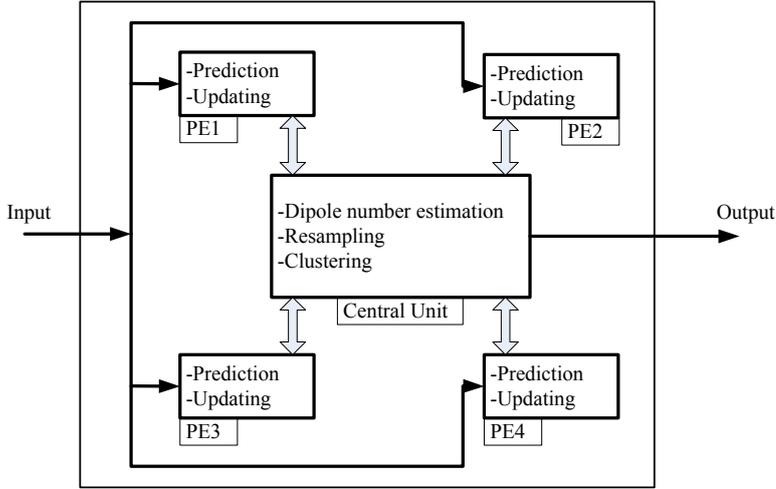
**Fig. 4** Amplitude of the EEG sensor signal as a function of the distance between the sensor and dipole source.

### 6.1 Architecture for PF-PHDF

The proposed PF-PHDF algorithm described in Section 4.3 is mapped onto the parallel architecture shown in Figure 5. It consists of  $\eta$  processing elements (PE) and one central unit (CU) connected by a global bus. Local processing steps, such as initialization (Step 1), prediction (Step 2) and part of updating (Step 3) are conducted in each PE. Global processing steps, such as computing normalization factors  $C_k$ , estimating the number of dipoles (Step 3), resampling (Step 4) and clustering (Step 5), are executed in the CU. Each PE communicates with the CU, but there is no communication among PEs.

The operation flow for then PF-PHDF is shown in Figure 6. Each PE processes  $R_k/\eta$  particles, where  $N_k$  is the number of particles at time step  $k$ . In each PE, first, the particles  $\mathbf{x}_{k-1}^{(\ell)}$  are processed at the prediction unit to generate the predicted particles  $\tilde{\mathbf{x}}_k^{(\ell)}$  and the predicted weights  $\tilde{w}_{k|k-1}^{(\ell)}$  based on Step 2. Next, the likelihoods  $p(\hat{\mathbf{z}}_{k,j}|\mathbf{x}_k^{(\ell)})$  are calculated for each individual measurement and particle. Since the calculations of  $p(\hat{\mathbf{z}}_{k,j}|\mathbf{x}_k^{(\ell)})$  for different individual measurements are independent, they are processed in parallel. Then, the  $i$ th PE sums up the likelihoods for all the particles  $\psi_i(\hat{\mathbf{z}}_{k,j}) = \sum_{\ell} p(\hat{\mathbf{z}}_{k,j}|\mathbf{x}_k^{(\ell)})$  and transmits  $\psi_i(\hat{\mathbf{z}}_{k,j})$  to the CU. After obtaining  $\{\psi_i(\hat{\mathbf{z}}_{k,j})\}_{i=1}^{\eta}$ , the CU calculates the normalization factor,  $C_k(\hat{\mathbf{z}}_{k,j}) = \sum_i \psi_i(\hat{\mathbf{z}}_{k,j})$ , and sends it back to each PE. Next, each PE computes the final weights  $\tilde{w}_k^{(\ell)}$  based on Equation (13).

Since the resampling step is operated in the central unit, the weights of all the particles have to be transferred from the PEs to the central unit, which results in a large communication overhead. In order to reduce this overhead, we employ



**Fig. 5** Block diagram of parallel PF-PHDF architecture with four processing elements.

the grouping method in [38] and add the group-and-mean unit in each PE. The main idea is as follows. The particles  $\tilde{\mathbf{x}}_k^{(\ell)}$ , and their corresponding weights  $\tilde{w}_k^{(\ell)}$ , are divided into  $G$  groups based on the range of the particles. The averages of each group,  $x_{k,\text{mean}}^{(g)}$ ,  $g = 1, \dots, G$ , are used as the new particles, and only the average weights  $w_{k,\text{mean}}^{(g)}$ ,  $g = 1, \dots, G$ , are transmitted to the central unit to be used as input to the resampling step. These particles are stored in the mean particle memory (MPMEM) for future use. Before the resampling step, we estimate the number of dipole sources by summing the final weights,  $\hat{N}_{d_k} = \sum_{\ell} \hat{w}_k^{(\ell)}$ . During the resampling step, the replication index  $\rho$  is calculated at the central unit based on the average weights. After the resampling step, the group averaged particles are read from MPMEM and sent to the prediction unit for computations in the next iteration. These particles are also sent to the clustering unit to estimate the dipole parameters. This procedure significantly reduces the communication between the PEs and the central unit.

The hardware resource for each block in Figure 6 is shown in Table 1. Note that the likelihood computation unit uses the largest number of resources.

**Table 1** Hardware operators for each block in Figure 6.

Unit	Block	+	×	÷	√	exp
Processing element	Prediction	6	0	0	0	0
	Likelihood	88	93	1	2	1
	Group mean	52	0	1	0	0
	Final weight	6	1	1	0	0
Central unit	Global	6	0	0	0	0
	Normal	9	0	0	0	0
	Resampling	2	3	1	0	0
	Clustering	6	3	1	1	0

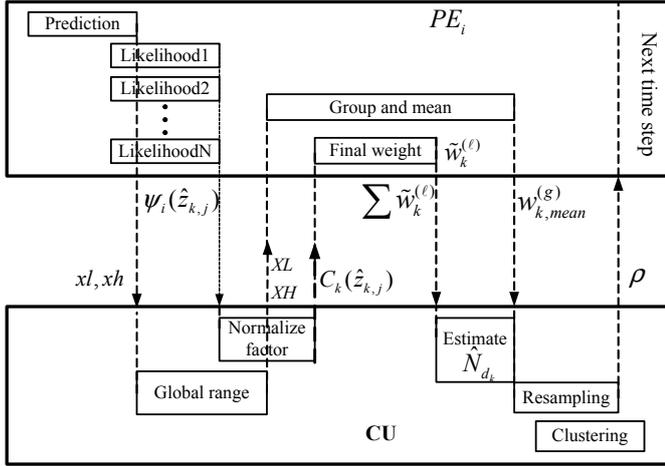


Fig. 6 Operation flow and communication between PE and CU at each time step.

## 6.2 Architecture for Sensor Scheduling

The overall block diagram of the hardware architecture for sensor scheduling is shown in Figure 7. At each time step  $k$ , we use the PF-PHDF to obtain the predicted dipole location  $\hat{\mathbf{r}}_k$ , and feed  $\hat{\mathbf{r}}_k$  as input to the sensor scheduling unit. First, the square of the predicted distance  $\hat{d}_{k,m}^2$  is calculated using three subtractors, three multipliers and two adders. Next, the sensors are sorted based on the predicted distance  $\hat{d}_{k,m}^2$  in the SORT unit. After sorting, the top  $n$  sensors are picked as the optimized sensor configuration for the next time step. Since we only need to find  $n$  sensors with the smallest  $\hat{d}_{k,m}^2$ , we maintain a sorted list of  $n$  elements and use the insertion-deletion sort algorithm [39] to update the list.

The architecture of the SORT unit is shown in Figure 8. It consists of  $n$  processors, one adder tree that adds  $n$  1-bit numbers and one rank register. All processors consist of one data register to store the distance value, one comparator to compare with the new distance and one rank register to store the parameter used to calculate the rank of new distance. The rank of the new distance is calculated by the adder tree and stored in the new rank register.

## 7 Simulation and FPGA Implementation Results

We now demonstrate the tracking performance of the proposed algorithm using synthetic EEG data sets and present FPGA implementation results.

### 7.1 Neural Activity Tracking Results

#### 7.1.1 Simulation Setup

In order to demonstrate the tracking performance of the proposed algorithm, we use the synthetic data from a previous study in [11] with three dipoles located

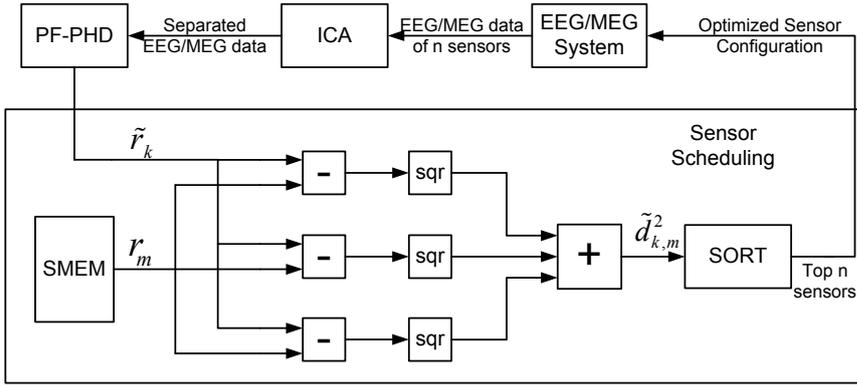


Fig. 7 Overall architecture of sensor scheduling.

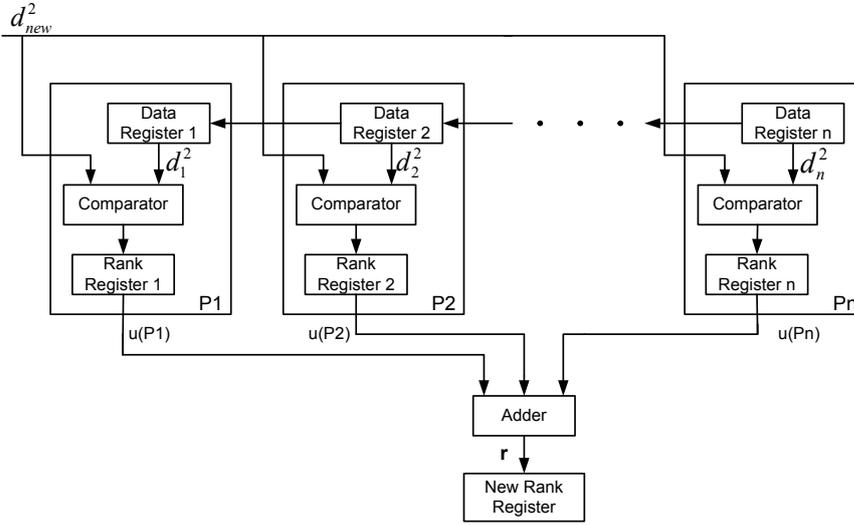


Fig. 8 Architecture of the SORT unit.

at  $V1$  (1.11, -5.34, 4.98),  $V5_R$  (4.36, -3.68, 4.44) and  $V5_L$  (-3.37, -4.85, 4.81) in an example where the brain volume hemisphere has a radius of 10 cm. The measurement noise is Gaussian with zero mean and standard deviation  $\sigma = 10^{-5}$ . For this simulation, we used uniformly distributed particles for each dipole. For a three dipole system, the maximum number of particles is  $3N + N_{\text{ext}} = 6,400$ , which is much less than the 100,000 particles used in [11]. The dipole state transition model in Equation (11) is a random walk model with Gaussian transition kernel,  $p(\mathbf{r}_k|\mathbf{r}_{k-1}) = \mathcal{N}(\mathbf{r}_{k-1}, \sigma_r)$  and  $p(\mathbf{d}_k|\mathbf{d}_{k-1}) = \mathcal{N}(\mathbf{d}_{k-1}, \sigma_d)$ , with  $\sigma_r=1$  cm and  $\sigma_d=2$  nA. Each existing dipole has a probability of survival  $\Pr_{k|k-1}(\mathbf{x}_{k-1})=0.8$  and a probability of detection  $\Pr_k^{\text{det}}(\mathbf{x}_k)=0.95$ .

### 7.1.2 Parameter Selection

The eigenvalue threshold  $V_{thr}$  described in Section 4.1 and window length  $L_w$  described in Section 4.4 are crucial parameters as they greatly impact the estimation results. Table 2 and Table 3 show the reconstruction RMSE with respect to different  $V_{thr}$  and  $L_w$ , respectively. The reconstruction RMSE is defined as the difference of estimated individual EEG data  $\hat{\mathbf{z}}_{k,j}$  and the true EEG data  $\mathbf{z}_{k,j}$ .

$$RMSE_j^{rec} = \sqrt{\frac{1}{K} \sum_{k=1}^K \frac{1}{M} (\mathbf{z}_{k,j} - \hat{\mathbf{z}}_{k,j})^T (\mathbf{z}_{k,j} - \hat{\mathbf{z}}_{k,j})},$$

where  $K$  is the number of time steps and  $M$  is the number of sensors. From Table 2, we can see that as the threshold value decreases, the RMSE increases and that there is a significant RMSE degradation when the threshold is smaller than 150. In addition, with thresholds smaller than 100, we can hardly distinguish dipole signals with noise which causes the FastICA algorithm to fail. Based on these results, we choose the eigenvalue threshold  $V_{thr}$  as 500 shown in Figure 9. From Table 3 we can see that if the window length is too long (much longer than the duration of dipole), the independent component analysis cannot capture the changing information of the dipole. As a result, the reconstruction error is large. If the window length is too small, the samples in the window cannot statistically represent the whole data, which also leads to larger reconstruction error. Based on these results, we choose a window length  $L_w = 100$  samples with sampling rate 1 kHz.

**Table 2** RMSE of reconstructed EEG data for different threshold values.

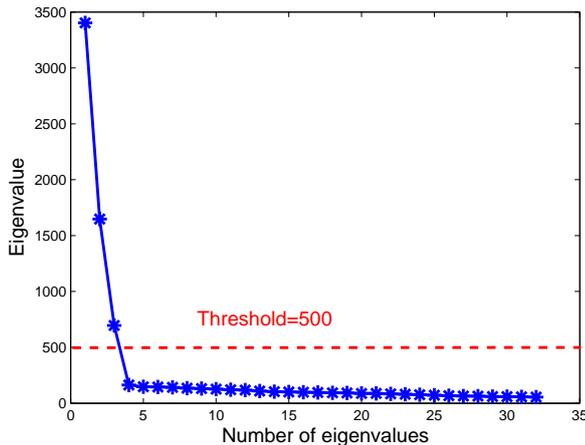
Threshold	500	150	130	100	50
Dipole1	2.68 $\mu V$	2.91 $\mu V$	4.97 $\mu V$	7.45 $\mu V$	NaN
Dipole2	2.67 $\mu V$	2.87 $\mu V$	5.08 $\mu V$	8.35 $\mu V$	NaN
Dipole3	2.53 $\mu V$	2.79 $\mu V$	4.16 $\mu V$	7.95 $\mu V$	NaN

**Table 3** RMSE of reconstructed EEG data for different window lengths.

Window length	50	100	250	500
Dipole1	2.78 $\mu V$	2.16 $\mu V$	2.33 $\mu V$	4.22 $\mu V$
Dipole2	2.68 $\mu V$	2.21 $\mu V$	2.42 $\mu V$	4.51 $\mu V$
Dipole3	2.60 $\mu V$	2.13 $\mu V$	2.44 $\mu V$	4.13 $\mu V$

### 7.1.3 Simulation results for sensor scheduling

The number of sensors used  $\mathfrak{N}_s$  after sensor scheduling impacts the power consumption of the EEG system and the tracking performance of the proposed algorithm. Table 4 shows the position tracking result in terms of RMSE with respect to  $\mathfrak{N}_s$ . As expected, use of fewer sensors result in higher RMSE but lower sensor power consumption. Thus,  $\mathfrak{N}_s$  should be chosen based on the power constraint



**Fig. 9** Eigenvalues of EEG covariance matrix for synthetic data and the threshold selection.

or the RMSE constraint depending on the application. Since the most important bottleneck of wearable EEG instruments is the battery size [22,23], here  $\mathfrak{N}_s$  is chosen based on the power constraint. In this paper, the power consumption of each wireless EEG sensor is set as  $C_m = 10mV$  [31,32] and the total power constraint is set as  $\mathcal{P} = 150 mW$ . Based on Equation (17),  $\mathfrak{N}_s$  is chosen to be 15.

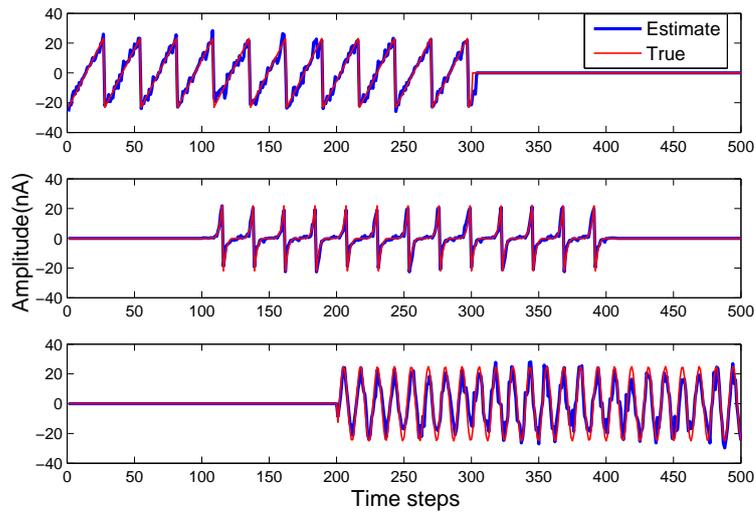
**Table 4** Position RMSE for different number of sensors  $\mathfrak{N}_s$ .

Number of sensors	6	9	15	24	32
RMSE	9.98 mm	6.95 mm	6.41 mm	6.33 mm	6.28 mm

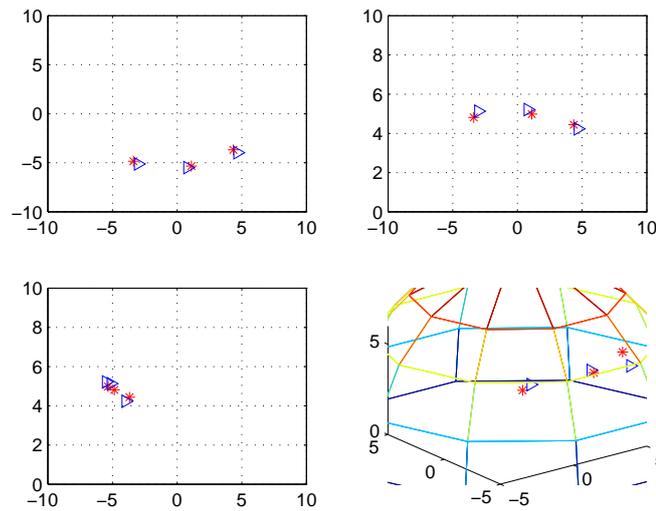
#### 7.1.4 Estimation results with sensor scheduling

In this section, we show the estimation results for synthetic EEG data. Here we choose the eigenvalue selection threshold as 500 and the window length  $L_w = 100$ . We use 2,000 particles for each existing dipole and 400 particles for the newborn dipole. The tracking result for the amplitudes of three dipoles are shown in Figure 10. The RMSE for the dipole current amplitude is 1.83 nA. Figure 11 shows the estimation results for the 3-D location of the three dipoles; the position RMSE is 6.41 mm.

We compare the performance of the proposed tracking algorithm with the results in [8,11,17] and the comparison is shown in Table 5. From Table 5, we can see that the proposed PF-PHDF algorithm has comparable tracking performance, with significantly reduced number of particles (only 6,400 compared to 100,000 in [19]). Furthermore, with sensor scheduling technique, the number of sensors is reduced from 32 to 15, which means about 50% reduction in sensor power consumption.



**Fig. 10** Amplitude estimation results for the three dipoles in the synthetic data using the PF-PHDF algorithm with sensor scheduling.



**Fig. 11** Estimated 3-D locations of dipoles for synthetic EEG data.

## 7.2 Hardware Implementation Evaluation

The proposed hardware architecture is implemented using Verilog HDL and synthesized on the Xilinx Virtex-5 device. The design is verified using Modelsim.

**Table 5** Comparison of neural activity tracking for synthetic data.

Approach	Number of particles	Number of dipoles	Knowledge of dipole number	RMSE of location
PF [17]	100,000	4	Known	5.4 mm
RB-PF [8]	50,000	2	Known	6.3 mm
D-PF [11]	100,000	3	Unknown	6.2 mm
PF-PHDF (with SS)	6,400	3	Unknown	6.4 mm

*Resource Utilization:* Table 6 summarizes the architecture resource utilization for the 4 PE parallel architecture for the PF-PHDF and sensor scheduling (SS). Since the total number of particles is 6,400, each PE processes 1,600 particles. For the likelihood calculation in the updating step, the exponential functions are implemented using CORDIC units, and the rest of the units are implemented using DSP cores.

**Table 6** Resource utilization for PF-PHDF on Xilinx XC5VSX240k

Unit	Occupied slices	Slice Reg.	Slice LUTs	Block Ram	DSP48Es
PF-PHDF	14,291 (39%)	43,637 (30%)	42,383 (29%)	134 (26%)	283 (27%)
SS	623 (1%)	1,714 (1%)	1,792 (1%)	3 (1%)	9 (1%)

*Execution Time:* Figure 12 shows the timing performance for one iteration of the proposed PF-PHDF and SS; the actual number of cycles is given in Table 7. Since some of the computations can be overlapped, the PF-PHDF takes  $N_{\text{PF-PHDF}} = (N_s + N_g + N_{gm} + N_c) = 9,150$  cycles. We choose a system clock rate of 100 MHz and so the processing time for one iteration of is  $T_{\text{PF-PHDF}} = N_{\text{PF-PHDF}} \times T_{\text{clk}} = 91.5 \mu\text{s}$ . Based on the pre-whitening and FastICA implementation in [29,37], the execution time of preprocessing step for 32 sensors is about  $265 \mu\text{s}$  and for 15 sensors is about  $66 \mu\text{s}$ . Thus, for a window with 100 samples, the total processing time without sensor scheduling is  $T_{\text{noSS}} = 265 + 91.5 \times 100 = 9,415 \mu\text{s}$ . Now for a system with sensor scheduling, the total time for one iteration is  $N_{\text{PF-PHDF}} + N_d + N_{\text{sort}} = 96,00$  cycles which translates to  $96 \mu\text{s}$  for 100 MHz clock rate. Thus, the total processing time with sensor scheduling is  $T_{\text{withSS}} = 66 + 96 \times 100 = 9,666 \mu\text{s}$ .

**Table 7** Execution cycles for each block in Figure 12

Unit	$N_s$	$N_l$	$N_g$	$N_n$	$N_w$	$N_{gm}$	$N_k$	$N_r$	$N_c$	$N_d$	$N_{\text{sort}}$
Cycles	1608	1654	5	4	1636	1664	4	53	5873	111	338

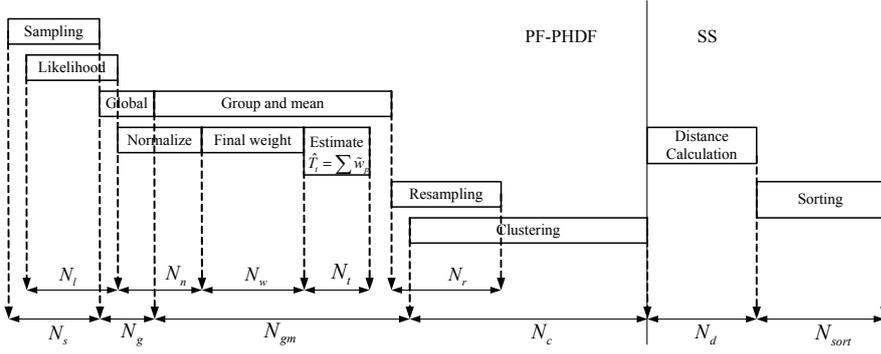


Fig. 12 Execution time breakdown for one PF-PHDF iteration.

### 7.3 Scalability

For the proposed tracking system, the number of particles used for each dipole  $N$  is a critical parameter as it impacts the estimation accuracy and processing time. Figure 13 shows the tracking performance in terms of RMSE for dipole location and the processing time for PF-PHDF as a function of  $N$ . Here we choose the number of PE  $\eta = 4$ . From Figure 13, we can see that as  $N$  increases, the RMSE decreases and the processing time increases. However, when  $N$  is greater than 2000, there is no significant improvement in the RMSE but the processing time increases rapidly. A good tradeoff between RMSE and processing time is obtained at  $N = 2000$ .

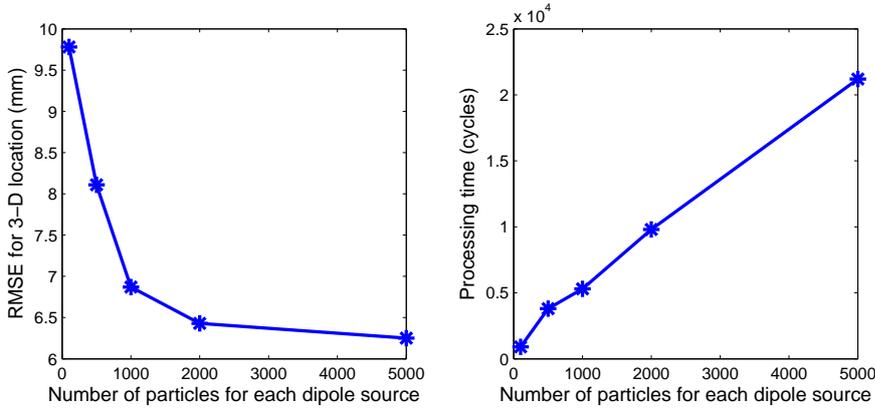
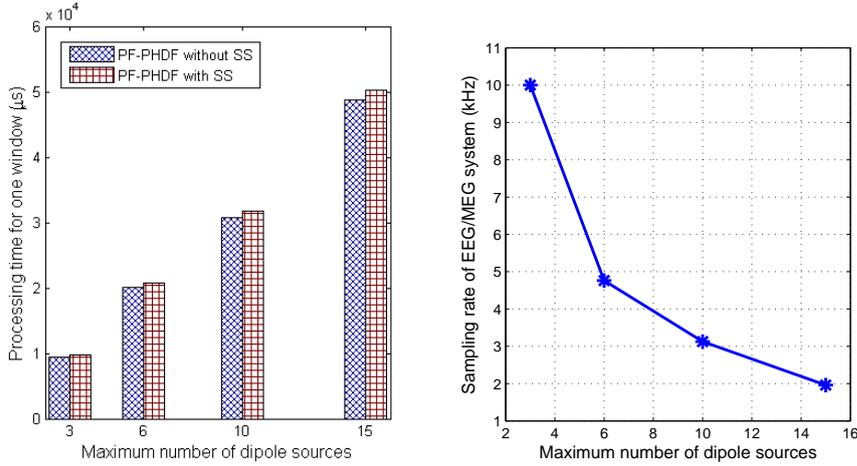


Fig. 13 RSME for location and processing time for PF-PHDF as a function of  $N$ .

In this paper, the maximum number of dipoles is small (less than 5). From the FPGA timing results, we project that if the maximum number of dipoles is 3, the proposed system can perform real-time processing at sampling rates of up to 10 kHz for window length  $L_w = 100$  samples. However, for epilepsy patients, the number of dipoles during seizure can be greater than 10 [40]. Figure 14 shows the timing performance of the proposed system with respect to the maximum num-

ber of dipoles for 2000 particles per dipole and  $\eta = 4$ . From Figure 14, we can see that as the maximum number of dipoles increases, the processing time for a window grows, as expected. Even when the number of dipoles is as large as 15, our system can still support real-time tracking with sampling rate of about 2 kHz. Furthermore, from Figure 14 (a), we can see that the processing time of PF-PHDF with sensor scheduling is slightly increased compared to PF-PHDF without sensor scheduling. However, the number of sensors used with sensor scheduling is significantly reduced (from 32 to 15), which means that the sensor power consumption is about two times lower.



**Fig. 14** Scalability of the proposed system with respect to maximum number of dipoles: Effect on (a) processing time for PF-PHDF with and without SS and (b) maximum sampling rate for real-time processing.

## 8 Conclusion

In this paper, we described the application of PF-PHDF for tracking an unknown number of neural dipole sources using EEG/MEG measurements and proposed an efficient sensor scheduling algorithm that significantly reduces the number of active sensors required. We demonstrated the improved performance of the proposed algorithm using numerical simulations on synthetic EEG data. The position RMSE of the proposed algorithm is 6.4 mm (vs. 6.3 mm) for tracking an unknown number of dipole sources, using significantly fewer number of particles (6,400 particles instead of 100,000 particles) and half the number of sensors (15 out of 32 sensors). The proposed algorithm was implemented on the Xilinx Virtex-5 FPGA platform. The processing time for a window with 100 samples using 6,400 particles was shown to be  $\sim 10$  ms and thus this method can support real-time EEG processing with sampling rates of up to 10 kHz.

## References

1. L. Miao, J. J. Zhang, C. Chakrabarti, A. Papandreou-Suppappola, and N. Kovvali, "Real-time closed-loop tracking of an unknown number of neural sources using probability hypothesis density particle filtering," in *IEEE Workshop on Signal Processing Systems*, October 2011, pp. 367–372.
2. D. Cohen, "Magnetoencephalography: Evidence of magnetic fields produced by alpha rhythm currents," *Science*, vol. 161, pp. 664–666, June 1972.
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, P. S. Lewis, and R. M. Leahy, "Multiple dipole modeling and localization from spatio-temporal MEG data," *IEEE Transactions on Biomedical Engineering*, vol. 39, pp. 541–557, June 1992.
5. 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.
6. B. D. Van Veen and K. Buckley, "Beamforming: A versatile approach to spatial filtering," *IEEE ASSP Magazine*, vol. 5, pp. 4–24, 1988.
7. 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.
8. C. Campi, A. Pascarella, A. Sorrentino, and M. Piana, "A Rao-Blackwellized particle filter for magnetoencephalography," *Inverse Problems and Imaging*, vol. 24, pp. 25–23, 2008.
9. H. R. Mohseni, F. Ghaderi, E. L. Wilding, and S. Sanei, "A beamforming particle filter for EEG dipole source localization," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 337–340.
10. A. Pascarella, A. Sorrentino, C. Campi, and M. Piana, "Particle filtering, beamforming and multiple signal classification for the analysis of MEG time series: A comparison of algorithms," *Inverse Problems and Imaging*, vol. 4, pp. 169–190, February 2010.
11. A. Sorrentino, L. Parkkonen, A. Pascarella, C. Campi, and M. Piana, "Dynamical MEG source modeling with multi-target Bayesian filtering," *Human Brain Mapping*, vol. 30, pp. 1911–1921, June 2009.
12. A. Pascarella and A. Sorrentino, "Statistical approaches to the inverse problem," in *Magnetoencephalography*, E. W. Pang, Ed. InTech, November 2011.
13. C. Campi, A. Pascarella, A. Sorrentino, and M. Piana, "Highly Automated Dipole ESTimation (HADES)," *Computational Intelligence and Neuroscience*, vol. 2011, 2011, article ID 982185.
14. A. Galka, O. Yamashita, T. Ozaki, R. Biscay, and P. Valde, "A solution to the dynamical inverse problem of EEG generation using spatiotemporal Kalman filtering," *Inverse Problems*, pp. 435–453, Oct. 2004.
15. J. Antelis and J. Minguez, "Dynamic solution to the EEG source localization problem using Kalman filters and particle filters," in *International Conference of the IEEE Engineering in Medicine and Biology Society*, 2009, pp. 77–80.
16. N. J. Gordon, D. J. Salmon, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," in *IEE Proceedings F: Radar and Signal Processing*, vol. 140, 1992, pp. 107–113.
17. A. Sorrentino, L. Parkkonen, and M. Piana, "Particle filters: A new method for reconstructing multiple current dipoles from MEG data," in *International Congress Series*, vol. 1300, 2007, pp. 173–176.
18. H. R. Mohseni, E. L. Wilding, and S. Sanei, "Sequential Monte Carlo techniques for EEG dipole placing and tracking," in *Sensor Array and Multichannel Signal Processing Workshop*, July 2008, pp. 95–98.
19. R. Mahler, "Multi-target Bayes filtering via first-order multi-target moments," *IEEE Transactions on Aerospace and Electronic System*, vol. 39, pp. 1152–1178, 2003.
20. B. N. Vo, S. Singh, and A. Doucet, "Sequential Monte Carlo implementation of the PHD filter for multi-target tracking," in *International Conference of Information Fusion*, vol. 2, 2003, pp. 792–799.
21. A. Hyvärinen and E. Oja, "A fast fixed-point algorithm for independent component analysis," *Neural Computation*, vol. 9, pp. 1483–1492, 1997.
22. A. J. Casson and E. Rodriguez-Villegas, "Data reduction techniques to facilitate wireless and long term AEEG epilepsy monitoring," in *IEEE EMBS conference on neural Engineering*, May 2007, pp. 298–301.

23. A. J. Casson, S. Smith, J. S. Duncan, and E. Rodriguez-Villegas, "Wearable EEG: what is it, why is it needed and what does it entail?" in *Engineering in Medicine and Biology Society*, Aug. 2008, pp. 5867–5870.
24. S. Baillet, J. C. Mosher, and R. M. Leahy, "Electromagnetic brain mapping," *IEEE Signal Processing Magazine*, vol. 18, pp. 14–30, 2001.
25. 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.
26. D. J. Daley and D. Vere-Jones, *An introduction to the theory of point processes, Volume II: General theory and Structure*, 2nd ed. Springer-Verlag, 1997.
27. M. Tobias and A. D. Lanterman, "Probability hypothesis density-based multi-target tracking with bistatic range and doppler observations," *IEEE Radar, Sonar and Navigation*, vol. 152, pp. 195–205, 2005.
28. D. E. Clark and J. Bell, "Bayesian multiple target tracking in forward scan sonar images using the PHD filter," *IEEE Radar, Sonar and Navigation*, vol. 152, pp. 327–334, 2005.
29. T. Chen, W. Liu, and L. Chen, "VLSI architecture of leading eigenvector generation for on-chip principal component analysis spike sorting system," in *Engineering in Medicine and Biology Society, 30th Annual International Conference of the IEEE*, 2008, pp. 3192–3195.
30. J. Cao, N. Murata, S. Amari, A. Cichocki, and T. Takeda, "Independent component analysis for unaveraged single-trial MEG data decomposition and single-dipole source localization," *Neurocomputing*, vol. 49, no. 1-4, pp. 255–277, 2002.
31. N. Verma, A. Shoeb, J. Bohorquez, J. Dawson, J. Guttag, and A. Chandrakasan, "A micro-power EEG acquisition SoC with integrated feature extraction processor for a chronic seizure detection system," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 4, pp. 804–816, April 2010.
32. Y. M. Chi, S. Deiss, and G. Cauwenberghs, "Non-contact low power EEG/ECG electrode for high density wearable biopotential sensor networks," in *Sixth International Workshop on Wearable and Implantable Body Sensor Networks*, June 2009, pp. 246–250.
33. W. Zhou, N. Kovvali, A. Papandreou-Suppappola, and A. Chattopadhyay, "Sensor optimization for progressive damage diagnosis in complex structures," in *Proc. of SPIE*, vol. 7650, 2010, pp. 1–10.
34. A. Chhetri, D. Morrell, and A. Papandreou-Suppappola, "Sensor scheduling using a 0-1 mixed integer programming framework," in *Fourth IEEE Workshop on Sensor Array and Multichannel Processing*, July 2006, pp. 471–475.
35. W. Welch, "Branch-and-bound search for experimental designs based on D-optimality and other criteria," *Technometrics*, vol. 24, pp. 41–48, 1982.
36. S. Joshi and S. Boyd, "Sensor selection via convex optimization," *IEEE Transaction on Signal Processing*, vol. 57, pp. 451–462, 2009.
37. S. Fujio, H. Shiomi, and Y. Okamura, "Acceleration of FPGA-based ICA processor for real-time processing," in *International Symposium on Antennas and Propagation*, July 2010, pp. 1–4.
38. L. Miao, J. J. Zhang, C. Chakrabarti, and A. Papandreou-Suppappola, "Multiple sensor sequential tracking of neural activity with FPGA implementation," in *Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA*, November 2010, pp. 369–373.
39. C. Chakrabarti, "High sample rate array architecture for median filters," *IEEE Transaction on Signal Processing*, vol. 57, pp. 707–712, Mar. 1994.
40. K. Shindo, A. Ikeda, T. Musha, K. Terada, H. Fukuyama, W. Taki, J. Kimura, and H. Shibasaki, "Clinical usefulness of the dipole tracing method for localizing interictal spikes in partial epilepsy," *Epilepsia*, vol. 39, pp. 371–379, 1998.