Signal Processing 166 (2020) 107220

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Distributed adaptive node-specific signal estimation in a wireless sensor network with noisy links☆

Fernando de la Hucha Arce[a,*], Marc Moonen[a], Marian Verhelst[b], Alexander Bertrand[a]

[a] Stadius Center for Dynamical Systems, Signal Processing and Data Analytics, KU Leuven, Dept. of Electrical Engineering (ESAT), Kasteelpark Arenberg 10, 3001 Leuven, Belgium
[b] MICAS Research Group, KU Leuven, Dept. of Electrical Engineering (ESAT), Kasteelpark Arenberg 10, 3001 Leuven, Belgium

ARTICLE INFO

Article history:
Received 8 January 2019
Revised 22 May 2019
Accepted 12 July 2019
Available online 22 July 2019

Keywords:
Wireless sensor networks
Signal estimation
Noisy links
Quantization

ABSTRACT

We consider a distributed signal estimation problem in a wireless sensor network where each node aims to estimate a node-specific desired signal using all sensor signals available in the network. In this setting, the distributed adaptive node-specific signal estimation (DANSE) algorithm is able to learn optimal fusion rules with which the nodes fuse their sensor signals, as the fused signals are then transmitted between the nodes. Under the assumption of transmission without errors, DANSE achieves the performance of centralized estimation. However, noisy communication links introduce errors in these transmitted signals, e.g., due to quantization or communication errors. In this paper we show fusion rules which take additive noise in the transmitted signals into account at almost no increase in computational complexity, resulting in a new algorithm denoted as 'noisy-DANSE' (N-DANSE). As the convergence proof for DANSE cannot be straightforwardly generalized to the case with noisy links, we use a different strategy to prove convergence of N-DANSE, which also proves convergence of DANSE without noisy links as a special case. We validate the convergence of N-DANSE and compare its performance with the original DANSE through numerical simulations, which demonstrate the superiority of N-DANSE over the original DANSE in noisy links scenarios.

© 2019 Elsevier B.V. All rights reserved.

1. Introduction

A wireless sensor network (WSN) consists of a set of nodes which collect information from the environment using their sensors, and which are able to exchange data over wireless communication links. The goal of the network is usually to infer information about a physical phenomenon from the sensor data gathered by the nodes.

A common paradigm for sensor data fusion in WSNs is the centralized approach, where the sensor data are transmitted to one node with a large energy budget and high computational power, usually called the fusion centre. However, wireless communication is often expensive in terms of energy and bandwidth, and nodes that are powered by batteries need to carefully manage their own energy budget to allow the network to function for a reasonable lifetime [1]. Distributed processing is an alternative paradigm where the computational task is divided among the nodes, as opposed to being carried out single-handedly by a fusion centre. Instead of transmitting their raw sensor data, nodes only transmit the results from their local computations, which allows for a reduction in the amount of data exchanged among nodes.

Here we focus on signal estimation, where the goal of each node is to continuously estimate a desired signal for each sample time at the sensors through a spatio-temporal filtering of all of the sensor signals available in the network. We assume that the desired signals are node-specific, yet all the desired signals from the different nodes are assumed to span a low-dimensional signal subspace, which defines a latent 'common interest'. A particular instance is where all the nodes estimate a local (node-specific) observation(s) of the same source signal(s). This is important in several applications where preserving spatial information is necessary, such as localization [2–5], speech enhancement in binaural hearing aids [6–8] and per-channel artifact removal in electroencephalography (EEG) sensor networks [9].

☆ This research work was carried out at the ESAT Laboratory of KU Leuven, in the frame of Research Fund KU Leuven C14/16/057, FWO projects nr. G.0931.14 and nr. 1.5.123.16N, and European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 802895). The scientific responsibility is assumed by the authors. The authors marked with [1] are members of EURASIP.

* Corresponding author.
E-mail addresses: fernando.delahuchaarce@esat.kuleuven.be (F. de la Hucha Arce), marc.moonen@esat.kuleuven.be (M. Moonen), marian.verhelst@esat.kuleuven.be (M. Verhelst), alexander.bertrand@esat.kuleuven.be (A. Bertrand).

https://doi.org/10.1016/j.sigpro.2019.07.013
0165-1684/© 2019 Elsevier B.V. All rights reserved.

Several algorithms have been designed for node-specific signal estimation that allow every node to learn the optimal fusion rules to fuse their own sensor signals and then transmit them to the other nodes. Under the assumption that the fused signals are transmitted without errors, every node converges to the centralized linear minimum mean squared error (MMSE) estimate of its node-specific desired signal. These algorithms are generally classified under the DANSE acronym, which stands for distributed adaptive node-specific signal estimation. The original DANSE algorithm has been designed for fully connected network topologies [10,11] and then extended to tree topologies [12], hybrid (tree plus clique) topologies [13] and eventually for any network topology [14]. For low SNR and non-stationary conditions, a low rank covariance matrix approximation based on a generalized eigenvalue decomposition (GEVD) has been incorporated into the DANSE algorithm [15], which also relaxes the assumptions on the desired signals spanning a low-dimensional subspace.

The transmission of linearly fused sensor signals allows the DANSE algorithm to significantly reduce the data exchange in the WSN while converging to the same node-specific desired signal estimates as the centralized approach. However, noise can be introduced in the transmitted signals when the communication links are noisy, for instance as a result of quantization of the fused signals [16] prior to transmission, or communication errors.

The effect of noisy links in WSNs has been studied extensively in the context of parameter estimation, where the estimation variable is a parameter vector of fixed dimension, which is generally assumed to be static or slowly varying over time. This allows for an iterative refinement process where intermediate estimates are exchanged between the nodes until convergence to a steady state regime is achieved. The distributed consensus-based estimation framework with noisy links has been studied in [17] for deterministic parameters and in [18] for random parameters, where the authors show the resilience of their algorithms to additive noise resulting from quantization and/or communication processes. The convergence of distributed consensus with dithered quantized observations and random link failures has been considered in [19]. The design of a quantizer whose quantization levels are progressively adapted to ensure the convergence of a distributed consensus algorithm has been studied in [20]. In the context of diffusion-based approaches to parameter estimation, the effect of noisy links has also been the subject of study. A study of diffusion adaptation with noisy links has been presented in [21], where the authors derive an optimal strategy for adjusting the combination weights for two-node networks. The effect of noisy links in the steady-state performance of diffusion least-mean-square (LMS) adaptive networks has been analyzed in [22], where convergence can still be proven but the performance is shown to depend non-monotonically on the step size. A similar analysis for the steady state for partial diffusion recursive LMS adaptation with noisy links is provided in [23]. More recently, a variable step-size diffusion LMS algorithm that explicitly takes into account the link noise has been proposed in [24]. Distributed estimation of a Gaussian parameter subject to unknown multiplicative noise and additive Gaussian noise has been studied in the context of quantization in a WSN with centralized architecture [25], where an analysis of different bit rate allocation methods is also provided.

In contrast to parameter estimation, in signal estimation a time series corresponding to the sensor sample times is estimated such that the dimension of the estimation variable grows with every new frame of sensor signal samples [7,10,26–28]. One possible approach is to treat each new frame of sensor signal samples as a new parameter vector to be estimated [27,29]. However, starting a new iterative parameter estimation process for every such frame would rapidly become expensive in terms of time and energy, particularly when a high sampling rate is required such as in audio signal processing applications. Therefore, signal estimation in WSNs often relies on the design of linear spatio-temporal fusion rules such as those mentioned by the DANSE algorithm [10–12,14,26]. Rather than iterating on the estimation variables directly, the iterations are performed on these fusion rules instead, in order to adapt them over time in a data-driven fashion, where a new frame of sensor observations can be used in each iteration. Unlike in the literature on parameter estimation in WSNs, the effect of noisy links, i.e., the presence of additional noise in the transmitted signals, is generally not considered in the existing literature on signal estimation in WSNs.

In this paper we focus on the DANSE algorithm for distributed signal estimation in a WSN with noisy links, i.e., when noise is introduced into the fused and transmitted signals due to, e.g., quantization or communication errors. We derive fusion rules that take this additional noise into account at almost no increase in computational complexity, resulting in a modified version of the DANSE algorithm, referred to as "noisy"-DANSE or N-DANSE for short. The convergence proof in [10] of the original DANSE algorithm cannot be straightforwardly generalized in the case with noisy links. Furthermore, as opposed to the original DANSE algorithm, the new N-DANSE algorithm minimizes an upper bound on the per-node mean squared errors. Therefore, we adopt a different strategy to prove convergence of the N-DANSE algorithm with noisy links. This new proof then also contains the convergence of DANSE without noisy links as a special case.

The paper is structured as follows. In Section 2 we formulate the problem statement and the signal model, and we briefly review the centralized approach to linear MMSE estimation. In Section 3 we review the DANSE algorithm, which facilitates the exposition of the rest of the paper. In Section 4 we derive the modified version of DANSE, named noisy-DANSE or N-DANSE, to account for noisy links, i.e., additive noise in the transmitted signals. In Section 5 we prove convergence of the N-DANSE algorithm to a unique point. In Section 6 we provide numerical simulations, supporting our analysis. Finally, we present the conclusions in Section 7.

2. Signal model and linear MMSE estimation

2.1. Signal model

We consider a WSN composed of K nodes, where the k-th node has access to M_k sensor signals. We denote the set of nodes by $\mathcal{K} = \{1, \ldots, K\}$ and the total number of sensors by $M = \sum_{k\in\mathcal{K}} M_k$. The sensor signal y_{km} captured by the m-th sensor of the k-th node is modelled as the combination of a node-specific desired signal component x_{km} and an undesired noise component v_{km}, which can be expressed mathematically as

$$y_{km}[t] = x_{km}[t] + v_{km}[t], \quad m \in \{1, \ldots, M_k\}, \tag{1}$$

where $t \in \mathbb{N}$ denotes the discrete time index of the sensor signal samples. In order to allow frequency domain representations, we assume that the sensor signals y_{km} are complex-valued, and we denote complex conjugation with the superscript $(\cdot)^*$. We assume that the desired signal components x_{km} are uncorrelated with the undesired noise components v_{km} for all nodes and sensors. It is noted that correlation may exist within or across nodes for the desired signal components and for the undesired noise components, i.e., $E\{x_{km}x_{qn}^*\}$ and $E\{v_{km}v_{qn}^*\}$ are not necessarily zero. We remark that no statistical distribution, Gaussian or otherwise, is assumed on the sensor signals y_{km} or their components x_{km}, v_{km}. Besides, we assume that all sensor signals are realizations of short-

term wide-sense stationary[1] and short-term ergodic stochastic processes.

We denote by $\mathbf{y}_k$ the $M_k \times 1$ vector containing the M_k sensor signals of node k, i.e.,

$$\mathbf{y}_k = [y_{k1}, \ldots, y_{kM_k}]^T , \tag{2}$$

where the superscript $(\cdot)^T$ denotes the transpose operator. For the sake of an easy exposition, we will omit the discrete time index t when referring to a signal, and include it only when referring to a specific observation, e.g., a sample of the M_k-channel sensor signal $\mathbf{y}_k$ collected by node k at sample time t is denoted by $\mathbf{y}_k[t]$. The $M_k \times 1$ vectors $\mathbf{x}_k$ and $\mathbf{v}_k$ are defined in a similar manner, such that

$$\mathbf{y}_k = \mathbf{x}_k + \mathbf{v}_k . \tag{3}$$

We assume that the node-specific desired signal components $\mathbf{x}_k$ are related to a desired source signal s through an unknown steering vector $\mathbf{a}_k$ such that

$$\mathbf{x}_k = \mathbf{a}_k s, \quad \forall k \in \mathcal{K}, \tag{4}$$

where $\mathbf{a}_k$ is an $M_k \times 1$ vector containing the transfer functions from the source to the sensors. Note that we assume a single desired source signal s to be present in order to simplify the exposition in Sections 3 and 4. However, all results can be extended to the case with multiple desired sources in a similar fashion as in the original DANSE algorithm [10].

The goal of each node k is to estimate the desired signal component $x_{k\tilde{m}}$ in its $\tilde{m}$-th sensor, where $\tilde{m}$ can be freely chosen. We only estimate one signal per node as this will simplify the notation later on. However, this is without loss of generality, as the optimal estimation of other channels of $\mathbf{x}_k$ can be obtained as a by-product in the (N-)DANSE algorithm without increasing the required communication bandwidth. We will explain this in Section 3 under Eq. (19). To simplify the notation, we denote by d_k the desired signal of the k-th node, i.e.,

$$d_k = x_{k\tilde{m}} . \tag{5}$$

Note that in (4) neither the source signal s nor the desired signal components $\mathbf{x}_k$ are observable, and that the steering vector $\mathbf{a}_k$ is also unknown. We do not attempt to estimate neither s nor $\mathbf{a}_k$ since we aim to preserve the characteristics of the desired signals as they are observed by each node. This is relevant in several applications where it is important to estimate signals at specific node locations, as explained in Section 1 and references therein.

Finally, we highlight that the signal model given by (1)-(4) includes convolutive time-domain mixtures, described as instantaneous mixtures in the frequency domain. In this case, the framework is applied in the short-term Fourier transform domain in each frequency bin separately [30].

2.2. Centralized linear MMSE estimation

We first consider the centralized estimation problem where every node has access to the network-wide $M \times 1$ sensor signal vector $\mathbf{y}$, given by

$$\mathbf{y} = \left[\mathbf{y}_1^T, \ldots, \mathbf{y}_K^T\right]^T . \tag{6}$$

The network-wide desired signal component vector $\mathbf{x}$ and noise component vector $\mathbf{v}$ are defined in an similar manner, such that $\mathbf{y} = \mathbf{x} + \mathbf{v}$. In this case, the goal for the k-th node is to estimate its desired signal d_k based on a linear MMSE estimator $\hat{\mathbf{w}}_k$ which minimizes the cost function

$$J_k(\mathbf{w}_k) = E\left\{ \left| d_k - \mathbf{w}_k^H \mathbf{y} \right|^2 \right\}, \tag{7}$$

where $E\{\cdot\}$ is the expectation operator and $(\cdot)^H$ denotes conjugate transpose. Assuming that the sensor signal correlation matrix $\mathbf{R}_{yy} = E\{\mathbf{y}\mathbf{y}^H\}$ has full rank,[2] the unique minimizer of (7) is given by

$$\hat{\mathbf{w}}_k = \mathbf{R}_{yy}^{-1} \mathbf{r}_{yd_k} , \tag{8}$$

where $\mathbf{r}_{yd_k} = E\{\mathbf{y}d_k^*\}$. The estimate of the desired signal d_k of the k-th node is given by

$$\hat{d}_k = \hat{\mathbf{w}}_k^H \mathbf{y} . \tag{9}$$

2.3. Estimation of signal statistics

The matrix $\mathbf{R}_{yy}$ can be estimated through sample averaging, for instance using a sliding window,

$$\mathbf{R}_{yy}[t] = \sum_{n=t-L+1}^{t} \mathbf{y}[n]\mathbf{y}[n]^H , \tag{10}$$

where L is the size of the sliding window.

Sample averaging is not possible for $\mathbf{r}_{yd_k}$ since the desired signals d_k are not observable, and hence its estimation has to be done indirectly [10]. Using (3), (5) and the fact that $\mathbf{x}$ and $\mathbf{v}$ are uncorrelated,[3] $\mathbf{r}_{yd_k}$ can be expressed as

$$\mathbf{r}_{yd_k} = \mathbf{R}_{xx}\mathbf{c}_k , \tag{11}$$

where $\mathbf{R}_{xx} = E\{\mathbf{x}\mathbf{x}^H\}$ and $\mathbf{c}_k$ is an $M \times 1$ selection vector whose entry corresponding to the $\tilde{m}$-th channel of $\mathbf{x}_k$ is one, and all other entries are zero.

In cases where the desired source has an 'on-off' behaviour, as in speech [6,30,31] or EEG signal enhancement [9], the noise correlation matrix $\mathbf{R}_{vv} = E\{\mathbf{v}\mathbf{v}^H\}$ can be estimated during periods when the desired source is not active, since then the sensor signal samples only contain a noise component. Since we assume that $\mathbf{x}$ and $\mathbf{v}$ are uncorrelated and $\mathbf{v}$ is zero-mean, it is then possible to use the relationship $\mathbf{R}_{xx} = \mathbf{R}_{yy} - \mathbf{R}_{vv}$ to obtain an estimate of $\mathbf{R}_{xx}$. More advanced data-driven techniques to estimate $\mathbf{R}_{xx}$ that rely on subspace methods have been developed in [15,31].

3. The DANSE algorithm

In this section we provide a brief review of the DANSE algorithm. For more details we refer the reader to [10,11].

In the context of WSNs, the k-th node only has access to its own sensor signals $\mathbf{y}_k$, and thus every node would need to exchange their complete set of sensor signals with every other node in order to compute the optimal linear MMSE estimator $\hat{\mathbf{w}}_k$ (8) and the corresponding optimal signal estimate $\hat{d}_k = \hat{\mathbf{w}}_k^H \mathbf{y}$ (9). This would require a significant amount of energy and bandwidth [1]. The DANSE algorithm allows to obtain the optimal linear MMSE estimates of the desired signals without requiring a full exchange of all the sensor signals.

For the sake of brevity and clarity of exposition, we consider the DANSE algorithm in a fully connected network as in [10]. However, it is noted that the DANSE algorithm has also been adapted to a tree topology [12] and to be topology independent [14][4].

[1] This assumption is added to simplify the theoretical derivations. In practice, the assumption is relaxed to stationarity of the spatial coherence between every pair of sensor signals y_{km} and y_{qn}. This means that non-stationary sources (such as speech) are allowed, as long as the transfer functions from sources to sensors remain static or vary only slowly compared to the tracking speed of the DANSE algorithm [30].

[2] This assumption is usually satisfied in practice due to the presence of a noise component in each sensor that is independent of other sensor signals, such as thermal noise. If this is not the case, the pseudoinverse has to be used.

[3] For the sake of easy exposition, we also assume that the noise components $\mathbf{v}$ are zero mean.

[4] In Section 6.8 we compare the N-DANSE and DANSE algorithms in a tree topology through numerical simulations.

The main idea behind the DANSE algorithm is that each node k can optimally fuse its own M_k-channel sensor signal vector $\mathbf{y}_k$ to generate the single-channel fused signal z_k, given by

$$z_k = \mathbf{f}_k^H \mathbf{y}_k \quad \forall k \in \mathcal{K}, \tag{12}$$

where the $M_k \times 1$ fusion vector $\mathbf{f}_k$ will be defined later in (19). Each node k then transmits its fused signal z_k to all other nodes in the network. As every z-signal is received by all the nodes in the network, a node k has access to an $(M_k + K - 1)$-channel signal, consisting of its own M_k sensor signals $\mathbf{y}_k$ and the $K-1$ z-signals from other nodes, which can be collected in the $(K-1)\times 1$ vector $\mathbf{z}_{-k} = [z_1, \ldots, z_{k-1}, z_{k+1}, \ldots, z_K]^T$, where the subscript '$-k$' refers to the signal z_k not being included. The $(M_k + K - 1)$-channel signal in node k is then defined as

$$\tilde{\mathbf{y}}_k = \begin{bmatrix} \mathbf{y}_k \\ \mathbf{z}_{-k} \end{bmatrix} = \tilde{\mathbf{x}}_k + \tilde{\mathbf{v}}_k \,. \tag{13}$$

Node k can use $\tilde{\mathbf{y}}_k$ to estimate its desired signal d_k using a local linear MMSE estimator $\tilde{\mathbf{w}}_k$ given by

$$\tilde{\mathbf{w}}_k = \arg\min_{\mathbf{w}} E\left\{ |d_k - \mathbf{w}^H \tilde{\mathbf{y}}_k|^2 \right\}. \tag{14}$$

Note that the DANSE algorithm needs to find the optimal fusion vectors $\mathbf{f}_k$ and the optimal estimators $\tilde{\mathbf{w}}_k$ for every node-specific signal $d_k \, \forall k \in \mathcal{K}$. To solve this, the DANSE algorithm iteratively updates the fusion vectors $\mathbf{f}_k$ in (12) for all nodes one by one in a round-robin fashion. To this end, we introduce the iteration index $i \in \mathbb{N}$ and write it in the subscript of all variables that are influenced by $\mathbf{f}_k$, e.g., $z_k^i = \mathbf{f}_k^{iH} \mathbf{y}_k$. In every iteration, each node $k \in \mathcal{K}$ updates its local estimator as

$$\tilde{\mathbf{w}}_k^{i+1} = \arg\min_{\mathbf{w}} E\left\{ \left| d_k - \mathbf{w}^H \tilde{\mathbf{y}}_k^i \right|^2 \right\}, \tag{15}$$

which is then given by (compare with (8)-(11))

$$\tilde{\mathbf{w}}_k^{i+1} = \left(\mathbf{R}^i_{\tilde{y}_k \tilde{y}_k} \right)^{-1} \mathbf{R}^i_{\tilde{x}_k \tilde{x}_k} \tilde{\mathbf{c}}_k \,, \tag{16}$$

where $\mathbf{R}^i_{\tilde{y}_k \tilde{y}_k} = E\{\tilde{\mathbf{y}}_k^i \tilde{\mathbf{y}}_k^{iH}\}$, $\mathbf{R}^i_{\tilde{x}_k \tilde{x}_k} = E\{\tilde{\mathbf{x}}_k^i \tilde{\mathbf{x}}_k^{iH}\}$ and $\tilde{\mathbf{c}}_k$ is the $(M_k + K - 1)\times 1$ selection vector whose $\tilde{m}$-th entry is one and all other entries are zero. The estimated desired signal at any node k is then

$$\tilde{d}_k^i = \left(\tilde{\mathbf{w}}_k^{i+1} \right)^H \tilde{\mathbf{y}}_k^i = \left(\boldsymbol{\psi}_k^{i+1} \right)^H \mathbf{y}_k + \left(\mathbf{g}_{k,-k}^{i+1} \right)^H \mathbf{z}_{-k}^i \,, \tag{17}$$

where we used the following partitioning of the node-specific estimator $\tilde{\mathbf{w}}_k^{i+1}$,

$$\tilde{\mathbf{w}}_k^{i+1} = \begin{bmatrix} \boldsymbol{\psi}_k^{i+1} \\ \mathbf{g}_{k,-k}^{i+1} \end{bmatrix}, \tag{18}$$

where $\boldsymbol{\psi}_k^{i+1}$ and $\mathbf{g}_{k,-k}^{i+1}$ are vectors of dimensions $M_k \times 1$ and $(K-1)\times 1$ respectively, and the elements of $\mathbf{g}_{k,-k}^{i+1}$ are given by $\mathbf{g}_{k,-k}^{i+1} = [g_{k1}^{i+1}, \ldots, g_{k,k-1}^{i+1}, g_{k,k+1}^{i+1}, \ldots, g_{kK}^{i+1}]^T$. After applying (16) in each node, one node, say node k, will also update its fusion vector based on its $\boldsymbol{\psi}_k^{i+1}$, i.e.,

$$\mathbf{f}_k^{i+1} = \boldsymbol{\psi}_k^{i+1} \,, \tag{19}$$

whereas the fusion vectors of all the other nodes remain unchanged[5] [10]. The updating node k changes in a round-robin fashion from 1 to K through the iterations. It is noted that, if the other channels of $\mathbf{x}_k$ would be included as desired signals in (5), the selection vector $\tilde{\mathbf{c}}_k$ in (16) would become a selection matrix with M_k columns, and similarly the estimator $\tilde{\mathbf{w}}_k$ would also become a matrix with M_k columns, one for each channel of $\mathbf{x}_k$. Nevertheless, only one column has to be selected to compute the fusion vector $\mathbf{f}_k$, since all columns would be the same up to scaling due to (4), and thus no extra data would need to be transmitted in that case.

[5] A version of the algorithm in which all the nodes can update their fusion rules simultaneously has been proposed in [11]. We consider this case through numerical simulations in Section 6.7.

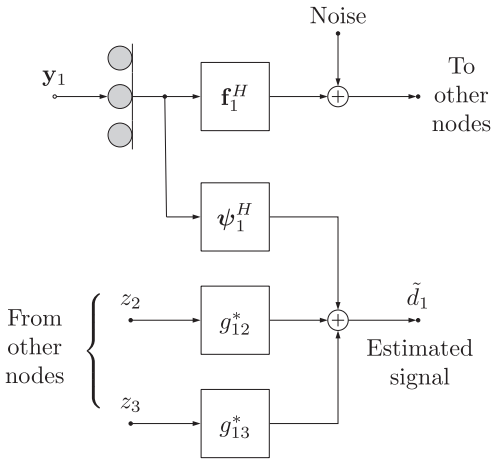


Fig. 1. Diagram of signal flow in node 1 for the (N-)DANSE algorithm in a network with three nodes ($K = 3$). The square boxes denote a multiplication from the left-hand side (i.e., $\boldsymbol{\psi}_1^H \mathbf{y}_1$).

Under Assumption 4, it is proven in [10] that the update (19) results in a sequence of node-specific estimators $\{\tilde{\mathbf{w}}_k^i, \forall k \in \mathcal{K}, \forall i \in \mathbb{N}\}$ which converges to a stable equilibrium as $i \to \infty$. In this convergence point, at each node k the estimated desired signal $\tilde{d}_k^i$ in (17) is equal to the centralized node-specific estimated signal $\hat{d}_k = \hat{\mathbf{w}}_k^H \mathbf{y}$, where $\hat{\mathbf{w}}_k$ is the node-specific estimator defined in (8).

As an example, a diagram of the signal flow inside node 1 in a network of $K = 3$ nodes is shown in Fig. 1. The additive noise in the fused signals z_k is introduced in Section 4, and is to be ignored for the time being.

We highlight the fact that, while due to the iterative nature of the DANSE algorithm it may appear that the same sensor signal observations are fused and transmitted several times over the sequence of iterations, this is not the case in practice. In practical applications the iterations are spread over time, such that the updates of $\mathbf{f}_k$ are performed over different sensor signal observations. These sensor signal observations are usually processed in frames. The updated fusion vectors and node-specific estimators are then only applied to the next incoming sensor signal observations. An explicit description of the processing in frames will be provided for the N-DANSE algorithm in Section 4 (Algorithm 1). This description is also valid for the DANSE algorithm as explained above, as we will show that the N-DANSE algorithm is a generalization of the DANSE algorithm.

We also note that, since the (N)-DANSE algorithm is intended to perform spatial filtering (or beamforming), there is an inherent assumption of synchronization across all y and z-signals that is only there if temporal filtering is included. As a consequence, clock drift needs to be handled either by an explicit synchronization protocol or by compensation within the algorithm itself. The latter is beyond the scope of the paper, but we refer the interested reader to [32–34].

4. The N-DANSE algorithm: additive noise in the transmitted signals

4.1. Noisy links

Let us now consider the presence of additive noise in the transmitted signals. We denote by z_{kq}^i the signal transmitted by the

k-th node and received by the q-th node at iteration i. With additive noise, it is given by

$$z_{kq}^i = \mathbf{f}_k^{iH}\mathbf{y}_k + e_{kq}^i\,, \tag{20}$$

where e_{kq}^i denotes the noise added during the communication process between node k and node q. In Fig. 1 a diagram of the signal flow for node 1 is depicted as an example.

We make the following assumptions about the additive noise:

- The additive noises e_{kl}^i, e_{qp}^i have zero mean and are mutually uncorrelated, i.e., $E\{e_{kl}^i(e_{qp}^i)^*\} = 0,\ \forall q \neq k$.
- The additive noise e_{kq}^i and the signals $\mathbf{y}_k$ are uncorrelated, i.e., $E\{\mathbf{y}_k(e_{kq}^i)^*\} = 0\,\forall k, q \in \mathcal{K}$.
- The second order moment of the additive noise e_{kq}^i is linearly related to the second order moment of the fused signal $\mathbf{f}_k^{iH}\mathbf{y}_k$, i.e.,

$$E\left\{|e_{kq}^i|^2\right\} = \beta_k E\left\{|\mathbf{f}_k^{iH}\mathbf{y}_k|^2\right\}\forall k, q \in \mathcal{K}\,. \tag{21}$$

We assume that the parameter β_k is known by node k.

Note that Assumption 21 is without loss of generality, as the signal $\mathbf{f}_k^H\mathbf{y}_k$ is usually scaled before transmission to maximally cover the available dynamic range. A scaling of z_{kq} has no influence in the dynamics of the algorithm, as the scaling will be compensated for by the $\mathbf{g}_{k,-k}$ coefficients in (18). Besides, it is also noted that (21) means that the variances of the additive noises e_{kq} depend only on the transmitting node k. Although each node q receives a different version of $\mathbf{f}_k^H\mathbf{y}_k$ with different decoding errors e_{kq}, their impact has comparable magnitude since wireless links are generally designed to satisfy a certain target bit error rate. Besides, the chosen coding scheme of each node has a comparable effect on all receiving nodes, e.g., a weak coding scheme would result in more decoding errors in all nodes which receive its signal. Furthermore, this model also covers quantization errors introduced at the transmitting node k. We also highlight the fact that no statistical distribution, Gaussian or otherwise, is assumed on the additive noises e_{kq}, which is also important to allow the modelling of different transmission errors such as communication and quantization noise.

In the particular case of uniform quantization, the mathematical properties of quantization noise have been extensively studied [16,35,36]. In our framework this would happen when the signals $\mathbf{f}_k^{iH}\mathbf{y}_k, \forall k \in \mathcal{K}$, are subject to uniform quantization prior to their transmission, in which case the parameter β_k in (21) can be shown to be given by [16]

$$\beta_k = \frac{\Delta_{b_k}^2}{12\,E\left\{\left|\mathbf{f}_k^H\mathbf{y}_k\right|^2\right\}}\,, \tag{22}$$

where $\Delta_{b_k} = A_k/2^{b_k}$. The parameter A_k is given by the dynamic range[6] of the fused signal $\mathbf{f}_k^{iH}\mathbf{y}_k$, and b_k is the number of bits used by the k-th node to quantize its fused signal $\mathbf{f}_k^{iH}\mathbf{y}_k$. Quantization in the frequency domain can also be considered following the model discussed above, as explained in [37].

In the remainder of this section we propose a modified version of the DANSE algorithm, referred to as noisy-DANSE or N-DANSE for short, for the noisy links case (20). A convergence proof for the N-DANSE algorithm is provided in Section 5, based on a different strategy than in [10].

[6] The dynamic range is usually chosen to be several standard deviations of the signal, i.e., $A_k^2 \propto E\{|\mathbf{f}_k^H\mathbf{y}_k|^2\}$, such that (22) is independent of $\mathbf{f}_k$.

4.2. Fusion vectors for N-DANSE

Fusion vectors govern how useful the z-signals are to the estimation problems of other nodes. In the original DANSE algorithm, each node finds its optimal fusion vector as part of the solution to its own local estimation problem, as given in (19). In the presence of noisy links, modelled by (20), the update of the fusion vector of node k must take into account the additional noise terms e_{kq} which are present in the estimation problems of other nodes $q \neq k$.

The main idea is to define an additional cost function that is minimized in the updating node k to define the fusion vector $\mathbf{f}_k$. Although this cost function can only contain information available to node k, let us first consider the case as if node k had access to all the noisy z-signals received by all the other nodes in the network, i.e. $\mathbf{z}_{-k,q} = [z_{1q}, \ldots, z_{k-1,q}, z_{k+1,q}, \ldots, z_{Kq}]^T$, for all[7] $q \neq k$. A proper fusion rule $\mathbf{f}_k$ would be one that minimizes the total estimation error across all other nodes q, assuming node q estimates d_q using all its received z-signals, including the -to be optimized- $z_{kq} = \mathbf{f}_k^H\mathbf{y} + e_{kq}$. This leads to the following cost function

$$J_k^s\left(\mathbf{f}_k, h_{1k}, \ldots, h_{Kk}, \mathbf{h}_{1,-k}, \ldots, \mathbf{h}_{K,-k}\right) = \sum_{q\in\mathcal{K}\setminus\{k\}} E\left\{|d_q - \left(\mathbf{f}_k^H\mathbf{y}_k + e_{kq}\right)h_{qk}^* - \mathbf{h}_{q,-k}^H\mathbf{z}_{-k,q}|^2\right\}, \tag{23}$$

where the h-coefficients are auxiliary optimization variables that mimic the choice of the g-coefficients at other nodes. Note that this is an upper bound on the actual achievable total mean squared error (MSE), as node q can use its local sensor signal $\mathbf{y}_q$ in its local estimation problem instead of z_{qq}, which offers more degrees of freedom and is free of additive noise. However, $\mathbf{y}_q$ cannot be included in (23), as the updating node k does not have access to it. Nevertheless, it is important to emphasize that the actual total MSE achieved by the network will always be *lower*, and thus better, than predicted by this bound. Note that finding the fusion vectors which minimize the total MSE would only be possible if nodes had access to all the information in the WSN, i.e., all sensor signals $\mathbf{y}_k$ and all additive noises e_{kq}. In Section 6.5, we demonstrate the impact of using this upper bound by comparing the result with a 'clairvoyant' algorithm where all this information would be available (see also Appendix C).

Using the assumptions on the noise statistics as listed in the previous subsection, we show in Appendix A that the cost function (23) is identical to a similar cost function in which all the $\mathbf{z}_{-k,q}$ can be replaced with $\mathbf{z}_{-k,k}$, i.e., the noisy version of $\mathbf{z}_{-k}$ as observed at node k. This means that the second subscript in $\mathbf{z}_{-k,q}$ is interchangeable in the cost function (23). Therefore, we replace $\mathbf{z}_{-k,k}$ with $\mathbf{z}_{-k}$ in the sequel for the sake of an easier exposition[8]. This leads to the new cost function

$$J_k^s\left(\mathbf{f}_k, h_{1k}, \ldots, h_{Kk}, \mathbf{h}_{1,-k}, \ldots, \mathbf{h}_{K,-k}\right) = \sum_{q\in\mathcal{K}\setminus\{k\}} E\left\{|d_q - \left(\mathbf{f}_k^H\mathbf{y}_k + e_{kq}\right)h_{qk}^* - \mathbf{h}_{q,-k}^H\mathbf{z}_{-k}|^2\right\}. \tag{24}$$

Note that node k has access to all signals in (24), except for the desired signals d_q. Nevertheless, due to (4) and (5), all node-specific desired signals d_q are the same up to a scaling, and therefore can be replaced with d_k, which can be compensated for by a similar scaling of the h_{qk} and $\mathbf{h}_{q,-k}$ variables. It then follows that the minimization of $\mathbf{f}_k$ over the sum of terms in (24) is the same as the

[7] The signal z_{qq} is here defined as if node q would send a noisy version of z_q to itself.

[8] This is with a slight abuse of notation, as the z-signals $\mathbf{z}_{-k}$ were originally defined without additional noise. In the sequel, we assume that the presence of this noise is clear from the context, i.e., the signal z_k is assumed to be noise-free as in (12) *before* transmission by node k, but becomes noisy as in (20) after being received by another node $q \neq k$.

minimization over a single term with $q = k$, i.e., minimizing the cost function

$$J_k^f(\mathbf{f}_k, h_k, \mathbf{h}_{-k}) = E\left\{|d_k - \left(\mathbf{f}_k^H\mathbf{y}_k + e_k\right)h_k^* - \mathbf{h}_{-k}^H\mathbf{z}_{-k}|^2\right\}, \tag{25}$$

where, with a slight abuse of notation, e_k represents any noise signal e_{kq} that satisfies the assumptions given in the previous subsection. It can be easily verified that these assumptions assure that the value of J_k^f is the same for any choice of q to define e_{kq} (based on similar arguments to those in Appendix A). Despite the fact that the cost function (25) is non-convex, a closed form expression can be found for its global minimum up to a scaling ambiguity. To see this, we first expand (25) as

$$\begin{aligned} J_k^f = {} & E\{|d_k|^2\} - \mathbf{r}_{y_k d_k}^H h_k\mathbf{f}_k - h_k^*\mathbf{f}_k^H\mathbf{r}_{y_k d_k} \\ & + h_k h_k^*(1+\beta_k)\mathbf{f}_k^H\mathbf{R}_{y_k y_k}\mathbf{f}_k - \mathbf{r}_{z_{-k}d_k}^H\mathbf{h}_{-k} \\ & - \mathbf{h}_{-k}^H\mathbf{r}_{z_{-k}d_k} + h_k^*\mathbf{f}_k^H\mathbf{R}_{y_k z_{-k}}\mathbf{h}_{-k} \\ & + \mathbf{h}_{-k}^H\mathbf{R}_{y_k z_{-k}}^H h_k\mathbf{f}_k + \mathbf{h}_{-k}^H\mathbf{R}_{z_{-k}z_{-k}}\mathbf{h}_{-k}, \end{aligned} \tag{26}$$

where $\mathbf{r}_{y_k d_k} = E\{\mathbf{y}_k d_k^*\}$, $\mathbf{r}_{z_{-k}d_k} = E\{\mathbf{z}_{-k}d_k^*\}$, $\mathbf{R}_{y_k y_k} = E\{\mathbf{y}_k\mathbf{y}_k^H\}$, $\mathbf{R}_{y_k z_{-k}} = E\{\mathbf{y}_k\mathbf{z}_{-k}^H\}$, $\mathbf{R}_{z_{-k}z_{-k}} = E\{\mathbf{z}_{-k}\mathbf{z}_{-k}^H\}$, and we have used the assumed statistical properties of e_k. Then, we define a new variable $\mathbf{p}_k$ given by

$$\mathbf{p}_k = \begin{bmatrix} h_k\mathbf{f}_k \\ \mathbf{h}_{-k} \end{bmatrix}, \tag{27}$$

which allows to rewrite (26) as

$$J_k^f(\mathbf{p}_k) = E\left\{|d_k|^2\right\} + \mathbf{p}_k^H\mathbf{R}_{\beta_k}\mathbf{p}_k - \mathbf{r}_{\tilde{y}_k d_k}^H\mathbf{p}_k - \mathbf{p}_k^H\mathbf{r}_{\tilde{y}_k d_k}^H, \tag{28}$$

where the matrix $\mathbf{R}_{\beta_k}$ is defined as

$$\mathbf{R}_{\beta_k} = \begin{bmatrix} (1+\beta_k)\mathbf{R}_{y_k y_k} & \mathbf{R}_{y_k z_{-k}} \\ \mathbf{R}_{y_k z_{-k}}^H & \mathbf{R}_{z_{-k}z_{-k}} \end{bmatrix}. \tag{29}$$

The cost function of (28) is quadratic with a positive definite matrix $\mathbf{R}_{\beta_k}$, and thus its global minimizer is given by

$$\begin{bmatrix} h_k\mathbf{f}_k \\ \mathbf{h}_{-k} \end{bmatrix} = \left(\mathbf{R}_{\beta_k}\right)^{-1}\mathbf{R}_{\tilde{x}_k\tilde{x}_k}\tilde{\mathbf{c}}_k. \tag{30}$$

The coefficients $\mathbf{h}_{-k}$ are a byproduct of the minimization of J_k^f and they do not need to be computed explicitly.

We can see from (30) that the fusion vector $\mathbf{f}_k$ is only defined up to an unknown scaling h_k. However, any choice of the scaling factor for $\mathbf{f}_k$ will be compensated for by the other nodes when they update their node-specific estimators, i.e., a scaling of $\mathbf{f}_k$, and hence z_{kq}, will be compensated for in node q by an inverse scaling of the corresponding entry in $\mathbf{g}_{-q}$ such that the product remains the same. For this reason, the scaling factor h_k can be absorbed in the fusion vector $\mathbf{f}_k$, which is equivalent to setting $h_{qk} = 1$ in (23). The update rule (30) can then be re-written as

$$\begin{bmatrix} \mathbf{f}_k^{i+1} \\ \mathbf{h}_{-k}^{i+1} \end{bmatrix} = \left(\mathbf{R}_{\beta_k}^i\right)^{-1}\mathbf{R}_{\tilde{x}_k\tilde{x}_k}^i\tilde{\mathbf{c}}_k, \tag{31}$$

where we have introduced the iteration index i since (30) defines the update rule for the fusion vector $\mathbf{f}_k$ in the N-DANSE algorithm.

Remark 1. Note that (31) is similar to the original DANSE update rule given in (16), with the matrix $\mathbf{R}_{\beta_k}^i$ replacing $\mathbf{R}_{\tilde{y}_k\tilde{y}_k}^i$, and that the structure of both matrices is the same except for the scaling of the block $\mathbf{R}_{y_k y_k}$ by $(1+\beta_k)$. In the case of $\beta_k = 0$, $\forall k \in \mathcal{K}$, i.e. without noise in the communication, it is readily seen that (16) and (30) yield the same result, which makes N-DANSE a generalization of the original DANSE algorithm.

Remark 2. We emphasize that, since the update rule (31) only requires a scaling of the block $\mathbf{R}_{y_k y_k}$, the increase in computational complexity in N-DANSE compared to DANSE is minimal, and hence taking into account additive noise in the transmitted signals does not increase the computational complexity of the algorithm, except for this additional pre-scaling of the submatrix $\mathbf{R}_{y_k y_k}$.

Remark 3. The estimation of the correlation matrices required for the N-DANSE algorithm can be done in the same way as described in Section 2.3 under the same conditions given there, e.g.

$$\mathbf{R}_{\tilde{y}_k\tilde{y}_k}^i[t] = \sum_{n=t-L+1}^{t}\tilde{\mathbf{y}}[n]\tilde{\mathbf{y}}[n]^H. \tag{32}$$

Under the assumption of a desired signal with 'on-off' behaviour, $\mathbf{R}_{\tilde{v}_k\tilde{v}_k}^i$ can be computed in the same way during noise-only segments. Then, $\mathbf{R}_{\tilde{x}_k\tilde{x}_k}^i$ can be obtained through, e.g., the substraction $\mathbf{R}_{\tilde{x}_k\tilde{x}_k}^i = \mathbf{R}_{\tilde{y}_k\tilde{y}_k}^i - \mathbf{R}_{\tilde{v}_k\tilde{v}_k}^i$ under the conditions given in Section 2.3, or using other methods referenced therein. Note that the estimation of $\mathbf{R}_{\beta_k}^i$ is not necessary since it can be obtained from $\mathbf{R}_{\tilde{y}_k\tilde{y}_k}^i$ using (29). The parameter β_k can either be computed through a model of the additive noise, like (22) for uniform quantization, or through the use of training sequences and (21).

We provide a summary of the N-DANSE algorithm in Algorithm 1. Note that setting $\beta_k = 0, \forall k \in \mathcal{K}$, yields the original DANSE algorithm as described in Section 3. While so far we have only considered sequential updates, the algorithm can be modified to allow for simultaneous updates, similar to [11]. We briefly study the case of simultaneous updates of the fusion vectors in Section 6.7 with numerical simulations.

Algorithm 1 N-DANSE algorithm.

1: Initialize $\mathbf{f}_q^0$, $\boldsymbol{\psi}_q^0$, $\mathbf{g}_{q,-q}^0$ withrandom entries $\forall q \in \mathcal{K}$.
2: Initialize the iteration index $i \leftarrow 0$ and the updating node index $k \leftarrow 1$.
3: Each node q transmits N samples of its fused signal,

$$z_q^i[iN+n] = \mathbf{f}_q^{iH}\mathbf{y}_q[iN+n],$$

where $n \in \{1, \ldots, N\}$ and the notation $[\cdot]$ indicates a sample index.Note that, in N-DANSE, each node $p \in \mathcal{K}$ receives z_q^i with noise e_{qp} added to it,according to (20), which is then also present in $\tilde{\mathbf{y}}_p$.
4: Each node q updates its estimates of $\mathbf{R}_{\tilde{y}_q\tilde{y}_q}^i$ and$\mathbf{R}_{\tilde{x}_q\tilde{x}_q}^i$ using the samples from $iN+1$ to $iN+N$.
5: Each node q (including the updating node k) computes its node-specific estimator

$$\tilde{\mathbf{w}}_q^{i+1} = \begin{bmatrix} \boldsymbol{\psi}_q^{i+1} \\ \mathbf{g}_{q,-q}^{i+1} \end{bmatrix} = \left(\mathbf{R}_{\tilde{y}_q\tilde{y}_q}^i\right)^{-1}\mathbf{R}_{\tilde{x}_q\tilde{x}_q}^i\tilde{\mathbf{c}}_q \quad \forall q \in \mathcal{K}.$$

6: The updating node k computes its fusion vector

$$\mathbf{f}_k^{i+1} = \left[\mathbf{I}_{M_k}\mathbf{O}_{M_k\times(K-1)}\right]\left(\mathbf{R}_{\beta_k}^i\right)^{-1}\mathbf{R}_{\tilde{x}_k\tilde{x}_k}^i\tilde{\mathbf{c}}_k$$

where $\mathbf{I}_{M_k}$ is the $M_k \times M_k$ identity matrix and $\mathbf{O}_{M_k\times(K-1)}$ is anall-zero matrix of the corresponding dimensions.For the other nodes $q \neq k$, $\mathbf{f}_q^{i+1} = \mathbf{f}_q^i$.
7: Each node $q \in \mathcal{K}$ (including the updating node k) estimates N samples of its desiredsignal d_q:

$$\tilde{d}_q^i[iN+n] = \boldsymbol{\psi}_q^{i+1\,H}\mathbf{y}_q[iN+n] + \mathbf{g}_{q,-q}^{i+1\,H}\mathbf{z}_{-q}^i[iN+n].$$

8: $i \leftarrow i+1$ and $k \leftarrow (k \bmod K)+1$
9: Return to step 3.

5. Convergence analysis

Let us now consider the convergence of the N-DANSE algorithm described in Section 4. Since the original convergence proof of the original DANSE algorithm in [10] cannot be generalized to the case of the N-DANSE algorithm with noisy links, we present a different strategy to prove convergence, which then also contains convergence of the DANSE algorithm without noisy links as a special case. For simplicity, we first consider the case where all nodes have the same desired signal, i.e., $d_k = d\ \forall k \in \mathcal{K}$, and then we show how to extend the proof to the node-specific case where $d_k = a_k d\ \forall k \in \mathcal{K}$, which fits with (4) and (5). Note that the convergence analysis will be an asymptotic analysis, in the sense that it is assumed that the covariance matrices are perfectly estimated. This is only an approximation of the practical situation, where covariance matrices are estimated over finite windows as explained in Section 2.2, and therefore contain estimation errors.

5.1. Convergence for $d_k = d,\ \forall k \in \|$

Before we begin we state the following Lemma which will be necessary later in this section.

Lemma 5.1. *Let $f(\mathbf{x})$ be a continuous function in $\mathcal{C} \to \mathbb{R}$, where $\mathcal{C} \subset \mathbb{C}^n$, and let $\hat{\mathbf{x}}$ be a unique global minimum of f. Then there exists a ball centered in $\hat{\mathbf{x}}$ with radius ε, denoted by $B(\hat{\mathbf{x}}, \varepsilon)$, within which $|\mathbf{x} - \hat{\mathbf{x}}|$ can be made arbitrarily small by making $f(\mathbf{x}) - f(\hat{\mathbf{x}})$ arbitrarily small. Formally, this means that $\forall\, \delta \in (0, \varepsilon)$, $\exists \rho > 0$ such that*

$$\forall \mathbf{x} \in \mathcal{C} : f(\mathbf{x}) - f(\hat{\mathbf{x}}) \leq \rho \Rightarrow |\mathbf{x} - \hat{\mathbf{x}}| \leq \delta\,. \tag{33}$$

The proof for Lemma 5.1 is provided in Appendix B.

Theorem 5.1. *If $d_k = d,\ \forall k \in \mathcal{K}$, the N-DANSE algorithm described in Section 4 converges to a unique point for any initialization of its parameters.*

Proof. An N-DANSE update at node k minimizes the cost function (25), in which d_k is here replaced with d, and where h_k is set to 1 (as explained above in (31)), i.e.,

$$J_k^f(\mathbf{f}_k, \mathbf{h}_{-k}) = E\left\{|d - \left(\mathbf{f}_k^H \mathbf{y}_k + e_k\right) - \mathbf{h}_{-k}^H \mathbf{z}_{-k}|^2\right\}. \tag{34}$$

We assume that all the nodes share a global vector $\mathbf{h}$ which contains auxiliary variables $h_1, \ldots, h_K$(note that these are virtual variables which are not used in the algorithm but only in the proof). When a node k minimizes (34), it will then replace the variables in this global vector $\mathbf{h}$ (except h_k) with the new optimized values. Using (31), the corresponding N-DANSE update is then given by

$$\begin{bmatrix} \mathbf{f}_k^{i+1} \\ \mathbf{h}_{-k}^{i+1} \end{bmatrix} = \begin{bmatrix} (1+\beta_k)\mathbf{R}_{y_k y_k}^i & \mathbf{R}_{y_k z_{-k}}^i \\ \mathbf{R}_{y_k z_{-k}}^{iH} & \mathbf{R}_{z_{-k} z_{-k}}^i \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{r}_{y_k d} \\ \mathbf{r}_{z_{-k} d}^i \end{bmatrix}, \tag{35}$$

and we define the update for h_k as $h_k^{i+1} = h_k^i$.

Let us now introduce some additional notation which we will use to re-write (35) with respect to the network-wide statistics. The $\overline{M}_k \times (k-1)$ matrix $\overline{\mathbf{F}}_k^i$ and the $\underline{M}_k \times (K-k)$ matrix $\underline{\mathbf{F}}_k^i$ are defined as

$$\overline{\mathbf{F}}_k^i = \mathrm{diag}\left(\mathbf{f}_1^i, \ldots, \mathbf{f}_{k-1}^i\right), \tag{36}$$

$$\underline{\mathbf{F}}_k^i = \mathrm{diag}\left(\mathbf{f}_{k+1}^i, \ldots, \mathbf{f}_K^i\right), \tag{37}$$

where $\overline{M}_k = \sum_{n=1}^{k-1} M_n$, $\underline{M}_k = \sum_{n=k+1}^{K} M_n$ and $\mathrm{diag}(\cdot)$ is the operator that generates a block diagonal matrix from its arguments. The $M \times (M_k + K - 1)$ matrix $\mathbf{F}_k^i$ is defined as

$$\mathbf{F}_k^i = \begin{bmatrix} \mathbf{O} & \overline{\mathbf{F}}_k^i & \mathbf{O} \\ \mathbf{I}_{M_k} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \underline{\mathbf{F}}_k^i \end{bmatrix}, \tag{38}$$

where $\mathbf{I}_{M_k}$ is the $M_k \times M_k$ identity matrix, and $\mathbf{O}$ denotes an all-zero matrix of appropriate dimensions. Expression (35) can then be re-written with respect to the network-wide statistics as

$$\mathbf{F}_k^{iH} \mathbf{R}_{yy}^{\beta} \mathbf{F}_k^i \begin{bmatrix} \mathbf{f}_k^{i+1} \\ \mathbf{h}_{-k}^{i+1} \end{bmatrix} = \mathbf{F}_k^{iH} \mathbf{r}_{yd}\,, \tag{39}$$

where $\mathbf{r}_{yd} = E\{\mathbf{y} d^*\}$ and the matrix $\mathbf{R}_{yy}^{\beta}$ is given by

$$\begin{bmatrix} (1+\beta_1)\mathbf{R}_{y_1 y_1} & \cdots & \mathbf{R}_{y_1 y_K} \\ \mathbf{R}_{y_2 y_1} & \ddots & \mathbf{R}_{y_2 y_K} \\ \vdots & \cdots & \vdots \\ \mathbf{R}_{y_K y_1} & \cdots & (1+\beta_K)\mathbf{R}_{y_K y_K} \end{bmatrix}, \tag{40}$$

where $\mathbf{R}_{y_n y_m} = E\{\mathbf{y}_n \mathbf{y}_m^H\}$. Note that $\mathbf{F}_k^{iH} \mathbf{R}_{yy}^{\beta} \mathbf{F}_k^i = \mathbf{R}_{\beta_k}^i$, where $\mathbf{R}_{\beta_k}$ was defined in (29). Equivalently, after an update at node k, (39) can be expressed as

$$\mathbf{F}_k^{iH} \mathbf{R}_{yy}^{\beta} \begin{bmatrix} h_1^{i+1}\mathbf{f}_1^i \\ \vdots \\ h_{k-1}^{i+1}\mathbf{f}_{k-1}^i \\ \mathbf{f}_k^{i+1} \\ h_{k+1}^{i+1}\mathbf{f}_{k+1}^i \\ \vdots \\ h_K^{i+1}\mathbf{f}_K^i \end{bmatrix} = \mathbf{F}_k^{iH} \mathbf{r}_{yd}\,. \tag{41}$$

The first M_k equations of (41) can be written as

$$\left[\mathbf{R}_{y_k y_1}, \ldots, (1+\beta_k)\mathbf{R}_{y_k y_k}, \ldots, \mathbf{R}_{y_k y_K}\right] \begin{bmatrix} h_1^{i+1}\mathbf{f}_1^i \\ \vdots \\ \mathbf{f}_k^{i+1} \\ \vdots \\ h_K^{i+1}\mathbf{f}_K^i \end{bmatrix} = \mathbf{r}_{y_k d}\,. \tag{42}$$

Now let us first assume that we are in a fixed point of the update rule of the fusion vectors, i.e. $\mathbf{f}_k^{i+1} = \mathbf{f}_k^i = \mathbf{f}_k^\star, \forall k \in \mathcal{K}$. Note that in a fixed point all the entries of the global auxiliary vector $\mathbf{h}$ must be identical to one. This can be explained as follows. We reiterate that each entry of $\mathbf{z}_{-k}$ in (34) is given by $z_q = \mathbf{f}_q^H \mathbf{y}_q + e_q, \forall q \neq k$. By sequentially updating (34) for each node $k \in \mathcal{K}$, and assuming that the fusion vectors $\mathbf{f}_k$ do not change (since we are in a fixed point), all coefficients h_k in $\mathbf{h}$ must by definition be equal to 1. This is a direct consequence of the assumption that $d_k = d, \forall k \in \mathcal{K}$. Hence the equations in (42) can be stacked $\forall k \in \mathcal{K}$ to obtain

$$\mathbf{R}_{yy}^{\beta} \begin{bmatrix} \mathbf{f}_1^\star \\ \vdots \\ \mathbf{f}_K^\star \end{bmatrix} = \mathbf{r}_{yd}\,, \tag{43}$$

which is a linear system of equations with a unique solution if $\mathbf{R}_{yy}^{\beta}$ is full rank. This assumption is satisfied, for any value $\beta_k \geq 0,\ \forall k \in \mathcal{K}$, due to the assumed full rank of $\mathbf{R}_{yy}$ in Section 2. This means that the fixed point is unique.

Our next step is to consider the opposite case, when the algorithm is not in a fixed point. In this case, $\mathbf{f}_k^{i+1} \neq \mathbf{f}_k^i$, or equivalently

$$\mathbf{f}_k^{i+1} = \mathbf{f}_k^i + \boldsymbol{\phi}_k^i\,, \tag{44}$$

for non-zero $\boldsymbol{\phi}_k^i$. If we replace $\mathbf{f}_k^{i+1}$ in (42) with its version in the current iteration i, $\mathbf{f}_k^i$, we need to add an error term, i.e.,

$$\left[\mathbf{R}_{y_k y_1}, \ldots, (1+\beta_k)\mathbf{R}_{y_k y_k}, \ldots, \mathbf{R}_{y_k y_K}\right] \begin{bmatrix} \mathbf{f}_1^i \\ \vdots \\ \mathbf{f}_k^i \\ \vdots \\ \mathbf{f}_K^i \end{bmatrix} = \mathbf{r}_{y_k d} + \boldsymbol{\epsilon}_k^i , \tag{45}$$

where the norm of $\boldsymbol{\epsilon}_k^i$ will vanish if and only if the norm of $\boldsymbol{\phi}_k^i$ in (44) will vanish as $i \to \infty$. Note that the error term ϵ_k^i also compensates for the fact that the coefficients in $\mathbf{h}^i$ have been replaced with ones, although they are not necessarily equal to one when the fixed point has not been reached. Stacking the equations in (45) gives

$$\mathbf{R}_{yy}^{\beta} \begin{bmatrix} \mathbf{f}_1^i \\ \vdots \\ \mathbf{f}_K^i \end{bmatrix} = \mathbf{r}_{yd} + \boldsymbol{\epsilon}^i , \tag{46}$$

or equivalently

$$\begin{bmatrix} \mathbf{f}_1^i \\ \vdots \\ \mathbf{f}_K^i \end{bmatrix} = \left(\mathbf{R}_{yy}^{\beta}\right)^{-1} \left(\mathbf{r}_{yd} + \boldsymbol{\epsilon}^i\right) . \tag{47}$$

Note that any fusion vector update given by (35), which optimizes (34), can be interpreted as one step of an alternating optimization (AO) [38] procedure on the cost function

$$J^f(\mathbf{f}_1, \ldots, \mathbf{f}_K, h_1, \ldots, h_K) = E\left\{ \left| d - \sum_{k \in \mathcal{K}} h_k^* \left(\mathbf{f}_k^H \mathbf{y}_k + e_k\right) \right|^2 \right\} , \tag{48}$$

in which in each iteration we select an index k for which we optimize over $\mathbf{f}_k$ and $h_q, \forall q \in \mathcal{K}$ (with the constraint $h_k = 1$ by convention), while all other variables $\mathbf{f}_q$ with $q \neq k$ remain fixed. This will result in a monotonic decrease in the values of J^f [39]. Since J^f is bounded from below, the result must converge to a finite value, and thus

$$\lim_{i \to \infty} \left(J^f\left(\mathbf{f}^i, \mathbf{h}^i\right) - J^f\left(\mathbf{f}^{i+1}, \mathbf{h}^{i+1}\right) \right) = 0 , \tag{49}$$

where $\mathbf{f} = [\mathbf{f}_1^T, \ldots, \mathbf{f}_K^T]^T$. From (49), it also holds that

$$\lim_{i \to \infty} \left(J_k^f\left(\mathbf{f}_k^i, \mathbf{h}_{-k}^i\right) - J_k^f\left(\mathbf{f}_k^{i+1}, \mathbf{h}_{-k}^{i+1}\right) \right) = 0 . \tag{50}$$

Note that $\mathbf{f}_k^{i+1}$ is the result of a *global* optimization process of the function J_k^f in (34), which has a unique minimum. Together with (50), this fact allows us to use Lemma 5.1 on the function J_k^f, which implies that the distance between fusion vectors in consecutive updates must necessarily vanish in the limit, i.e.,

$$\lim_{i \to \infty} \left\| \mathbf{f}_k^{i+1} - \mathbf{f}_k^i \right\| \to 0 , \forall k \in \mathcal{K} . \tag{51}$$

From (51) we conclude that $\boldsymbol{\phi}_k^i$ in (44) will vanish, and as a result also $\boldsymbol{\epsilon}_k^i$ in (45) will vanish as $i \to \infty$. Thus we arrive to the following statement

$$\lim_{i \to \infty} \begin{bmatrix} \mathbf{f}_1^i \\ \vdots \\ \mathbf{f}_K^i \end{bmatrix} = \left(\mathbf{R}_{yy}^{\beta}\right)^{-1} \mathbf{r}_{yd} . \tag{52}$$

This shows that the fusion vectors $\mathbf{f}_k^i$ converge to a unique point. □

5.2. Convergence for $d_k = a_k d, \ \forall k \in \|$

Theorem 5.2. *If $d_k = a_k d, \ \forall k \in \mathcal{K}$, the N-DANSE algorithm described in Section 4 converges to a unique point for any initialization of its parameters.*

Proof. An update of $\mathbf{f}_k$ at node k based on (25) now depends on the desired signal d_k of the updating node, leading to the update (31), which can then be rewritten as

$$\begin{bmatrix} \mathbf{f}_k^{i+1} \\ \mathbf{h}_{k,-k}^{i+1} \end{bmatrix} = \begin{bmatrix} (1+\beta_k)\mathbf{R}_{y_k y_k}^i & \mathbf{R}_{y_k z_{-k}}^i \\ \mathbf{R}_{y_k z_{-k}}^{iH} & \mathbf{R}_{z_{-k} z_{-k}}^i \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{r}_{y_k d_k} \\ \mathbf{r}_{z_{-k} d_k}^i \end{bmatrix} = \begin{bmatrix} (1+\beta_k)\mathbf{R}_{y_k y_k}^i & \mathbf{R}_{y_k z_{-k}}^i \\ \mathbf{R}_{y_k z_{-k}}^{iH} & \mathbf{R}_{z_{-k} z_{-k}}^i \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{r}_{y_k d} \\ \mathbf{r}_{z_{-k} d}^i \end{bmatrix} a_k , \tag{53}$$

where we used $d_k = a_k d$ in the last step. By comparing (53) with (35), we see that the node-specific case (in (53)) results in the same fusion vector $\mathbf{f}_k$ as in the case where the desired signal is the same at each node (in (35)), up to an unknown scaling. However, this scaling has no impact on the algorithm dynamics and future updates of other fusion vectors, as the scaling will be compensated for at each node $k \in \mathcal{K}$ by the corresponding coefficient in $\mathbf{g}_{k,-k}$, and hence will also not affect the update of $\mathbf{f}_q$ at the next updating node q. Thus, up to a scaling factor a_k, the same sequence of fusion vectors $\mathbf{f}_k^i$ will be generated as for the case where $d_k = d, \forall k \in \mathcal{K}$. As a result, the convergence result in (52) also holds for the update (53), up to a scaling a_k for every $\mathbf{f}_k$, i.e.,

$$\lim_{i \to \infty} \begin{bmatrix} \frac{1}{a_1}\mathbf{f}_1^i \\ \vdots \\ \frac{1}{a_K}\mathbf{f}_K^i \end{bmatrix} = \left(\mathbf{R}_{yy}^{\beta}\right)^{-1} \mathbf{r}_{yd} . \tag{54}$$

□

Corollary 5.1. *The convergence of the DANSE algorithm without noisy links as presented in [10] follows from Theorem 5.2 by combining its proof with Remark I from Section 4.2.*

6. Simulation results

In this section we analyze the behaviour of the N-DANSE algorithm through numerical simulations.

6.1. Data generation

We consider scenarios in a two-dimensional 5×5 m area where the positions of nodes and both desired and undesired sources are randomly generated such that each coordinate follows a uniform distribution in [0, 5]. The minimum distance between any pair of positions is 0.5 m. In each scenario there are three noise sources and one desired source present. The network in any scenario consists of K nodes with $M_k = 3$ sensors each, where the number of nodes K will be specified later for each simulation. The three sensors are placed parallel to the y-axis, spaced with a constant distance of $l = 10$ cm. All source signals consist of 10^5 complex samples drawn from a uniform distribution with zero mean and unit variance, i.e., the real and imaginary parts are generated independently from a uniform distribution in $(-\frac{\sqrt{6}}{2}, \frac{\sqrt{6}}{2})$. The entries of the steering vectors $\mathbf{a}_k$ are generated according to a narrow-band propagation model such that

$$\mathbf{a}_k = \frac{1}{\sqrt{r_k}} \left[1, e^{-i2\pi \frac{l\cos(\theta_k)}{\lambda}}, \ldots, e^{-i2\pi (M_k - 1) \frac{l\cos(\theta_k)}{\lambda}}\right]^T , \tag{55}$$

where r_k is the distance between the source and the first sensor of the k-th node, θ_k is the angle between the first sensor of the k-th node and the source, and $\lambda = \frac{c}{f}$ is the wavelength corresponding

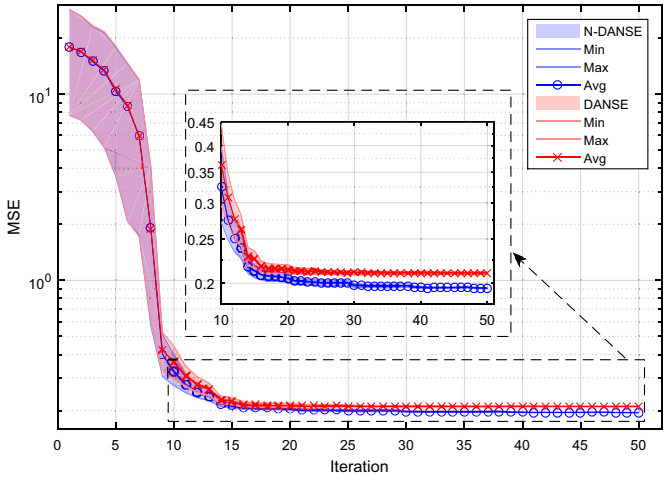


Fig. 2. Convergence of the N-DANSE and DANSE algorithms in a noisy links scenario with $K = 9$ nodes for 100 different initializations. The MSE is shown in a logarithmic scale. The central graph of the figure is a magnified version of the area bounded by the lowermost rectangle.

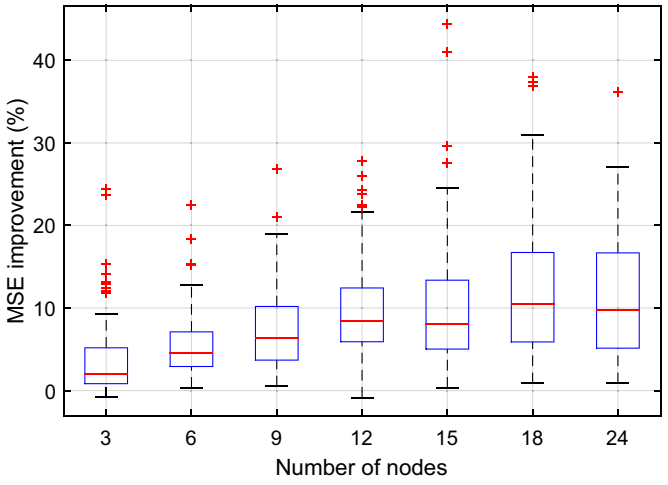


Fig. 3. Box plots of the MSE improvement μ_{MSE} of N-DANSE over DANSE for networks with a number of nodes between $K = 3$ and $K = 24$ and 100 scenarios per each different K. The red crosses indicate outliers. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

to $f = 1$ kHz for a propagation speed of $c = 331$ m/s. Likewise, for each noise source a steering vector is generated in a similar way to (55). The node sensor signals $\mathbf{y}_k$ are mixtures of desired and noise source signals defined by the corresponding steering vectors, plus independent zero-mean white Gaussian noise whose power is 1% of the power of the corresponding channel of $\mathbf{y}_k$, which represents local sensor noise such as thermal noise. The desired signal d_k for node k is the desired source signal component $x_{k\tilde{m}}$ at the channel $\tilde{m}$ with the highest signal-to-noise ratio (SNR). The additional noise in the z-signals is drawn from a zero-mean uniform distribution with second order moment defined by β. The parameter β can be different in each node, and different values are simulated as will be explained in the sequel.

The network-wide second order statistics $\mathbf{R}_{yy}$ and $\mathbf{r}_{yd_k}$ are estimated through sample averaging using all 10^5 samples of the sensor signals. It is noted that, in practical implementations, nodes will have to estimate the necessary second order statistics on the fly based on sample averaging from finite length segments of the sensor signals, as the full length signals are rarely available.

6.2. Validation of convergence

Our first point of interest is to study the convergence of the N-DANSE algorithm with the goal to validate the results presented in Section 5. To this end we simulate 100 different initializations for N-DANSE and DANSE, choosing the same initialization for both algorithms each time, in a scenario generated according to Section 6.1 with $K = 9$ nodes and $\beta_1, \ldots, \beta_9$ generated at random from a uniform distribution in (0, 1). In Fig. 2 we show the resulting MSE for each initialization at each iteration of N-DANSE and DANSE in the corresponding shaded area, whose limits mark the maximum and minimum MSE achieved at each iteration. The marked line is the average MSE of the respective algorithm for all initializations. We can observe that both the N-DANSE and DANSE algorithms converge to a unique point for all initializations, which for N-DANSE is expected from the theoretical results presented in Section 5.

6.3. Performance in different scenarios

We are now interested in comparing the performance of the N-DANSE algorithm and the original DANSE algorithm in noisy links scenarios. For this comparison we analyze 100 different scenarios where the position of nodes and sources are randomly generated in each scenario, as explained in Section 6.1. Each transmitted signal z_{kq} has a corresponding additive noise defined by β_k, which is generated at random in each scenario from a uniform distribution in (0, 1). To measure the performance of the algorithms we consider the MSE improvement across all nodes, defined as

$$\mu_{\text{MSE}} = 100 \cdot \left(\frac{\sum_{k=1}^{K} \text{MSE}_{k,\text{DANSE}}}{\sum_{k=1}^{K} \text{MSE}_{k,\text{N-DANSE}}} - 1 \right), \tag{56}$$

where the MSE is considered at the convergence point. This figure of merit shows the total MSE improvement, expressed as a percentage of the total MSE achieved at convergence by N-DANSE, that can be obtained by using the N-DANSE algorithm instead of the original DANSE algorithm in the case of noisy links for each specific scenario (i.e., the same node and source locations, β_k, etc). Fig. 3 shows box plots for the MSE improvement μ_{MSE} after convergence for networks of different numbers of nodes K ranging from 3 to 24, where 100 different scenarios are generated for each specific K. It can be observed that the MSE improvement μ_{MSE} of N-DANSE over DANSE is almost always positive, showing the superiority of the former over the latter. The MSE improvement of N-DANSE over DANSE is observed to increase for larger networks, which can be intuitively explained since, in larger networks, there are more nodes doing a better optimization of the MSE with respect to the additive noise when using the N-DANSE algorithm, while when using the original DANSE algorithm more nodes are doing an imperfect optimization with respect to the additive noise, hence the increasing MSE improvement with increased number of nodes.

6.4. Influence of the power of the additive noises

We turn our attention here to the effect of the ratio of additive noise power to fused signal power β_k on the achieved MSE after convergence. In order to illustrate this effect, we analyze 100 scenarios with $K = 12$ nodes, where the positions of nodes and sources are generated at random as described in Section 6.1. To study the influence of changing β_k, in each scenario we generate each β_k at random from a uniform distribution in $(\bar{\beta}, 1.5\bar{\beta})$, where $\bar{\beta}$ is then swept from 0 to 4.

Fig. 4 shows the resulting MSE improvement μ_{MSE} after convergence as a function of $\bar{\beta}$. We observe that the MSE improve-

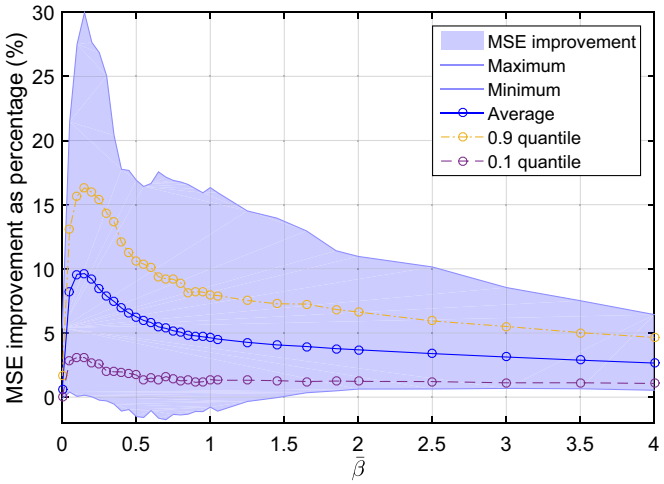


Fig. 4. Influence of the ratio of additive noise power to fused signal power β_k in the MSE improvement of N-DANSE over DANSE. The data corresponds to 100 scenarios with $K = 12$ nodes.

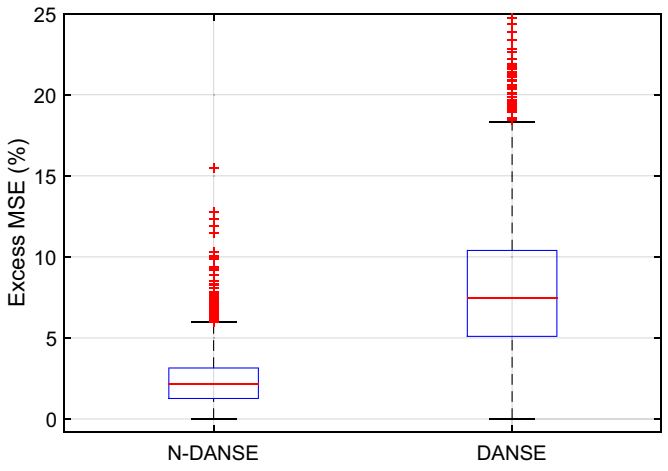


Fig. 5. Box plots of the excess MSE between (N-)DANSE and the centralized clairvoyant scheme, as percentage of the centralized clairvoyant MSE. The excess MSE of N-DANSE appears on the left and the excess MSE of DANSE on the right. The red crosses indicate outliers. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

ment of the N-DANSE algorithm over the original DANSE algorithm starts growing as $\bar{\beta}$ increases, then peaks around $\bar{\beta} = 0.2$ (in this set of scenarios), and then it starts to decay. This is explained by the fact that for moderate values of additive noise power the fusion vector obtained by N-DANSE provides a more useful signal z_k for the estimation problems of the rest of the nodes than the z_k signal obtained by DANSE (note that $\beta_k = 1$ corresponds to the additive noise e_k being as powerful as the fused signal $\mathbf{f}_k^H \mathbf{y}_k$). As the additive noises e_k grow more powerful, the signals z_k become less helpful for the rest of the nodes, hence why the MSE improvement between N-DANSE and DANSE decays, although the N-DANSE algorithm remains generally superior to the original DANSE algorithm. Note that the largest improvement happens for $\bar{\beta} \ll 1$, which is the most practical range for quantization and communication noise.

6.5. Comparison with the centralized clairvoyant scheme.

We now focus on the difference between the N-DANSE and DANSE algorithms and the optimal fusion vectors for the case of noisy links, which could be obtained if nodes would have access to all local sensor signals $\mathbf{y}_k$ free of noise and to all additive noises e_k, $\forall k \in \mathcal{K}$. Since the optimal scheme requires each node to have access to all information available in the network, we refer to it as the centralized clairvoyant scheme. The cost function to be minimized in this centralized clairvoyant scheme is given by the sum of the MSE at each node, where the k-th node uses its local sensor signal free of additive noise $\mathbf{y}_k$ and the noisy transmitted signals from the rest of nodes $\mathbf{z}_{-k}$. In Appendix C we provide the mathematical form of the centralized clairvoyant cost function in (63), and we derive expressions for its minimization using an alternating optimization scheme, since a closed form solution for its minimizer is not possible. In Fig. 5 we show box plots of the excess MSE, i.e., the difference between the MSE achieved by either N-DANSE or DANSE and the optimal MSE achieved by the centralized clairvoyant scheme, expressed as a percentage of the optimal MSE value, which is given by

$$\mu_{\text{Clairvoy}} = 100 \cdot \left(\frac{\sum_{k=1}^{K} \text{MSE}_{k,(\text{N-})\text{DANSE}}}{\sum_{k=1}^{K} \text{MSE}_{k,\text{Clairvoyant}}} - 1 \right). \tag{57}$$

This provides a measure of how close the corresponding algorithm gets to the clairvoyant scheme. The data is obtained from the same 100 scenarios used in Section 6.4. We observe that the N-DANSE algorithm is consistently closer to the optimal centralized clairvoyant scheme than the DANSE algorithm, which shows that it generally provides superior performance, i.e. lower MSE, than the original DANSE algorithm in noisy link scenarios. A Wilcoxon signed-rank test shows that the MSE improvement between N-DANSE and DANSE is statistically significant (p-value $= 6.17 \cdot 10^{-7} < 0.05$).

6.6. Influence of the bit depth in uniform quantization of the z-signals

A particularly interesting case for the source of additive errors in the transmitted z-signals is uniform quantization, as mentioned in Section 4.1. To study this case, we consider that the z-signals are subject to uniform quantization, prior to transmission, with a bit depth randomly generated between 1 and $\bar{b}$ bits, where we simulate for $\bar{b}$ between 4 and 16. We generate 100 random scenarios as described in Section 6.1 with $K = 12$ nodes. Since the signals are complex-valued, we quantize their real and imaginary parts independently, but with the same bit depth for both. The dynamic ranges of the real and imaginary parts of z_k are respectively denoted by $A_{k,r}$ and $A_{k,i}$. An estimate of the quantization noise power is needed in the N-DANSE algorithm, and it is here calculated from the statistical model for uniform quantization given in (22). As explained under Assumption 21, we choose a scaling factor for each signal z_k such that

$$\beta_k = \frac{\Delta_{b_k}^2}{12\, E\left\{ |z_k|^2 \right\}} = \frac{1}{12\, 2^{2b_k}} \frac{A_{k,r}^2 + A_{k,i}^2}{E\left\{ |z_k|^2 \right\}}. \tag{58}$$

Fig. 6 shows the MSE improvement μ_{MSE} between N-DANSE and DANSE (as defined in (56) for each $\bar{b}$. It can be seen that N-DANSE is consistently superior to DANSE for scenarios with uniform quantization where β_k is estimated from (58) for a wide range of bit depths.

6.7. N-DANSE with simultaneous updates

Up to this point we have focused on sequential updates of the fusion vectors. However, the original DANSE algorithm has also been shown to converge to the optimal solution with simultaneous and asynchronous updates if a proper relaxation of the updates is applied. Rather than performing a hard update, this relaxation consists of making a convex combination between the current and the

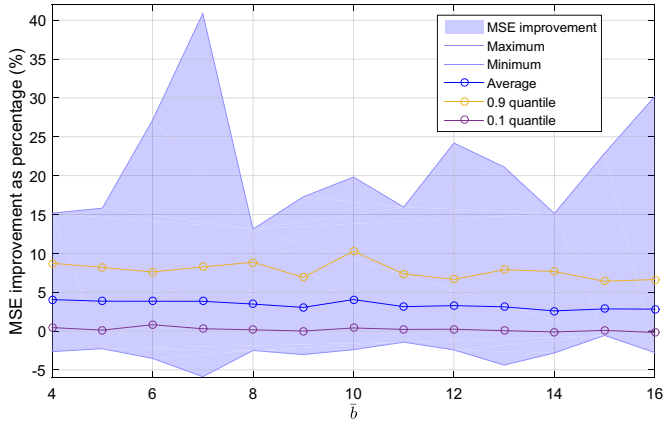


Fig. 6. Influence of the bit depth in the MSE improvement of N-DANSE over DANSE. The data corresponds to 100 scenarios with K = 12 nodes, where the bit depth is randomly chosen from $\{1, \ldots, \bar{b}\}$.

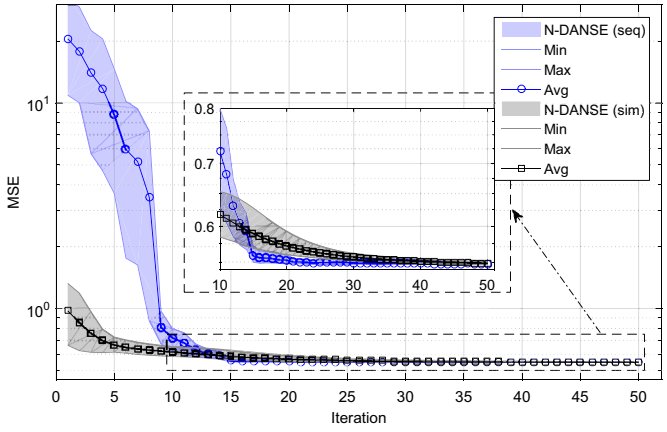


Fig. 7. Convergence of the N-DANSE algorithm with simultaneous and sequential updates in a noisy links scenario with $K = 9$ nodes for 100 different initializations. The MSE is shown in a logarithmic scale. The central graph of the figure is a magnified version of the area bounded by the lowermost rectangle.

newly computed fusion vector, which adds some memory to the updating [11]. The same relaxation strategy can be applied to the N-DANSE algorithm with simultaneous (or asynchronous) updates, i.e.

$$\mathbf{f}_k^{i+1} = (1-\lambda)\mathbf{f}_k^i + \lambda \mathbf{f}_k^{\text{new}}, \tag{59}$$

where $0 < \lambda < 1$ and

$$\mathbf{f}_k^{\text{new}} = \left[\mathbf{I}_{M_k} \mathbf{O}_{M_k \times (K-1)}\right] \left(\mathbf{R}_{\beta_k}^i\right)^{-1} \mathbf{R}_{\tilde{\mathbf{x}}_k \tilde{\mathbf{x}}_k}^i \tilde{\mathbf{c}}_k . \tag{60}$$

For more technical details we refer the reader to [11].

In order to show that this strategy is also valid for the N-DANSE algorithm, we have simulated 100 different initializations of the N-DANSE algorithm with both sequential and simultaneous updates, as given by (59), under the same conditions as Section 6.2. The value of λ was chosen to be $\lambda = 0.4$. In Fig. 7 we show the resulting MSE at each iteration in the corresponding shaded area, in the same way as in Fig. 2. We can observe that both algorithms converge to the same MSE, and since N-DANSE with sequential updates has been shown to converge to a lower MSE than DANSE, the same conclusion can be reached for simultaneous updates.

6.8. Other topologies

As we have noted before, the DANSE algorithm has been extended to tree [12] and generic topologies [14]. In these topologies, nodes can only communicate with a subset of other nodes

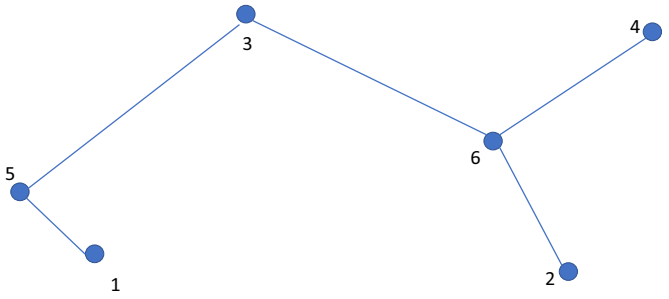


Fig. 8. Tree topology with noisy links with $K = 6$ nodes for Fig. 9.

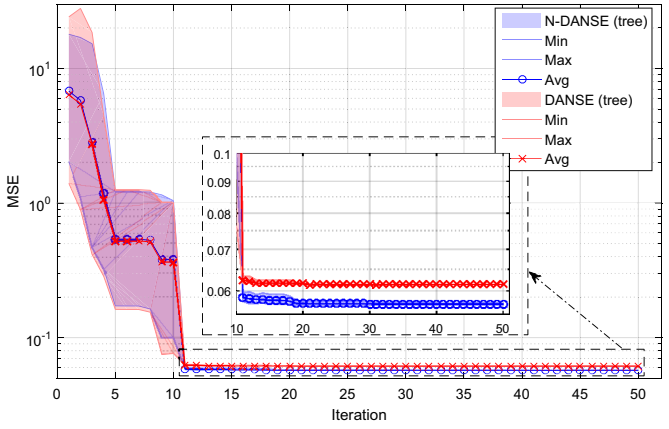


Fig. 9. Convergence of the N-DANSE and DANSE algorithms in a noisy links scenario with a tree topology with $K = 6$ nodes for 100 different initializations. The MSE is shown in a logarithmic scale. The central graph of the figure is a magnified version of the area bounded by the lowermost rectangle.

(their neighbours). The extensions to tree (T-DANSE) and generic (TI-DANSE) topologies design the signal flow such that nodes only need to communicate with their neighbours and that the amount of data each node must transmit is independent of the number of neighbours. The extension of these algorithms to the noisy links scenarios follows the same reasoning we have explained in Section 4. While a detailed description is beyond the scope of this paper (for more details we refer the reader to [12,14]), we show the validity of N-DANSE in a tree topology through a simulation with 100 different initializations of the N-DANSE and DANSE algorithms. The tree topology with $K = 6$ nodes is shown in Fig. 8, and the scenario is generated under the same conditions as Section 6.2 for sequential updates in a fully connected network.

In Fig. 9 we show the resulting MSE at each iteration in the corresponding shaded area, in the same way as in Fig. 2. We can again observe that both algorithms converge to a unique point and that N-DANSE converges to a lower MSE than DANSE.

7. Conclusions

In this paper we have tackled the problem of distributed signal estimation in a WSN in the presence of noisy links, i.e., with additive noise in the signals transmitted between the nodes, e.g., due to quantization or due to noise in the communication channel, within the framework of the distributed adaptive signal estimation (DANSE) algorithm. We have provided a modification of the fusion rules which incorporates knowledge of the presence of noisy links at almost no increase in computational complexity, resulting in a modified version of the DANSE algorithm, referred to as noisy-DANSE or N-DANSE algorithm. Additionally, we have provided a proof of the convergence of the N-DANSE algorithm to a unique point using a different strategy from the proof of original DANSE,

which cannot be straightforwardly generalized to our problem. To conclude, we have used numerical simulations to validate the convergence of the N-DANSE algorithm and to demonstrate its superiority over the original DANSE algorithm in the case of noisy links. In particular the MSE achieved after convergence by the N-DANSE algorithm is closer to the clairvoyant optimal MSE than the MSE achieved by the original DANSE algorithm, and the improvement in MSE provided by the N-DANSE algorithm increases for networks with more nodes.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Equivalence of cost functions (23) and (24)

We start by repeating the cost function (23) for convenience,

$$J_k^s\big(\mathbf{f}_k, h_{1k}, \ldots, h_{Kk}, \mathbf{h}_{1,-k}, \ldots, \mathbf{h}_{K,-k}\big) = \sum_{q\in\mathcal{K}\setminus\{k\}} E\Big\{\big|d_q - \big(\mathbf{f}_k^H\mathbf{y}_k + e_{kq}\big)h_{qk}^* - \mathbf{h}_{q,-k}^H\mathbf{z}_{-k,q}\big|^2\Big\}, \tag{61}$$

where $\mathbf{z}_{-k,q} = [z_{1q}, \ldots, z_{k-1,q}, z_{k+1,q}, \ldots, z_{Kq}]^T$ contains the z-signals received by node q with the exception of z_{kq}. The q-th term of (61) can be expanded as

$$\begin{aligned} E\{|d_q|^2\} &- \mathbf{r}_{y_k d_q}^H h_{qk}\mathbf{f}_k - h_{qk}^*\mathbf{f}_k^H\mathbf{r}_{y_k d_q} \\ &+ h_{qk}h_{qk}^*(1+\beta_k)\mathbf{f}_k^H\mathbf{R}_{y_k y_k}\mathbf{f}_k - \mathbf{r}_{z_{-k,q}d_q}^H\mathbf{h}_{q,-k} \\ &- \mathbf{h}_{q,-k}^H\mathbf{r}_{z_{-k}d_q} + h_{qk}^*\mathbf{f}_k^H\mathbf{R}_{y_k z_{-k,q}}\mathbf{h}_{q,-k} \\ &+ \mathbf{h}_{q,-k}^H\mathbf{R}_{y_k z_{-k,q}}^H h_{qk}\mathbf{f}_k + \mathbf{h}_{q,-k}^H\mathbf{R}_{z_{-k,q}z_{-k,q}}\mathbf{h}_{q,-k}, \end{aligned} \tag{62}$$

where $\mathbf{r}_{y_k d_q} = E\{\mathbf{y}_k d_q^*\}$, $\mathbf{R}_{y_k y_k} = E\{\mathbf{y}_k\mathbf{y}_k^H\}$, $\mathbf{r}_{z_{-k,q}d_q} = E\{\mathbf{z}_{-k,q}d_q^*\}$ $\mathbf{R}_{y_k z_{-k,q}} = E\{\mathbf{y}_k\mathbf{z}_{-k,q}^H\}$, $\mathbf{R}_{z_{-k,q}z_{-k,q}} = E\{\mathbf{z}_{-k,q}\mathbf{z}_{-k,q}^H\}$, and we have used the statistical properties of e_{kq} assumed in Section 4.1. It can be readily seen that (62) depends on $\mathbf{y}_k$ and $\mathbf{z}_{-k,q}$ only through their second order statistics.

Note that in $\mathbf{r}_{z_{-k,q}d_q}$ and $\mathbf{R}_{y_k z_{-k,q}}$ the variable $\mathbf{z}_{-k,q}$ can be replaced by $\mathbf{z}_{-k,k} = [z_{1k}, \ldots, z_{k-1,k}, z_{k+1,k}, \ldots, z_{Kk}]^T$, since $E\{\mathbf{y}_k z_{pq}^*\} = E\{\mathbf{y}_k z_{pk}^*\}, \forall p, q, k \in \mathcal{K}$ due to the assumption that $\mathbf{y}_k$ and e_{kq} are uncorrelated for all $k, q \in \mathcal{K}$, which also implies that d_q and e_{kq} are uncorrelated due to (3)-(5). Similarly, $\mathbf{z}_{-k,q}$ can be replaced by $\mathbf{z}_{-k,k}$ in $\mathbf{R}_{z_{-k,q}z_{-k,q}}$, since $E\{z_{pq}z_{rq}^*\} = E\{z_{pk}z_{rk}^*\}, \forall p, r, q, k \in \mathcal{K}$ due to (21). As a result, the variable $\mathbf{z}_{-k,q}$ can be replaced by $\mathbf{z}_{-k,k}$ in each term of (61) without affecting its value, showing that (23) and (24) are equivalent.

Appendix B. Proof of Lemma 5.1

Proof. We show how a ball $B(\hat{\mathbf{x}}, \varepsilon)$ can be constructed for which (33) holds. We denote by $f_{\min 2}$ the smallest non-global minimum of $f(\mathbf{x})$ in $\mathcal{C}$, if it exists. Let us choose ε such that the maximum of $f(\mathbf{x})$ in $B(\hat{\mathbf{x}}, \varepsilon)$ is lower than $f_{\min 2}$, i.e., $\max_{\mathbf{x}\in B(\hat{\mathbf{x}},\varepsilon)\cap\mathcal{C}} f(\mathbf{x}) = f_\varepsilon < f_{\min 2}$. Then, for any $\delta \in (0, \varepsilon)$, we can choose $\rho = \max_{\mathbf{x}\in B(\hat{\mathbf{x}},\delta)\cap\mathcal{C}} (f(\mathbf{x}) - f(\hat{\mathbf{x}}))$. Since $\hat{\mathbf{x}}$ is unique and f is continuous, all $\mathbf{x} \in \mathcal{C}$ that satisfy $f(\mathbf{x}) - f(\hat{\mathbf{x}}) \le \rho$ are in the closed ball of radius δ centered in $\hat{\mathbf{x}}$, which proves (33).

If $f_{\min 2}$ does not exist, ε can be freely chosen, and the same reasoning presented above can be applied. □

Appendix C. Minimization of the centralized clairvoyant cost function

Here we provide the cost function of the centralized clairvoyant scheme whose minimization provides the optimal fusion vectors and local linear MMSE estimators for each node in the case of noisy links, in the same framework of Section 4 and used as a benchmark in Section 6.5. We also provide a detailed derivation of the steps to find its minimizer through the alternating optimization method [38]. Note that a practical implementation is not possible in a real WSN due to the assumption that all nodes have access to the local sensor signals of other nodes free of additive noises e_k.

The optimal set of fusion vectors and linear MMSE estimators can be found by minimizing the sum of the MSE of each node. Mathematically this cost function is given by

$$J_c(\mathbf{w}^c, \mathbf{f}^c, \mathbf{h}^c) = \sum_{k\in\mathcal{K}} E\left\{\left|d_k - \mathbf{w}_k^{cH}\mathbf{y}_k - \sum_{j\neq k}\big(\mathbf{f}_j^{cH}\mathbf{y}_j + e_j\big)h_j^{(k)*}\right|^2\right\}, \tag{63}$$

where $\mathbf{w}_k^c$ is the estimator applied to the sensor signals of the k-th node, $\mathbf{f}_j^c$ is the vector used to fuse the sensor signals of the j-th node and $h_j^{(k)}$ is the j-th entry of the estimator applied to the k-th estimation problem. The variables $\mathbf{w}^c$, $\mathbf{f}^c$ and $\mathbf{h}^c$ denote the corresponding stacked versions of the mentioned variables, i.e., $\mathbf{w}^c = [\mathbf{w}_1^{cT}, \ldots, \mathbf{w}_K^{cT}]^T$, $\mathbf{f}^c = [\mathbf{f}_1^{cT}, \ldots, \mathbf{f}_K^{cT}]^T$ and $\mathbf{h}^c = [h_1^{(1)}, h_2^{(1)}, \ldots, h_K^{(1)}, \ldots, h_1^{(K)}, \ldots, h_K^{(K)}]^T$. Note that the variables $h_k^{(k)}$ are never used since the fused signal from the k-th node is not used for the k-th estimation problem, so they can be ignored.

A closed form solution of the minimizer of (63) is not possible, and the problem is non-convex. However, note that fixing either the fusion vectors $\mathbf{f}^c$ or the estimator variables $\mathbf{w}^c$, $\mathbf{h}^c$ results in a linear MMSE problem, which is convex, so we choose an alternating optimization scheme [38] to perform the minimization. For the sake of clarity, we will refer to the cost function simply as J_c throughout the rest of the appendix. For the same reason we will omit the iteration index i, assuming that the derivation presented here corresponds to a single iteration.

First we will expand the expression of J_c given in (63), which yields the equivalent expression

$$\begin{aligned} J_c = \sum_{k\in\mathcal{K}}\Big[&E\{|d_k|^2\} - \mathbf{r}_{y_k d_k}^H\mathbf{w}_k^c - \mathbf{w}_k^{cH}\mathbf{r}_{y_k d_k} \\ &- \sum_{j\neq k} h_j^{(k)*}\mathbf{f}_j^{cH}\mathbf{r}_{y_j d_k} - \sum_{j\neq k}\mathbf{r}_{y_j d_k}^H\mathbf{f}_j^c h_j^{(k)} + \mathbf{w}_k^{cH}\mathbf{R}_{y_k y_k}\mathbf{w}_k^c \\ &+ \sum_{j\neq k}\mathbf{w}_k^{cH}\mathbf{R}_{y_k y_j}\mathbf{f}_j^c h_j^{(k)} + \sum_{j\neq k} h_j^{(k)*}\mathbf{f}_j^{cH}\mathbf{R}_{y_k y_j}^H\mathbf{w}_k^c \\ &+ \sum_{j\neq k}\sum_{m\neq k}\mathbf{f}_j^{cH}\mathbf{R}_{y_j y_m}\mathbf{f}_m^c h_j^{(k)*}h_m^{(k)} \\ &+ \sum_{j\neq k} h_j^{(k)}h_j^{(k)*}\beta_j\mathbf{f}_j^{cH}\mathbf{R}_{y_j y_j}\mathbf{f}_j^c\Big], \end{aligned} \tag{64}$$

where $\mathbf{r}_{y_k d_k} = E\{\mathbf{y}_k d_k^*\}$, $\mathbf{R}_{y_j y_m} = E\{\mathbf{y}_k\mathbf{y}_m^H\}$, and we have used the statistical properties of e_k, $\forall k \in \mathcal{K}$, presented in Section 4.1.

Now we can compute the gradient vectors with respect to each variable, assuming that the others remain fixed. For $\mathbf{w}_k^c$ we have

$$\frac{\partial J_c}{\partial \mathbf{w}_k^{c*}} = -\mathbf{r}_{y_k d_k} + \mathbf{R}_{y_k y_k}\mathbf{w}_k^c + \sum_{j\neq k}\mathbf{R}_{y_k y_j}\mathbf{f}_j^c h_j^{(k)}. \tag{65}$$

By setting (65) to zero we can find the update rule for $\mathbf{w}_k^c$ as

$$\mathbf{w}_k^c = \mathbf{R}_{y_k y_k}^{-1}\left(\mathbf{r}_{y_k d_k} - \sum_{j\neq k}\mathbf{R}_{y_k y_j}\mathbf{f}_j^c h_j^{(k)}\right). \tag{66}$$

Similarly for $\mathbf{f}_j^c$ we have

$$\frac{\partial J_c}{\partial \mathbf{f}_j^{c*}} = -\sum_{k\neq j}\mathbf{r}_{y_j d_k} h_j^{(k)*} + \sum_{k\neq j} h_j^{(k)*}\mathbf{R}_{y_k y_k}\mathbf{w}_k^c + \sum_{k\neq j}\sum_{m\neq k,j}\mathbf{R}_{y_j y_m}\mathbf{f}_m^c h_j^{(k)*} h_m^{(k)} + \sum_{k\neq j} h_j^{(k)*} h_j^{(k)}\mathbf{R}_{y_j y_j}\mathbf{f}_j^c + \sum_{k\neq j} h_j^{(k)*} h_j^{(k)}\beta_j \mathbf{R}_{y_j y_j}\mathbf{f}_j^c. \quad (67)$$

Again, setting (67) to zero we find the update rule for $\mathbf{f}_j^c$ as

$$\mathbf{f}_j^c = \frac{1}{(1+\beta_j)\sum_{k\neq j} h_j^{(k)} h_j^{(k)*}} \mathbf{R}_{y_j y_j}^{-1}\sum_{k\neq j} h_j^{(k)*}\left(\mathbf{r}_{y_j d_k} - \mathbf{R}_{y_j y_k}\mathbf{w}_k^c - \sum_{m\neq k,j}\mathbf{R}_{y_j y_m}\mathbf{f}_m^c h_m^{(k)}\right). \quad (68)$$

Finally, for $h_j^{(k)}$ we have

$$\frac{\partial J_c}{\partial h_j^{(k)*}} = -\mathbf{f}_j^{cH}\mathbf{r}_{y_j d_k} + \mathbf{f}_j^{cH}\mathbf{R}_{y_j y_k}\mathbf{w}_k^c + h_j^{(k)}\mathbf{f}_j^{cH}\mathbf{R}_{y_j y_j}\mathbf{f}_j^c + \sum_{m\neq k,j} h_m^{(k)}\mathbf{f}_j^{cH}\mathbf{R}_{y_j y_m}\mathbf{f}_m^c + h_j^{(k)}\beta_j\mathbf{f}_j^{cH}\mathbf{R}_{y_j y_j}\mathbf{f}_j^c. \quad (69)$$

Setting (69) to zero we obtain

$$h_j^{(k)} = \frac{1}{(1+\beta_j)\mathbf{f}_j^{cH}\mathbf{R}_{y_j y_j}\mathbf{f}_j^c}\left(\mathbf{f}_j^{cH}\mathbf{r}_{y_j d_k} - \mathbf{f}_j^{cH}\mathbf{R}_{y_j y_k}\mathbf{w}_k^c - \sum_{m\neq k,j} h_m^{(k)}\mathbf{f}_j^{cH}\mathbf{R}_{y_j y_m}\mathbf{f}_m^c\right). \quad (70)$$

Note that the update rule (70) is not very useful in practice, since $\mathbf{w}_k^c$ and $[h_1^{(k)},\ldots,h_K^{(k)}]^T$ can be updated together by solving a linear MMSE problem where the fusion vectors $\mathbf{f}^c$ are fixed.

References

[1] G. Anastasi, M. Conti, M. Di Francesco, A. Passarella, Energy conservation in wireless sensor networks: a survey, Ad Hoc Netw. 7 (3) (2009) 537–568 https://doi.org/10.1016/j.adhoc.2008.06.003.

[2] J.C. Chen, K. Yao, R.E. Hudson, Source localization and beamforming, IEEE Signal Process. Mag. 19 (2) (2002) 30–39, doi:10.1109/79.985676.

[3] A. Griffin, A. Alexandridis, D. Pavlidi, Y. Mastorakis, A. Mouchtaris", Localizing multiple audio sources in a wireless acoustic sensor network, Signal Process. 107 (2015) 54–67, doi:10.1016/j.sigpro.2014.08.013.

[4] A. Hassani, A. Bertrand, M. Moonen, Cooperative integrated noise reduction and node-specific direction-of-arrival estimation in a fully connected wireless acoustic sensor network, Signal Process. 107 (2015) 68–81, doi:10.1016/j.sigpro.2014.09.001.

[5] M. Cobos, F. Antonacci, A. Alexandridis, A. Mouchtaris, B. Lee, A survey of sound source localization methods in wireless acoustic sensor networks, Wirel. Commun. Mob. Comput. 2017 (Article ID 3956282, 24 pages) (2017), doi:10.1155/2017/3956282.

[6] S. Doclo, M. Moonen, T.V. den Bogaert, J. Wouters, Reduced-bandwidth and distributed MWF-based noise reduction algorithms for binaural hearing aids, IEEE Trans. Audio Speech Lang. Process. 17 (1) (2009) 38–51, doi:10.1109/TASL.2008.2004291.

[7] S. Markovich-Golan, S. Gannot, I. Cohen, A reduced bandwidth binaural MVDR beamformer, in: Proceedings of the International Workshop on Acoustic Echo and Noise Control (IWAENC), 2010. Tel-Aviv, Israel

[8] D. Marquardt, V. Hohmann, S. Doclo, Interaural coherence preservation in multi-channel wiener filtering-based noise reduction for binaural hearing aids, IEEE/ACM Trans. Audio Speech Lang. Process. 23 (12) (2015) 2162–2176, doi:10.1109/TASLP.2015.2471096.

[9] A. Bertrand, Distributed signal processing for wireless EEG sensor networks, IEEE Trans. Neural Syst. Rehabil. Eng. 23 (6) (2015) 923–935.

[10] A. Bertrand, M. Moonen, Distributed adaptive node-specific signal estimation in fully connected sensor networks–Part I: sequential node updating, IEEE Trans. Signal Process. 58 (10) (2010) 5277–5291, doi:10.1109/TSP.2010.2052612.

[11] A. Bertrand, M. Moonen, Distributed adaptive node-specific signal estimation in fully connected sensor networks–part II: simultaneous and asynchronous node updating, IEEE Trans. Signal Process. 58 (10) (2010) 5292–5306, doi:10.1109/TSP.2010.2052613.

[12] A. Bertrand, M. Moonen, Distributed adaptive estimation of node-specific signals in wireless sensor networks with a tree topology, IEEE Trans. Signal Process. 59 (5) (2011) 2196–2210.

[13] J. Szurley, A. Bertrand, M. Moonen, Distributed adaptive node-specific signal estimation in heterogeneous and mixed-topology wireless sensor networks, Signal Process. vol.117 (2015) 44–60.

[14] J. Szurley, A. Bertrand, M. Moonen, Topology-independent distributed adaptive node-specific signal estimation in wireless sensor networks, IEEE Tran. Signal Inf. Process. Over Netw. 3 (1) (2017) 130–144, doi:10.1109/TSIPN.2016.2623095.

[15] A. Hassani, A. Bertrand, M. Moonen, GEVD-Based low-rank approximation for distributed adaptive node-specific signal estimation in wireless sensor networks, IEEE Trans. Signal Process. 64 (10) (2016) 2557–2572, doi:10.1109/TSP.2015.2510973.

[16] R.M. Gray, D.L. Neuhof, Quantization, IEEE Trans. Inf. Theory 44 (6) (1998) 2325–2383, doi:10.1109/18.720541.

[17] I.D. Schizas, A. Ribeiro, G.B. Giannakis, Consensus in ad hoc WSNs with noisy links - Part I: distributed estimation of deterministic signals, IEEE Trans. Signal Process. 56 (1) (2008) 350–364, doi:10.1109/TSP.2007.906734.

[18] I.D. Schizas, G.B. Giannakis, S.I. Roumeliotis, A. Ribeiro, Consensus in ad hoc WSNs with noisy links - Part II: distributed estimation and smoothing of random signals, IEEE Trans. Signal Process. 56 (4) (2008) 1650–1666, doi:10.1109/TSP.2007.908943.

[19] S. Kar, J.M.F. Moura, Distributed consensus algorithms in sensor networks: quantized data and random link failures, IEEE Trans. Signal Process. 58 (3) (2010) 1383–1400, doi:10.1109/TSP.2009.2036046.

[20] D. Thanou, E. Kokiopoulou, Y. Pu, P. Frossard, Distributed average consensus with quantization refinement, IEEE Trans. Signal Process. 61 (1) (2013) 194–205, doi:10.1109/TSP.2012.2223692.

[21] S.Y. Tu, A.H. Sayed, Adaptive networks with noisy links, in: Proceedings of the IEEE Global Telecommunications Conference - GLOBECOM, 2011, pp. 1–5, doi:10.1109/GLOCOM.2011.6134038.

[22] A. Khalili, M.A. Tinati, A. Rastegarnia, J.A. Chambers, Steady-state analysis of diffusion LMS adaptive networks with noisy links, IEEE Trans. Signal Process. 60 (2) (2012) 974–979, doi:10.1109/TSP.2011.2173338.

[23] V. Vahidpour, A. Rastegarnia, A. Khalili, S. Sanei, Analysis of partial diffusion recursive least squares adaptation over noisy links, IET Signal Process. 11 (2017) 749–757. 8

[24] W. Xia, Y. Wang, A variable step-size diffusion LMS algorithm over networks with noisy links, Signal Process. 148 (2018) 205–213, doi:10.1016/j.sigpro.2018.02.012.

[25] A. Sani, A. Vosoughi, On distributed linear estimation with observation model uncertainties, IEEE Trans. Signal Process. 66 (12) (2018) 3212–3227, doi:10.1109/TSP.2018.2824279.

[26] S. Markovich-Golan, S. Gannot, I. Cohen, Distributed multiple constraints generalized sidelobe canceler for fully connected wireless acoustic sensor networks, IEEE Trans. Audio Speech Lang. Process. 21 (2) (2013) 343–356, doi:10.1109/TASL.2012.2224454.

[27] Y. Zeng, R.C. Hendriks, Distributed delay and sum beamformer for speech enhancement via randomized gossip, IEEE/ACM Trans. Audio Speech Lang. Process. 22 (1) (2014) 260–273, doi:10.1109/TASLP.2013.2290861.

[28] T.C. Lawin-Ore, S. Doclo, Analysis of rate constraints for MWF-based noise reduction in acoustic sensor networks, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2011, pp. 269–272, doi:10.1109/ICASSP.2011.5946392.

[29] A.I. Koutrouvelis, T.W. Sherson, R. Heusdens, R.C. Hendriks, A low-cost robust distributed linearly constrained beamformer for wireless acoustic sensor networks with arbitrary topology, IEEE/ACM Trans. Audio Speech Lang. Process. (2018) 1–1, doi:10.1109/TASLP.2018.2829405.

[30] A. Bertrand, M. Moonen, Robust distributed noise reduction in hearing aids with external acoustic sensor nodes, EURASIP J. Adv. Signal Process. 2009 (2009) 14. Article ID 530435

[31] R. Serizel, M. Moonen, B.V. Dijk, J. Wouters, Low-rank approximation based multichannel wiener filter algorithms for noise reduction with application in cochlear implants, IEEE/ACM Trans. Audio Speech Lang. Process. 22 (4) (2014) 785–799, doi:10.1109/TASLP.2014.2304240.

[32] S. Markovich-Golan, S. Gannot, I. Cohen, Blind sampling rate offset estimation and compensation in wireless acoustic sensor networks with application to beamforming, in: Proceedings of the International Workshop on Acoustic Signal Enhancement, IWAENC, 2012, pp. 1–4.

[33] S. Miyabe, N. Ono, S. Makino, Blind compensation of inter-channel sampling frequency mismatch with maximum likelihood estimation in STFT domain, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, 2013, pp. 674–678, doi:10.1109/ICASSP.2013.6637733.

[34] M.H. Bahari, A. Bertrand, M. Moonen, Blind sampling rate offset estimation for wireless acoustic sensor networks through weighted least-squares coherence drift estimation, IEEE/ACM Trans. Audio Speech Lang. Process. 25 (3) (2017) 674–686, doi:10.1109/TASLP.2016.2647713.

[35] A. Sripad, D. Snyder, A necessary and sufficient condition for quantization errors to be uniform and white, IEEE Trans. Acoust. Speech Signal Process. 25 (5) (1977) 442–448, doi:10.1109/TASSP.1977.1162977.
[36] R.M. Gray, Quantization noise spectra, IEEE Trans. Inf. Theory 36 (6) (1990) 1220–1244, doi:10.1109/18.59924.
[37] F. de la Hucha Arce, M. Moonen, M. Verhelst, A. Bertrand, Adaptive quantization for multichannel wiener filter-based speech enhancement in wireless acoustic sensor networks, Wirel. Commun. Mob. Comput. 2017 (2017) 15. doi: 10.1155/2017/3173196. Article ID 3173196.
[38] J.C. Bezdek, R.J. Hathaway, Some Notes on Alternating Optimization, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 288–300.
[39] J.C. Bezdek, R.J. Hathaway, Convergence of alternating optimization, Neural Parallel Sci. Comput. 11 (4) (2003) 351–368.