sensors

MDPI

Article

A Robust Diffusion Estimation Algorithm with Self-Adjusting Step-Size in WSNs

Xiaodan Shao [1,2], Feng Chen [1,*,†], Qing Ye [2] and Shukai Duan [2]

[1] College of Electronic and Information Engineering, School of Mathematics and Statistics, Southwest University, and Key Laboratory of Nonlinear Circuits and Intelligent Information Processing, Chongqing 400715, China; xiaodanshao3@126.com
[2] College of Electronic and Information Engineering, Southwest University, Chongqing 400715, China; qingye6@126.com (Q.Y.); duansk@swu.edu.cn (S.D.)
* Correspondence: fengchenmit@swu.edu.cn
† Current address: Key Laboratory of Nonlinear Circuits and Intelligent Information Processing, No.2, Tiansheng Road, Beibei District, Chongqing 400715, China

Academic Editor: Leonhard M. Reindl
Received: 24 February 2017; Accepted: 7 April 2017; Published: 10 April 2017

Abstract: In wireless sensor networks (WSNs), each sensor node can estimate the global parameter from the local data in a distributed manner. This paper proposed a robust diffusion estimation algorithm based on a minimum error entropy criterion with a self-adjusting step-size, which are referred to as the diffusion MEE-SAS (DMEE-SAS) algorithm. The DMEE-SAS algorithm has a fast speed of convergence and is robust against non-Gaussian noise in the measurements. The detailed performance analysis of the DMEE-SAS algorithm is performed. By combining the DMEE-SAS algorithm with the diffusion minimum error entropy (DMEE) algorithm, an Improving DMEE-SAS algorithm is proposed for a non-stationary environment where tracking is very important. The Improving DMEE-SAS algorithm can avoid insensitivity of the DMEE-SAS algorithm due to the small effective step-size near the optimal estimator and obtain a fast convergence speed. Numerical simulations are given to verify the effectiveness and advantages of these proposed algorithms.

Keywords: robust diffusion estimation; self-adjusting step-size; non-Gaussian noise; wireless sensor networks

1. Introduction

The problem of parameter estimation, which is the indirect determination of the unknown parameters from measurements of other quantities [1–6], is a key issue in the signal processing field. Distributed estimation has become very popular for parameter estimation in wireless sensor networks. The objective is to enable the nodes to estimate a vector of parameters of interest in a distributed manner from the observed data. Distributed estimation schemes over adaptive networks can be mainly classified into incremental strategies [7–9], consensus strategies [10,11], and diffusion strategies [12–22]. In the incremental strategies, data is processed in a cyclic fashion through the network. The consensus strategies rely on the fusion of intermediate estimates of multiple neighboring nodes. In the Diffusion strategies, information is processed at all nodes while the nodes communicate with all their neighbors to share their intermediate estimates. The diffusion strategies are particularly attractive because they are robust, flexible and fully-distributed, such as the diffusion least mean squares (DLMS) algorithm [12]. In this paper, we focus on the diffusion estimation strategies.

The performance of distributed estimation degrades severely when the signals are perturbed by non-Gaussian noise. Non-Gaussian noise may be natural, due to atmospheric phenomena, or man-made, due to either electric machinery present in the operation environment, or multipath telecommunications signals [23–25]. Recently, some researchers focus on improving robustness for non-Gaussian noise of distributed estimation methods. The efforts are mainly directed at searching for a more robust cost function to replace the MSE criterion, which is optimal only when the measurement noise is Gaussian. To address this problem, the diffusion least mean p-power (DLMP) based on p-norm error criterion was proposed to estimate the parameters of the wireless sensor networks [26]. The correntropy as a nonlinear similarity measure has been successfully used as a robust and efficient cost function for non-Gaussian signal processing [27–30]. In [27], two robust MCC based diffusion algorithms, namely the Adapt-then-Combine (ATC) and Combine-then-Adapt (CTA) diffusion maximum correntropy criterion (DMCC) algorithms, are developed to improve the performance of the distributed estimation over network in impulsive noise environments.

The error entropy criterion based on the minimum error entropy (MEE) method also has shown its ability to achieve more accurate estimates than mean-square error (MSE) under non-Gaussian noise [31–37]. In [31], the diffusion minimum error entropy (DMEE) was proposed. The DMEE algorithm achieved improved performance for non-Gaussian noise with the fixed step-size, but it still suffers from conflicting requirements between convergence rate and the steady-state mean square error. A large step-size leads to a fast convergence rate but a large mean-square error at the steady state. For this problem, variable step-size techniques have been widely used to improve the convergence of diffusion LMS algorithms remarkably by adjusting the step-size appropriately [38–41]. Lee et al. [38] proposed a novel variable step-size diffusion LMS algorithm which controls the step-size suboptimally to attain the minimum mean square error at each time instant. In [41], Abdolee investigated the effect of adaptation step-sizes on the tracking performance of DLMS algorithms in networks under non-stationary signal conditions. However, to the best of our knowledge, the variable step-size technique has not been extended to the field of distributed minimum error entropy estimation for non-Gaussian noise yet.

In this paper, we incorporate the minimum error entropy criterion with self-adjusting step-size (MEE-SAS) [42] into the cost function in diffusion distributed estimation. Then, we figure out the diffusion-strategy solutions, which are referred to as the diffusion MEE-SAS (DMEE-SAS) algorithm. Numerical simulation results show that the DMEE-SAS algorithm outperforms DLMS, DLMP and DMEE algorithms when the noise is modeled to be non-Gaussian noise. We also design an Improving DMEE-SAS algorithm by using a switching scheme between DMEE-SAS and DMEE algorithms for a non-stationary environment, which tracks the changing estimator very effectively. The Improving DMEE-SAS algorithm can avoid the small effective step-size of the DMEE-SAS algorithm when it is close to the optimal estimator.

We organize the paper as follows. In Section 2, we briefly revisit the minimization error entropy criterion. In Section 3, firstly, we propose the DMEE-SAS algorithm and analyze the mean, mean square and instantaneous MSD performance for the DMEE-SAS algorithm. Then, we propose the Improving DMEE-SAS algorithm for a non-stationary scenario. Simulation results are shown in Section 4. Finally, we draw conclusions in Section 5.

2. Minimization Error Entropy Criterion

Considering the limited computational capability and limited memory space for nodes in real distributed networks, this paper is based on an MEE criterion, which is simple enough and has good estimation accuracy. Important properties of MEE can be found in [32,35,37]. In many real world applications, the MEE estimator can outperform significantly the well-known MSE estimator and show strong robustness to noises, especially when data are contaminated by non-Gaussian noises. In this subsection, we introduce an MEE criterion, which could be used to derive a robust diffusion estimation algorithm with a self-adjusting step-size (DMEE-SAS) algorithm.

The aim of the adaptive signal processing problem is to minimize the difference between the desired and the system outputs, which is defined as error e. For the evaluation of the error entropy, we seek to estimate entropy directly from the error samples. Therefore, system parameters can be estimated by minimizing the *Renyi's* entropy of the error e. *Renyi's* entropy is given by

$$H_\alpha\left(e\right) = \frac{1}{1-\alpha} log \int q^\alpha\left(e\right) de, \tag{1}$$

where $q^\alpha\left(e\right)$ is the probability density function of a continuous error e, and α is a parameter. When parameter α is set as 2, Equation (1) is quadratic *Renyi's* entropy. Using a Gaussian kernel with kernel size σ, we can obtain a convenient evaluation of the integral operator in the formulation of quadratic *Renyi's* entropy as follows:

$$\begin{aligned} H_2(e) &= -log \int q^2\left(e\right) de \\ &= -log \int \left(\frac{1}{N}\sum_{i=1}^{N} G_\sigma\left(e-e_i\right)\right)^2 de \\ &= -\log(V(e)), \end{aligned} \tag{2}$$

where $e = \left[e_1, e_2, \cdots, e_N\right]$ is N independent and identically distributed samples, and the Gaussian kernel is defined as

$$G_{\sigma\sqrt{2}}(e-e_i) = \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{1}{2\sigma^2}(e-e_i)^2).$$

The information $V(e)$ is quadratic information potential and is defined as the expectation of probability density function, $V\left(e\right) = E\left[q\left(e\right)\right]$. The quadratic information potential $V(e)$ can be easily estimated by using a simple and effective nonparametric estimator

$$V(e) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} G_{\sigma\sqrt{2}}(e_j - e_i) \leq V(0) = \frac{1}{\sigma\sqrt{2\pi}}. \tag{3}$$

The maximum value of the quadratic information potential $V(0)$ will be achieved when $e_1 = e_2 = \cdots = e_N$. The above results are obtained in the case of batch mode, where the N data points are fixed. For online training methods, in order to reduce calculation costs, the estimate of quadratic information potential can be approximated stochastically by dropping the time average in (3), leading to

$$V(e_i) \approx \frac{1}{L}\sum_{j=i-L+1}^{i} G_{\sigma\sqrt{2}}(e_i - e_j), \tag{4}$$

where L is the latest L samples at time i.

Obviously, to minimize the error entropy is equivalent to maximizing the quadratic information potential since the log is a monotonic function. Therefore, the cost function for the MEE criterion is given by

$$J_{MEE}\left(e\right) = max V(e). \tag{5}$$

The selection of the kernel size σ is an important step in estimating the information potential and is critical to the success of information theoretic criteria. In particular, increasing the kernel size leads to a stretching effect on the performance surface in the weight space, which results in increased accuracy of the quadratic approximation around the optimal point [43]. In order to ensure accuracy, in the following, a large enough kernel size can be used during the adaptation process, which is commonly used in information theoretic criteria [42,44].

3. Proposed Algorithms

As mentioned in the Introduction, the diffusion minimum error entropy algorithm achieved improved performance for non-Gaussian noise with the fixed step-size, but it still suffers from conflicting requirements between convergence rate and the steady-state mean square error. Therefore, we consider a new cost function, which can achieve fast convergence speed and strong robustness against non-Gaussian noise.

3.1. Diffusion MEE-SAS Algorithm

Consider a connected wireless sensor networks with K nodes. $k \in \{1, 2, \ldots, K\}$ is the node index and i is the time index. To proceed with the analysis, we assume a liner measurement model as follows:

$$d_{k,i} = u_{k,i}^T w^0 + v_{k,i}, \tag{6}$$

where w^0 is a $M \times 1$ deterministic but unknown vector, $d_{k,i}$ is a scalar measurement of some random process, $u_{k,i}$ is the $M \times 1$ regression vector at time i with zero mean, and $v_{k,i}$ is the random noise signal at time i with zero mean. For each node k, we have

$$V(e_k) = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} G_{\sigma\sqrt{2}}(e_{k,j} - e_{k,i}) \leq V(0) = \frac{1}{\sigma\sqrt{2\pi}}, \tag{7}$$

where $e_{k,i} = d_{k,i} - u_{k,i}^T w$. The maximum value $V(0)$ will be achieved when $e_{k,i} = e_{k,j}$, $j = i - L + 1, i - L + 2, \cdots, i$.

We seek an estimate of w^0 by minimizing a linear combination of local information. As explained in Section 2, minimizing a linear combination of the local information is equivalent to maximizing a linear combination of the local quadratic information potential $V(e_{k,i})$. To maximize the information potential is equivalent to minimizing the following cost function:

$$\begin{aligned} J_k(w) &= \sum_{l \in N_k} c_{l,k} [V(0) - V(e_l)]^2 \\ &= \sum_{l \in N_k} F_l(w), \end{aligned} \tag{8}$$

where

$$F_l(w) = [V(0) - V(e_l)]^2.$$

N_k denotes the one-hop neighbor set of node k, and $\{c_{lk}\}$ are some non-negative cooperative coefficients satisfying $c_{lk} = 0$ if $l \notin N_k$, $1_N^T \bar{C} = 1_N^T$ and $\bar{C} 1_N = 1_N$. Here, $\bar{C}$ is a $N \times N$ matrix with individual entries $\{c_{lk}\}$ and 1_N is a $N \times 1$ all-unity vector. The gradient of the individual local cost function is given by

$$\nabla J_k(w) = \sum_{l \in N_k} c_{lk} f_l(w), \tag{9}$$

where

$$f_l(w) = (\frac{2}{\sigma^2 N^2})(V(0) - V(e_k)) \sum_{i=1}^{N} \sum_{j=1}^{N} G_{\sigma\sqrt{2}}(e_{k,j} - e_{k,i})(e_{k,j} - e_{k,i})(u_{k,i} - u_{k,j}). \tag{10}$$

We can replace the estimate of quadratic information potential by the stochastic quadratic information potential, leading to

$$\nabla \hat{J}_k(w) = \sum_{l \in N_k} c_{lk} \hat{f}_l(w), \tag{11}$$

where

$$\hat{f}_l(w) = (\frac{2}{\sigma^2 L})(V(0) - V(e_{k,i})) \sum_{j=i-L+1}^{i} G_{\sigma\sqrt{2}}(e_{k,i} - e_{k,j})(e_{k,i} - e_{k,j})(u_{k,j} - u_{k,i}), \tag{12}$$

where

$$V(e_{l,i}) \approx \frac{1}{L} \sum_{j=i-L+1}^{i} G_{\sigma\sqrt{2}}(e_{l,i} - e_{l,j}). \tag{13}$$

Iterative steepest-descent solution for estimating w^0 at each node k can thus be derived as

$$\begin{aligned} w_{k,i+1} &= w_{k,i} - \mu_k \nabla \hat{J}_k(w) \\ &= w_{k,i} - \mu_k \frac{2}{\sigma^2 L} \sum_{l \in N_k} c_{lk} [V(0) - V(e_{l,i})] \sum_{j=i-L+1}^{i} G_{\sigma\sqrt{2}}(e_{l,i} - e_{l,j})(e_{l,i} - e_{l,j})(u_{l,j} - u_{l,i}), \end{aligned} \tag{14}$$

where μ_k is a positive step size. Using the general framework for diffusion-based distributed adaptive optimization [13], an adapt-then-combine (ATC) strategy for diffusion MEE-SAS algorithm can be formulated as

$$\begin{cases} \varphi_{k,i+1} = w_{k,i} - \mu_k \frac{2}{\sigma^2 L} [V(0) - V(e_{k,i})] \sum_{j=i-L+1}^{i} G_{\sigma\sqrt{2}}(e_{k,i} - e_{k,j})(e_{k,i} - e_{k,j})(u_{k,j} - u_{k,i}), \\ w_{k,i+1} = \sum_{l \in N_k} c_{lk} \varphi_{l,i+1}. \end{cases} \tag{15}$$

According to Equation (15), the DMEE-SAS algorithm can be seen as a diffusion estimation algorithm with variable step size $\mu_k(i)$, where

$$\mu_k(i) = 2\mu_k [V(0) - V(e_{k,i})]. \tag{16}$$

The DMEE-SAS algorithm is described formally in Algorithm 1.

Algorithm 1: DMEE-SAS Algorithm

```
Initialize: $w_{k,i} = 0$
for $i = 1 : T$
for each node $k$:
  Adaptation
  $\mu_k(i) = 2\mu_k [V(0) - V(e_{k,i})]$
  $\varphi_{k,i+1} = w_{k,i} - \mu_k(i) \frac{1}{\sigma^2 L} \sum_{j=i-L+1}^{i} G_{\sigma\sqrt{2}}(e_{k,i} - e_{k,j})(e_{k,i} - e_{k,j})(u_{k,j} - u_{k,i})$
  Combination
  $w_{k,i+1} = \sum_{l \in N_k} c_{lk} \varphi_{l,i+1}$
end for
```

In the adaption step of DMEE-SAS algorithm, $V(0) - V(e_{k,i})$ is close to $V(0)$ when the algorithm starts, and it is close to 0 when the algorithm begins to converge. $V(0) - V(e_{k,i})$ is always a non-negative scalar quantity, which can accelerate the rate of convergence and achieve small steady-state estimation errors. The fast convergence rate and the small steady-state estimation errors of the DMEE-SAS algorithm can be established against non-Gaussian noise in the measurements.

3.2. Performance Analysis

In this section, we analyze the mean, mean-square and instantaneous MSD performance of the DMEE-SAS algorithm. For tractability of the analysis, here we focus on the case of batch mode. To briefly present the convergence property of the proposed algorithm in terms of global quantities, the following notations are introduced: $M = diag\{\mu_1 I_M, \ldots, \mu_K I_M\}$, $W_i = col\{w_{1,i}, \cdots w_{K,i}\}$, $w^{(0)} = col\{w^0, \cdots, w^0\}$, $\tilde{W}_i = col\{\tilde{w}_{1,i} \cdots \tilde{w}_{K,i}\}$, $S = col\{s_1(w^0), \cdots, s_K(w^0)\}$, $C = \bar{C}^T \otimes I_M$, I_M is the identity matrix.

In order to make the analysis tractable, the followings are assumed:

Assumption 1: The regressor $u_{k,i}$ is independent identically distributed (i.i.d) in time and spatially independent, and $E[u_{k,i}] = 0$, $R_k = E[u_{k,i}^T u_{k,i}]$.

Assumption 2: The input noise $v_{k,i}$ is super-Gaussian noise. In addition, $v_{k,i}$ and the regressor $u_{k,i}$ are independent from each other. We have $E[v_{k,i}] = 0$ and $E[v_{k,i}^2] = \xi_k$.

Assumption 3: The step-sizes, μ_k, $\forall k$, are small enough such that their squared values are negligible.

3.2.1. Mean Performance

Because the input signal and output noises are generated from stationary and ergodic processes, the double time average in Equation (10) can be replaced by the expectation, leading to

$$f_k(w) \approx \frac{2}{\sigma^2}(V(0) - E[G_{\sigma\sqrt{2}}(e_{k,j} - e_{k,i})])E[G_{\sigma\sqrt{2}}(e_{k,j} - e_{k,i})(e_{k,j} - e_{k,i})(u_{k,i} - u_{k,j})]. \tag{17}$$

We consider the gradient error caused by approximating the quadratic information potential $V(e_{k,i})$ with their instantaneous values [45]. The gradient error at iteration i and each node k is defined as follows:

$$s_k(w_{k,i}) = \hat{f}_k(w_{k,i}) - f_k(w_{k,i}). \tag{18}$$

Using Equation (15), the update equation of the intermediate estimate can be rewritten as

$$\varphi_{k,i+1} = w_{k,i} - \mu_k(f_k(w_{k,i}) + s_k(w_{k,i})). \tag{19}$$

According to [44], when input signal-to-noise ratio is not too low, the error should be small on the whole. Therefore, for a relative large kernal size σ, when $w = w^0$, $((e_{k,i} - e_{k,j})/\sigma \approx 0$ and $G_{\sigma\sqrt{2}}(e_{k,i} - e_{k,j}) \approx \frac{1}{\sigma\sqrt{2\pi}}$. Therefore, the Hessian matrix function $H_k(w^0)$ of $F_l(w)$ is calculated as:

$$\begin{aligned}
H_k(w^0) &= \frac{\partial f_k(w)}{\partial w}|_{w^0} \\
&= \frac{\partial}{\partial w}\frac{2}{\sigma^2}(V(0) - E[G_{\sigma\sqrt{2}}(e_{k,j} - e_{k,i})])E[G_{\sigma\sqrt{2}}(e_{k,j} - e_{k,i})(e_{k,j} - e_{k,i})(u_{k,i} - u_{k,j})] \\
&= \frac{2}{\sigma^2}(V(0) - E[G_{\sigma\sqrt{2}}(e_{k,j} - e_{k,i})])E[G_{\sigma\sqrt{2}}(e_{k,j} - e_{k,i})(u_{k,i} - u_{k,j})^T(u_{k,i} - u_{k,j}) - \\
&\quad \frac{1}{\sigma^2}G_{\sigma\sqrt{2}}(e_{k,j} - e_{k,i})(e_{k,j} - e_{k,i})^2(u_{k,i} - u_{k,j})^T(u_{k,i} - u_{k,j})] \\
&\quad + \frac{2}{\sigma^4}E[G_{\sigma\sqrt{2}}(e_{k,j} - e_{k,i})(e_{k,j} - e_{k,i})(u_{k,i} - u_{k,j})]^T E[G_{\sigma\sqrt{2}}(e_{k,j} - e_{k,i})(e_{k,j} - e_{k,i})(u_{k,i} - u_{k,j})] \\
&= \frac{1}{\sigma^6\pi}E[v_{k,i}^2 + v_{k,j}^2]E[u_{k,i}^T u_{k,i} + u_{k,j}^T u_{k,j}] \\
&= \frac{4\xi_k R_k}{\sigma^6\pi}.
\end{aligned} \tag{20}$$

Based on the Theorem 1.2.1 of [46], we obtain

$$\begin{aligned}
f_k(w_{k,i}) &= f_k(w^0) - (\int_0^1 H_k(w^0 - x\tilde{w}_{k,i})dx)\tilde{w}_{k,i} \\
&= -(\int_0^1 H_k(w^0 - x\tilde{w}_{k,i})dx)\tilde{w}_{k,i},
\end{aligned} \tag{21}$$

where $\tilde{w}_{k,i} = w^0 - w_{k,i}$ is the weight error vector for node k. We assume that the estimate of each node converges to the vicinity of the unknown vector w^0. Therefore, $\tilde{w}_{k,i}$ is small enough such that it is negligible, yielding

$$\begin{aligned} f_k(w_{k,i}) &\approx -(\int_0^1 H_k(w^0)dx)\tilde{w}_{k,i} \\ &= -H_k(w^0)\tilde{w}_{k,i}. \end{aligned} \tag{22}$$

We can also obtain the approximation of the gradient error at the vicinity of w^0, which is given by

$$\begin{aligned} s_k(w_{k,i}) &\approx s_k(w^0) \\ &= \hat{f}_k(w^0) - f_k(w^0) \\ &= \frac{2}{\sigma^2 L}(V(0) - \frac{1}{L}\sum_{j=i-L+1}^{i} G_{\sigma\sqrt{2}}(v_{l,i} - v_{l,j})) \sum_{j=i-L+1}^{i} \\ &\quad G_{\sigma\sqrt{2}}(v_{k,i} - v_{k,j})(v_{k,i} - v_{k,j})(u_{k,j} - u_{k,i}). \end{aligned} \tag{23}$$

Substituting Equations (22) and (23) into Equation (19), an approximation of intermediate estimate can be obtained at the vicinity of

$$\varphi_{k,i+1} = w_{k,i} + \mu_k(H_k(w^0)\tilde{w}_{k,i} - s_k(w^0)). \tag{24}$$

By substituting Equation (24) into the second equation of Equation (15), we get the estimate of unknown parameter as follows:

$$w_{k,i+1} = \sum_{l \in N_k} c_{lk}[w_{l,i} + \mu_k(H_k(w^0)\tilde{w}_{l,i} - s_l(w^0))]. \tag{25}$$

Using global quantities defined above gives the update equation for the network estimate vector as

$$W_{i+1} = C(W_i + MH\tilde{W}_i - MS), \tag{26}$$

where H collects the Hessian matrix across the network into the global vector $H = diag(H_1(w^0), \cdots, H_N(w^0))$. Noting that $Cw^{(0)} = w^{(0)}$, subtraction of both sides of Equation (26) from $w^{(0)}$ gives

$$\tilde{W}_{i+1} = C(I_{MN} - MH)\tilde{W}_i + CMS. \tag{27}$$

In view of assumptions A1 and A2, $\tilde{W}_i$, H and C are independent of each other. Hence, taking expectation of both sides of Equation (27) leads to

$$E[\tilde{W}_{i+1}] = E[C](I_{MN} - MH)E[\tilde{W}_i] + CME[S]. \tag{28}$$

We can easily find that $E[S] = col\{E[s_1(w^0), \cdots, s_N(w^0)]\} = 0$, and Equation (28) has therefore been reduced to this form

$$E[\tilde{W}_{i+1}] = E[C](I_{MN} - MH)E[\tilde{W}_i]. \tag{29}$$

From Equation (29), we observe that, in order to be stable for Algorithm 1 in the mean sense, the matrix $E[C](I_{MN} - MH)$ should be stable. All the entries of $E(C)$ are non-negative and all the rows of it add up to unity. Therefore, to ensure the stability in the mean, it should hold that

$$|\lambda_{\max}\{I_{MN} - MH\}| < 1. \tag{30}$$

We use the notion $\lambda_{\max}(A)$ to denote the maximum eigenvalue of a Hermitian matrix A. Thus, we note that a sufficient condition for unbiasedness is

$$0 < \mu_k < \frac{2}{\lambda_{\max}\{H_k(w^0)\}} = \frac{\sigma^6 \pi}{2\lambda_{max}\{R_k \xi_k\}}. \tag{31}$$

3.2.2. Mean-Square Performance

In order to make the presentation clearer, we shall introduce the following notation

$$\mathbf{\Gamma} = (I_{MN} - MH)C^T \mathbf{\Sigma} C(I_{MN} - MH).$$

Performing weighted energy balance on both sides of Equation (27) and taking expectations gives

$$E[\|\tilde{W}_{i+1}\|^2_{\Sigma}] = E[\|\tilde{W}_i\|^2_{\mathbf{\Gamma}}] + E[S^T MC^T \mathbf{\Sigma} CMS], \tag{32}$$

where $\mathbf{\Sigma}$ is an arbitrary symmetric nonnegative-definite matrix, and the notion $\|a\|^2_{\mathbf{\Sigma}} = a^T \mathbf{\Sigma} a$ represents a weighted vector norm for any Hermitian $\mathbf{\Sigma} > 0$. By defining

$$r = vec\{E[\mathbf{\Gamma}]\}, \theta = vec\{\mathbf{\Sigma}\},$$

where the vec(.) notation stacks the columns of its matrix argument on top of each other. We can modify Equation (32) to

$$E[\|\tilde{W}_{i+1}\|^2_{\theta}] = E[\|\tilde{W}_i\|^2_{r}] + E[S^T MC^T \mathbf{\Sigma} CMS]. \tag{33}$$

Using the following relationship of the vectorization operator and the Kronecker product [47]:

$$vec(ABC) = (C^T \otimes A)vec\{B\}.$$

We can obtain that

$$r = \phi\theta, \tag{34}$$

where

$$\phi = E[(I_{MN} - MH) \otimes (I_{MN} - MH)]\beta, \tag{35}$$
$$\beta = E[C^T \otimes C^T].$$

Considering Assumption 3, we can approximate Equation (35) as

$$\begin{aligned} \phi &\approx (I_{M^2N^2} - I_{MN} \otimes MH - MH \otimes I_{MN})\beta \\ &= (I_{MN} - MH) \otimes (I_{MN} - MH)\beta. \end{aligned} \tag{36}$$

Using the following relationship of the vectorization operator and the matrix trace [47]:

$$Tr\{A^T B\} = vec^T(B)vec(A).$$

We find that

$$E[S^T MC^T \mathbf{\Sigma} CMS] = vec^T\{Q\}\beta\theta, \tag{37}$$

where

$$Q = E[MSS^T M].$$

Substituting Equations (34) and (37) into Equation (33), we can then reformulate recursion as follows:

$$E[\|\tilde{W}_{i+1}\|^2_{\theta}] = E[\|\tilde{W}_i\|^2_{\phi\theta}] + vec^T\{Q\}\beta\theta. \tag{38}$$

It is known that Equation (38) is stable and convergent if the matrix ϕ is stable [48], form the Equation

$$\beta^T 1_{M^2N^2} = E[(C \otimes I_M)1_{MN} \otimes (C \otimes I_M)1_{MN}] = 1_{M^2N^2},$$

We know that all the entries of β in Equation (37) are non-negative, and all its columns sum up to unity. Using the property $\lambda(\mathbf{A} \otimes \mathbf{A}) = \lambda^2(\mathbf{A})$, the stability of ϕ has the same conditions as the stability of $I_{MN} - MH$. Therefore, we choose the step size in accordance with Equation (31), which can keep the DMEE-SAS stable in the mean-square sense.

3.2.3. Instantaneous MSD

In order to analyze instantaneous mean-square-error (MSD), we can exploit the liberty of choosing θ at time i. Then, Expression (38) gives:

$$E[\|\tilde{W}_{n+1}\|^2_{\phi^{i-n}\theta}] = E[\|\tilde{W}_n\|^2_{\phi^{i-n+1}\theta}] + vec^T\{Q\}\,\beta\phi^{i-n}\theta. \tag{39}$$

The sum of both sides of Equation (39) for $n = 0, 1, ..., i-1$ can be given by

$$E[\|\tilde{W}_i\|^2_{\theta}] = E[\|\tilde{W}_0\|^2_{\phi^{i}\theta}] + vec^T\{Q\}\sum_{n=0}^{i-1}\beta\phi^{n}\theta. \tag{40}$$

We can also adopt a similar way to describe the time instant $i+1$, given by

$$E[\|\tilde{W}_{i+1}\|^2_{\theta}] = E[\|\tilde{W}_0\|^2_{\phi^{i+1}\theta}] + vec^T\{Q\}\sum_{n=0}^{i}\beta\phi^{n}\theta. \tag{41}$$

Subtraction of both sides of Equation (40) from Equation (41) gives

$$E[\|\tilde{W}_{i+1}\|^2_{\theta}] = E[\|\tilde{W}_i\|^2_{\theta}] - E[\|\tilde{W}_0\|^2_{\phi^{i}(I_{M^2N^2}-\phi)\theta}] + vec^T\{Q\}\,\beta\phi^{n}\theta. \tag{42}$$

By setting

$$\theta = vec\{I_{MN}\}$$

in Equation (44) and dividing both sides of it by N, the instantaneous MSD for the whole network are computed by:

$$\frac{1}{N}\sum_{k=1}^{N} E\left[\|\tilde{w}_{k,i}\|^2\right] = \frac{1}{N}E\left[\|\tilde{W}_i\|^2\right], \tag{43}$$

where $\tilde{W}_i$ can be obtained by the following iteration:

$$\frac{1}{N}E[\|\tilde{W}_{i+1}\|^2] = \frac{1}{N}E[\|\tilde{W}_i\|^2] - \frac{1}{N}E[\|\tilde{W}_0\|^2_{\phi^{i}(I_{M^2N^2}-\phi)vec\{I_{MN}\}}] + \frac{1}{N}vec^T\{Q\}\,\beta\phi^{i}vec\{I_{MN}\}. \tag{44}$$

3.3. An Improving Scheme for the DMEE-SAS Algorithm

The too small effective step size near the optimal estimator will hinder the tracking ability of the DMEE-SAS algorithm in a non-stationary environment. In a non-stationary environment, the optimal estimator has small changes. A random-walk model is commonly used in the literature to describe the non-stationarity of the weight vector [48].

Therefore, we try to combine the DMEE-SAS algorithm with the DMEE algorithm [31] in a non-stationary environment where tracking is important. The DMEE-SAS algorithm should be used due to the faster convergence when the algorithm starts, and the DMEE algorithm should be used when the algorithm begins to converge. We use the Lyapunov stability theory [49] to analyze the switching time for each node.

The Lyapunov energy function is a method for analyzing the convergence characteristics of dynamic systems. The cost function can be viewed as a Lyapunov energy function. For the DMEE-SAS algorithm, the continuous-time learning rule is

$$\dot{w} = -\mu_{DMEE-SAS} \frac{\partial J_k(w)_{DMEE-SAS}}{\partial w}. \tag{45}$$

The temporal dynamics for the Lyapunov energy that describes the DMEE-SAS algorithm can be obtained as follows:

$$\begin{aligned} \dot{J}_k(w)_{DMEE-SAS} &= \sum_{l \in N_k} c_{lk}(-2)[V(0) - V(e_{l,i})] \frac{\partial V(e_{l,i})^T}{\partial w} \dot{w} \\ &= \sum_{l \in N_k} c_{lk}(-4)\mu_{k,DMEE-SAS}[V(0) - V(e_{l,i})]^2 \left\| \frac{\partial V(e_{l,i})}{\partial w} \right\|^2. \end{aligned} \tag{46}$$

The individual local energy function for DMEE algorithm can be written as

$$J_k(w)_{DMEE} = -\sum_{l \in N_k} c_{lk} V(e_{l,i}). \tag{47}$$

For the DMEE algorithm, the continuous-time learning rule is

$$\dot{w} = -\mu_{DMEE} \frac{\partial J_k(w)_{DMEE}}{\partial w}. \tag{48}$$

In a similar way, the temporal dynamics for the Lyapunov energy that describes the DMEE algorithm can be obtained as follows:

$$\begin{aligned} \dot{J}_k(w)_{DMEE} &= \sum_{l \in N_k} c_{lk} \frac{\partial V(e_{l,i})^T}{\partial w} \dot{w} \\ &= \sum_{l \in N_k} c_{lk}(-\mu_{l,DMEE}) \left\| \frac{\partial V(e_{l,i})}{\partial w} \right\|^2. \end{aligned} \tag{49}$$

The switching time is determined as

$$\left|\dot{J}_k(w)_{DMEE-SAS}\right| > \left|\dot{J}_k(w)_{DMEE}\right| \Leftrightarrow V(e_{l,i}) < V(0) - \frac{1}{2}\sqrt{\frac{\mu_{l,DMEE}}{\mu_{l,DMEE-SAS}}} (l \in N_k). \tag{50}$$

When the condition of Equation (50) is met, we should switch from the DMEE-SAS algorithm to the DMEE-SAS algorithm. We introduce the following auxiliary variable:

$$s_{k,i} = \begin{cases} 1, V(e_{k,i}) < V(0) - \frac{1}{2}\sqrt{\frac{\mu_{k,DMEE}}{\mu_{k,DMEE-SAS}}}, \\ 0, \text{otherwise}. \end{cases}$$

This yields the following algorithm, which we refer to as the improving DMEE-SAS algorithm:

$$\begin{cases} \varphi_{k,i+1} = w_{k,i} - s_{k,i}\mu_{k,DMEE-SAS} \frac{2}{\sigma^2 L}[V(0) - V(e_{k,i})] \sum\limits_{j=i-L+1}^{i} G_{\sigma\sqrt{2}}(e_{k,i} - e_{k,j})(e_{k,i} - e_{k,j}) \\ \qquad (u_{k,j} - u_{k,i}) - (1 - s_{k,i})\mu_{k,DMEE} \frac{1}{\sigma^2 L} \sum\limits_{j=i-L+1}^{i} G_{\sigma\sqrt{2}}(e_{k,i} - e_{k,j})(e_{k,i} - e_{k,j})(u_{k,j} - u_{k,i}) \\ w_{k,i+1} = \sum\limits_{l \in N_k} c_{lk}\varphi_{l,i+1}. \end{cases} \tag{51}$$

For the purpose of clarity, we summarize the procedure of the Improving DMEE-SAS algorithm in Algorithm 2.

Algorithm 2: Improving DMEE-SAS Algorithm

```
Initialize:
$w_{k,i} = 0$
  for $i = 1 : T$
for each node $k$:
  Adaptation
  each node calculates the switching time using Equation (50).
  each node updates intermediate estimate $\varphi_{k,i}$ according to the first equation of
Equation (51).
  Combination
  $w_{k,i+1} = \sum_{l \in N_k} c_{lk} \varphi_{l,i+1}$
end for
```

4. Simulation Results

Twenty sensors are randomly placed in a square 100 × 100 shown in Figure 1. The communication distance is set as 50. In this paper, the performance of the steady-state network MSD [12] is adopted for performance comparison. All of the performance measures are averaged over 100 trials.

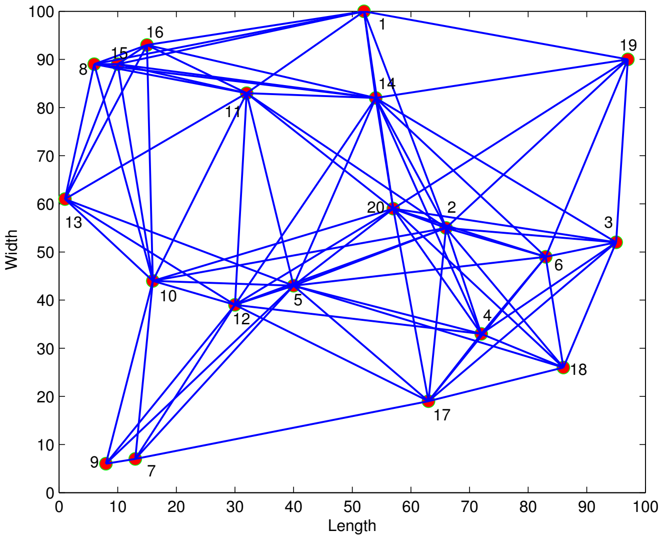

Figure 1. Network topology.

We employ the super-Gaussian distribution as the noise model in our simulations. We generate the noise from the zero-mean generalized Gaussian distribution of probability density function $q_V(v) =\propto \exp(-|v|^p)$, where p is a positive shape parameter of probability density function [50]. We set $p = 0.6$ to make the noise distribution be super-Gaussian.

(a) In Stationary Environment

Here, the proposed DMEE-SAS algorithm performance is compared with that of some existing algorithms in the literature. We assume the communication link is the ideal link. The unknown parameter vector w^0 is set to $[\frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}]^T$.

We set the window length $L = 8$ and kernel size σ=1.5 for both DMEE and DMEE-SAS algorithms. Furthermore, the p is 1.2 for the DLMP algorithm. The steady state MSD curves are plotted in Figure 2. It is found that the DMEE-SAS algorithm is robust to the non-Gaussian noises and performs better than the DLMP algorithm [26] and DLMS [12]. The DMEE-SAS algorithm achieves a better convergence performance than the DMEE [31] algorithm when the DMEE-SAS and DMEE algorithms achieve comparable performance.

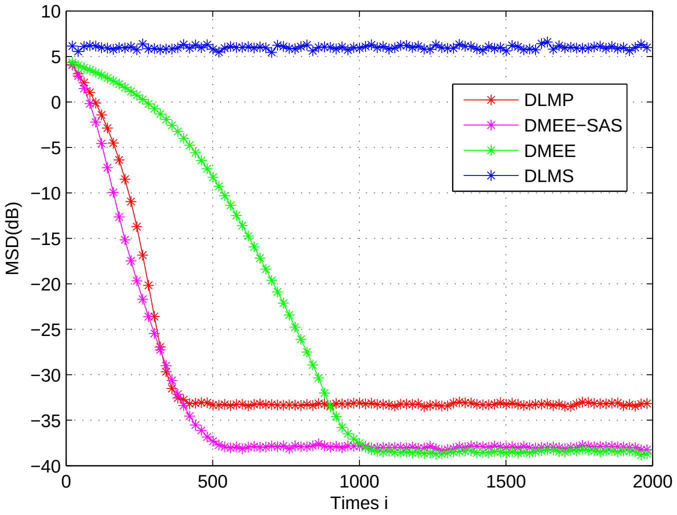

Figure 2. Transient mean-square-error (MSD) curves.

(b) In Non-stationary Environment

Here, the simulations are carried out in the same environments as those shown in stationary environment, except for the optimal estimator w^0. We compare the proposed Improving DMEE-SAS algorithm with other algorithms.

Motivated by [51], we assume a time-varying w^0 of length 6 as follows:

$$w_i^0 = \frac{1}{2}[a_{1,i}, a_{2,i}, a_{3,i}, a_{4,i}, a_{5,i}, a_{6,i}]^T,$$

where $a_{k,i} = [\cos(wi + \frac{(k-1)}{2}\pi)]$ for $k = 1, 2, 3, 4, 5, 6$ and $w = \frac{\pi}{3000}$.

The unknown link is assumed to change at time 6000. In Figure 3, the Improving DMEE-SAS algorithm can detect the weight vector change and the performance of it is better than the DLMS algorithm. We observe that Improving DMEE-SAS and DMEE algorithms achieve comparable performance and Improving DMEE-SAS achieves better convergence performance than the DMEE algorithm. When compared with the DMEE-SAS algorithm, the Improving DMEE-SAS algorithm exhibits a significant improvement in performance when the estimate is close to the optimal estimator. The Improving DMEE-SAS algorithm achieves a low MSD and fast rate of convergence in the non-stationary environment.

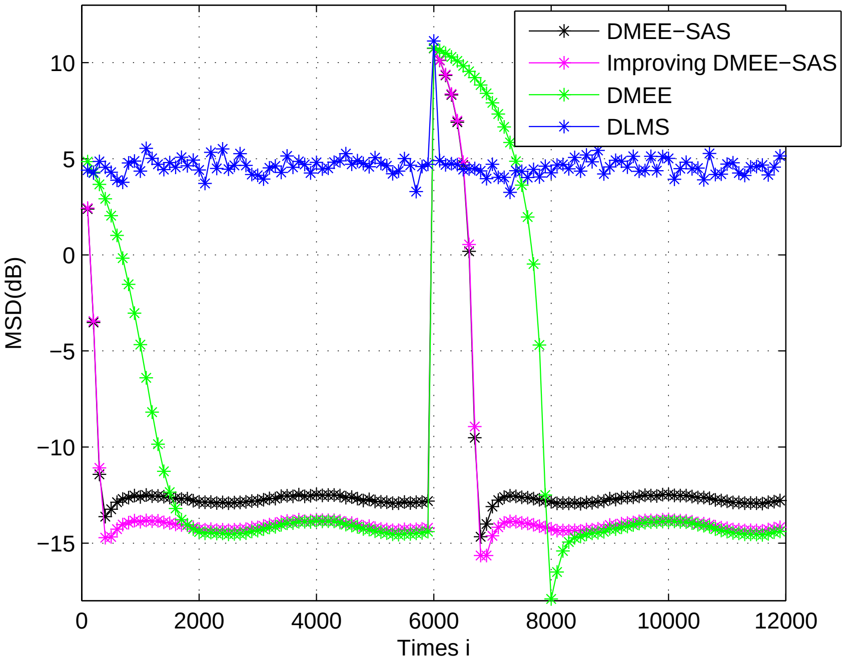

Figure 3. MSD learning curves in a non-stationary environment.

5. Conclusions

In this paper, a robust diffusion estimation algorithm with self-adjusting step-size is developed which is called the DMEE-SAS algorithm. The mean and mean square convergence analysis of this new algorithm are carried out, and a sufficient condition for ensuring the stability is obtained. Simulation results illustrate that the DMEE-SAS algorithm can achieve better performance than the DLMS, robust DLMP, and DMEE algorithms in non-Gaussian noise scenario. In addition, we propose the Improving DMEE-SAS algorithm, using it in the non-stationary scenario where the unknown parameter is changing over time. The Improving DMEE-SAS algorithm combined the DMEE-SAS algorithm with the DMEE algorithm, and it can avoid the small effective step-size of the DMEE-SAS algorithm when close to the optimal estimator.

Acknowledgments: This work was supported in part by the Doctoral Fund of Southwest University (No. SWU113067), the National Natural Science Foundation of China (Grant Nos. 61672436, 61571372, 61372139, 61101233, 60972155), Fundamental Research Funds for the Central Universities (Grant Nos. XDJK2014A009, XDJK2016A001, XDJK2014C018) and Program for New Century Excellent Talents in University (Grant Nos. 201347).

Author Contributions: Xiaodan Shao conceived and designed the study and wrote the paper; Feng Chen did the simulations and gave the data analysis of the proposed algorithms. Qing Ye and Xiaodan Shao revised the paper; Shukai Duan checked the algorithms mentioned in this paper and helped to polish the English; all authors approved the final manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wang, G.; Li, C.; Dong, L. Noise estimation using mean square cross prediction error for speech enhancement. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2010**, *57*, 1489–1499.
2. Xia, Y.; Jahanchahi, C.; Nitta, T. Performance bounds of quaternion estimators. *IEEE Trans. Neural Netw. Learn. Syst.* **2015**, *26*, 3287–3292.
3. Xia, Y.; Douglas, S,C.; Mandic, D.P. Adaptive frequency estimation in smart grid applications: Exploiting noncircularity and widely linear adaptive estimators. *IEEE Signal Process. Mag.* **2012**, *29*, 44–54.

4. Liu, Y.; Li, C. Complex-valued Bayesian parameter estimation via Markov chain Monte Carlo. *Inf. Sci.* **2016**, *326*, 334–349.
5. Xia, Y.; Mandic, D.P. Widely linear adaptive frequency estimation of unbalanced three-phase power systems. *IEEE Trans. Instrum. Meas.* **2012**, *61*, 74–83.
6. Xia, Y.; Blazic, Z.; Mandic, D.P. Complex-valued least squares frequency estimation for unbalanced power systems. *IEEE Trans. Instrum. Meas.* **2015**, *64*, 638–648.
7. Lopes, C.G.; Sayed, A.H. Incremental Adaptive Strategies over Distributed Networks. *IEEE Trans. Signal Process.* **2007**, *55*, 4064–4077.
8. Rabbat, M.G.; Nowak, R.D. Quantized incremental algorithms for distributed optimization. *IEEE J. Sel. Areas Commun.* **2006**, *23*, 798–808.
9. Nedic, A.; Bertsekas, D.P. Incremental subgradient methods for nondifferentiable optimization. *SIAM J. Optim.* **2001**, *12*, 109–138.
10. Kar, S.; Moura, J.M.F. Distributed Consensus Algorithms in Sensor Networks With Imperfect Communication: Link Failures and Channel Noise. *IEEE Trans. Signal Process.* **2009**, *57*, 355–369.
11. Carli, R.; Chiuso, A.; Schenato, L. Distributed Kalman filtering based on consensus strategies. *IEEE J. Sel. Areas Commun.* **2008**, *26*, 622–633.
12. Cattivelli, F.S.; Sayed, A.H. Diffusion LMS Strategies for Distributed Estimation. *IEEE Trans. Signal Process.* **2010**, *58*, 1035–1048.
13. Chen, J.; Sayed, A.H. Diffusion adaptation strategies for distributed optimization and learning over networks. *IEEE Trans. Signal Process.* **2012**, *60*, 4289–4305.
14. Chen, F.; Shao, X. Broken-motifs Diffusion LMS Algorithm for Reducing Communication Load. *Signal Process.* **2017**, *133*, 213–218.
15. Xia, Y.; Mandic, D.P.; Sayed, A.H. An Adaptive Diffusion Augmented CLMS Algorithm for Distributed Filtering of Noncircular Complex Signals. *IEEE Signal Process. Lett.* **2011**, *18*, 659–662.
16. Kanna, S.; Mandic, D.P. Steady-State Behavior of General Complex-Valued Diffusion LMS Strategies. *IEEE Signal Process. Lett.* **2016**, *23*, 722–726.
17. Kanna, S.; Talebi, S.P.; Mandic, D.P. Diffusion widely linear adaptive estimation of system frequency in distributed power grids. In Proceedings of the IEEE International Energy Conference, Leuven, Belgium, 8 April 2014; pp. 772–778.
18. Jahanchahi, C.; Mandic, D.P. An adaptive diffusion quaternion LMS algorithm for distributed networks of 3D and 4D vector sensors. In Proceedings of the IEEE Signal Processing Conference, Napa, CA, USA, 11 August 2013; pp. 1–5.
19. Abdolee, R.; Champagne, B.; Sayed, A.H. A diffusion LMS strategy for parameter estimation in noisy regressor applications. In Proceedings of the Signal Processing Conference, Ann Arbor, MI, USA, 5 August 2012; pp. 749–753.
20. Abdolee, R.; Champagne, B. Diffusion LMS Strategies in Sensor Networks With Noisy Input Data. *IEEE/ACM Trans. Netw.* **2016**, *24*, 3–14.
21. Abdolee, R.; Champagne, B.; Sayed, A. Diffusion Adaptation over Multi-Agent Networks with Wireless Link Impairments. *IEEE Trans. Mob. Comput.* **2016**, *15*, 1362–1376.
22. Kanna, S.; Dini, D.H.; Xia, Y. Distributed Widely Linear Kalman Filtering for Frequency Estimation in Power Networks. *IEEE Trans. Signal Inf. Process. Netw.* **2015**, *1*, 45–57.
23. Middleton, D. Non-Gaussian noise models in signal processing for telecommunications: New methods an results for class A and class B noise models. *IEEE Trans. Inf. Theory* **1999**, *45*, 1129–1149.
24. Zoubir, A.M.; Koivunen, V.; Chakhchoukh, Y.; Muma, M. Robust estimation in signal processing: A tutorial-style treatment of fundamental concepts. *IEEE Signal Process. Mag.* **2012**, *29*, 61–80.
25. Zoubir, A.M.; Brcich, R.F. Multiuser detection in heavy tailed noise. *Digit. Signal Process.* **2002**, *12*, 262–273.
26. Wen, F. Diffusion least-mean P-power algorithms for distributed estimation in alpha-stable noise environments. *Electron. Lett.* **2013**, *49*, 1355–1356.
27. Ma,W.; Chen, B.; Duan, J.; Zhao, H. Diffusion maximum correntropy criterion algorithms for robust distributed estimation. *Digit. Signal Process.* **2016**, *58*, 10–19.
28. Chen, B.; Xing, L.; Zhao, H.; Zheng, N.; Principe, J.C. Generalized correntropy for robust adaptive filtering. *IEEE Trans. Signal Process.* **2016**, *64*, 3376–3387.

29. Ma,W.; Qu, H.; Gui, G.; Xu, L.; Zhao, J.; Chen, B. Maximum correntropy criterion based sparse adaptive filtering algorithms for robust channel estimation under non-Gaussian environments. *J. Frankl. Inst.* **2015**, *352*, 2708–2727.
30. Chen, B.; Xing, L.; Liang, J.; Zheng, N.; Principe, J.C. Steady-state Mean-square Error Analysis for Adaptive Filtering under the Maximum Correntropy Criterion. *IEEE Signal Process. Lett.* **2014**, *21*, 880–884.
31. Li, C.; Shen, P. ; Liu, Y.; Zhang, Z. Diffusion information theoretic learning for distributed estimation over network. *IEEE Trans. Signal Process.* **2013**, *61*, 4011–4024.
32. Chen, B.; Zhu, Y.; Hu, J. Mean-Square Convergence Analysis of ADALINE Training with Minimum Error Entropy Criterion. *IEEE Trans. Neural Netw.* **2010**, *21*, 1168–1179.
33. Chen, B.; Hu, H.; Pu, L.; Sun, Z. Stochastic gradient algorithm under (h, phi)-entropy criterion. *Circuits Syst. Signal Process.* **2007**, *26*, 941–960.
34. Chen, B.; Zhu, P.; Principe, J.C. Survival information potential: A new criterion for adaptive system training. *IEEE Trans. Signal Process.* **2012**, *60*, 1184–1194.
35. Chen, B.; Principe, J.C. Some further results on the minimum error entropy estimation. *Entropy* **2012**, *14*, 966–977.
36. Chen, B.; Yuan, Z.; Zheng, N.; Principe, J.C. Kernel minimum error entropy algorithm. *Neurocomputing* **2013**, *121*, 160–169.
37. Chen, B.; Xing, L.; Xu, B.; Zhao, H.; Principe, J.C. Insights into the Robustness of Minimum Error Entropy Estimation. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, 1–7, doi:10.1109/TNNLS.2016.2636160.
38. Lee, H.S.; Kim, S.E.; Lee, J.W. A Variable Step-Size Diffusion LMS Algorithm for Distributed Estimation. *IEEE Trans. Signal Process.* **2015**, *63*, 1808–1820.
39. Ghazanfari-Rad, S.; Labeau, F. Optimal variable step-size diffusion LMS algorithms. *IEEE Stat. Signal Process. Workshop* **2014**, 464–467, doi:10.1109/SSP.2014.6884676.
40. Ni, J.; Yang, J. Variable step-size diffusion least mean fourth algorithm for distributed estimation. *Signal Process.* **2016**, *122*, 93–97.
41. Abdolee, R.; Vakilian, V.; Champagne, B. Tracking Performance and Optimal Adaptation Step-Sizes of Diffusion-LMS Networks. *IEEE Trans. Control Netw. Syst.* **2016**, doi:10.1109/TCNS.2016.2578044.
42. Han, S.; Rao, S.; Erdogmus, D.; Jeong, K.H.; Principe, J. A minimum-error entropy criterion with self-adjusting step-size (MEE-SAS). *Signal Process.* **2007**, *87*, 2733–2745.
43. Erdogmus, D.; Principe, J.C. Convergence properties and data efficiency of the minimum error entropy criterion in ADALINE training. *IEEE Trans. Signal Process.* **2003**, *51*, 1966–1978.
44. Shen, P.; Li, C. Minimum Total Error Entropy Method for Parameter Estimation. *IEEE Trans. Signal Process.* **2015**, *63*, 4079–4090.
45. Arablouei, R.; Werner, S. Analysis of the gradient-descent total least-squares adaptive filtering algorithm. *IEEE Trans. Signal Process.* **2014**, *62*, 1256–1264.
46. Kelley, C.T. *Iterative Methods for Optimization*; SIAM: Philadelphia, PA, USA, 1999; Volume 9, pp. 878–878.
47. Abadir, K.M.; Magnus, J.R. *Matrix Algebra*; Cambridge University Press: Cambridge, UK, 2005.
48. Sayed, A.H. *Adaptive Filters*; John Wiley & Sons: Hoboken, NJ, USA, 2008.
49. Khalil, H.K. *Nonlinear Systems*; Macmillan: New York, NY, USA, 1992.
50. Li, H.; Li, X.; Anderson. M. A class of adaptive algorithms based on entropy estimation achieving CRLB for linear non-Gaussian filtering. *IEEE Trans. Signal Process.* **2012**, *60*, 2049–2055.
51. Zhao, X.; Tu, S.; Sayed, A.H. Diffusion adaptation over networks under imperfect information exchange and non-stationary data. *IEEE Trans. Signal Process.* **2012**, *60*, 3460–3475.