



Chundong Xu^{1,*}, Qinglin Li¹ and Dongwen Ying²

- ¹ School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou 341000, China; qinglinli1997@163.com
- ² Institute of Acoustics, Chinese Academy of Sciences, Beijing 100000, China; building2009@163.com
- Correspondence: xuchundong@jxust.edu.cn; Tel.: +86-135-0707-8202

Abstract: In this paper, we develop a modified adaptive combination strategy for the distributed estimation problem over diffusion networks. We still consider the online adaptive combiners estimation problem from the perspective of minimum variance unbiased estimation. In contrast with the classic adaptive combination strategy which exploits orthogonal projection technology, we formulate a non-constrained mean-square deviation (MSD) cost function by introducing Lagrange multipliers. Based on the Karush–Kuhn–Tucker (KKT) conditions, we derive the fixed-point iteration scheme of adaptive combiners. Illustrative simulations validate the improved transient and steady-state performance of the diffusion least-mean-square LMS algorithm incorporated with the proposed adaptive combination strategy.

Keywords: distributed estimation; diffusion strategy; adaptive combination strategy; KKT conditions



Citation: Chundong, X.; Qinglin, L.; Dongwen Y. An Effective Adaptive Combination Strategy for Distributed Learning Network. *Appl. Sci.* **2021**, *11*, 5723. https://dx.doi.org/10.3390/ app11125723

Academic Editor: Christos Verikoukis

Received: 9 March 2021 Accepted: 31 May 2021 Published: 20 June 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/).

1. Introduction

It is generally beneficial to exploit diffusion strategies for distributed parameter estimation issues over adaptive networks [1–6]. Specifically, the diffusion least-mean-square (LMS)-based methods have already been used in many contexts, such as biological behavior modeling [7,8], distributed detection [9], distributed localization [10] and target tracking and escaping from predators [11], where scalability, robustness, and low power consumption are the desirable features [1]. For the diffusion strategy, each node of the network is allowed to receive the intermediate estimates from its neighboring nodes to improve the accuracy of its local estimate. Such cooperation enables each node to leverage the spatial diversity of noise profile over the entire network. From this point of view, the performance of distributed diffusion methods can be further enhanced by using suitable combination weights (combiners).

There have been several static combination rules [1,12], e.g., Metropolis rule, Laplacian rule, Uniform rule and Relative-degree rule. However, these static combiners are designed based solely on the topology of network, so they would generally be unadjustable to adapt to the spatial variation of signal and noise statistics. To address this problem, many studies resort to the adaptive combination (AC) strategies [12–18], most of which are developed for the adapt-then-combine (ATC) diffusion LMS algorithm [1].

Based on the minimum variance unbiased estimation (MVUE), the classic AC strategy [12] outperforms existing static combiners when applying onto the diffusion LMS algorithms. An optimal adaptive combination scheme is derived by estimating the variances of the measurement noises adaptively [13]. Simulation results validate the superior steady-state performance of the diffusion LMS algorithm regarding the optimal combiners in [13], as compared to previous static and adaptive combiners. Based on the adaptive combination rule in [13], an optimal combination rule regarding the channel distortion is also proposed [16]. To achieve both the accelerated convergence rate and good steady-state network performance, the combination switching mechanisms [14,15] are proposed, i.e.,



static combination scheme in the converging stage and AC scheme when approaching the steady state.

In addition, a decoupled adapt-then-combine (D-ATC) algorithm is proposed, for, the least-squares (LS)-based AC scheme is developed [17,18], which could achieve rather approximate performance as the ATC algorithm with the classic AC in homogeneous networks.

Motivation and Contribution

As mentioned above, the classic AC strategy is derived based on MVUE, which is validated to be a feasible criterion [12]. In fact, one of the key techniques in the classic AC strategy [12] is orthogonal projection which is exploited to guarantee the combiners adding up to 1. However, the orthogonal projection technique actually limits the update direction of combiners at each iteration, which can be relaxed to further improve the performance of diffusion LMS algorithm further ahead in Section 3.2.

In this paper, we still formulate the online adaptive combiners estimation of the ATC algorithm from the perspective of MVUE. Instead of directly exploiting the orthogonal projection technology in [12], we present a non-constrained mean-square deviation (MSD) cost function based on Lagrange multipliers. According to the fixed-point iteration methodology and KKT necessary conditions, we develop an effective adaptive combination strategy, which solely relies on the previous instantaneous intermediate weight estimates without resorting to the knowledge of measurement data and noises. The proposed AC strategy can be seen as the modified and extended version of the classic AC in [12]. Simulations validate the superior performance of the diffusion LMS algorithm when using the proposed AC strategy.

Notation: \mathbb{R} and \mathbb{C} denote the fields of real and complex numbers, respectively. Scalars are denoted by lower-case letters, and vectors and matrices respectively by lower- and upper-case boldface letters. The transpose and conjugate transpose are denoted by $(\cdot)^{T}$ and $(\cdot)^{H}$, respectively. $E\{\cdot\}$ represents expectation. $\Re(\cdot)$ means the real part. $\operatorname{col}\{\cdot\}$ stands for the vector obtained by stacking its arguments on top of one another. diag $\{\cdot\}$ generates a diagonal matrix using the given diagonal arguments. $[\cdot]_i$ stands for the *i*th element of a vector. The min $\{\cdot\}$ denotes the minimum element of a vector. We define the eigenvalue set of the square matrix F as $\{\lambda(F)\}$, with $\lambda_{\max}(F)$ denoting the maximum eigenvalue. The spectral radius of the square matrix F is denoted by $\varrho(F) \triangleq \max\{|\lambda(F)|\}$.

2. The ATC Diffusion LMS Algorithm

2.1. Model Assumption

Consider a network containing *N* nodes, which are used to estimate an *M*-dimensional unknown parameter $w_o \in \mathbb{C}^M$ collectively. \mathcal{N}_k denotes the set of neighbors of node *k*, including *k* itself. The cardinality of \mathcal{N}_k is n_k . For each node *k*, at the time instant *t*, the regressor $u_k(t) \in \mathbb{C}^M$ and the measurement signal $d_k(t) \in \mathbb{C}$ are available. The signal model is given by

$$d_k(t) \triangleq \boldsymbol{w}_o^{\mathrm{H}} \boldsymbol{u}_k(t) + \boldsymbol{v}_k(t), \tag{1}$$

where $v_k(t)$ denotes the additive zero-mean white Gaussian measurement noise at node k, with variance $\sigma_{v,k}^2$. For any k and t, $v_k(t)$ is independent from $u_k(t)$, and for all $k \neq l$ or $i \neq j$, $v_k(i)$ is independent from $v_l(j)$.

2.2. ATC Algorithm

The main target of distributed estimation algorithm is to generate an estimate $w_k(t)$ of w_o at each node k and time t in a distributed manner. For the diffusion strategy, generally, each node k executes a local adaptation step to obtain an intermediate estimate $\psi_k(t)$, then all the nodes share their intermediate estimates to their neighbors, and finally, each node k

linearly combines all the intermediate estimates received from its neighbors under some combination weights. The detailed steps of ATC diffusion LMS algorithm are

$$\boldsymbol{\psi}_k(t) = \boldsymbol{w}_k(t-1) + \mu_k \boldsymbol{u}_k(t) \boldsymbol{e}_k^*(t), \qquad (2)$$

$$\boldsymbol{w}_{k}(t) = \sum_{l \in \mathcal{N}_{k}} a_{l,k} \boldsymbol{\psi}_{l}(t), \qquad (3)$$

where $e_k(t) \triangleq d_k(t) - w_k^H(t-1)u_k(t)$ is the prior estimate error and $\mu_k > 0$ is the step size for $k \in \{1, 2, ..., N\}$. The combiner $a_{l,k}$ is the weight of intermediate estimation from node l during the combination step of node k. Moreover, the non-negative combination matrix $A = [a_{l,k}]$ satisfies [1,12]

$$a_{l,k} \ge 0 \text{ if } l \in \mathcal{N}_k, \ a_{l,k} = 0 \text{ if } l \notin \mathcal{N}_k, \ a_k^{\mathrm{T}} \mathbf{1} = 1,$$

$$\tag{4}$$

with a_k denoting the *k*th column of *A*. Notice that *A* is left-stochastic since the entries of each column are non-negative and sum to 1 [19,20].

3. Adaptive Combination Scheme

3.1. Minimum Variance Unbiased Estimation

Consider the signal model and the ATC diffusion LMS algorithm in Section 2. Assume that for each node $k \in \{1, ..., N\}$, the intermediate estimate $\psi_k(t)$ in the diffusion LMS algorithm satisfies

$$E\{\boldsymbol{\psi}_k(t)\} = \boldsymbol{w}_o, \quad \text{for all } k \in \{1, \dots, N\}, \tag{5}$$

We define

$$\Psi(t) \triangleq [\psi_1(t), \ldots, \psi_N(t)], \ k = 1, \cdots, N$$

Following [12], we have the minimum variance unbiased estimation problem for each $k \in \{1, ..., N\}$,

$$\min_{\boldsymbol{a}_{k} \in \mathbb{R}^{N}} \boldsymbol{a}_{k}^{T} \Re(\boldsymbol{Q}_{\boldsymbol{\Psi}}) \boldsymbol{a}_{k}$$
st. $\mathbf{1}_{N}^{T} \boldsymbol{a}_{k} = 1; \boldsymbol{a}_{l\,k} = 0$, for all $l \notin \mathcal{N}_{k}$,
(6)

where $Q_{\Psi} \triangleq E\{(\Psi(t) - E\{\Psi(t)\})^{H}(\Psi(t) - E\{\Psi(t)\})\}$ and $\mathbf{1}_{N}$ denotes the $N \times 1$ vector with unit entries. Applying the convex combination strategy for all $k \in \{1, ..., N\}$, we also require $a_{l,k} \ge 0$.

3.2. Fixed-Point Iteration Solution

First, we introduce a transform matrix P_k , defined as $P_k \triangleq [l\text{th column of } I_N]_{l \in \mathcal{N}_k}$. Then, a_k in (6) can be expressed by $a_k = P_k b_k$, with $b_k \in \mathbb{R}^{n_k}$. Therefore, the minimization problem (6) can be transformed into

$$\min_{\boldsymbol{b}_{k}} \quad J(\boldsymbol{b}_{k}) \triangleq \boldsymbol{b}_{k}^{T} \Re(\boldsymbol{Q}_{\boldsymbol{\Psi}_{k}}) \boldsymbol{b}_{k}$$
st. $\mathbf{1}_{n_{k}}^{T} \boldsymbol{b}_{k} = 1; b_{k,i} \geq 0, \text{ for } i = 1, \dots, n_{k},$

$$(7)$$

where $Q_{\Psi_k} \triangleq P_k^T Q_{\Psi} P_k = E\{(\Psi_k(t) - E\{\Psi_k(t)\})^H(\Psi_k(t) - E\{\Psi_k(t)\})\}$ with $\Psi_k(t) \triangleq \Psi(t)P_k$, and $b_{k,i}$ is the *i*th element of vector b_k . By introducing the Lagrange multipliers α and ω , we obtain the cost function

$$\mathcal{L}(\boldsymbol{b}_k, \boldsymbol{\omega}, \boldsymbol{\alpha}) \triangleq J(\boldsymbol{b}_k) - \boldsymbol{\omega}^T \boldsymbol{b}_k + \boldsymbol{\alpha} (\boldsymbol{1}_{n_k}^T \boldsymbol{b}_k - 1).$$
(8)

Taking the gradient of (8) with respect to b_k yields

$$\nabla_{\boldsymbol{b}_{k}} \mathcal{L}(\boldsymbol{b}_{k}, \boldsymbol{\omega}, \boldsymbol{\alpha}) = \nabla_{\boldsymbol{b}_{k}} J(\boldsymbol{b}_{k}) - \boldsymbol{\omega} + \mathbf{1}_{n_{k}} \boldsymbol{\alpha}$$
$$= \Re(\boldsymbol{Q}_{\boldsymbol{\Psi}_{k}}) \boldsymbol{b}_{k} - \boldsymbol{\omega} + \mathbf{1}_{n_{k}} \boldsymbol{\alpha}.$$
(9)

According to the Karush–Kuhn–Tucker (KKT) condition [21], the optimal value of the tuple (b_k , ω , α) should obey

$$\begin{cases} \Re(\mathbf{Q}_{\mathbf{Y}_k})\mathbf{b}_k - \boldsymbol{\omega} + \mathbf{1}_{n_k}\boldsymbol{\alpha} = 0\\ [\boldsymbol{\omega}]_i b_{k,i} = 0 \end{cases}$$
(10)

or equivalently,

$$[\Re(\mathbf{Q}_{\Psi_k})\mathbf{b}_k + \mathbf{1}_{n_k}\alpha]_i b_{k,i} = 0.$$
⁽¹¹⁾

We introduce a positive definite diagonal matrix $D_k(t-1)$, whose *i*th diagonal element is an arbitrary positive function with respect to $b_k(t-1)$, denoted as $f_i(b_k(t-1))$. Obviously, $D_k(t-1)(-\nabla_{b_k}\mathcal{L}(b_k, \omega, \alpha))$ is still the descent direction of the cost function (8) [21]. Therefore, we have the adaptive solution of problem (7) through the fixed-point iteration method $h_{i-1}(t) = h_{i-1}(t-1) - n_{i-1}(t)f_i(b_i(t-1))h_{i-1}(t-1)$

$$\begin{aligned}
b_{k,i}(t) &= b_{k,i}(t-1) - \eta_{k,i}(t) f_i(\boldsymbol{b}_k(t-1)) b_{k,i}(t-1) \\
& [\Re(\boldsymbol{Q}_{\boldsymbol{\Psi}_k}) \boldsymbol{b}_k(t-1) + \mathbf{1}_{n_k} \alpha]_i,
\end{aligned} \tag{12}$$

where $b_{k,i}(t)$ is the estimate of $b_{k,i}$ at time instant t and $\eta_{k,i}$ is the learning factor of $b_{k,i}(t)$.

To simplify the problem, we choose the same learning factor $\eta_{k,i}$ for all *i* at any time *t*. Hence, we can rewrite (12) as a vector,

$$\boldsymbol{b}_{k}(t) = \boldsymbol{b}_{k}(t-1) - \eta_{k}(t)\boldsymbol{\Gamma}_{k}(t-1)[\Re(\boldsymbol{Q}_{\boldsymbol{\Psi}_{k}})\boldsymbol{b}_{k}(t-1) + \boldsymbol{1}_{n_{k}}\boldsymbol{\alpha}].$$
(13)

where $\Gamma_k(t-1) \triangleq \text{diag}\{D_k(t-1)b_k(t-1)\}.$

Applying the constraint $\mathbf{1}_{n_k}^T \boldsymbol{b}_k(t) = 1$ and pre-multiplying both sides of (13) by $\mathbf{1}_{n_k}^T$ yields the Lagrange multiplier α

$$\alpha = \frac{\mathbf{1}_{n_k}^T (\boldsymbol{b}_k(t-1) - \eta_k(t) \boldsymbol{\Gamma}_k(t-1) \Re(\boldsymbol{Q}_{\boldsymbol{\Psi}_k}) \boldsymbol{b}_k(t-1)) - 1}{\eta_k(t) \mathbf{1}_{n_k}^T \boldsymbol{\Gamma}_k(t-1) \mathbf{1}_{n_k}}.$$
(14)

Substituting α into (13), and using the constraint $\mathbf{1}_{n_k}^T \mathbf{b}_k(t-1) = 1$ again, we can obtain the update of combiners $\mathbf{b}_k(t)$,

$$\boldsymbol{b}_{k}(t) = \boldsymbol{b}_{k}(t-1) - \eta_{k}(t)\boldsymbol{G}_{k}(t-1)\boldsymbol{\Gamma}_{k}(t-1)\Re(\boldsymbol{Q}_{\boldsymbol{\Psi}_{k}})\boldsymbol{b}_{k}(t-1).$$
(15)

where $G_k(t-1) \triangleq I_{n_k} - \frac{D_k(t-1)b_k(t-1)\mathbf{1}_{n_k}^T}{\mathbf{1}_{n_k}^T D_k(t-1)b_k(t-1)}$ with I_{n_k} denoting the $n_k \times n_k$ identity matrix. The adaptive combiners (15) can be undeted by two incremental steps

The adaptive combiners (15) can be updated by two incremental steps

$$\boldsymbol{g}_k(t) = \boldsymbol{\zeta}_k(t-1)\Re(\boldsymbol{Q}_{\boldsymbol{\Psi}_k})\boldsymbol{b}_k(t-1)$$
(16)

$$\boldsymbol{b}_k(t) = \boldsymbol{b}_k(t-1) - \eta_k(t)\boldsymbol{g}_k(t) \tag{17}$$

where

$$\boldsymbol{\zeta}_k(t-1) = \boldsymbol{G}_k(t-1)\boldsymbol{\Gamma}_k(t-1). \tag{18}$$

Then, we can obtain the combiner $a_k(t)$

$$\boldsymbol{a}_k(t) = \mathbf{P}_k \boldsymbol{b}_k(t) \tag{19}$$

which can be used in (3) to update the local weight estimate adaptively. We also define the adaptive combination matrix $A(t) \in \mathbb{R}^{N \times N}$ where $a_k(t)$ is the *k*th column vector of it.

Please note that $-g_k(t)$ in (16) can be seen as the product of $\zeta_k(t-1)$ and the gradient $\nabla_{b_k} J(b_k(t-1)) = \Re(Q_{\Psi_k})b_k(t-1)$, which means $\zeta_k(t-1)$ can be seen an auxiliary matrix to adjust the update direction $\nabla_{b_k} J(b_k(t-1))$. On the one hand, note that $G_k(t-1)$ in (18) is a projection matrix [22], which enables the update direction $-g_k(t)$ orthogonal to the vector $\mathbf{1}_{n_k}$, i.e., $\mathbf{1}_{n_k}^T g_k(t) = 0$. Then we have $\mathbf{1}_{n_k}^T b_k(t) = \mathbf{1}_{n_k}^T b_k(t-1) = \cdots = \mathbf{1}_{n_k}^T b_k(0) = 1$ if the initial combiners satisfy $\mathbf{1}_{n_k}^T b_k(0) = 1$. On the other hand, since as a whole, $\zeta_k(t-1)$ is

a positive semi-definite symmetric matrix, the update direction $-g_k(t)$ is still the descent direction of the cost function (7) [21]. Instead of using the positive semi-definite symmetric matrix $\zeta_k(t-1) = G_k(t-1)\Gamma_k(t-1)$, the classic AC [12] consider $\zeta_k(t-1)$ being replaced by the orthogonal projection matrix characterized by $\mathbf{1}_{n_k}$, i.e., $I_{n_k} - \frac{\mathbf{1}_{n_k}\mathbf{1}_{n_k}^T}{n_k}$, which actually limits the update direction of the adaptive combiners to be lie in the hyperplane spanned by $\mathbf{1}_{n_k}$ and $\nabla_{\mathbf{b}_k} J(\mathbf{b}_k(t-1))$. In fact, we find that (16) and (17) could reduce to the classic AC [12] when $\mathbf{D}_k(t) = \text{diag}\{\mathbf{b}_k(t)\}^{-1}$.

We now consider optimizing the learning factor $\eta_k(t)$. Substituting (16) and (17) into (7) yields a cost function regarding the learning factor $\eta_k(t)$,

$$h(\eta_k(t)) = \mathbf{g}_k^I(t) \Re(\mathbf{Q}_{\mathbf{\Psi}_k}) \mathbf{g}_k(t) \eta_k^Z(t) - 2\mathbf{g}_k^T(t) \Re(\mathbf{Q}_{\mathbf{\Psi}_k}) \mathbf{b}_k(t-1) \eta_k(t) + \mathbf{b}_k^T(t-1) \Re(\mathbf{Q}_{\mathbf{\Psi}_k}) \mathbf{b}_k(t-1).$$
(20)

Obviously, $h(\eta_k(t))$ is a quadratic (convex) function with respect to $\eta_k(t)$. Thus, its minimum value can be readily obtained if and only if

$$\eta_k^o(t) = \frac{\boldsymbol{g}_k^T(t)\Re(\boldsymbol{Q}_{\boldsymbol{\Psi}_k})\boldsymbol{b}_k(t-1)}{\boldsymbol{g}_k^T(t)\Re(\boldsymbol{Q}_{\boldsymbol{\Psi}_k})\boldsymbol{g}_k(t)} = \frac{\boldsymbol{b}_k^T(t-1)\Re(\boldsymbol{Q}_{\boldsymbol{\Psi}_k})\boldsymbol{\zeta}_k(t-1)\Re(\boldsymbol{Q}_{\boldsymbol{\Psi}_k})\boldsymbol{b}_k(t-1)}{\boldsymbol{g}_k^T(t)\Re(\boldsymbol{Q}_{\boldsymbol{\Psi}_k})\boldsymbol{g}_k(t)}.$$
(21)

Note that the optimal learning factor $\eta_k^o(t)$ is non-negative since $\zeta_k(t-1)$ is a positive semi-definite matrix.

To guarantee that the combiners are non-negative, we set the upper bound of $\eta_k(t)$ as [12],

$$\eta_k^{\max}(t) = \frac{\min(\boldsymbol{b}_k(t))}{\|\boldsymbol{g}_k(t)\|_{\infty} + \varepsilon},$$
(22)

where $\varepsilon > 0$ is a small constant. Please note that $\|\cdot\|_{\infty}$ represents the maximum norm. Thus, we choose the learning factor in (17) at time instant *t*,

$$\eta_k(t) = \min\{\eta_k^{\max}(t), \eta_k^o(t)\}.$$
(23)

Please note that the Q_{Ψ_k} is usually unavailable in practical applications. As what done in [12], Q_{Ψ_k} can be replaced by its approximation

$$\hat{Q}_{\Psi_k}(t) \approx \frac{1}{2} \Delta \Psi_k^H(t) \Delta \Psi_k(t).$$
(24)

where $\Delta \Psi_k(t) = \Psi_k(t) - \Psi_k(t-1)$. To make it smoother, we consider a forgetting factor λ . Then, the iterative expression of \hat{Q}_{Ψ_k} can be written as

$$\hat{\mathbf{Q}}_{\boldsymbol{\Psi}_{k}}(t) = \lambda \hat{\mathbf{Q}}_{\boldsymbol{\Psi}_{k}}(t-1) + \frac{1}{2} \Delta \boldsymbol{\Psi}_{k}^{H}(t) \Delta \boldsymbol{\Psi}_{k}(t).$$
(25)

In practical applications, we use $\hat{Q}_{\Psi_k}(t)$ in (25) to replace the aforementioned statistical quantity Q_{Ψ_k} for each node *k* at time instant *t*.

Finally, the implementation of the ATC algorithm with the proposed AC strategy is summarized in Algorithm 1.

Algorithm 1 ATC with the proposed AC strategy

For each node k, set $\psi_k(0) = w_k(0) = \mathbf{0}_M$ and choose $b_k(0) \in \mathbb{R}^{n_k}$ so that $\mathbf{1}_{n_k}^T b_k(0) = 1$. Given a small positive constant ε and step size μ_k , at each time instant t > 0, compute at each node k:

1. Update the intermediate weight estimate $\psi_k(t)$ through (2).

2. Update combiner $a_k(t)$ consecutively through (25), (16), (23), (17) and (19).

3. Update the local weight estimate $w_k(t)$ through (3).

4. Mean Convergence

We now analyze the mean convergence of the diffusion LMS algorithm with the proposed adaptive combiners.

We now introduce the following independence assumptions:

Assumption 1 (Independence). All regressors $u_k(t)$ are spatially and temporally independent ([1], Assumption 1).

Assumption 2. The combination matrix A(t) is independent of all regressors $u_k(t)$ and all local weight estimates $w_k(t-1)$ at time t-1 ([12], Assumption 4.3).

Theorem 1. *Under Assumptions 1 and 2, a sufficient condition to guarantee the convergence of the diffusion LMS algorithm is given by,*

$$0 < \mu_k < \frac{2}{\varrho(\boldsymbol{R})},\tag{26}$$

where $\varrho(\mathbf{R})$ denotes the spectrum radius of the matrix \mathbf{R} .

The proof is referred to Appendix A. Please note that the sufficient condition (26) is consistent with ([1], Equation (37)) and ([12], Equation (25))

5. Simulation Results

We evaluate herein the MSD of the proposed algorithm. Without loss of generality, the unknown weight vector w_0 is set to be $\mathbf{1}_M/M$ with M = 5. The initial weight vector estimations are $w_k(0) = \mathbf{0}_M$ for each node k. The constant ε used in (22) is set to be 0.5×10^{-4} and the forgetting factor $\lambda = 0$ or $\lambda = 0.95$. We consider $D_k(t) = \text{diag}\{b_k(t)\}^{\gamma}$ in terms of the proposed AC, where γ can be 0, -1 or -2.

We use the empirical MSDs as the performance metrics. Both the transient and steadystate empirical network MSDs hereinafter are obtained by averaging $\mathcal{L} = 500$ independent trials over all nodes of the network,

$$\mathcal{MSD}(t) \triangleq \frac{1}{\mathcal{L}N} \sum_{k=1}^{N} \sum_{\ell=1}^{\mathcal{L}} \|\tilde{\boldsymbol{w}}_{k}^{(\ell)}(t)\|_{2}^{2},$$
(27)

$$\mathcal{MSD} \triangleq \frac{1}{\mathcal{L}N} \sum_{k=1}^{N} \sum_{\ell=1}^{\mathcal{L}} \| \tilde{\boldsymbol{w}}_{k}^{(\ell)}(\infty) \|_{2}^{2},$$
(28)

where $\tilde{w}_{k}^{(\ell)}(t) \triangleq w_{o} - w_{k}^{(\ell)}(t)$ with $w_{k}^{(\ell)}(t)$ denoting the transient weight estimate of the ℓ th trial and $\|\tilde{w}_{k}^{(\ell)}(\infty)\|_{2}^{2}$ is obtained by averaging $\|\tilde{w}_{k}^{(\ell)}(t)\|_{2}^{2}$ of 100 iterations after convergence.

Example 1. We consider the same topology of the network with N = 15 nodes as ([12], Figure 5) ([17], Figure 6a), illustrated in Figure 1a. The measurement noise is real Gaussian white noise with its variance $\sigma_{v,k}^2$ at each node k presented in Figure 1b. According to the mean convergence condition, we set the step size to $\mu_k = 0.01$ for each node k. We herein consider $\gamma = 0$. Each regressor $u_k(t)$ at each node k is real Gaussian white noise sequence with their covariance matrices $\mathbf{R}_{u,k} = \sigma_{u,k}^2 \mathbf{I}_M$

with $\sigma_{u,k}^2 = 1$ for all k. The noise power of node 5, 12, 13 at the 1500th iteration suddenly goes up to 5.

As demonstrated in Figure 2, the proposed AC strategy outperforms the classic AC [12] and the uniform combination [1] in terms of superior steady-state performance and similar convergence rates, and outperforms LS-based AC [17] and the optimal AC [13] in terms of accelerated convergence rate. We also observe that the forgetting factor can further enhance the performance of the proposed AC. After the sudden change of the noise power of some nodes, the proposed AC exhibits rather high reconvergence rate and good steady-state performance.



Figure 1. Network topology and noise profile: (a) network topology; (b) noise profile.



Figure 2. The MSD learning curves.

Example 2. The initial measurement noise variance $\sigma_{v,k}^2$ at each node k is presented in Figure 1b. In this simulation, we compare the proposed AC scheme with the uniform combination and the classic AC in terms of the steady-state MSD at different noise variances $\tau \sigma_{v,k}^2$ at each node k. The other simulation conditions are same as Example 1. The steady-state MSDs with respect to the noise variances are illustrated in Table 1.

The fairness of this experiment is endorsed by the relatively approximate convergence behavior of the transient MSD curves plotted in Figure 3. As shown in Table 1, with the noise variances increasing, the ATC algorithm incorporated with the proposed AC exhibits superior performance compared to the ATC algorithm with the uniform combination or the classic AC strategy. Additionally, we can also find that the performance gain brought by the forgetting factor is limited when the noise variances increase to some certain limit.

Example 3. We now consider target tracking model ([17], Equation (52)), namely

$$\boldsymbol{w}_o(t) = \boldsymbol{w}_o + \boldsymbol{\theta}(t) \tag{29}$$

$$\boldsymbol{\theta}(t) = 0.99\boldsymbol{\theta}(t-1) + \boldsymbol{\xi}(t) \tag{30}$$

where $\boldsymbol{\xi}(t)$ is a sequence of independent identically distributed perturbations with zero mean and covariance matrix $\boldsymbol{\Xi}$, independent of the input regressors and measurement noise at every iteration. We herein consider $\boldsymbol{\xi}(t)$ is Gaussian white with $\boldsymbol{\Xi} = \sigma_{\boldsymbol{\xi}}^2 \mathbf{I}_M$ and $\sigma_{\boldsymbol{\xi}}^2 = 1 \times 10^{-7}$, i.e., the unknown weight vector is varying slowly. The other simulation conditions are same as Example 1.

Table 1. The steady-state MSD with respect to the variation of the noise power.

	Combiners	Uniform	Classic AC	1 - 0	$\lambda = 0.95$
Noise $ au$		Unitorini	Classic AC	$\lambda = 0$	n = 0.95
$\tau = 0.5$		-49.0	-50.2	-51.2	-53.0
au = 1		-46.0	-48.0	-49.1	-50.0
$\tau = 1.5$		-44.3	-46.6	-47.7	-48.3
au = 2		-43.3	-45.9	-46.9	-47.1
au = 2.5		-42.1	-45.1	-46.1	-46.1
$\tau = 3$		-41.2	-44.3	-45.2	-45.3
$\tau = 3.5$		-40.6	-43.7	-44.4	-44.4
au = 4		-40.2	-43.5	-44.1	-44.0
au = 4.5		-39.6	-42.9	-43.5	-43.3
au = 5		-39.2	-42.7	-43.3	-43.2



Figure 3. The MSD learning curves under different noise levels.

As illustrated in Figure 4, similar to Example 1, the proposed AC strategy outperforms the uniform combination rule, the classic AC and LS-based AC in terms of superior steadystate performance, and outperforms the optimal AC in terms of accelerated convergence rate, under tracking scenarios. We could also observe that the forgetting factor can further enhance the performance of the proposed AC in the tracking scenario.



Figure 4. The MSD learning curves under tracking scenarios.

Example 4. We now consider the impact of the factors λ and γ in the performance of the proposed AC strategy. Without loos of generality, we consider the step sizes $\mu_k = 0.02$ for each node k. The simulation result is shown in Figure 5.

It can be seen from Figure 5 that with specific γ , the larger λ brings the higher performance gain for the proposed AC. We could also observe that choosing $\gamma = -2$ brings better steady-state performance than the other two choices, especially for small forgetting factor λ . In particular, as compared to Example 1, for the two scenarios $\gamma = 0, \lambda = 0$ and $\gamma = 0, \lambda = 0.95$, we find that larger step sizes in this example lead to accelerated convergence rate at the cost of degraded steady-state performance.



Figure 5. The MSD learning curves

Example 5. We consider the sparse network with N = 15 nodes with its topology same as ([12], Figure 8), depicted in Figure 6a. The measurement noise is real Gaussian white noise with its variance $\sigma_{v,k}^2$ at each node presented in Figure 6c. We consider the heterogenous network with the step sizes $\mu_k = 0.004$ for orange shaded nodes and $\mu_n = 0.02$ for the rest. Each regressor $u_k(t)$ at each node k is real Gaussian white noise sequence with their covariance matrices $\mathbf{R}_{u,k} = \sigma_{u,k}^2 \mathbf{I}_M$ with $\sigma_{u,k}^2$ presented in Figure 6b for all k.



Figure 6. The topology, regressor power and noise profile: (a) network topology; (b) regressor power; (c) noise profile.

As illustrated in Figure 7, for the sparse network, the proposed AC scheme outperforms the classic AC and LS-based AC in terms of superior steady-state performance while keeping rather approximate convergence rate. It also outperforms the optimal AC scheme in terms of the accelerated convergence rate. Moreover, the introduction of forgetting factor can further enhance the performance of the proposed AC scheme.



Figure 7. The MSD learning curves for the sparse network.

6. Conclusions

In this paper, we present a modified adaptive combination strategy for the distributed estimation problem over diffusion networks to improve robustness against the spatial variation of signal and noise statistics over the network. Considering the Karush–Kuhn–Tucker conditions and fixed-point iteration methodology, we derive an effective adaptive combination strategy for the ATC diffusion LMS algorithm. We also invoke the forgetting factor and optimize the learning factor to further enhance the performance of the proposed adaptive combination strategy. Illustrative simulations validate the improved performance of the diffusion LMS algorithm with the proposed adaptive combination strategy.

Author Contributions: Data curation, Q.L.; Funding acquisition, C.X.; Methodology, Q.L.; Software, C.X.; Supervision, C.X. and D.Y.; Writing—Original draft, Q.L.; Writing—Review & editing, Q.L. and D.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was jointly funded by the National Natural Science Foundation of China under Grant 11864016, 61671442 and Social Science Key Research Base Project of Jiangxi Province under Grant JD19042.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: In this section you can acknowledge any support given which is not covered by the author contribution or funding sections. This may include administrative and technical support, or donations in kind (e.g., materials used for experiments).

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

Appendix A. Mean Convergence Analysis

We define the network weight error

$$\tilde{w}(t) \triangleq \operatorname{col}\{\tilde{w}_1(t), \cdots, \tilde{w}_N(t)\} \in \mathbb{C}^{MN},\tag{A1}$$

where $\tilde{w}_k(t) \triangleq w_0 - w_k(t), k = 1, 2, \dots, N$. We define the global regressor,

$$\boldsymbol{u}(t) \triangleq \operatorname{col}\{\boldsymbol{u}_1(t), \cdots, \boldsymbol{u}_N(t)\},\tag{A2}$$

and the global covariance matrix,

$$\mathbf{R} \triangleq \operatorname{diag}\{\mathbf{R}_1, \cdots, \mathbf{R}_N\},\tag{A3}$$

where the covariance matrix $\mathbf{R}_k \triangleq \mathbb{E} \{ \mathbf{u}_k(t) \mathbf{u}_k^{\mathrm{H}}(t) \}, k = 1, 2, \cdots, N.$ We also introduce the following diagonal matrices,

$$\boldsymbol{\mathcal{V}}(t) \triangleq \operatorname{diag}\{v_1(t), \cdots, v_N(t)\} \otimes \boldsymbol{I}_M,\tag{A4}$$

$$\mathcal{D} \triangleq \operatorname{diag}\{\mu_1, \cdots, \mu_N\} \otimes I_M, \tag{A5}$$

and the extended combination matrix

$$\boldsymbol{\mathcal{A}}(t) \triangleq \boldsymbol{A}(t) \otimes \boldsymbol{I}_{M}. \tag{A6}$$

Following [1,12], we can obtain the iteration of the network weight error

$$\tilde{\boldsymbol{w}}(t) = \hat{\boldsymbol{F}}(t)\tilde{\boldsymbol{w}}(t-1) - \boldsymbol{\mathcal{A}}^{\mathrm{T}}(t)\boldsymbol{\mathcal{D}}\boldsymbol{z}(t), \tag{A7}$$

where $\hat{F}(t) \triangleq \mathcal{A}^{\mathrm{T}}(t) (\mathbf{I}_{MN} - \mathcal{D}\hat{R}(t))$ with $\hat{R}(t) \triangleq \mathrm{diag}\{\hat{R}_1(t), \cdots, \hat{R}_N(t)\}$ denoting the instantaneous estimate of the global covariance matrix R and $\hat{R}_k(t) \triangleq u_k(t)u_k^{\mathrm{H}}(t)$, and $z(t) \triangleq \mathcal{V}^*(t)u(t)$.

Obviously, we have $\mathbb{E}\{z(t)\} = \mathbf{0}$. According to Assumption 1 and Assumption 2, taking mathematical expectation of (A7) yields

$$\mathbb{E}\{\tilde{\boldsymbol{w}}(t)\} = \boldsymbol{F}(t)\mathbb{E}\{\tilde{\boldsymbol{w}}(t-1)\},\tag{A8}$$

12 of 13

where

$$\mathbf{F}(t) \triangleq \mathbb{E}\left\{\hat{\mathbf{F}}(t)\right\} = \mathbb{E}\left\{\mathbf{\mathcal{A}}^{\mathrm{T}}(t)\right\} (\mathbf{I}_{MN} - \mathbf{\mathcal{D}}\mathbf{R}).$$
(A9)

(A8) can be further expressed as

$$\mathbb{E}\{\tilde{\boldsymbol{w}}(t)\} = \boldsymbol{\mathcal{F}}(t)\mathbb{E}\{\tilde{\boldsymbol{w}}(0)\},\tag{A10}$$

where $\mathcal{F}(t) \triangleq \prod_{i=t}^{1} F(i) \triangleq F(t) \cdots F(1)$. To facilitate our analysis, we herein introduce the submultiplicative matrix norm (A submultiplicative matrix norm satisfies $||AB|| \leq ||A|| ||B||$ [23]). For any square matrix X, and any $\epsilon > 0$, there exists a submultiplicative matrix norm $|| \cdot ||_{\varrho}$ such that $\varrho(X) \leq ||X||_{\varrho} \leq \varrho(X) + \epsilon$, where $\varrho(X)$ denotes the spectrum radius of X [23,24]. Accordingly, we have $||\mathcal{F}(t)||_{\varrho} \leq \prod_{i=t}^{1} ||F(i)||_{\varrho}$. Notice that the diffusion LMS algorithm converges if and only if $\lim_{t\to\infty} ||\mathcal{F}(t)||_{\varrho} = 0$. Hence, a sufficient condition for the diffusion LMS algorithm to converge is $||F(t)||_{\varrho} \leq \varrho(F(t)) + \epsilon < 1$. Therefore, the diffusion LMS algorithm converges if $\varrho(F(t)) < 1$ for all t with a sufficiently small ϵ chosen. Thus, the diffusion LMS algorithm converges if F(t) in (A8) is stable at each t.

 $\mathbb{E}\left\{\mathcal{A}^{T}(t)\right\}$ in (A9) is left-stochastic since each column of it sums to 1. Thus, according to ([1], Appexdix I), F(t) is stable if, and only if $\mathbf{I}_{MN} - \mathcal{D}\mathbf{R}$ is stable, i.e., $\max\{|1 - \lambda(\mathcal{D}\mathbf{R})|\} < 1$. Notice that $\mathcal{D}\mathbf{R}$ is positive semi-definite Hermitian since \mathbf{R} is block diagonal and \mathcal{D} is positive definite diagonal. In light of [25], we have $\lambda_{\max}(\mathcal{D}\mathbf{R}) \leq \lambda_{\max}(\mathcal{D})\lambda_{\max}(\mathbf{R}) = \mu_{\max}\lambda_{\max}(\mathbf{R})$, where μ_{\max} is the maximum step size used in the network. Thus, $\max\{|1 - \lambda(\mathcal{D}\mathbf{R})|\} < 1$ holds if, and only if $\mu_{\max}\lambda_{\max}(\mathbf{R}) < 2$, i.e., $0 < \mu < 2/\lambda_{\max}(\mathbf{R})$.

References

- Cattivelli, F.S.; Sayed, A.H. Diffusion LMS Strategies for Distributed Estimation. *IEEE Trans. Signal Process.* 2010, 58, 1035–1048. [CrossRef]
- Lee, H.S.; Kim, S.E.; Lee, J.W.; Song, W.J. A Variable Step-Size Diffusion LMS Algorithm for Distributed Estimation. *IEEE Trans. Signal Process.* 2015, 63, 1808–1820. [CrossRef]
- Ahn, D.C.; Lee, J.W.; Shin, S.J.; Song, W.J. A new robust variable weighting coefficients diffusion LMS algorithm. *Signal Process*. 2017, 131, 300–306. [CrossRef]
- Huang, W.; Li, L.; Li, Q. Diffusion Robust Variable Step-Size LMS Algorithm Over Distributed Networks. *IEEE Access* 2018, 6, 47511–47520. [CrossRef]
- 5. Ashkezari-Toussi, S.; Sadoghi-Yazdi, H. Robust diffusion LMS over adaptive networks. *Signal Process.* **2019**, *158*, 201–209. [CrossRef]
- Nassif, R.; Vlaski, S.; Sayed, A.H. Adaptation and Learning Over Networks Under Subspace Constraints—Part I: Stability Analysis. *IEEE Trans. Signal Process.* 2020, 68, 1346–1360. [CrossRef]
- Tu, S.Y.; Sayed, A.H. Mobile adaptive networks with self-organization abilities. In Proceedings of the 2010 7th International Symposium on Wireless Communication Systems, York, UK, 19–22 September 2010; pp. 379–383. [CrossRef]
- Cattivelli, F.S.; Sayed, A.H. Modeling Bird Flight Formations Using Diffusion Adaptation. *IEEE Trans. Signal Process.* 2011, 59, 2038–2051. [CrossRef]
- Cattivelli, F.S.; Sayed, A.H. Distributed detection over adaptive networks using diffusion adaptation. *IEEE Trans. Signal Process.* 2011, 59, 1917–1932. [CrossRef]
- Chen, J.; Sayed, A.H. Diffusion Adaptation Strategies for Distributed Optimization and Learning Over Networks. *IEEE Trans. Signal Process.* 2012, 60, 4289–4305. [CrossRef]
- 11. Tu, S.; Sayed, A.H. Mobile Adaptive Networks. IEEE J. Sel. Top. Signal Process. 2011, 5, 649–664. [CrossRef]
- Takahashi, N.; Yamada, I.; Sayed, A.H. Diffusion Least-Mean Squares with adaptive combiners: formulation and performance analysis. *IEEE Trans. Signal Process.* 2010, 58, 4795–4810. [CrossRef]
- Tu, S.; Sayed, A.H. Optimal combination rules for adaptation and learning over networks. In Proceedings of the 2011 4th IEEE International Workshop Computational Advances in Multi-Sensor Adaptive Processing, San Juan, Puerto Rico, 13–16 December 2011; pp. 317–320. [CrossRef]
- Yu, C.; Sayed, A.H. A strategy for adjusting combination weights over adaptive networks. In Proceedings of the 2013 IEEE International Conference Acoustic, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; pp. 4579–4583. [CrossRef]

- Fernandez-Bes, J.; Arenas-García, J.; Sayed, A.H. Adjustment of combination weights over adaptive diffusion networks. In Proceedings of the 2014 IEEE International Conference Acoustic, Speech and Signal Processing, Florence, Italy, 4–9 May 2014; pp. 6409–6413. [CrossRef]
- 16. Abdolee, R.; Vakilian, V. An Iterative Scheme for Computing Combination Weights in Diffusion Wireless Networks. *IEEE Wireless Commun. Lett.* 2017, *6*, 510–513. [CrossRef]
- 17. Fernandez-Bes, J.; Azpicueta-Ruiz, L.A.; Arenas-García, J. Distributed estimation in diffusion networks using affine least-squares combiners. *Dig. Signal Process.* **2015**, *36*, 1–14. [CrossRef]
- Fernandez-Bes, J.; Arenas-García, J.; Silva, M.T.M. Adaptive Diffusion Schemes for Heterogeneous Networks. *IEEE Trans. Signal Process.* 2017, 65, 5661–5674. [CrossRef]
- 19. Sayed, A.H. Adaptive Networks. Proc. IEEE 2014, 102, 460-497. [CrossRef]
- 20. Zhao, X.; Sayed, A.H. Asynchronous Adaptation and Learning Over Networks-Part I: Modeling and Stability Analysis. *IEEE Trans. Signal Process.* 2015, *63*, 811–826. [CrossRef]
- Chen, J.; Richard, C.; Bermudez, J.C.M. Nonnegative Least-Mean-Square Algorithm. IEEE Trans. Signal Process. 2011, 59, 5225–5235. [CrossRef]
- 22. Behrens, R.T.; Scharf, L.L. Signal processing applications of oblique projection operators. *IEEE Trans. Signal Process.* **1994**, *42*, 1413–1424. [CrossRef]
- 23. Kailath, T.; Sayed, A.H.; Hassibi, B. Linear Estimation; Prentice-Hall: Englewood Cliffs, NJ, USA, 2000.
- 24. Sayed, A.H. Fundamentals of Adaptive Filtering; Wiley: New York, NY, USA, 2003.
- 25. Marshall, A.W.; Olkin, I.; Arnold, B.C. Matrix Theory. In *Inequalities: Theory of Majorization and Its Applications*; Springer: New York, NY, USA, 2011; Chapter 9; pp. 338–347. [CrossRef]