

# REAL-TIME CLOSED-LOOP TRACKING OF AN UNKNOWN NUMBER OF NEURAL SOURCES USING PROBABILITY HYPOTHESIS DENSITY PARTICLE FILTERING

L. Miao, J. J. Zhang, C. Chakrabarti, A. Papandreou-Suppappola, N. Kovvali

School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ  
{Lifeng.Miao, Jun.Zhang.EE, chaitali, papandreou, Narayan.Kovvali}@asu.edu

## ABSTRACT

Probability hypothesis density (PHD) filtering, implemented using particle filters, is a Bayesian technique used to non-linearly track multiple objects. In this paper, we propose a new approach based on PHD particle filters (PHD-PF) to automatically track the number of magnetoencephalography (MEG) neural dipole sources and their unknown states. In particular, by separating the MEG measurements using independent component analysis, PHD-PF is applied in a *closed-loop* to first estimate the number of sources and then recover their amplitude, location and orientation. We also reduce the processing time and computational complexity by employing window-based processing and multi-channel decomposition. We simulate the overall system using synthetic data and show that the proposed algorithm has tracking performance similar to existing approaches with significantly fewer number of particles. We also map the algorithm on to a Xilinx Virtex-5 field-programmable gate array (FPGA) platform. The processing period for one iteration using 3,200 particles is only about 314  $\mu$ s, which makes this implementation suitable for real-time tracking.

**Index Terms**— PHD filter, independent component analysis, parallel architecture, FPGA implementation.

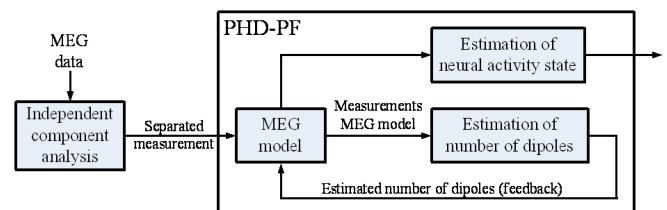
## 1. INTRODUCTION

Magnetoencephalography (MEG) is a non-invasive neurophysiological diagnostic tool that is used to measure magnetic fields generated by the brain's neural activity [1]. MEG measurements can be used to track neural activity and have been used during epilepsy surgery. Since most MEG signals result from intracellular longitudinal currents along active cortex regions, equivalent current dipole models have been successfully used to determine the brain's electrical current distribution [2, 3].

Sequential Bayesian estimation techniques, such as particle filtering (PF) and multiple PF, have been used to solve the nonlinear problem of tracking a known number of MEG dipole sources [4, 5]. However, in realistic scenarios, the number of neural sources at each time step is unknown and

has to be estimated together with the MEG sources amplitudes, locations and orientations. In [6], an approach was proposed to dynamically estimate the number of dipoles and neural activity. In particular, the number of neural sources at each time step was modeled as a random finite set (RFS) and a PF was used to estimate the number and states of dipole sources. This approach required a large number of particles (in the order of 100,000) to achieve good estimation performance, and is not amenable to real-time implementation.

In this paper, we propose a new approach to estimate both the unknown number of neural sources and their states at each time step. We base our approach on the PHD-PF that is used to estimate the state of multiple objects [7]. We first separate the MEG measurements into independent components corresponding to different neural sources [8], and we use the PHD-PF to estimate the number of dipoles at each time step in a closed-loop configuration before estimating their states. The closed-loop approach simplifies the dual estimation to a known-number of dipoles problem, and thus decreases the number of required particles and reduces the computational complexity. The overall approach is depicted in Figure 1. For real-time neural tracking, we propose the use of window-based processing along with multi-channel decomposition. The proposed algorithm is implemented on a Xilinx Virtex-5 FPGA platform; for the 4-processor architecture, the processing time per iteration using 3,200 particles is shown to be only 314  $\mu$ s, which makes it feasible for real-time tracking.



**Fig. 1.** Block diagram of proposed closed-loop tracking system.

The paper is organized as follows. In Section 2, we introduce the neural activity tracking problem, and in Section 3, we review the PHD-PF. We propose a new approach to estimate the unknown number of dipole sources and their states in Section 4, and in Section 5, we present the hardware implementation of the proposed algorithm. We provide simulations and demonstrate our algorithmic and hardware performance results in Section 6.

## 2. NEURAL ACTIVITY TRACKING PROBLEM

A *current dipole* is used to model the primary current from a set of neurons due to their combined activation in response to a sensory stimulus. The  $j$ th current dipole,  $j=1, \dots, N_d$ , at time  $t, t=1, \dots, T$ , can be represented by its amplitude  $s_{j,t}$ , three-dimensional (3-D) location vector  $\mathbf{m}_{j,t}$ , and 3-D orientation vector  $\mathbf{p}_{j,t}$ . If  $M$  is the number of sensors, the measured magnetic field outside the human head can be represented by [9]

$$\mathbf{B} = \mathbf{A} \mathbf{S} + \mathbf{N} \quad (1)$$

where  $\mathbf{S}$  is an  $N_d \times T$  matrix whose  $(j, t)$ th element is the  $j$ th dipole amplitude  $s_{j,t}$  at time  $t$ ,  $\mathbf{N}$  is the measurement noise,  $\mathbf{B}$  is an  $M \times T$  matrix whose  $(m, t)$ th element is the MEG measurement  $\mathbf{b}_t(\mathbf{p}_{s_m})$ ,  $m=1, \dots, M$ ,  $\mathbf{p}_{s_m}$  is the position of the  $m$ th sensor,  $\mathbf{A}$  is an  $M \times N_d$  matrix whose  $(m, j)$ th element is  $[\mathbf{g}(\mathbf{p}_{s_m}, \mathbf{e}_{s_m}, \mathbf{p}_j) \cdot \mathbf{m}_j]$ ,  $\mathbf{e}_{s_m}$  is the orientation of the  $m$ th sensor,

$$\begin{aligned} \mathbf{g}(\mathbf{p}_{s_m}, \mathbf{e}_{s_m}, \mathbf{p}_j) &= \sum_{j=1}^{N_d} \frac{\mu_0}{4\pi F_j^2} (F_j \mathbf{m}_j \times \mathbf{p}_j \\ &\quad - (\mathbf{m}_j \times \mathbf{p}_j \cdot \mathbf{p}_s) \nabla F_j). \end{aligned} \quad (2)$$

Here,  $F_j = a_j(p_s a_j + p_s^2 - (\mathbf{p}_j \cdot \mathbf{p}_s))$ ,  $a_j = |\mathbf{p}_s - \mathbf{p}_j|$ ,  $p_s = |\mathbf{p}_s|$  and  $\nabla F_j$  is defined in [9]. The inverse problem is to dynamically estimate  $s_{j,t}$ ,  $\mathbf{p}_j$  and  $\mathbf{m}_j$  from the MEG measurements matrix  $\mathbf{B}$  using (1). Note that we assume that the location and orientation of the dipoles do not change much with time.

## 3. PHD PARTICLE FILTER

**PHD Filter.** The PHD filter is an efficient approach for solving the multiple object tracking problem. It can be used to dynamically estimate the unknown parameters or states of an unknown number of objects based on noisy observations. The PHD is the first moment of the multiple target posterior distribution. Thus, integrating it over a region  $R$  yields an estimate of the number of objects in  $R$ ; the peaks of the PHD provide an estimate of the objects' parameters [10].

In a multi-object tracking formulation, the state and measurement at time  $t$  are modeled as RFS  $\mathbf{X}_t = \{\mathbf{x}_{t,1}, \dots, \mathbf{x}_{t,T_t}\}$  and  $\mathbf{Z}_t = \{\mathbf{z}_{t,1}, \dots, \mathbf{z}_{t,O_t}\}$ , where  $\mathbf{x}_{t,i}$  is the state vector of the  $i$ th single object,  $T_t$  is the number of objects at time  $t$ ,  $\mathbf{z}_{t,j}$  is the measurement vector and  $O_t$  is the total number of measurements at time  $t$ . The PHD filter recursively approximates the PHD of multiple objects using prediction and updating steps [7]:

$$\begin{aligned} D_{t|t-1}(\mathbf{x}_t | \mathbf{Z}_{1:t-1}) &= \gamma_t(\mathbf{x}_t) + \int \phi_{t|t-1}(\mathbf{x}_t, \mathbf{x}_{t-1}) \\ &\quad D_{t-1|t-1}(\mathbf{x}_{t-1} | \mathbf{Z}_{1:t-1}) d\mathbf{x}_{t-1} \end{aligned} \quad (3)$$

$$\begin{aligned} D_{t|t}(\mathbf{x}_t | \mathbf{Z}_{1:t}) &= [\nu(\mathbf{x}_t) + \sum_{\mathbf{z} \in \mathbf{Z}_t} \frac{\psi_{t,\mathbf{z}}(\mathbf{x}_t)}{\kappa_t(\mathbf{z}) + \langle D_{t|t-1}, \psi_{t,\mathbf{z}} \rangle}] \\ &\quad D_{t|t-1}(\mathbf{x}_t | \mathbf{Z}_{1:t-1}) \end{aligned} \quad (4)$$

where  $\phi_{t|t-1}(\mathbf{x}_t, \mathbf{x}_{t-1}) = P_S f_{t|t-1}(\mathbf{x}_t | \mathbf{x}_{t-1}) + b_{t|t-1}$ ,  $\nu(\mathbf{x}_t) = 1 - P_D$ ,  $\kappa_t(\mathbf{z}) = \lambda_t c_t(\mathbf{z})$  and  $\psi_{t,\mathbf{z}} = P_D g(\mathbf{z} | \mathbf{x}_t)$ . In (3),  $\gamma_t$  denotes the PHD of the spontaneous birth of a new object at time  $t$ ,  $b_t$  is the PHD of a spawned object,  $P_S$  is the probability that the object is still present, and  $f_{t|t-1}(\cdot | \cdot)$  represents the transition probability density of a single object. In (4),  $g(\cdot | \cdot)$  is the likelihood of a single-object,  $P_D$  is the probability of detection,  $\lambda_t$  is the average number of clutters, and  $c_t$  is the clutter probability density. The inner product is given by  $\langle D_{t|t}, \varphi \rangle = \int D_{t|t}(\mathbf{x}_t | \mathbf{Z}_{1:t}) \varphi(\mathbf{x}_t) d\mathbf{x}_t$ .

**PHD Particle Filter Implementation.** A PF implementation of the PHD filter has been proposed in [7]. Since it is robust and computationally inexpensive, the PHD-PF has been widely used. In [7],  $D_{t-1|t-1}(\mathbf{x}_{t-1} | \mathbf{Z}_{1:t-1})$  is represented by a set of particles and weights  $\{w_{t-1}^{(i)}, x_{t-1}^{(i)}\}_{i=1}^{N_{t-1}}$  as

$$D_{t-1|t-1}(\mathbf{x}_{t-1} | \mathbf{Z}_{1:t-1}) = \sum_{i=1}^{N_{t-1}} w_{t-1}^{(i)} \delta_{x_{t-1}^{(i)}}(\mathbf{x}_{t-1}). \quad (5)$$

Substituting (5) into (3), we can obtain

$$D_{t|t-1}(\mathbf{x}_t | \mathbf{Z}_{1:t-1}) = \sum_{i=1}^{N_{t-1}} w_{t-1}^{(i)} \phi_t(\mathbf{x}_t, x_{t-1}^{(i)}) + \gamma_t(\mathbf{x}_t).$$

As a result,  $D_{t|t-1}(\mathbf{x}_t | \mathbf{Z}_{1:t-1})$  can be represented by

$$\begin{aligned} D_{t|t-1}(\mathbf{x}_t | \mathbf{Z}_{1:t-1}) &= \sum_{i=1}^{N_{t-1} + M_{\text{ext}}} w_{t|t-1}^{(i)} \delta_{x_t^{(i)}}(\mathbf{x}_t) \\ w_{t|t-1}^{(i)} &= \begin{cases} \frac{\phi_t(x_t^{(i)}, x_{t-1}^{(i)}) w_{t-1}^{(i)}}{q_t(x_t^{(i)} | x_{t-1}^{(i)}, \mathbf{Z}_t)}, & i = 1, \dots, N_{t-1} \\ \frac{\gamma_t(x_t^{(i)})}{M_{\text{ext}} p_t(x_t^{(i)} | \mathbf{Z}_t)}, & i = N_{t-1} + 1, \dots, N_{t-1} + M_{\text{ext}} \end{cases} \end{aligned}$$

$N_t$  is the number of particles at time  $t$  and  $M_{\text{ext}}$  is the number of particles for the new born target. Similarly, in Equation (4),  $D_{t|t}(\mathbf{x}_t | \mathbf{Z}_{1:t})$  can be represented as

$$D_{t|t}(\mathbf{x}_t | \mathbf{Z}_{1:t}) = \sum_{i=1}^{N_{t-1} + M_{\text{ext}}} w_t^{(i)} \delta_{x_t^{(i)}}(\mathbf{x}_t), \quad (6)$$

$$w_t^{(i)} = [v(x_t^{(i)}) + \sum_{\mathbf{z} \in \mathbf{Z}_t} \frac{\psi_{t,\mathbf{z}}(x_t^{(i)})}{\kappa_t(\mathbf{z}) + C_t(\mathbf{z})}] w_{t|t-1}^{(i)},$$

$$C_t(\mathbf{z}) = \sum_{i=1}^{N_{t-1} + M_{\text{ext}}} \psi_{t,\mathbf{z}}(x_t^{(i)}) w_{t|t-1}^{(i)}.$$

As a result, the PHD of a random set is represented by a set of particles and its corresponding weights.

## 4. TRACKING NEURAL ACTIVITY USING PHD-PF

### 4.1. Separation of Dipole Source Components

In order to apply the PHD-PF algorithm, the measurement vector  $\mathbf{z}_{t,j}$  must be generated by a single target. However, for the MEG dipole model in Section 2, the MEG measurement  $\mathbf{b}_t$  is due to multiple neural sources. Thus, we first need to separate the mixed MEG measurements into components

so that each component corresponds to an individual neural source. Since distinct neural sources are mutually independent [11], the independent component analysis (ICA) technique can be used to separate the MEG measurements. Here, we use fast ICA approach presented in [8] to separate the mixed MEG measurements.

We first need to pre-whiten the data by transforming the MEG measurement matrix  $\mathbf{B}$  to obtain the white matrix  $\mathbf{Z} = \mathbf{P}\mathbf{B}$ , such that the elements in  $\mathbf{Z}$  are uncorrelated and  $E\{\mathbf{Z}\mathbf{Z}^T\} = \mathbf{I}$ , where  $E\{\cdot\}$  denotes statistical expectation and  $\mathbf{I}$  is the identity matrix. The fast ICA is an iterative process. At the  $k$  iteration, using an initial random vector  $\mathbf{w}(0)$ , we let  $\mathbf{w}(k) = E\{\mathbf{Z}[\mathbf{Z}^T\mathbf{w}(k-1)]^3\} - 3\mathbf{w}(k-1)$ , and we divide  $\mathbf{w}(k)$  by its norm. If  $|\mathbf{w}^T(k)\mathbf{w}(k-1)|$  is not close enough to 1, then we go to the next iteration. Otherwise, the algorithm converges to  $\mathbf{w}(k)$ , where  $\mathbf{w}$  is the separated vector used to estimate the independent source  $\hat{\mathbf{s}}$  by  $\hat{\mathbf{s}} = \mathbf{w}^T\mathbf{Z}$  and  $\hat{\mathbf{S}} = [\hat{\mathbf{s}}_1, \dots, \hat{\mathbf{s}}_N]$ . Since  $\hat{\mathbf{S}} = \mathbf{W}^T\mathbf{Z} = \mathbf{W}^T\mathbf{P}\mathbf{B}$ , we can use the unmixing matrix  $\mathbf{A}_g = \mathbf{P}^{-1}\mathbf{W}$  to recover the MEG signal as

$$\hat{\mathbf{B}} = \mathbf{A}_g\hat{\mathbf{S}} \quad \text{or} \quad \hat{\mathbf{b}}_j = \mathbf{a}_j\hat{\mathbf{s}}_j, \quad (7)$$

where  $\mathbf{a}_j$  is the  $j$ th column of matrix  $\mathbf{A}_g$ ,  $\hat{\mathbf{s}}_j$  is the  $j$ th estimated source, and  $\hat{\mathbf{b}}_j$  is the estimated magnetic field generated by the  $j$ th source. After the ICA source separation, we obtain the set of separated MEG measurements  $\{\hat{\mathbf{b}}_1(t), \hat{\mathbf{b}}_2(t), \dots, \hat{\mathbf{b}}_J(t)\}_{t=1}^T$ , where  $J > N_d$  is the total number of sources (dipole or noise), and each measurement corresponds to an individual source. We use these new measurements as the input to the PHD-PF.

Implementing the pre-whitening step in real-time is quite challenging. Here, we need to calculate the eigenvalues and eigenvectors of the matrix  $\mathbf{C}_B = E\{\mathbf{B}\mathbf{B}^T\}$ , where  $\mathbf{B}$  is  $M \times T$  [8]. In an MEG data acquisition system, the number of sensors  $M$  is always greater than 50. In Section 5.2, we propose an approach which divides the MEG sensors into sub-channels to reduce the measurement dimension.

## 4.2. Multi-dipole Estimation Using PHD-PF

We consider the MEG source estimation as a dynamical Bayesian problem. Its state-space model is given by

$$\mathbf{X}_{t+1} = \mathbf{X}_t + \mathbf{u}_t, \quad \mathbf{Z}_t = h(\mathbf{X}_t) + \mathbf{v}_t,$$

where  $\mathbf{X}_t$  represents the current dipole states  $\{s_{t,j}, \mathbf{m}_{t,j}, \mathbf{p}_{t,j}\}$ , the signal-dipole evolution process is modeled as a random walk with Gaussian transition kernel,  $\mathbf{Z}_t = \{\mathbf{z}_1(t), \mathbf{z}_2(t), \dots, \mathbf{z}_J(t)\}^T = \{\hat{\mathbf{b}}_1(t), \hat{\mathbf{b}}_2(t), \dots, \hat{\mathbf{b}}_J(t)\}^T$  is the separated measurement,  $h(\cdot)$  is the measurement function determined by (1), and  $\mathbf{v}_t$  is the Gaussian measurement noise with standard deviation  $\sigma_m$ .

We propose a new approach to efficiently estimate the parameters of each dipole source using PPHD filtering. The general scheme is as follows:

**Step 1 Initialization:** Draw samples  $\{x_0^{(i)}\}_{i=1}^{N_0}$  from the prior PHD  $D_{0|0}$ , where  $N_0 = T_0 \times N$  and  $T_0$  is the initial number of dipoles and  $N$  is the number of particles used for each

dipole. Assign particle weight  $w_0^{(i)} = T_0/N$ .

**Step 2 Prediction:** For the existing dipole, sample  $\{\tilde{x}_t^{(i)}\}_{i=1}^{N_{t-1}}$  from a proposal density  $q_t(\cdot|x_{t-1}^{(i)})$ . Evaluate the predicted weights

$$\tilde{w}_{t|t-1}^{(i)} = \frac{\phi_{t|t-1}(\tilde{x}_t^{(i)}, x_{t-1}^{(i)})}{q_t(\tilde{x}_t^{(i)}|x_{t-1}^{(i)})} w_{t-1}^{(i)}.$$

Here we choose  $q_t(\cdot|\cdot)$  equal to the transition probability density  $f_{t|t-1}(\cdot|\cdot)$  and assume no dipole spawns. As a result,  $\tilde{w}_{t|t-1}^{(i)} = P_S w_{t-1}^{(i)}$ . For the new born dipole, sample  $\{\tilde{x}_t^{(i)}\}_{i=N_{t-1}+1}^{N_{t-1}+M_{ext}}$  from another proposal density  $p_t(\cdot|Z_t)$ . Compute the weights for the new born particles

$$\tilde{w}_{t|t-1}^{(i)} = \frac{1}{M} \frac{\gamma_t(\tilde{x}_t^{(i)})}{p_t(\tilde{x}_t^{(i)}|Z_t)}.$$

Here we choose  $p_t(\cdot)$  equal to  $\frac{1}{a}\gamma_t(\cdot)$ , thus  $\tilde{w}_{t|t-1}^{(i)} = \frac{a}{M}$ , where  $a$  is a constant.

**Step 3 Updating:** Set  $R_t = N_{t-1} + M_{ext}$  and update the weights  $\{\tilde{w}_t^{(i)}\}_{i=1}^{R_t}$  as

$$\tilde{w}_t^{(i)} = [1 - P_D + \sum_{j=1}^J \frac{\psi_{t,\mathbf{z}_j}(\tilde{x}_t^{(i)})}{\kappa_t(\mathbf{z}_j) + C_t(\mathbf{z}_j)}] \tilde{w}_{t|t-1}^{(i)}, \quad (8)$$

$$C_t(\mathbf{z}_j) = \sum_{i=1}^{R_t} \psi_{t,\mathbf{z}_j}(\tilde{x}_t^{(i)}) \tilde{w}_{t|t-1}^{(i)}. \quad (9)$$

Here  $\{\psi_{t,\mathbf{z}_j}(\tilde{x}_t^{(i)})\}_{j=1}^J$  are the likelihoods obtain by

$$\psi_{t,\mathbf{z}_j} = \exp\left\{-\frac{1}{2\sigma_m^2} [\mathbf{z}_j(t) - h(\tilde{x}_t^{(i)})]^T \cdot [\mathbf{z}_j(t) - h(\tilde{x}_t^{(i)})]\right\}$$

**Step 4 Resampling:** Estimate the number of dipoles  $\hat{T}_t = \sum_{i=1}^{R_t} \tilde{w}_t^{(i)}$  and resample  $\{\tilde{x}_t^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^{R_t}$  to get  $\{x_t^{(i)}, w_t^{(i)}\}_{i=1}^{N_t}$ , where  $N_t = N \cdot \text{round}(\hat{T}_t)$  and  $\text{round}(a)$  denotes the nearest integer to  $a$ . After resampling, each of the new particles has weight  $\hat{T}_t/N_t$ .

**Step 5 Estimating:** Cluster the resampled particles and estimate the state parameters. The clustering is performed in a 3-dimensional position space and we use k-means clustering algorithm.

From the algorithm, we can see the number of particles changes over time and is proportional to the number of dipoles. Thus, at time  $t$ ,  $N_t \propto T_t$ . Different from standard particle filtering, there is a summation among sub-measurements  $\{\mathbf{z}_j\}_{j=1}^J$  when updating the weights in Step 3. These independent sub-measurements are separated from the mixed measurement and each sub-measurement corresponds to an individual source. After Step 3, the posterior PHD at time  $t$  is approximated by a set of weighted particles  $\{\tilde{x}_t^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^{R_t}$  and it contains all the available information about the dipoles. For example, the number of dipoles can be obtained by computing the integral of the posterior PHD, which is equal to the summation of the weights  $\sum_{i=1}^{R_t} \tilde{w}_t^{(i)}$

and the estimate of dipole parameters can be found from the peaks of the PHD. In the resampling step, the new weights  $w_t^{(i)}$  are not normalized to 1, but sum to  $\hat{T}_t$ .

## 5. REAL-TIME HARDWARE IMPLEMENTATION

### 5.1. Window Processing

Most of the existing methods apply the ICA algorithm on the whole MEG/EEG data [12], which can only be done off-line. In order to process the MEG data on-line, we window the MEG data, and apply the proposed algorithm on the data in each window of length  $W$ , as shown in Fig. 2. If  $W$  is small, the computations will be done faster, but the estimation error will be larger, as shown in Table 2. In this scheme, the window length has to be larger than the execution time of FastICA as shown in Fig. 2.

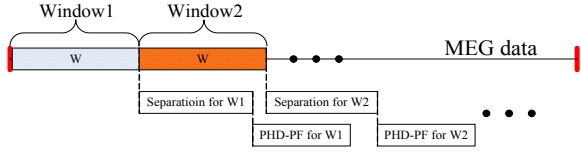


Fig. 2. Window processing of the MEG data.

### 5.2. Multi-channel Decomposition

From Section 4.1, the computational complexity of pre-whitening is very high since the  $M \times T$  MEG measurement matrix  $\mathbf{B}$  is quite large. To facilitate fast processing, we propose to split the  $M$  channels into  $N_s$  sub-channels with  $\alpha = M/N_s$  channels per sub-channel. Based on the ICA algorithm [8], if  $\alpha$  is greater than the number of sources, we can still recover the source signal using  $\alpha$  channels. Since the  $N_s$  sub-channel computations are independent, they can be easily mapped on to a parallel architecture. In Fig. 3, measurement matrix  $\mathbf{B}_{M \times T}$  is decomposed into  $N_s$  sub-matrices  $\{\mathbf{B}^n\}_{n=1}^{N_s}$ . Each sub-matrix corresponds to  $\alpha$  channels in a sub-channel. The  $N_s$  source separation units implement the pre-processing steps in Section 4.1. FastICA unit operates on the sub-matrix  $\mathbf{B}^n$  and outputs the independent component matrix  $\mathbf{S}$  and the unmixing matrix  $\mathbf{A}^n$ . Next, the separated magnetic fields  $\hat{\mathbf{b}}^n$  are recovered by the Recover unit. The magnetic fields of all  $M$  channels are combined and fed to the PHD-PF unit.

Care must be taken when choosing  $\alpha$ . For smaller  $\alpha$ , the computational complexity will be lower and the processing speed will be faster; however, the number of parallelized FastICA hardware units will be higher and thus the resource utilization will increase. Smaller  $\alpha$  values also result in a degradation of the estimation performance, as shown in Table 3.

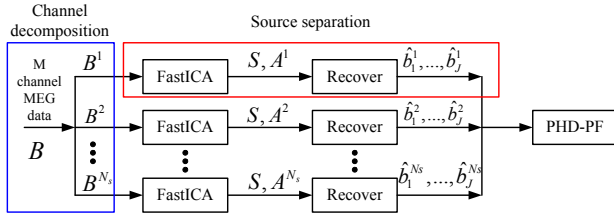


Fig. 3. Block diagram of parallel sub-channel processing system.

### 5.3. Architecture of Particle PHD filter

In this section, we propose an efficient multi-processor architecture for the PHD-PF. The high-level block diagram is shown in Fig. 4(a). It consists of  $P$  processing elements (PE) connected by a global bus. Each PE processes  $R_t/P$  particles, where  $R_t$  is the total number of particles in time  $t$ . Local processing steps, such as initialization (Step 1), prediction (Step 2) and part of updating (Step 3) are operated in each PE as shown in Fig. 4(b). Global processing steps, such as computing normalization factors  $C_t$ , estimating the dipole number (step3)  $\hat{T}_t$ , resampling (Step 4) and clustering (Step 5), are executed in the CU.

Each PE communicates with the CU, but there is no inter-PE communication. Fig. 4(b) also shows the data transferred between the PE and CU. Since the resampling needs the weights of all the particles, in order to reduce the communication overhead between PE and CU, we employ the grouping method in [5]. The main idea is as follows. After obtaining the particles  $\{\tilde{x}_t^{(i)}, \tilde{w}_t^{(i)}\}$ , we divide them into  $G$  groups; we use the averages of each group  $\mu x_g, g = 1, \dots, G$  as the new particles and only send the average weights  $\mu w_g$  from PE to CU as shown in Fig. 4(b). The effect of the grouping procedure was studied in detail in [5].

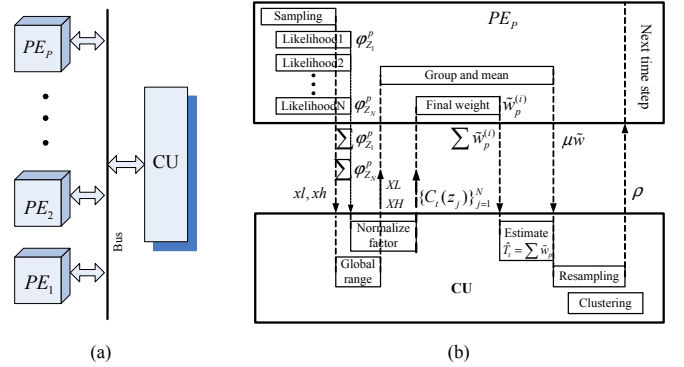


Fig. 4. (a) Parallel architecture. (b) Operations in PE and CU.

From Fig. 4, the likelihoods  $\psi_{t, \mathbf{z}_j}(\tilde{x}_t^{(i)})$  are calculated  $J$  times for different individual measurements  $\mathbf{z}_j$  for each particle  $\tilde{x}_t^{(i)}$ . Since they are independent, we implement them in parallel. The normalization factor for each individual measurement  $C_t(\mathbf{z}_j)$  is computed in the CU by summing the likelihoods from every PE  $\sum_p \sum_j \psi_{t, \mathbf{z}_j}^p$  in Equation (8) and the sum of updated weights is used to estimate the dipole number  $\hat{T}_t$ . Next, using  $\hat{T}_t$ , particles are resampled in CU based on their weights and then the resampled particles are clustered to obtain the estimation of different dipole states. The hardware resource for each block in Fig. 4 is shown in Table 1.

## 6. SIMULATIONS AND FPGA IMPLEMENTATION

### 6.1. Neural Activity Tracking Result

In order to demonstrate the tracking performance of the proposed PHD-PF, we use synthetic data from a previous study in [6] as well as three dipoles localized at V1 (1.11, 5.34, 4.98),

**Table 1.** Hardware operators for each block in a PE.

Block	+	×	÷	√	exp
Sampling	6	0	0	0	0
Likelihood	88	93	1	2	1
Global	6	0	0	0	0
Normal	9	0	0	0	0
Final weight	6	1	1	0	0
Group mean	52	0	1	0	0
Resampling	2	3	1	0	0
Clustering	6	3	1	1	0

$V5_R$  (4.36, 3.68, 4.44) and  $V5_L$  (3.37, 4.85, 4.81). For this simulation, we used  $M = 1,000$  particles for each dipole and the maximum number of particles was  $3M + M_{\text{ext}} = 3,200$ , which is much less than the 100,000 particles used in [6]. The particles are initially uniformly distributed in the brain volume hemisphere with radius 10 cm. The dipole evolution model in Equation (8) is a random walk with Gaussian transition kernel  $f_{t+1|t}(p_{t+1}|p_t) = \mathcal{N}(p_{t+1} - p_t, \sigma_p)$  and  $f_{t+1|t}(m_{t+1}|m_t) = \mathcal{N}(m_{t+1} - m_t, \sigma_m)$ , with  $\sigma_p=1$  cm and  $\sigma_m=2$  nA. Each existing dipole has a probability of survival  $P_S=0.8$  and probability of detection  $P_D=0.95$ .

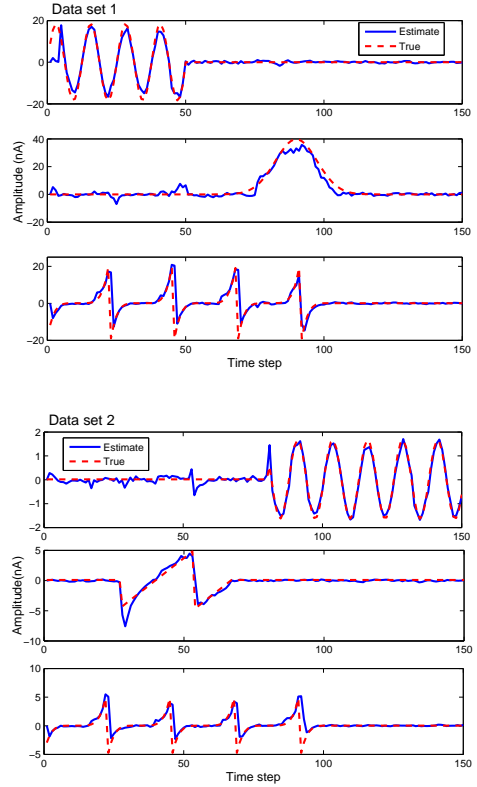
The dipole source amplitudes for each data set are shown in Fig. 5 (red dashed curves). In both data sets, we observe that the number of dipoles changes over time. For instance, in Fig. 5 (a) there are two dipoles for  $t = 0, \dots, 49$ , one dipole for  $t = 50, \dots, 69$ , two dipoles for  $t = 70, \dots, 99$ , and one dipole for  $t = 100, \dots, 120$ . Fig. 5 also shows the source amplitudes estimation result and Fig. 6 shows the 3-D position estimation result for 100 Monte Carlo simulations, for  $W=50$  ms and  $\alpha=4$ . The average position RMSE is 7.93 mm for data set 1 and 7.24 mm for data set 2, which are comparable with the 7mm result in [6]. Furthermore, since only 3,200 particles are used here instead of the 10,000 in [6], the resource utilization and processing time of proposed PHD-PF approach is significantly lower.

Table 2 shows the root mean-squared error (RMSE) tracking performance with respect to the window length  $W$ . We see that as  $W$  decreases, the RMSE increases; when  $W$  is smaller than 50 ms, there is a significant degradation in the RMSE. Also, since for smaller  $W$  the processing is faster, we choose  $W=50$  ms as a compromise between RMSE performance and processing speed.

The choice of sub-channel factor  $\alpha$  can also impact the RMSE performance. Since the number of MEG channels is 60,  $\alpha$  is chosen to be 60, 6, 4 and 3. Table 3 shows that as  $\alpha$  decreases, the RMSE increases. In order to reduce the computation complexity without performance degradation, and keep the RMSE to less than 10%, we choose  $\alpha = 4$ .

**Table 2.** RMSE for different window lengths  $W$ 

$W$ , ms	150	100	50	25
Amplitude, nA	2.20	2.73	3.37	5.17
	(5.5%)	(6.8%)	(8.4%)	(13%)
Position, mm	5.37	5.81	6.62	8.24
	(5.4%)	(5.8%)	(6.6%)	(8.2%)

**Fig. 5.** Amplitude estimation as a function of time for two data sets.**Table 3.** RMSE for different sub-channel factors  $\alpha$ 

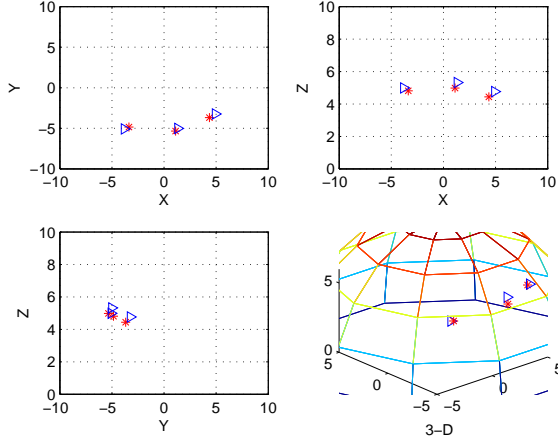
$\alpha$	60	6	4	3
Amplitude, nA	2.20	3.53	3.67	4.72
	(5.5%)	(8.8%)	(9.2%)	(12%)
Position, mm	5.37	7.19	7.63	9.35
	(5.4%)	(7.2%)	(7.6%)	(9.4%)

## 6.2. Hardware Implementation Evaluation

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

*Resource Utilization:* Table 4 summarizes the architecture resource utilization for the 4 PE parallel architecture for PHD-PF. Since the total number of particles is 3,200, each PE processes 800 particles. The number of sub-channels  $N_s = 60/4 = 15$ . The exponential functions are implemented using CORDIC units, and the rest of the units are implemented using DSP cores. For the pre-processing part, we use the FPGA implementation of FastICA in [13]. The total slice utilization for the FastICA is 47,760 (32%).

*Execution Time:* Fig. 7 shows the timing performance for one iteration of the proposed method; the actual number of cycles is given in Table 5. One iteration takes  $T_{\text{total}}=(N_s + N_g + N_{gm} + N_c) \times T_{\text{clk}}=4,852$  cycles. We choose the system clock rate as 100 MHz. The total processing period for one iteration is only  $T_{\text{total}}=48.52\mu\text{s}$ . Based on FastICA implementation in

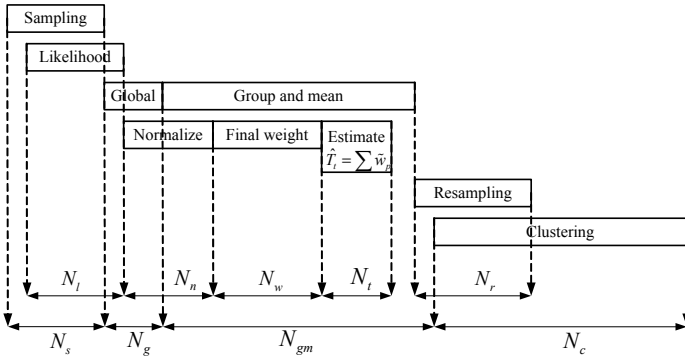


**Fig. 6.** 3-D dipole positions tracking result for window length  $W=50$  ms, segments factor  $\alpha=4$  and 100 Monte Carlo simulations (red star–True; blue triangle–Reconstructed).

**Table 4.** PPHD resource utilization on Xilinx XC5VSX240T

Unit	Occupied slices	Slice Reg.	Slice LUTs	Block Ram	DSP48Es
4-PE	28,250 (76%)	92,490 (62%)	93,270 (63%)	324 (63%)	295 (28%)

[13], the execution time for source separation is about  $265\mu\text{s}$ . As a result, the total processing time for one iteration is about  $314\mu\text{s}$ . Since one window includes 50 iterations, the processing time for one window is 15.7ms which is much shorter than the window length 50ms. Thus the proposed approach can meet the requirements of real-time processing.



**Fig. 7.** Execution time breakdown for one iteration of PPHD.

## 7. CONCLUSION

In this paper, we proposed the application of particle PHD filtering for tracking unknown number of neural dipoles and demonstrated its performance using numerical simulations on synthetic data. This method achieves good performance in terms of RMSE using significantly fewer number of particles compared to [6]. We also presented window based processing method and multi-channel decomposition to enable real-time processing. The proposed method was implemented on the Xilinx Virtex-5 FPGA platform. The processing time for one

**Table 5.** Execution cycles for each block

Unit	$N_s$	$N_l$	$N_g$	$N_n$	$N_w$	$N_{gm}$	$N_t$	$N_r$	$N_e$
Cycles	804	827	5	4	818	832	4	53	3211

iteration using 3,200 particles was shown to be only  $314\mu\text{s}$ , which makes this feasible for real-time processing.

## 8. REFERENCES

- [1] D. Cohen, “Magnetoencephalography: Evidence of magnetic fields produced by alpha rhythm currents,” *Science*, vol. 161, pp. 664–666, June 1972.
- [2] M. Hämäläinen and et al., “Magnetoencephalography–theory, instrumentation and applications to noninvasive studies of the working human brain,” *Reviews of Modern Physics*, vol. 65, pp. 413–497, 1993.
- [3] 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.
- [4] E. Somersalo, A. Voutilainen, and J. P. Kaipio, “Non-stationary magnetoencephalography by Bayesian filtering of dipole models,” *Inverse Problems*, vol. 19, pp. 1047–1063, 2004.
- [5] L. Miao and et al., “Multiple sensor sequential tracking of neural activity: Algorithm and FPGA implementation,” in *Asilomar Conf. Signals, Systems and Computers*, November 2010.
- [6] A. Sorrentino and et al., “Dynamical MEG source modeling with multi-target Bayesian filtering,” *Human Brain Mapping*, vol. 30, pp. 1911–1921, June 2009.
- [7] B.-N. Vo, S. Singh, and A. Doucet, “Sequential Monte Carlo implementation of the PHD filter for multi-target tracking,” *Proceedings of the Sixth International Conference of Information Fusion*, vol. 2, pp. 792–799, 2003.
- [8] A. Hyvärinen and E. Oja, “A Fast Fixed-Point Algorithm for Independent Component Analysis,” *Neural Computation*, vol. 9, pp. 1483–1492, 1997.
- [9] A. Pascarella and et al., “Particle filtering, beamforming and multiple signal classification for the analysis of meg time series: A comparison of algorithms,” *Inverse Problems and Imaging*, vol. 4, pp. 169–190, Feb. 2010.
- [10] 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.
- [11] S. Baillet, J. C. Mosher, and R. M. Leahy, “Electromagnetic brain mapping,” *IEEE Signal Processing Magazine*, vol. 18, pp. 14–30, 2001.
- [12] J. Cao and et al., “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.
- [13] S. Fujio, H. Shiomi, and Y. Okamura, “Acceleration of FPGA-based ICA processor for real-time processing,” *International Symposium on Antennas and Propagation (APSURSI)*, pp. 1–4, July 2010.