Talebi, Sayed Pouria; Werner, Stefan; Mandic, Danilo

Distributed Adaptive Filtering of α-Stable Signals

Published in:
IEEE Signal Processing Letters

DOI:
10.1109/LSP.2018.2862639

Published: 01/10/2018

Document Version
Peer reviewed version

Please cite the original version:
Talebi, S. P., Werner, S., & Mandic, D. (2018). Distributed Adaptive Filtering of -Stable Signals. *IEEE Signal Processing Letters*, *25*(10), 1450 - 1454. https://doi.org/10.1109/LSP.2018.2862639

Distributed Adaptive Filtering of α-Stable Signals

Sayed Pouria Talebi, Stefan Werner, and Danilo P. Mandic

***Abstract*—A cost-effective framework for distributed adaptive filtering of α-stable signals over sensor networks is proposed. First, the filtering paradigm of α-stable signals through multiple observations made over a network of sensors is revisited and an optimal solution is formulated. Then, an adaptive gradient descent based algorithm for distributed real-time filtering of α-stable signals via multi-agent networks is derived. This not only provides an approximation of the formulated optimal solution, but also a cost-effective algorithm which scales with the size of the network. Moreover, performance of the derived algorithm is analyzed and convergence conditions are established.**

***Index Terms*—Sensor networks, distributed adaptive filtering, consensus fusion, fractional differential, α-stable random signals.**

I. Introduction

Most signal processing and machine learning approaches, assume a Gaussian model for the signal, as this often leads to mathematically tractable and computationally efficient solutions. However, in an increasing number of modern applications, the encountered signals exhibit sharp spikes, resulting in distributions that decay slower than the Gaussian case [1]–[8]. For modeling such signals, α-stable random processes have proven to be advantageous [3,6,7].

A general closed-form expression for the probability distribution function of the entire class of α-stable random vectors does not exist. However, the class of real-valued α-stable random vectors with elliptically symmetric distributions, referred to as SαS, admit characteristic function of the form [1,9]

$$\boldsymbol{\Phi}_{\boldsymbol{Z}}(\mathbf{s}) = \mathbb{E}\left\{e^{i\mathbf{s}^{\mathsf{T}}\mathbf{z}}\right\} = e^{i\mathbf{s}^{\mathsf{T}}\boldsymbol{\xi}} e^{-\left(\frac{1}{2}\mathbf{s}^{\mathsf{T}}\boldsymbol{\Gamma}_{\mathbf{z}}\mathbf{s}\right)^{\frac{\alpha}{2}}} \tag{1}$$

where $i^2 = -1$, while $(\cdot)^{\mathsf{T}}$ and $\mathbb{E}\{\cdot\}$ denote the transpose and statistical expectation operators, with $\boldsymbol{\Phi}_{\boldsymbol{Z}}(\cdot)$ denoting the characteristic function of the random vector $\boldsymbol{Z}$, and the positive semi-definite covariation matrix $\boldsymbol{\Gamma}_{\mathbf{z}}$ determining the elliptical shape of the distribution of $\boldsymbol{Z}$ which is centered at $\boldsymbol{\xi}$. The characteristic exponent $\alpha \in (0, 2]$ in (1) specifies tail heaviness of the density function [1,3]. A small value of α indicates severe impulsiveness, resulting in heavier tails, while a value close to 2 indicates a more Gaussian type behavior. Indeed when $\alpha = 2$, random vector $\boldsymbol{Z}$ has a Gaussian distribution with covariance matrix $\boldsymbol{\Gamma}_{\mathbf{z}}$ and mean vector $\boldsymbol{\xi}$.

In addition to their trade-mark stability property [6], SαS random processes also admit a generalized version of the central limit theorem [2], which makes them attractive for modeling a large variety of signals. However, with the exception of the Gaussian case, SαS random processes only have finite statistical moments of orders strictly less than α [2]–[4]. Therefore, in filtering applications it is implicitly implied that $\alpha \in (1, 2]$, to ensure that finite conditional expectations do exist. Indeed, we limit our study to real-valued SαS processes with $\alpha \in (1, 2]$, where the mean vector corresponds to $\boldsymbol{\xi}$ in (1).

Since out of the entire class of SαS random variables only the Gaussian case has well-defined variance and higher-order statistical moments, filtering techniques based on minimizing the second-order moment of an error measure, such as the least mean square (LMS), do not perform well when applied to the generality of SαS random processes [10]–[13]. The distributed particle filter in [13] addresses this issue; however, the solution is too cumbersome for many filtering applications. Although a number of gradient based techniques for filtering SαS signals have been proposed [11,14,15], a distributed approach suitable for implementation over a large-scale sensor network accompanied with a rigorous convergence and stability analysis is still lacking.

In recent years, fractional-order calculus is experiencing a renaissance, finding applications in statistics and control [16]–[18], where they have been used to derive fractional-order statistical moments and describe complex physical systems. However, the potential of fractional-order calculus in adaptive learning remains largely untapped.

In light of the advantages that distributed learning techniques offer [19]–[25], this work introduces a class of practical distributed adaptive filtering solutions for SαS signals. To this end, based on the characteristic function in (1), an optimal solution to such a problem is formulated. However, calculation of such a solution requires knowledge of the signal statistics and is impractical for implementation over large-scale networks. Therefore, an adaptive filtering algorithm, where agents of the network collectively minimize the fractional square of an error measure in a gradient descent manner using fractional-order differentials is derived. Moreover, performance of the so derived algorithm is analyzed and convergence criteria are established. Finally, the introduced concept is validated through a simulation example.

Mathematical Notations: Scalars, column vectors, and matrices are denoted respectively by lowercase, bold lowercase, and bold uppercase letters. Multi-variate random processes are represented by uppercase bold Italic letters. The transpose and statistical expectation operators are denoted by $(\cdot)^{\mathsf{T}}$ and $\mathbb{E}\{\cdot\}$, while $\otimes$ denotes the Kronecker product. Finally, $(\cdot)^{\langle\tau\rangle}$ denotes the element wise implementation of the function $f(z) = |z|^{\tau}\text{sign}(z)$, with $\text{sign}(\cdot)$ denoting the sign function.

Sayed Pouria Talebi is with The Department of Signal Processing and Acoustics, Aalto University, Espoo FI-02150 Finland, E-mail: pouria.talebi@aalto.fi. Stefan Werner is with The Department of Electronic Systems, Norwegian University of Science and Technology, Trondheim NO-7491 Norway, E-mail: stefan.werner@ntnu.no. Danilo P. Mandic is with The Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2AZ, U.K. E-mail: d.mandic@imperial.ac.uk. The work of Sayed Pouria Talebi and Stefan Werner was supported in part by Academy of Finland under Grant 296849.

II. Problem Formulation

We consider the classical distributed filtering problem [23, 26]–[29] from the more general perspective of SαS random signals. This includes Gaussian random signals as a special case, for which $\alpha = 2$.

A. The Network

The sensor network is modeled as a connected graph $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$, where the node set $\mathcal{N}$ represents agents of the sensor network, while the edge set $\mathcal{E}$ represents bidirectional communication links between agents of the sensor network. The neighborhood of a node, l, is denoted by $\mathcal{N}_l$ and is defined as the set of nodes that are connected to node l, including itself. The number of nodes in the set $\mathcal{N}_l$, i.e., its cardinality, is denoted by $|\mathcal{N}_l|$, with $|\mathcal{N}|$ denoting the total number of nodes in the network.

B. The Filtering Problem

The aim is collaborative estimation of a parameter matrix, $\mathbf{H}$, through observations made via sensors of the network. The observations and desired parameter vector are interrelated as

$$\forall l \in \mathcal{N}: \ \mathbf{y}_{l,n} = \mathbf{H}\mathbf{x}_{l,n} + \mathbf{w}_{l,n} \tag{2}$$

where, for node l and time instant n, $\mathbf{y}_{l,n}$ and $\mathbf{w}_{l,n}$ denote the observation and background noise vectors, with $\mathbf{x}_{l,n}$ representing the regression vector used to identify the system. The regression vectors $\{\mathbf{x}_{l,n} : \forall l \in \mathcal{N}\}$ and background noise vectors $\{\mathbf{w}_{l,n} : \forall l \in \mathcal{N}\}$ are assumed to be temporally and spatially independent white zero-mean SαS random sequences.[1]

III. Optimal Solution

In order to derive an optimal solution to the proposed filtering problem, the parameter matrix, $\mathbf{H}$, is first formulated in terms of the regression and observation vector statistics. To this end, the observation, regression, and background noise vectors are organized into the column vectors

$$\begin{aligned} \mathbf{y}_n =& [\mathbf{y}_{1,n}^{\mathsf{T}}, \ldots, \mathbf{y}_{|\mathcal{N}|,n}^{\mathsf{T}}]^{\mathsf{T}} \\ \mathbf{x}_n =& [\mathbf{x}_{1,n}^{\mathsf{T}}, \ldots, \mathbf{x}_{|\mathcal{N}|,n}^{\mathsf{T}}]^{\mathsf{T}} \\ \mathbf{w}_n =& [\mathbf{w}_{1,n}^{\mathsf{T}}, \ldots, \mathbf{w}_{|\mathcal{N}|,n}^{\mathsf{T}}]^{\mathsf{T}} \end{aligned} \tag{3}$$

which allows the expression in (2) to be formulated from a network-wide perspective as

$$\mathbf{y}_n = (\boldsymbol{\mathcal{I}} \otimes \mathbf{H})\, \mathbf{x}_n + \mathbf{w}_n \tag{4}$$

where $\boldsymbol{\mathcal{I}}$ is an $|\mathcal{N}| \times |\mathcal{N}|$ identity matrix.

Now, from (1) the joint characteristic function of $\mathbf{y}_n$ and $\mathbf{x}_n$ can be expressed as

$$\begin{aligned} \boldsymbol{\Phi}_{\boldsymbol{Y},\boldsymbol{X}}(\mathbf{s}_y, \mathbf{s}_x) =& \mathbb{E}\left\{ e^{i\left(\mathbf{s}_{\mathbf{y}}^{\mathsf{T}}\mathbf{y}_n + \mathbf{s}_{\mathbf{x}}^{\mathsf{T}}\mathbf{x}_n\right)} \right\} \\ =& \mathbb{E}\left\{ e^{i\left(\mathbf{s}_{\mathbf{y}}^{\mathsf{T}}(\boldsymbol{\mathcal{I}}\otimes\mathbf{H}) + \mathbf{s}_{\mathbf{x}}^{\mathsf{T}}\right)\mathbf{x}_n} e^{i\mathbf{s}_{\mathbf{y}}^{\mathsf{T}}\mathbf{w}_n} \right\} \\ =& \boldsymbol{\Phi}_{\boldsymbol{X}}\left((\boldsymbol{\mathcal{I}} \otimes \mathbf{H}^{\mathsf{T}})\mathbf{s}_{\mathbf{y}} + \mathbf{s}_{\mathbf{x}}\right) \boldsymbol{\Phi}_{\boldsymbol{W}}(\mathbf{s}_{\mathbf{y}}). \end{aligned} \tag{5}$$

[1]The assumptions on temporal and spatial independence are congruent with the seminal work in [23,29]. In addition, without loss of generality, we assume that $\boldsymbol{\Gamma}_{\mathbf{w}}$, statistic of the system zero input response, is known. In essence, the classical distributed filtering problem (see [23,26]–[29]) is generalized to an SαS setting.

On the other hand, considering the general characteristic function of SαS random processes in (1) we have

$$\boldsymbol{\Phi}_{\boldsymbol{X}}\left((\boldsymbol{\mathcal{I}} \otimes \mathbf{H}^{\mathsf{T}})\mathbf{s}_{\mathbf{y}} + \mathbf{s}_{\mathbf{x}}\right) = e^{-\left(\frac{1}{2}\boldsymbol{\Theta}_{\mathbf{x}}(\mathbf{s}_y, \mathbf{s}_x)\right)^{\frac{\alpha}{2}}} \tag{6}$$

where

$$\begin{aligned} \boldsymbol{\Theta}_{\mathbf{x}}(\mathbf{s}_y, \mathbf{s}_x) =& \mathbf{s}_{\mathbf{y}}^{\mathsf{T}}(\boldsymbol{\mathcal{I}} \otimes \mathbf{H})\boldsymbol{\Gamma}_{\mathbf{x}}(\boldsymbol{\mathcal{I}} \otimes \mathbf{H}^{\mathsf{T}})\mathbf{s}_{\mathbf{y}} + \mathbf{s}_{\mathbf{x}}^{\mathsf{T}}\boldsymbol{\Gamma}_{\mathbf{x}}\mathbf{s}_{\mathbf{x}} \\ &+ \mathbf{s}_{\mathbf{y}}^{\mathsf{T}}(\boldsymbol{\mathcal{I}} \otimes \mathbf{H})\boldsymbol{\Gamma}_{\mathbf{x}}\mathbf{s}_{\mathbf{x}} + \mathbf{s}_{\mathbf{x}}^{\mathsf{T}}\boldsymbol{\Gamma}_{\mathbf{x}}(\boldsymbol{\mathcal{I}} \otimes \mathbf{H}^{\mathsf{T}})\mathbf{s}_{\mathbf{y}}. \end{aligned} \tag{7}$$

In a similar fashion, we have

$$\boldsymbol{\Phi}_{\boldsymbol{W}}(\mathbf{s}_{\mathbf{y}}) = e^{-\left(\frac{1}{2}\boldsymbol{\Theta}_{\mathbf{w}}(\mathbf{s}_y)\right)^{\frac{\alpha}{2}}} \ \text{ with } \ \boldsymbol{\Theta}_{\mathbf{w}}(\mathbf{s}_y) = \mathbf{s}_{\mathbf{y}}^{\mathsf{T}}\boldsymbol{\Gamma}_{\mathbf{W}}\mathbf{s}_{\mathbf{y}}. \tag{8}$$

From the expressions in (5)-(8), $\boldsymbol{\Phi}_{\boldsymbol{Y},\boldsymbol{X}}(\mathbf{s}_{\mathbf{y}}, \mathbf{s}_{\mathbf{x}})$ can be formulated as

$$\boldsymbol{\Phi}_{\boldsymbol{Y},\boldsymbol{X}}(\mathbf{s}_y, \mathbf{s}_x) = e^{-\left(\frac{1}{2}\boldsymbol{\Theta}_{\mathbf{y},\mathbf{x}}(\mathbf{s}_y, \mathbf{s}_x)\right)^{\frac{\alpha}{2}}} \tag{9}$$

with $\boldsymbol{\Theta}_{\mathbf{y},\mathbf{x}}^{\frac{\alpha}{2}}(\mathbf{s}_y, \mathbf{s}_x) = \boldsymbol{\Theta}_{\mathbf{x}}^{\frac{\alpha}{2}}(\mathbf{s}_y, \mathbf{s}_x) + \boldsymbol{\Theta}_{\mathbf{w}}^{\frac{\alpha}{2}}(\mathbf{s}_y)$ and

$$\boldsymbol{\Theta}_{\mathbf{x}}(\mathbf{s}_y, \mathbf{s}_x) = \begin{bmatrix} \mathbf{s}_{\mathbf{y}}^{\mathsf{T}} & \mathbf{s}_{\mathbf{x}}^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Gamma}_{\mathbf{y}} & \boldsymbol{\Gamma}_{\mathbf{yx}} \\ \boldsymbol{\Gamma}_{\mathbf{xy}} & \boldsymbol{\Gamma}_{\mathbf{x}} \end{bmatrix} \begin{bmatrix} \mathbf{s}_{\mathbf{y}} \\ \mathbf{s}_{\mathbf{x}} \end{bmatrix}$$

where $\boldsymbol{\Gamma}_{\mathbf{y}} = (\boldsymbol{\mathcal{I}} \otimes \mathbf{H})\boldsymbol{\Gamma}_{\mathbf{x}}(\boldsymbol{\mathcal{I}} \otimes \mathbf{H})^{\mathsf{T}}$, $\boldsymbol{\Gamma}_{\mathbf{xy}} = \boldsymbol{\Gamma}_{\mathbf{x}}(\boldsymbol{\mathcal{I}} \otimes \mathbf{H})^{\mathsf{T}}$, and $\boldsymbol{\Gamma}_{\mathbf{yx}} = (\boldsymbol{\mathcal{I}} \otimes \mathbf{H})\boldsymbol{\Gamma}_{\mathbf{x}}$. Therefore, it can be shown that

$$(\boldsymbol{\mathcal{I}} \otimes \mathbf{H}) = \boldsymbol{\Gamma}_{\mathbf{yx}}\boldsymbol{\Gamma}_{\mathbf{x}}^{-1}. \tag{10}$$

The temporal and spatial independent assumption of the regression and background noise sequences leads to $\boldsymbol{\Gamma}_{\mathbf{yx}}$ and $\boldsymbol{\Gamma}_{\mathbf{x}}$ being block diagonal matrices so that

$$\begin{aligned} \boldsymbol{\Gamma}_{\mathbf{yx}} =& \text{block-diag}\{\boldsymbol{\Gamma}_{\mathbf{y}_l\mathbf{x}_l} : \forall l \in \mathcal{N}\} \\ \boldsymbol{\Gamma}_{\mathbf{x}} =& \text{block-diag}\{\boldsymbol{\Gamma}_{\mathbf{x}_l} : \forall l \in \mathcal{N}\}. \end{aligned} \tag{11}$$

Finally, upon replacing (11) into (10) and summing the block diagonal elements we have

$$\mathbf{H} = \frac{1}{|\mathcal{N}|} \sum_{\forall l \in \mathcal{N}} \boldsymbol{\Gamma}_{\mathbf{y}_l\mathbf{x}_l}\boldsymbol{\Gamma}_{\mathbf{x}_l}^{-1}. \tag{12}$$

Akin to what was performed in [13], sufficient statistics for calculating the solution in (12) can be found from empirical estimates of characteristic functions of the regression and observation vectors. Thus the expression in (12) is considered as the optimal solution to the defined filtering problem. However, this approach is neither cost-effective nor is it suitable for implementation over a network. To this end, an adaptive gradient descent based technique is next developed.

IV. Distributed Adaptive Approach

Consider $\hat{\mathbf{y}}_{l,n}$, the estimate of $\mathbf{y}_{l,n}$, obtained through the strictly linear filter

$$\hat{\mathbf{y}}_{l,n} = \hat{\mathbf{H}}_n\mathbf{x}_{l,n} \tag{13}$$

where $\hat{\mathbf{H}}_n$ is the estimate of desired parameter matrix $\mathbf{H}$ at time instant n. The estimates $\{\hat{\mathbf{H}}_n : n = 1, 2, \ldots\}$ are selected to iteratively minimize expected value of the cost function

$$\mathcal{J}_n = \frac{1}{|\mathcal{N}|} \sum_{\forall l \in \mathcal{N}} \boldsymbol{\epsilon}_{l,n}^{\mathsf{T}}\boldsymbol{\epsilon}_{l,n}^{\langle\alpha'-1\rangle} \ \text{ with } \ \boldsymbol{\epsilon}_{l,n} = \mathbf{y}_{l,n} - \hat{\mathbf{y}}_{l,n} \tag{14}$$

where $1 < \alpha' < \alpha$.

Remark 1. The fractional order of "$\alpha'-1$" ensures that the cost function, $\mathcal{J}_n$, retains a convex shape. This also ensures that the terms $\forall l \in \mathcal{N} : \boldsymbol{\epsilon}_{l,n}^{\mathsf{T}} \boldsymbol{\epsilon}_{l,n}^{\langle \alpha'-1 \rangle}$ have finite statistical expectations.

At each time instant, the estimate of the parameter matrix is updated in a steepest descent manner expressed as

$$\hat{\mathbf{H}}_{n+1} = \hat{\mathbf{H}}_n - \eta \nabla^{\alpha'-1} \mathcal{J}_n \tag{15}$$

where η is a positive real-valued gain and $\nabla^{\alpha'-1}$ denotes the $(\alpha'-1)$-order gradient operator.[2] Adopting the framework introduced in [30,31] for calculating fractional differentials, the update term in (15) yields

$$\hat{\mathbf{H}}_{n+1} = \hat{\mathbf{H}}_n + \eta \sum_{\forall l \in \mathcal{N}} \boldsymbol{\epsilon}_{l,n} \left(\mathbf{x}_{l,n}^{\langle \alpha'-1 \rangle} \right)^{\mathsf{T}} = \frac{1}{|\mathcal{N}|} \sum_{\forall l \in \mathcal{N}} \boldsymbol{\Phi}_{l,n} \tag{16}$$

where constant multiplicative terms are absorbed into η, while

$$\boldsymbol{\Phi}_{l,n} = \hat{\mathbf{H}}_n + \mu \boldsymbol{\epsilon}_{l,n} \left(\mathbf{x}_{l,n}^{\langle \alpha'-1 \rangle} \right)^{\mathsf{T}} \tag{17}$$

represents the intermediate estimate of $\mathbf{H}$ at time instant n at node l, whereas $\mu = \eta/|\mathcal{N}|$ is an adaptation gain.

The average in (16) can be calculated in a distributed manner using an average consensus filter (ACF) [32,33]. In the context of this work, the state of the ACF after κ iterations is expressed as

$$\mathbf{F}_{i,(\kappa)} = \mathbf{F}_{i,(\kappa-1)} + \sum_{\forall j \in \mathcal{N}_i} m_{i,j} \big(\mathbf{F}_{j,(\kappa-1)} - \mathbf{F}_{i,(\kappa-1)} \big) \tag{18}$$

where $\mathbf{F}_{i,(\kappa)}$ is the state of the filter at node i after κ iterations and $m_{i,j}$ denotes a positive real-valued weight. This can be expressed in a more mathematically convenient fashion as

$$\boldsymbol{\mathcal{F}}_{(\kappa)} = (\boldsymbol{\mathcal{M}} \otimes \mathbf{I}) \boldsymbol{\mathcal{F}}_{(\kappa-1)} = (\boldsymbol{\mathcal{M}}^{\kappa} \otimes \mathbf{I}) \boldsymbol{\mathcal{F}}_{(0)} \tag{19}$$

where $\boldsymbol{\mathcal{F}}_{(\kappa)} = [\mathbf{F}_{1,(\kappa)}^{\mathsf{T}}, \ldots, \mathbf{F}_{|\mathcal{N}|,(\kappa)}^{\mathsf{T}}]^{\mathsf{T}}$, whereas $\mathbf{I}$ is an identity matrix of appropriate size, while the element on the i^{th} row and j^{th} column of $\boldsymbol{\mathcal{M}}$ is selected so that

$$\boldsymbol{\mathcal{M}}_{i,j} = \begin{cases} 1 + m_{i,i} - \sum\limits_{\forall l \in \mathcal{N}_i} m_{i,l} & \text{if } i = j, \\ m_{i,j} & \text{if } i \in \mathcal{N}_j \backslash j \\ 0 & \text{otherwise.} \end{cases}$$

Furthermore, assuming the weights are selected so that $\boldsymbol{\mathcal{M}}$ is also doubly stochastic and meets the conditions in [32], from the work in [32,33], we have

$$\lim_{\kappa \to \infty} \mathbf{F}_{i,(\kappa)} = \frac{1}{|\mathcal{N}|} \sum_{\forall j \in \mathcal{N}} \mathbf{F}_{j,(0)} \tag{20}$$

which is the ACF required to calculate the averages in (16). The operation of the ACF at node i after κ iterations is represented via the following schematic

$$\mathbf{F}_{i,(k)} \leftarrow \boxed{\text{ACF}} \leftarrow \{\forall j \in \mathcal{N} : \mathbf{F}_{j,(0)}\}.$$

The ACF described in (18)-(20) allows the update operation in (16) to be performed in a distributed manner. The operations of such a distributed fractional least mean square (DFLMS) filter are summarized in Algorithm 1, where $\hat{\mathbf{H}}_{l,n}$ denotes the estimate of $\mathbf{H}$ obtained at time instant n at node l. For the case of $\alpha = 2$, as $\alpha' \to 2$; then, the proposed DFLMS (Algorithm 1) simplifies into the distributed least mean square (DLMS) in [34,35].

[2]For more information on application of fractional differentials, the reader is referred to [16]–[18,30,31].

Algorithm 1. Distributed Fractional Least Mean Square

For nodes $l = \{1, \ldots, |\mathcal{N}|\}$:

Estimate Filter Output & Error Term:

$$\hat{\mathbf{y}}_{l,n} = \hat{\mathbf{H}}_{l,n} \mathbf{x}_{l,n} \tag{21}$$

$$\boldsymbol{\epsilon}_{l,n} = \mathbf{y}_{l,n} - \hat{\mathbf{y}}_{l,n} \tag{22}$$

Perform Local Update:

$$\boldsymbol{\Phi}_{l,n} = \hat{\mathbf{H}}_{l,n} + \mu \boldsymbol{\epsilon}_{l,n} \left(\mathbf{x}_{l,n}^{\langle \alpha'-1 \rangle} \right)^{\mathsf{T}} \tag{23}$$

Perform Consensus Update:

$$\hat{\mathbf{H}}_{l,n+1} \leftarrow \boxed{\text{ACF}} \leftarrow \{\forall i \in \mathcal{N} : \boldsymbol{\Phi}_{i,n}\} \tag{24}$$

V. Convergence and Stability Analysis

From the error term defined in (14) we have

$$\boldsymbol{\epsilon}_{l,n} = \mathbf{y}_{l,n} - \hat{\mathbf{y}}_{l,n} = \underbrace{\left(\mathbf{H} - \hat{\mathbf{H}}_{l,n} \right)}_{\boldsymbol{\Upsilon}_{l,n}} \mathbf{x}_{l,n} + \mathbf{w}_{l,n}. \tag{25}$$

Now, substitution of $\boldsymbol{\epsilon}_{l,n}$ from (25) into (23) yields

$$\boldsymbol{\Phi}_{l,n} = \hat{\mathbf{H}}_{l,n} + \mu \boldsymbol{\Upsilon}_{l,n} \mathbf{x}_{l,n} \left(\mathbf{x}_{l,n}^{\langle \alpha'-1 \rangle} \right)^{\mathsf{T}} + \mu \mathbf{w}_{l,n} \left(\mathbf{x}_{l,n}^{\langle \alpha'-1 \rangle} \right)^{\mathsf{T}}. \tag{26}$$

Subsequently, considering $\boldsymbol{\Phi}_{l,n}$ in the formulation in (26) gives

$$\begin{aligned} \mathbf{H} - \boldsymbol{\Phi}_{l,n} = & \boldsymbol{\Upsilon}_{l,n} \left(\mathbf{I} - \mu \mathbf{x}_{l,n} \left(\mathbf{x}_{l,n}^{\langle \alpha'-1 \rangle} \right)^{\mathsf{T}} \right) \\ & - \mu \mathbf{w}_{l,n} \left(\mathbf{x}_{l,n}^{\langle \alpha'-1 \rangle} \right)^{\mathsf{T}}. \end{aligned} \tag{27}$$

From (24) and (27), the error of the estimated parameter matrix can be expressed in a regressive fashion as

$$\boldsymbol{\Upsilon}_{l,n+1} \leftarrow \boxed{\text{ACF}} \leftarrow \{\forall i \in \mathcal{N} : \mathbf{H} - \boldsymbol{\Phi}_{i,n}\}. \tag{28}$$

Replacing the expression in (19) as the ACF in (28) yields

$$\boldsymbol{\mathcal{E}}_{n+1} = \boldsymbol{\mathcal{A}}_n \boldsymbol{\mathcal{E}}_n - \mu \left(\boldsymbol{\mathcal{M}}^{\kappa} \otimes \mathbf{I} \right) \boldsymbol{\mathcal{Q}}_n \tag{29}$$

where $\boldsymbol{\mathcal{A}}_n = (\boldsymbol{\mathcal{M}}^{\kappa} \otimes \mathbf{I}) (\boldsymbol{\mathcal{I}} \otimes \mathbf{I} - \mu \boldsymbol{\mathcal{X}}_n)$, while

$$\begin{aligned} \boldsymbol{\mathcal{E}}_n &= [\boldsymbol{\Upsilon}_{1,n}, \ldots, \boldsymbol{\Upsilon}_{|\mathcal{N}|,n}]^{\mathsf{T}} \\ \boldsymbol{\mathcal{X}}_n &= \text{block-diag}\{\mathbf{x}_{l,n}^{\langle \alpha'-1 \rangle} \mathbf{x}_{l,n}^{\mathsf{T}} : \forall l \in \mathcal{N}\} \\ \boldsymbol{\mathcal{Q}}_n &= [\mathbf{Q}_{l,n}^{\mathsf{T}}, \ldots, \mathbf{Q}_{|\mathcal{N}|,n}^{\mathsf{T}}]^{\mathsf{T}} \text{ with } \mathbf{Q}_{l,n} = \mathbf{x}_{l,n}^{\langle \alpha'-1 \rangle} \mathbf{w}_{l,n}^{\mathsf{T}}. \end{aligned}$$

Alternatively, (29) can be formulated as

$$\begin{aligned} \text{vec}(\boldsymbol{\mathcal{E}}_{n+1}) = & (\mathbf{I} \otimes \boldsymbol{\mathcal{A}}_n) \, \text{vec}(\boldsymbol{\mathcal{E}}_n) \\ & - \mu \text{vec} \left((\boldsymbol{\mathcal{M}}^{\kappa} \otimes \mathbf{I}) \, \boldsymbol{\mathcal{Q}}_n \right) \end{aligned} \tag{30}$$

where $\text{vec}(\cdot)$ transforms a matrix into a column vector by stacking its columns.[3]

[3]The effect of network topology and selection of ACF parameters, i.e., number of iterations and weights, on the estimation error dynamics are indicated by $\boldsymbol{\mathcal{M}}^{\kappa}$.

Considering the expressions in (29) and (30), for the case of $1 < p < \alpha$, $\|\text{vec}(\boldsymbol{\mathcal{E}})\|_p^p$ will converge if the eigenvalues of $\boldsymbol{\mathcal{A}}_n$ lie within the unit circle. In statistical terms, $\mathbb{E}\left\{\|\text{vec}(\boldsymbol{\mathcal{E}}_n)\|_p^p\right\}$ will converge to a stabilizing solution if the eigenvalues of $\mathbb{E}\{\boldsymbol{\mathcal{A}}_n\}$ lie within the unit circle. Given that

$$\mathbb{E}\{\boldsymbol{\mathcal{A}}_n\} = (\boldsymbol{\mathcal{M}}^{\kappa} \otimes \mathbf{I})\left(\boldsymbol{\mathcal{I}} \otimes \mathbf{I} - \mu\mathbb{E}\{\boldsymbol{\mathcal{X}}_n\}\right)$$

and $\boldsymbol{\mathcal{M}}$ was assumed doubly stochastic, resulting in the eigenvalues of $\boldsymbol{\mathcal{M}}^{\kappa}$ lying on or within the unit circle; then, the sufficient condition for convergence is given by

$$\mu < \frac{1}{\lambda_{\max}(\mathbb{E}\{\boldsymbol{\mathcal{X}}_n\})} \tag{31}$$

where $\lambda_{\max}(\mathbb{E}\{\boldsymbol{\mathcal{X}}_n\})$ is the largest eigenvalue of $\mathbb{E}\{\boldsymbol{\mathcal{X}}_n\}$.

Remark 2. The criterion $1 < p < \alpha$ guarantees that all statistical expectations exist and are finite.

Remark 3. If statistics of the regression vectors are not available, a straightforward technique for guaranteeing convergence would be to normalize the adaptation step-size at each time instant, that is, to replace (23) from Algorithm 1 with

$$\boldsymbol{\Phi}_{l,n} = \hat{\mathbf{H}}_{l,n} + \mu\boldsymbol{\epsilon}_{l,n}\left(\mathbf{x}_{l,n}^{\langle\alpha'-1\rangle}\right)^{\mathsf{T}}\left(\mathbf{x}_{l,n}^{\mathsf{T}}\mathbf{x}_{l,n}^{\langle\alpha'-1\rangle}\right)^{-1} \tag{32}$$

where $\mu < 1$ will guarantee convergence.

Remark 4. Taking into account that the background noise and regression processes were assumed to be temporally and spatially independent white zero-mean SαS sequences, taking the statistical expectation of (29) yields

$$\mathbb{E}\{\boldsymbol{\mathcal{E}}_{n+1}\} = (\boldsymbol{\mathcal{M}}^{\kappa} \otimes \mathbf{I})\left(\boldsymbol{\mathcal{I}} \otimes \mathbf{I} - \mu\mathbb{E}\{\boldsymbol{\mathcal{X}}_n\}\right)\mathbb{E}\{\boldsymbol{\mathcal{E}}_n\}. \tag{33}$$

Therefore, if the condition in (31) is satisfied, any misadjustment in initialization values decreases exponentially fast.

VI. Performance Evaluation

In order to demonstrate the performance of the proposed algorithm, the filtering problem defined in Section II-B was considered, where $\alpha = 1.7$, whereas the parameter matrix was

$$\mathbf{H} = \begin{bmatrix} 1 & 0.5 & -1.5 \\ 0.6 & 1 & -0.6 \end{bmatrix}$$

while the covariation matrices of the regression and background noise sequences were

$$\boldsymbol{\Gamma}_{\mathbf{x}} = \boldsymbol{\mathcal{I}} \otimes \begin{bmatrix} 1 & 0.2 & 0.2 \\ 0.2 & 2 & 0.5 \\ 0.2 & 0.5 & 1 \end{bmatrix} \quad \text{and} \quad \boldsymbol{\Gamma}_{\mathbf{w}} = 10^{-2} \times \boldsymbol{\mathcal{I}} \otimes \begin{bmatrix} 1 & 0.1 \\ 0.1 & 1 \end{bmatrix}$$

with the adaptation gain set to $\mu = 0.01$ and $\alpha' = 1.6$. The network with $|\mathcal{N}| = 20$ and the topology shown in Figure 1 was used to estimate the parameter matrix.

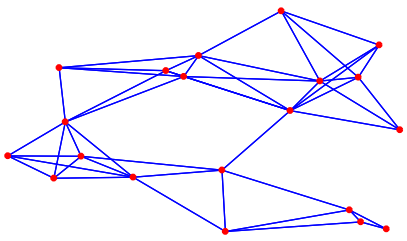

Figure 1. The sensor network with 20 nodes and 48 edges used in simulations.

Two performance indicators were considered: i) the mean absolute error (MAE) term at each node, that is, $\mathbb{E}\{|\boldsymbol{\epsilon}_{l,n}|\}$; ii) the mean absolute deviation (MAD) term at each node, that is, $\mathbb{E}\left\{|\mathbf{H} - \hat{\mathbf{H}}_{l,n}|\right\}$. The mean-values were calculated empirically via averaging of the results obtained from 10^3 independent realizations of the experiment. In addition, at each time instant the ACF was iterated $\kappa = 4$ times to approximate the required averages and the initial value for the parameter matrix estimate was $\forall l \in \mathcal{N} : \hat{\mathbf{H}}_{l,0} = \mathbf{0}$.

In Figure 2, the performance of the developed DFLMS is compared to that of the traditional DLMS in [35]. Observe that the proposed DFLMS, both in its formulation in Algorithm 1 and with the normalized step-size given in (32), performed well in a 1.7-stable environment and achieved lower MAE and MAD than the DLMS. In addition, the DFLMS framework outperformed the DLMS in terms of convergence rate. Finally, notice that the DLMS exhibited large jitters (sharp spikes) in its MAE (*cf.* MAD) behavior as a result of regression and background noise sequences that were heavy-tailed, whereas such behavior was greatly attenuated in the developed DFLMS framework. To enable a fair performance comparison, for DFLMS with normalized step-size, adaptation gain was chosen as $\mu = 0.08$ to achieve a similar convergence rate to that of the DFLMS in Algorithm 1.

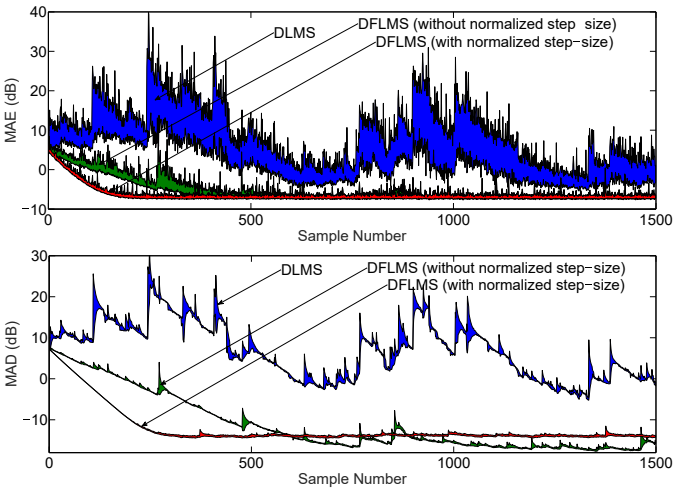

Figure 2. The MAE (top) and MAD (bottom) performance of the DFLMS and DLMS. Performance of the DFLMS across all nodes lies within the region in green, while performance of the DFLMS with normalized step-size lies within the region in red, whereas performance of the DLMS lies within the region in blue.

VII. Conclusion

Adaptive filtering of real-valued SαS signals over sensor networks has been considered where, based on the characteristic function of such signals, an optimal filtering solution has been derived. In addition, an algorithm for filtering SαS signals in a distributed fashion, based on minimizing real-time fractional powers of an error measure using differentials of order "$\alpha' - 1$", where $1 < \alpha' < \alpha$, has been proposed. The performance of the proposed distributed filtering algorithm has been analyzed and convergence bound on the adaptation gain has been established.

REFERENCES

[1] G. Samorodnitsky and M. S. Taqqu, *Stable non-Gaussian random processes*. Chapman & Hall, 1994.

[2] V. V. Uchaikin and V. M. Zolotarev, *Chance and stability: Stable distributions and their applications*. VSP, 1999.

[3] M. Shao and C. L. Nikias, "Signal processing with fractional lower order moments: Stable processes and their applications," *In Proceedings of the IEEE*, vol. 81, no. 7, pp. 986–1010, July 1993.

[4] J. P. Nolan and D. Ojeda-Revah, "Linear and nonlinear regression with stable errors," *Journal of Econometrics*, vol. 172, no. 2, pp. 186–194, February 2013.

[5] N. Azzaoui and L. Clavier, "Statistical channel model based on α-stable random processes and application to the 60 GHz ultra wide band channel," *IEEE Transactions on Communications*, vol. 58, no. 5, pp. 1457–1467, May 2010.

[6] J. P. Nolan, "Modeling financial data with stable distributions," *Handbook of Heavy Tailed Distributions in Finance, Handbooks in Finance, S. T. Rachev, Ed. Elsevier-North Holland, Amsterdam*, 2005.

[7] S. Leglaive, U. Simsekli, A. Liutkus, R. Badeau, and G. Richard, "Alpha-stable multichannel audio source separation," *In Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 576–580, March 2017.

[8] Y. Gao, J. Ni, J. Chen, and X. Chen, "Steady-state and stability analyses of diffusion sign-error LMS algorithm," *Signal Processing*, vol. 149, pp. 62–67, 2018.

[9] J. P. Nolan, "Multivariate elliptically contoured stable distributions: Theory and estimation," *Computational Statistics*, vol. 28, no. 5, pp. 2067–2089, October 2013.

[10] E. Masry, "Alpha-stable signals and adaptive filtering," *IEEE Transactions on Signal Processing*, vol. 48, no. 11, pp. 3011–3016, November 2000.

[11] ——, "Adaptive filtering algorithms and alpha stable signals," *In Proceeding of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 9–12, 2000.

[12] L. Mihaylova, P. Brasnett, A. Achim, D. Bull, and N. Canagarajah, "Particle filtering with alpha-stable distributions," *In Proceedings of IEEE/SP* 13^{th} *Workshop on Statistical Signal Processing*, pp. 381–386, July 2005.

[13] S. P. Talebi and D. P. Mandic, "Distributed particle filtering of α-stable signals," *IEEE Signal Processing Letters*, vol. 24, no. 12, pp. 1862–1866, December 2017.

[14] A. Navia-Vazquez and J. Arenas-Garcia, "Combination of recursive least p-norm algorithms for robust adaptive filtering in alpha-stable noise," *IEEE Transactions on Signal Processing*, vol. 60, no. 3, pp. 1478–1482, March 2012.

[15] G. Aydin, O. Arikan, and A. E. Cetin, "Robust adaptive filtering algorithms for α-stable random processes," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 46, no. 2, pp. 198–202, February 1999.

[16] D. Cafagna, "Fractional calculus: A mathematical tool from the past for present engineers," *IEEE Industrial Electronics Magazine*, vol. 1, no. 2, pp. 35–40, Summer 2007.

[17] G. Cottone, M. D. Paola, and R. Metzler, "Fractional calculus approach to the statistical characterization of random variables and vectors," *Physica A: Statistical Mechanics and its Applications*, vol. 389, no. 5, pp. 909–920, 2010.

[18] Y. Chen, I. Petras, and D. Xue, "Fractional order control - a tutorial," *In Proceeding of American Control Conference*, pp. 1397–1411, June 2009.

[19] A. H. Sayed, "Adaptive networks," *In Proceedings of the IEEE*, vol. 102, no. 4, pp. 460–497, April 2014.

[20] S. P. Talebi, S. Kanna, and D. P. Mandic, "A distributed quaternion Kalman filter with applications to smart grid and target tracking," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 4, pp. 477–488, December 2016.

[21] H. Zhu, G. B. Giannakis, and A. Cano, "Distributed in-network channel decoding," *IEEE Transactions on Signal Processing*, vol. 57, no. 10, pp. 3970–3983, October 2009.

[22] S. P. Talebi, S. Kanna, Y. Xia, and D. P. Mandic, "Cost-effective diffusion Kalman filtering with implicit measurement exchanges," *In Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 4411–4415, March 2017.

[23] A. H. Sayed, "Adaptation, learning, and optimization over networks," *Foundations and Trends in Machine Learning*, vol. 7, no. 4-5, pp. 311–801, 2014.

[24] V. Kekatos and G. B. Giannakis, "Distributed robust power system state estimation," *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 1617–1626, May 2013.

[25] D. P. Mandic, Y. Xia, and D. Dini, "Frequency estimation," Patent US 9,995,774 B2, June, 2018.

[26] Y. Xia, D. P. Mandic, and A. H. Sayed, "An adaptive diffusion augmented CLMS algorithm for distributed filtering of noncircular complex signals," *IEEE Signal Processing Letters*, vol. 18, no. 11, pp. 659–662, November 2011.

[27] S. Werner, M. L. R. de Campos, and P. S. R. Diniz, "Partial-update NLMS algorithms with data-selective updating," *IEEE Transactions on Signal Processing*, vol. 52, no. 4, pp. 938–949, April 2004.

[28] R. Arablouei, S. Werner, Y. F. Huang, and K. Dogancay, "Distributed least mean-square estimation with partial diffusion," *IEEE Transactions on Signal Processing*, vol. 62, no. 2, pp. 472–484, January 2014.

[29] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3122–3136, July 2008.

[30] G. Jumarie, "On the derivative chain-rules in fractional calculus via fractional difference and their application to systems modelling," *Central European Journal of Physics*, vol. 11, no. 6, pp. 617–633, 2013.

[31] ——, "Modified Riemann-Liouville derivative and fractional Taylor series of nondifferentiable functions further results," *Computers & Mathematics with Applications*, vol. 51, no. 9, pp. 1367–1376, 2006.

[32] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," *In Proceedings of Fourth International Symposium on Information Processing in Sensor Networks*, pp. 63–70, 2005.

[33] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.

[34] F. S. Cattivelli and A. H. Sayed, "Multi-level diffusion adaptive networks," pp. 2789–2792, April 2009.

[35] ——, "Diffusion LMS strategies for distributed estimation," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1035–1048, March 2010.